3. Argitaraldia

TCP/IP sareak

J.Mª. Rivadeneyra

Udako Euskal Unibertsitatea

TCP/IP SAREAK
3. ARGITARALDIA

José Mª Rivadeneyra

Udako **Euskal Unibertsitatea**
Bilbo, 2009

TCP/IP Sareak, 3. argitaraldia

ISBN: 978-84-8438-235-5
Lege-gordailua: BI-2002-09

Inprimategia: Lightning Source
Azalaren diseinua: Iñigo Ordozgoiti
Hizkuntza-zuzenketen arduraduna: Ander Altuna Gabiola

Banatzaileak: UEU. Erribera 14, 1. D BILBO telf. 946790546 Faxa. 944793039
Helbide elektronikoa: argitalpenak@ueu.org
www.ueu.org

Aurkibidea

Hitzaurrea

Liburu honen 2. bertsioa argitaratu zenetik 4 urte eta erdi igaro dira. Denbora horretan, liburua erabili dugunok hainbat gabezia eta hobetzekoak atzeman dizkiogu. Horien ondorioa da begi aurrean duzun 3. bertsio hau. Liburuaren edukiak eta egitura aldatu dira, baina ez, ordea, liburua argitaratzeko asmoa: helburu akademikoa eta euskararen garapenaren aldekoa hasierakoak dira. Interneten teknologiak eta barrukoak ikasi nahi dituzten eta irakasten dituzten euskaldunek bidea errazagoa izango dute liburu honen eskutik. Liburu honi esker, euskaldunoi errazagoa egiten bazaigu TCP/IP teknologiaz hitz egitea eta idaztea, helburua beteta egongo da.

NORI DAGOKIO LIBURU HAU?

Helburu akademikoa izanik, liburua bereziki interesatuko zaie Internet inguruko teknologiak ikasten dituztenei. Oso aproposa da Ingeniaritza Informatikoa ikasten dutenentzat, baita Telematika eta Telekomunikazioko titulazio unibertsitarioetan dabiltzanentzat ere. Ertaineko irakaskuntza teknikoetan ere oso erabilgarria izango da, batez ere irakasleriaren formazioan, euskaraz idatzitako testuen eskasia kontuan harturik.

Hala ere, liburua ahalik eta irakurtzen errazena egiten saiatu naiz, gaian sakondu nahi duen edozein euskaldunentzat erabilgarria izateko. Internet eta konputagailu-sareak barrutik ezagutzeko jakin-mina duen edozeinentzat idatzi dut liburua.

BERRITASUNAK

Aurreko bertsioa zeharo berrikusi dut, datu asko eguneratuz (adibidez, erreferentziatutako RFC guztiak berrikusita daude) eta azalpen batzuk argituz. Beste kontu batzuk, aldiz, kendu ditut, zaharkituta zeudelako. Baina xehetasunetatik harago, egiturako eta ikuspegiko aldaketak ere egon dira. Hasteko, liburuaren itxura klasikoagoa da, aurreko bertsioetan agertzen ziren ikonorik gabekoa. Ikono horiek liburuaren lehenengo argitaraldiaren osagarriak ziren animazioetarako sartu nituen. Bigarren erabilera estetikoa izan zen: liburuaren itxura arinagoa izateko zeuden, irakurketan laguntzeko asmoz. Argitaraldi honetan kendu ditut, animaziorik ez dagoelako eta egitura homogeneoa eman nahi izan diodalako testuari. Irakurtzeko erraztasuna ez galtzeko, irudi gehiago sartu ditut.

Edukiari dagokionez, eta Informatikan ezinbestekoak diren eguneraketez gain, zenbait aldaketa ditugu. Interneten egitura aztertzen da sakonago lehenengo

eta bigarren kapituluan, alde praktikoak eta azpiegiturari dagozkionak (hornitzaileak eta beraien arteko harremanak) gehiago jorratuz, eta teorikoak (komunikazioko arkitekturaren ingurukoak) arinduz. Bigarren kapituluan dezente zabaldu eta hobetu dut bideratzeari dagokiona, baita IPv6ri buruzko atala ere. Aplikazioei eta segurtasunari dagozkien kapituluak oso aldatuta daude. Lehenengoan, sare-aplikazioen diseinuari buruzko atala sartu dut, P2P eredua eta aplikazioak kontuan hartu ditut, DNSri buruzko atala dezente hobetuta dago, web aplikazioen deskribapena berria da, eta IP telefonia gehitu dut. FTPri buruzko atala kendu dut, haren erabilera behera doalako eta kapitulua handiegi ez bilakatzeko. Segurtasunari buruzko kapitulua berrantolatu dut, sare seguruak lortzeko teknikak eta komunikazio seguruak lortzekoak bereiziz. Sare-segurtasunaren kudeaketari buruzko atala gehitu dut, eta sarrera-kontrolari zegokiona asko zabaldu eta hobetu dut. Alabaina, kapitulua luzeegia ez izateko, posta elektronikoaren segurtasunari buruzko atala kendu dut. Azkenik, aplikazio banatuen programazioari buruzko kapitulua sinplifikatu eta eranskin moduan utzi dut.

IRAKURLEAREN EKARPENAK

Irakurlearen komentarioak eta gomendioak ongi etorriak izango ditut. Esadazue, zuen ustetan, zer sartu beharko nukeen hurrengo edizio batean, eta zer ez nukeen sartu behar, zertan zaudeten gustura liburuarekin eta zertan ez. Mesedez, bidali iezazkidazue zuen mezuak tcpipsareak@ehu.es helbidera.

ESKERRAK

Lehenik eta behin, funtsezkoa izan da Donostiako Informatika Fakultateko irakasle euskaldun asko aspalditik egiten ari diren lan izugarria Informatikaren arloan hiztegi teknikoaren sorreran. Haietako edozein beti prest laguntzeko agertu da nire zalantzak argitzeko. Alex Mendiburu irakasleak oso eskertzeko lana egin du testu osoa berrikusita; zenbait adierazpen argiago daude berari esker. Ana Gonzalez irakasleari eta nire fakultateko ikasleei esker ere, hainbat huts garbitu ditut.

UPV/EHU euskal unibertsitate publikoaren baliabideak eta azpiegiturak erabili ditut liburua idazteko. Testuaren berrikuspen linguistikoa eta argitaratzeko lanak Udako Euskal Unibertsitateak egin ditu. Interneten denok erabiltzeko moduan dauden baliabide linguistiko asko erabili ditut, baina Elhuyarrek argitaratutako hiztegia izan da gehien kontsultatu dudana. Mila esker instituzio horiei guztiei.

Testua idazteko OpenOffice 2.3.0 erabili dut, Ubuntu 7.10 (*Gutsy Gibbon*) sistema batean. Testuaren ortografia berrikusteko Xuxen zuzentzaile automatikoa erabili dut, eta irudiak sortzeko Dia 0.96.1 diagrama-editorea. Software hori guztia garatu eta eskura jarri digutenei, eskerrik asko.

J. M. Rivadeneyra
Donostian, 2009ko otsailean

1. Sarrera

Kapitulu hau ikasi eta gero, ikasleak ondo ulertu beharko ditu honako kontzeptu hauek:

- Zein den Interneten egitura fisikoa: nolako sareek osatzen duten eta nolakoak diren haien arteko erlazioak.
- Zer diren sare-arkitektura bat, mailak, zerbitzuak, protokoloak, mailen arteko interfazeak, eta maila bateko entitateak.
- Zeintzuk diren TCP/IP arkitekturaren mailak, eta zeintzuk diren maila bakoitzeko ezaugarri nagusiak.

1.1. INTERNETEN EGITURA FISIKOA

Internet mundu mailako konputagailu-sare bat da, hau da, munduan zehar dauden milioika konputagailu elkarrekin konektatzeko erabiltzen den sare bat. Konputagailuek sarea erabiltzen dute beraien artean «hitz egiteko», informazioa eta aginduak trukatzeko alegia, eta, horrela, konputagailu horiek erabiltzen dituzten gizakiei zerbitzuak emateko. Sarea beste gailu askok eta beraien arteko konexioek osatzen dute. Sare barruko gailu horiek, askotan konputagailu bereziak direnak, sare-gailuak deituko ditugu. Haien artean kontzentragailuak, kommutagailuak, eta bereziki garrantzitsuak diren bideratzaileak ditugu. Sare-gailuen arteko konexio gehienak kable bidezkoak dira (kobrezkoak, zuntz optikokoak, kable ardazkidekoak...), baina kablerik gabekoak ere badaude.

Gure konputagailua beste konputagailuekin komunikatzeko erabiltzen dugu Internet, baina Internet ez da sare bakar bat, sare askok osatutako sarearte bat baizik. Munduko beste puntan dagoen konputagailuraino heltzeko, gure makinatik ateratzen den informazioak honako sare hauek zeharkatuko ditu (ikusi 1.1. irudia):

- Gure sarea (jatorrizko sarea), eta helburua den konputagailuaren sarea (helburuko sarea). Hauek erakunde, enpresa edo partikular batzuen sareak izango dira. Hor daude fisikoki konektatuta (kable edo uhinen bidez) jatorrizko eta helburuko konputagailuak. Batzuetan etxeko edo enpresa txiki bateko sare lokala izango da, konputagailu gutxi batzuk kommutagailu (edo *switch*) baten bidez elkarturik osatuta. Beste batzuetan erakunde baten

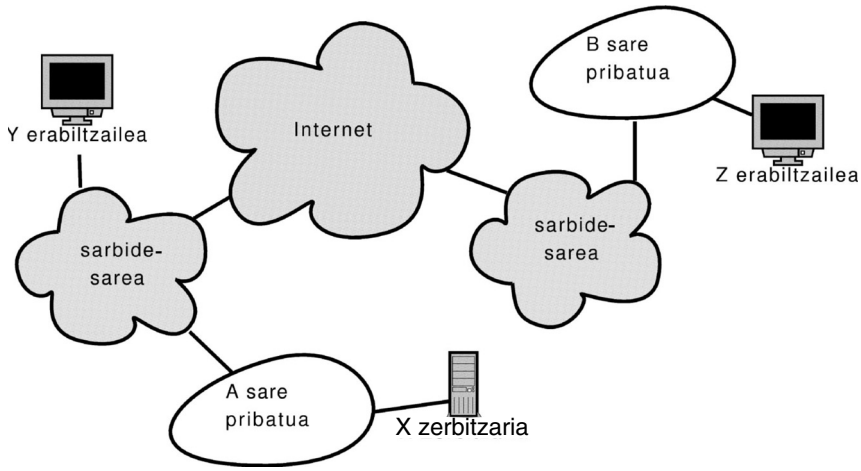
egoitza desberdinetan dauden sare lokalak telekomunikazio-konpainia bati kontratatutako konexioen bidez osatutako sareartea izango da. Edonola ere, Interneti lotutako sare pribatuak dira.

- Aurreko sareak eta Internet bera lotzen dituzten sareak. Hauei Interneterako sarbide-sareak deitzen zaie. Beste telekomunikazio-zerbitzuak (telefonia, telebista) eskuratzeko erabiltzen diren sare berak dira. Adibidez, etxean eta enpresa txikietan telefonoa konektatzeko dugun sare telefonikoa edo telefono/telebista zerbitzua ematen digun kable-sarea erabiltzen dira Internet atzitzeko. Lehen, etxeko testuinguru horretan, modem bidezko konexioak eta ISDN[1] konexioak erabiltzen ziren gehienbat. Gaur, sare telefonikoaren kasuan, teknologia horiek xDSLk ordezkatu ditu, eta kablerik gabeko teknologiak (WiMax nagusiki) aukera kabledunetarako alternatiba sendo bezala ari dira suertatzen. Etxekoak baino transmisio-behar handiagoak dituzten erakunde eta enpresen sareen kasuan, Interneterako sarbiderako espresuki eskaintzen diren beste konexio batzuk kontratatzen zaizkie telekomunikazio-konpainiei. Horretarako zenbait aukera teknologiko dituzte.

  Sarbide-sare hauen garrantzia izugarria da: sarbide txarra badugu, jai dugu Interneten.

- Internet hornitzaileen sareak. Hornitzaile hauek ingelesezko ISP (*Internet Service Provider*) siglaz dira ezagunak. Hauek dira Interneterako atea. Sarbide-sareak guk aukeratutako ISPren sarearekin konektatzen du gure sarea. Badaude ISP txikiak (edo Internet txikizkariak) eta handiak (edo Internet handizkariak). Txikizkariek Interneterako sarrera ematen diete erabiltzaileen sareei zuzenean. Handizkariek ISP txikiei ematen diete zerbitzua, trafikoa beraien artean mugituz. Handizkarien artean ere badaude mailak. Maila gorenekoek *Tier1* izena hartzen dute (*tier* hitzak, ingelesez, *maila* esan nahi du). Oso gutxi dira munduan, dozena bat inguru. Beraien sareak guztiz interkonektatuta daude, hau da, Tier1 baten sareak konexio zuzena du beste Tier1 sare guztiekin, inongo bitartekaririk gabe. Tier1 guztien sareek horrela osatzen duten sareartea Interneten ardatza da (ingelesez, *Internet backbone*). Bigarren mailako Internet handizkariek Tier2 izena hartzen dute. Tier2 sareek Tier1 batekin gutxienez izango dute konexioa, Internet osorako bidea edukitzeko. Batzuetan Tier3 terminoa ere erabiltzen da, Tier1 batekin konexiorik ez duten hornitzaileak izendatzeko. Horiek dira, gehienetan, zuzenean erabiltzaileei Interneterako sarrera ematen dietenak, hau da, guk Internet txikizkariak deitu ditugunak.

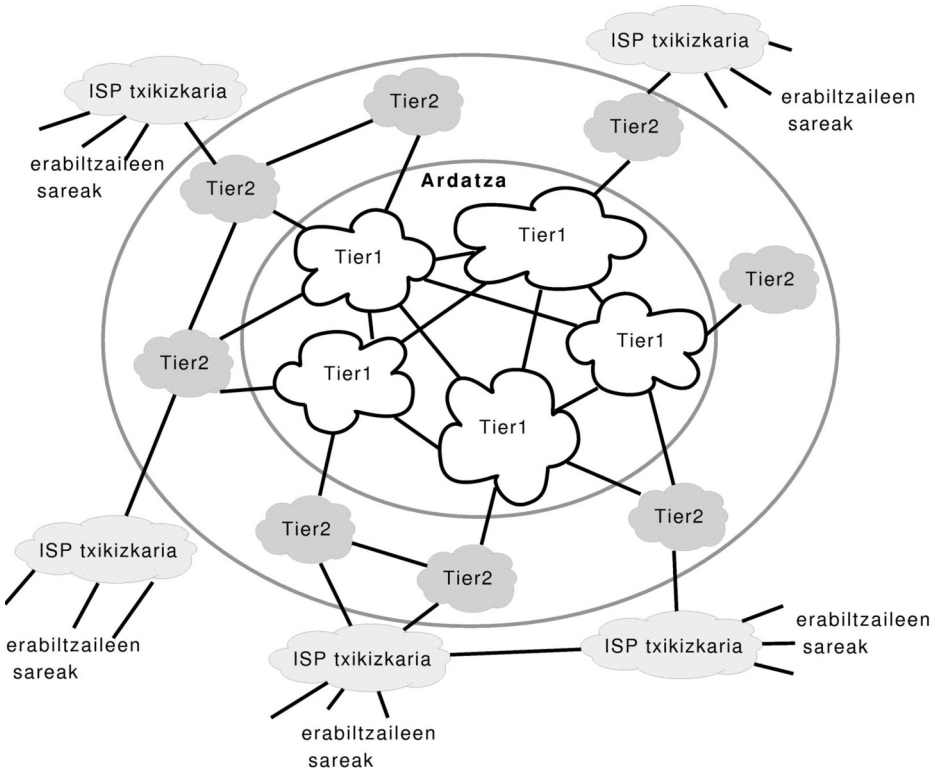
1. Hego Euskal Herrian RDSI izenarekin komertzializatu zen teknologia hau.



1.1. irudia. Tokiko sareak, sarbide-sareak, eta Internet.

1.1. irudian hiru konputagailu agertzen dira, X, Y, eta Z izenekoak. X eta Z sare pribatuetan daude kokatuta (etxeko edo laneko Ethernet sarea izan daiteke). Y erabiltzailea, aldiz, zuzenean konektatuta dago Internetera sarbide-sare baten bidez, inongo bertako sarerik gabe. Y-ren konfigurazioa ez zen arraroa orain dela urte batzuk (etxeko modem bidezko konexioak horrelakoak ziren eta), baina gaur egun gutxitan aurkituko dugu horrelako konexio zuzena duen konputagailu bat.

1.2. irudian Interneten barruko egitura azaltzen zaigu, hau da, Internet hornitzaileen sareen interkonexioak osatzen duena. Horren muinean ardatz-sarea dugu, eta horri lotuta Tier2 mailako hornitzaileen sareak. Horiek, Tier1 eta Tier2 sareak, igarobideak dira, hau da, ez dute Interneten trafikoa txertatzen ezta jasotzen ere. Erabiltzaileen sareei Interneterako konexioa ematen dieten Internet txikizkarien sareak dira trafikoaren iturburuak eta helburuak. Irudiak adierazten duenez, ardatz-saretik kanpo dagoen hornitzaile baten sareak beste sareekiko konexio bat baino gehiago izan ditzake. Maila bereko beste sareekiko loturak (bi Tier2 edo bi txikizkarien sareen artekoak) zirkuitulaburrak dira, gainkargatuta egon daitekeen Internet ardatza igaro gabe trafikoa bideratzeko bidezidorrak, alegia. Hurrengo atalean, IP maila aztertzen dugunean, berriro ikusiko ditugu bidezidor horiek. Goragoko sareekiko lotura anitzek (Tier2 batek bi Tier1 sareekiko loturak, adibidez) trafikoa beren helburura arinago heltzea eta *backup*arena egitea dute helburu.



1.2. irudia. Internet barruko egitura. Maila goreneko (Tier1) hornitzaile handizkariek ardatza osatzen dute. Txikizkariek zerbitzua ematen diete Interneteko erabiltzaileei, irudian agertzen ez diren sarbide-sareen bidez.

1.2. INTERNETEN EGITURA LOGIKOA: TCP/IP PROTOKOLOAK

1.2.1. Sare-arkitekturak

Demagun sare baten bidez lotutako bi konputagailu, eta demagun fitxategi bat eraman nahi dugula konputagailu batetik bestera, sarea erabiliz. Horretarako aplikazio bat sortu beharko dugu, bi programek osatuta: bata konputagailu batean egikarituko da, eta bestea beste konputagailuan. Aplikazio bat sarearen bidez lotutako konputagailu desberdinetan egikaritzen diren programek osatzen dutenean, **aplikazio banatu** edo **sare-aplikazio** bat dela diogu. Askotan (baina ez beti) aplikazioaren zati bati **bezero** deitzen diogu, eta besteari **zerbitzari**. Demagun gure zerbitzariak bezeroari bidaliko diola fitxategia. Horretarako, honako bi arazo hauek konpondu behar dira:

- Bidalketa burutzeko, bezeroak eta zerbitzariak hitz egin beharko diote elkarri. Adibidez, bezeroak adierazi beharko dio zerbitzariari fitxategi bat

jaso nahi duela, fitxategi horren izena edota kokapena adieraziz. Eta, agian, zerbitzariak erabiltzaile baten izena eta pasahitza eskatuko dizkio bezeroari ezer bidali baino lehen. Elkarrizketa hori nolakoa izango den definitu eta programatu beharko dugu. Hau da, zehaztu beharko da zeintzuk diren komando posibleak eta beren erantzunak. Sare-aplikazioaren zatiek, bezeroak eta zerbitzariak alegia, elkarrizketa hori gauzatuko dute fitxategia trukatzeko.

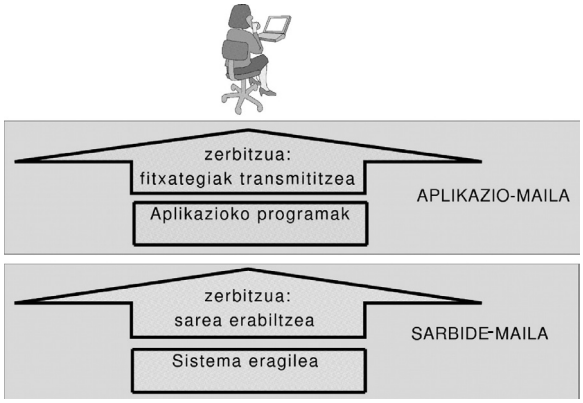
- Elkarrizketa hori gauzatzeko erabiliko diren komandoak eta erantzunak, baita gero fitxategia osatzen duten bitak ere, sareari eman beharko zaizkio, sareak bit horiek eraman ditzan bere helburuko konputagailura. Sareari informazioa emateko eta saretik informazioa jasotzeko sare-txartela dugu. Sare-txartelari informazioa emateko eta hortik informazioa jasotzeko, sistema eragilearekin integratuta dagoen sare-txartelaren *driverra* (edo kontroladorea) erabili behar dute aplikazioek.

Bi arazo horien konponketak guztiz desberdinak eta independenteak dira, edo, beste era batean esateko, *maila desberdin*eko arazoak dira. Lehenengo arazoa aplikazioak konpondu beharko duenez, **aplikazio-maila**ko arazoa dela esango dugu. Bigarren arazoa sarea atzitzea denez, **sarbide-maila**n kokatuko dugu.

Zerbitzuak

Gure programek, bezeroak eta zerbitzariak, **zerbitzu** bat eskaintzen diote erabiltzaileari. Gure adibide honetan, zerbitzu hori fitxategiak konputagailu batetik bestera eramatea da.

Erabiltzaileak aplikazio-mailak eskaintzen duen zerbitzua erabiltzen duen era berean, programa batek beste konputagailura zer edo zer bidali nahi duenean, txartela kontrolatzen duen driverra erabiliko du. Sarbideak eskaintzen duen zerbitzua aplikazioak erabiliko du, alegia. Beraz, nahiz eta bi mailetako arazoak era independentean konpondu, batak (aplikazioak) bestearen gainean (sarbidea) egin behar du lana.



1.3. irudia. Mailak eta zerbitzuak.

Entitateak eta protokoloak

Aplikazioa osatzen duten programei **aplikazio-mailako entitateak** esaten zaie, aplikazio-mailako arazoa konpontzen baitute. Lehen ikusi dugu aplikazio-mailako entitate horiek, beren zerbitzua gauzatzeko, elkarren artean hitz egin behar dutela. Elkarrizketa hori nolakoa izango den definitzea, hau da, komando eta erantzun posibleen definizioa, aplikazio-mailako protokolo bat da. Erabiltzaileari ematen zaion zerbitzu bakoitzeko, aplikazio bat garatu behar da eta, horrekin batera, **aplikazio-mailako protokolo** bat definitu behar da. Gure kasuan, fitxategiak trukatzeko sare-aplikazio bat diseinatuko dugu, eta aplikazioa programatzeko beharrezkoa den aplikazio-protokoloa ere definitu beharko dugu.

Era berean, sare-mailako arazoa konpontzeko erabilitako txartela **sarbide-mailako entitate** bat da. Aplikazio-mailan gertatzen den bezala, makina desberdinetan dauden sarbide-mailako entitateen artean ere elkarrizketa bat egingo da, eta elkarrizketa hori arautzen duen **sarbide-mailako protokoloa** sortu behar da. Bi maila horien protokoloak guztiz ezberdinak dira. Aplikazio-mailako protokoloan truka daitezkeen mezuak zeintzuk diren (komandoak eta erantzunak), nolakoak diren (haien sintaxia), eta haien erabilera adierazten dira. Sarbide-mailan, aldiz, erabilitako sarean informazio-unitateek duten egitura eta igortzeko/hartzeko prozedurak definitu behar dira. Adibidez, Ethernet sareetan, informazioa *trama* izeneko egituretan bidali behar da. Trama horiek zein eremu dituzten, eta trama horiek nola igorri eta hartu behar diren definitzen da Ethernet sare baterako sarbide-protokoloan (Ethernet izen bera duen protokoloa). Gero, Ethernet sarea erabiliz egikaritzen dugun aplikazioko programek beren artean trukatutako komandoak eta erantzunak (adibidez, GET komandoa eta bere erantzuna web orri bat jaisteko) aplikazio horren protokoloak definitzen ditu.

Dena batera: sare-arkitektura

Gure bi konputagailuen artean fitxategiak trukatzeko definitutako maila, zerbitzu eta protokoloek **sare-arkitektura** bat eratzen dute. Maila bakoitzean arazo gutxi batzuk konpontzen dira, eta beste mailetako arazoak ahazten dira. Maila bakoitzak bere zerbitzuak eskaintzen dizkio gainean dagoen mailari. Arkitekturako goi-goian dagoen mailak, aplikazio-mailak alegia, erabiltzaileari zuzenean eskaintzen dizkio bere zerbitzuak. Eta maila bakoitzean komunikazio-protokoloak erabiltzen dira, maila bereko entitateen arteko komunikazioa arautzeko eta, horrela, mailak eskaini behar dituen zerbitzuak gauzatzeko.

Orain arte erabili dugun arkitektura bi mailakoa da: sarbide-maila eta aplikazio-maila. Lehenengoak konpontzen duen arazoa hau da: bitak makina batetik bestera mugitzeko sare bat erabiltzea. Hori bera da sarbide-mailak goiko mailari ematen dion zerbitzua: informazioa makina batetik bestera eramatea. Erabilitako

sarbide-mailako protokoloa erabilitako sareak ezartzen duena izango da: Ethernet eta PPP (lotura zuzenetan erabiltzen dena) dira horretarako gehienbat erabiltzen direnak.

Aplikazio-mailak konpontzen duen arazoa da fitxategiak konputagailu batetik bestera mugitzea. Hori da gure aplikazioak erabiltzaileari ematen dion zerbitzua, hain zuzen ere. Zerbitzu hori gauzatzeko, aplikazioko protokoloa erabiltzen dute zerbitzariek eta bezeroek (aplikazio-mailako entitateek, alegia).

Beraz, ondo bereizi kontzeptu hauek guztiak:

- Sare-arkitektura baten osagaiak honako hauek dira: mailak, zerbitzuak eta protokoloak.
- Maila bakoitzak bere gaineko mailari eskaintzen dizkio bere zerbitzuak, eta aplikazio-mailak, erabiltzaileari.
- Zerbitzuak gauzatzeko, maila bereko entitateek egiten dute lan elkarrekin, eta beraien arteko komunikaziorako protokoloak behar dira arkitekturako maila bakoitzean.

Mailen arteko interfazeak

Gure arkitekturako aplikazioko entitateek (bezero- eta zerbitzari-programek, alegia) sarbide-mailako zerbitzuak eskuratzeko sarbide-mailako interfazea erabili beharko dute. Sarbide-mailako entitate bakoitzak, hau da, sare-teknologia bakoitzak, bere interfaze propioa definitzen du. Aplikazio-mailako programa batek zuzenean interfaze hori erabiltzea bideragarria ez denez (programatzaileek jakin beharko lukete nola erabiltzen den txartela bakoitza), driverrak erabiltzen dira. Driverrak programatzaile espezializatuek sortutako programak dira, txartelarekin «hitz egiten» dutenak, informazioa igorri eta hartzeko. Bezeroek eta zerbitzariek sarea erabili behar dutenean, sistema eragilearen dei berezi batzuk erabiliko dituzte, txartelaren driverra erabiltzen duten sistema-deiak, hain zuzen ere. Sistema-dei horiek osatuko dute sarbide-mailako interfazea.

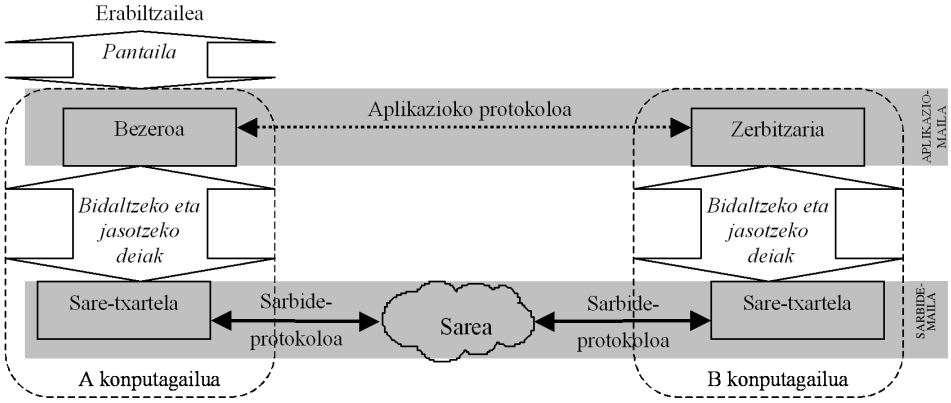
Era berean, erabiltzaileak, aplikazio-mailako zerbitzuak eskura ditzan, aplikazioko interfazea erabili beharko du. Gaur egun aplikazio gehienen interfazea grafikoa da, baina oraindik ere karaktere moduko interfazeak behin baino gehiagotan ikusiko ditugu.

Oro har, konputagailu berean dauden bi maila ezberdineko entitateen arteko komunikazioa beheko mailaren interfazearen bidez egiten da. Ez nahastu protokoloak eta interfazeak: protokoloak **makina desberdinetan** kokatutako **maila bereko entitateak** komunikatzeko **arauak** dira; interfazeak, aldiz, **makina**

berean kokatutako **maila desberdineko entitateak** komunikatzeko **deiak** edo mekanismoak[2] dira.

Entitate bakoitzak bere interfaze propioa du: interfazeak ez dira sare-arkitekturan definitzen.

Ondoko irudi honetan gure arkitekturako maila, entitate, protokolo, interfaze eta zerbitzuen arteko erlazioak ageri dira.



1.4. irudia. Bi mailako sare-arkitektura.

Aplikazioko protokoloari dagokion gezia ez da jarraitua irudian. Sarbide-protokoloari dagokiona, aldiz, bai. Horrekin hauxe adierazi nahi da: aplikazio-mailako entitateen komunikazioa ez da zuzena, beheko mailak ematen dizkion zerbitzuen zeharkakoa baizik. Sarbide-mailak, ostera, ez du inoren laguntza behar, bere entitateen arteko komunikazioa zuzena delako, bitartekaririk gabekoa.

1.2.2. TCP/IP sare-arkitektura[3]

1.4. irudian agertzen den arkitektura ez da nahikoa aplikazio banatuek dituzten komunikazio-beharrak asetzeko. Sare-arkitektura horren sarbide-mailaren zerbitzua, edozein sare mota erabilita, sare berean dagoen beste konputagailuraino informazioa eramatea da. Baina honako bi arazo hauek konpondu gabe gelditzen dira arkitektura horrekin:

2. Liburu honetan ez dugu azalduko, baina, sare-teknologiaren arabera, sarbide-maila ere beste azpimaila batzuetan zatitzen da. Maila horietako baxuena fisikoa da beti, eta maila fisikoa erabiltzeko ez dago «deiak» egitea, baizik eta seinaleak gaitzea edo desgaitzea. Horiek dira esaldi honetan aipatzen diren «mekanismoak».

3. Interneten ezaugarri teknikoak definitzen duen erakundeak (IAB -Internet Architecture Board) ez du inoiz TCP/IP arkitektura bat definitu, baina askotan, baita liburu honetan ere, kontzeptu hori erabiltzen da TCP/IP protokoloen egitura adierazteko. Ikusi RFC 1958.

- Aplikazio banatuaren beste aldea (bezeroa edo zerbitzaria) beste mota bateko sare batean badago, ez dago haraino ailegatzerik.
- Aplikazio banatuaren beste aldea kokatuta dagoen konputagailua atzituta ere, une berean egikaritzen ari diren prozesuen artean, nola jakin zeini eman behar zaion informazioa?

Sarearte-maila

Lehenengo arazoa konpontzeko, bi urrats eman behar dira:

1. Sareen artean kokatu konputagailu bereziak, sareen arteko konexio fisikoa gauzatzeko. Konputagailu berezi horiek **bideratzaileak** izango dira (ingelesez, *router*). Horietako bideratzaile batek bi sare konektatzen baditu, bi sare-txartela beharko ditu, bakoitza dagokion teknologiarena. Hiru sare interkonektatzen badira bideratzaile baten bidez, orduan bideratzaile horrek hiru sare-txartel izango ditu. Oro har, bideratzaileak interkonektatzen dituen sare adina txartel izango ditu.
2. Sarean txertatzen dugun informazioari bere bidea aurkitzeko (hau da, zein bideratzailetatik pasatu behar duen helmugako sareraino iristeko) kontrol-datu batzuk gehitu beharko dizkiogu. Bidali nahi dugun informazioak gehi bideratzerako datuek **datagrama** izeneko datu-egitura osatzen dute. Bideratze-informazio hori bideratzaileek erabiliko dute erabakitzeko txartel batetik hartutako datagrama zein txarteletatik birbidali behar duten.

TCP/IP arkitekturan, bideratze-informazio hori zein den **IP protokoloa**k definitzen du (IP – *Internet Protocol*, hau da, sarearterako protokoloa). Hurrengo atalean aztertuko dugu protokolo hori; oraingoz nahikoa da datagramaren formatua definitzen duela jakitea.

Informazioaren jatorrizko konputagailuan, nork osatuko ditu IP datagramak, bideratze-informazioa gehituz? Aplikazioko entitateek (bezeroek eta zerbitzariek, alegia)? Horrek suposatuko luke aplikazio-programatzaile guztiek ezagutu beharko luketela IP protokoloa, bere xehetasun guztiekin. Hobe da lan-zama hori beraien lepotik kentzea. Horregatik, sistema eragilearen zati bat den programa multzo batek egiten du lan hori. Programa multzo hori **sarearte-entitatea** edo **IP entitatea** da.

Konturatu gabe, 1.4. irudiko arkitekturari maila bat gehitu diogu: **sarearte-maila**, edo **IP maila**. Maila horrek ematen duen zerbitzua da informazioa sarearteko konputagailu batetik bestera eramatea. Adi: gure sarearen mugak gainditu ditugu, hau da, sarbide-mailak ematen digun zerbitzua baino zerbait gehiago dugu orain. IP mailak, bere zerbitzua emateko, sarbide-mailak erabiliko ditu. Hau da, datagrama bat bidaltzeko edo jasotzeko, IP entitatea osatzen duten programek sare-txartela atzitzeko interfazea (sarbide-mailako interfazea, alegia) erabiliko dute.

Beste alde batetik, aplikazioko entitateek (bezeroek eta zerbitzariek, alegia) sistema eragilearen deiak erabiliko dituzte IP mailako zerbitzuak atzitzeko, hau da, informazioa sarean txertatzeko edo jasotzeko.

Datagramak bere sareartean zeharreko bidean bisitatutako bideratzaile bakoitzean beste IP entitate bat izango dugu. Horiek datagramak jaso, bere IP **goiburukoak** aztertu, eta birbidaliko dituzte. Gainera, bideratze-erabakiak hartzeko behar dituzten datuak bilduko dituzte.

Garraio-maila

Hurrengo atalean ikusiko dugunez, IP mailak ematen duen zerbitzua ez da fidagarria. Hau da, ez du bermatzen datagrama guztiak helduko direnik beren helburura, ezta ateratako ordena berean ailegatuko direnik ere. Adibidez, gerta daiteke bidean dagoen bideratzaile batek bideratze-informazio okerra edukitzea eta datagrama gaizki birbidaltzea. Edo okerrago, gerta daiteke bideratzaile hori gainezka egotea, eta datagrama hori zakarrontzira zuzenean botatzea, birbidali gabe.

Horrelako fidagarritasunik gabeko zerbitzua nahikoa da aplikazio askotarako (adibidez, IP telefoniarako), baina guztiz onartezina da beste askotarako (adibidez, posta-mezuak bidaltzeko). Horregatik, IP mailaren lana gainbegiratzen duen beste maila bat behar da, fidagarritasuna behar duten aplikazio horien beharra asetzeko. Maila horri **garraio-maila** deitzen zaio, eta haren lana fidagarritasuna behar duten aplikazioei datagramak heltzen direla bermatzea da, eta gainera ordena mantenduz. Horrenbesteko fidagarritasuna behar ez duten aplikazioei beste zerbitzu xumeago bat eskainiko die garraio-mailak. TCP/IP arkitekturan, garraio-mailako entitateek zerbitzu fidagarria **TCP protokoloa** erabiliz gauzatzen dute, eta zerbitzu ez fidagarria, **UDP protokoloa**ren bidez. Horregatik, TCP/IP arkitekturako garraio-mailako entitateei TCP/UDP entitateak deitzen zaie askotan.

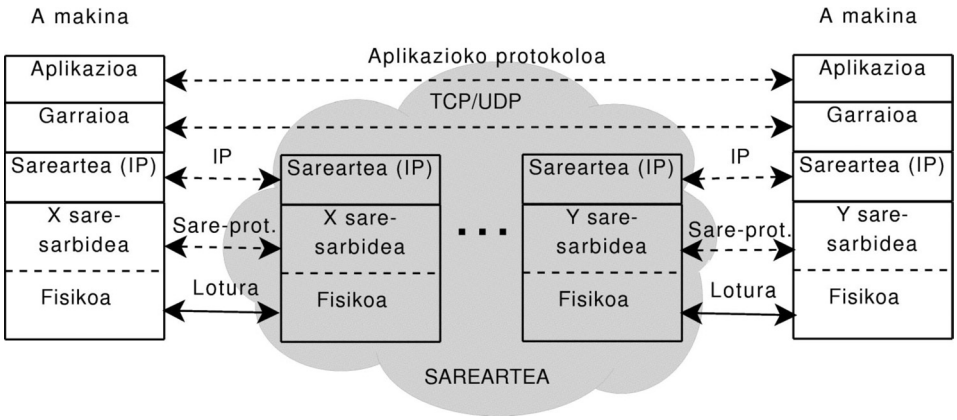
Hala ere, oraindik badugu beste arazo bat konpontzeke: datagrama bere helburura iristen denean, nola jakingo du horko TCP/UDP entitateak zein bezero/zerbitzariri (hau da, zein aplikazioko entitateri) eman behar dion datagrama heldu berriak bere barruan garraiatzen duen aplikazio-mailako informazioa? Hori da garraio-mailak (bai TCPk, baita UDPk ere) ematen duen beste zerbitzu bat: aplikazioen identifikazioa. Hirugarren kapituluan ikusiko dugu nola egiten duen.

IP mailarekin gertatzen den bezala, **TCP/UDP entitatea** sistema eragilearen zati bat den programa multzo bat da. Garraio-mailak aplikazioei eskaintzen dien **zerbitzua** hauxe da: informazioa eramatea sareartean dagoen konputagailu batean egikaritzen ari den aplikazio batetik sarearteko beste edozein konputagailutan egikaritzen ari den beste aplikazio-entitate batera (adi: IP mailak ematen duen zerbitzuak antzekoa dirudi, baina oso desberdina da). Aplikazioak erabakiko du ea zerbitzu hori era fidagarrian nahi duen (hau da, TCP erabiliz), ala era ez-

fidagarrian (hau da, UDP erabiliz). Edonola ere, garraio-mailak IP zerbitzua erabiltzen du bere lana egiteko.

Arkitekturan zeharreko bidaia osoa

Ondoko irudian ditugu TCP/IP arkitekturako osagai guztiak: haren mailak eta dagozkion protokolo nagusiak. Maila bakoitzak eskaintzen dituen zerbitzuak aurreko paragrafoetan deskribatu ditugu, nahiko sinplifikatuta.



1.5. irudia. TCP/IP arkitektura.

Irudian bi motatako konputagailu hauek agertzen dira: erabiltzailearen konputagailuak —komunikazioaren bi muturretan kokatzen direnak— eta bideratzaileak —tartean daudenak—. Irudian ikusten denez, bideratzaileetan ez dago garraio-mailako entitaterik, ezta aplikazio-mailakorik ere. Gabetasun horren zergatia laster jakingo dugu.

Ikus dezagun, era sinplifikatu batean, arkitekturako osagai bakoitzaren zeregina eta beraien arteko elkarrekintza bidalketa batean. Ondokoa da prozesua, urratsez urrats:

- Demagun A makinan dugula aplikazio banatu baten bezeroa (aplikazioko entitatea). Eta demagun bezero horren interfazea erabiliz (adibidez, pantailan klik batzuk eginez), aplikazioaren erabiltzaileak (ziur aski, pertsona batek) zerbitzu bat eskatzen duela (adibidez, B makinan dagoen fitxategi bat ekartzea).
- Bezeroak, erabiltzailearen eskaera osatzen duten datuekin, eta aplikazio-mailako protokoloari jarraituz, komando bat sortuko du. Komando hori da aplikazio-mailako informazio-unitate bat. Gogoan izan aplikazio-mailako protokoloak definitzen duela nolakoak diren komandoak eta erantzunak,

edo, beste era formalago batean esanda, nolakoak diren aplikazio-mailako informazio-unitateak.

- Bezeroak, garraio-mailako interfazea erabiliz, A makinako TCP/UDP entitateari helaraziko dio komandoa. TCP/UDP entitatea (edo garraio-mailako entitatea, izen hori nahiago baduzu) sistema eragilearen zati bat da, eta berarekin komunikatzeko gehien erabiltzen den interfazea *socket* izenekoa da.
- TCP/UDP entitateak bezeroak helarazitako komandoari garraio-mailako informazioa gehituko dio, komandoa B makinan dagoen zerbitzariraino ailegatuko dela (TCP zerbitzua erabiltzen bada soilik) bermatzeko, eta zerbitzari hori identifikatzeko B makinan lanean ari diren zerbitzari guztien artean. Horrela garraio-mailako informazio-unitatea sortuko du. Informazio-unitate horren izena **segmentua** da. Garraio-mailak garraio-mailako informazio-unitateetan (batzuetan batean baino gehiagotan) biltzen du komandoa, garraio-mailako **goiburukoak** gehituz. Nolakoak diren goiburuko horiek, erabilitako protokoloak (TCPk edo UDPk) definituko du.
- TCP/UDP entitateak IP mailako zerbitzua erabiliko du sortutako segmentua B makinako TCP/UDP entitateari helarazteko. Hau da, segmentua A makinako IP entitateari emango dio. IP entitate hori ere sistema eragilearen zati bat da. Bi entitateen arteko komunikazioa IP mailako interfazearen bidez egingo da. Interfaze hori sistema eragilearen araberakoa izango da; azken finean sistema eragilearen parte diren bi programen arteko komunikazioa besterik ez da.
- IP entitateak, bideratzaileen lana ahalbidetzeko, segmentuari beste goiburuko propio bat erantsiko dio, IP goiburukoa. Horrela sortuko du **datagrama** izeneko informazio-unitatea, IP mailari dagokiona, IP protokoloak definitzen duena. Beraz, segmentua datagrama batean (edo askotan) kapsulatzen du.
- A makinako IP entitateak ebatzi behar du zein den hurrengo bideratzailea datagrama horren B makinarako bidean. Horretarako bere bideratze-taulak erabiliko ditu. Gero, bere sare-txartelaren driverrari emango dio datagrama IP entitateak. Hau da, sarbide-mailako entitateari emango dio datagrama, eta esango dio bere sareko zein konputagailuri (gogoratu, bideko lehenengo bideratzaileari) igorri behar dion emandakoa.
- Driverrak sarbide-mailako informazio-unitatea sortuko du, datagramari beste goiburuko bat gehituz. Aukeratutako txartela Ethernet bada, **trama** izeneko unitatea sortuko du, edo, beste era batean esanda, datagrama trama batean (edo askotan) kapsulatuko du. Erabilitako sarearen araberakoak izango dira unitate horren egitura eta izena: Frame Relay-n edo PPPn, Ethernet-en bezala, trama du izena, baina ATMn zelula deitzen diote, eta X.25 zaharretan paketea. Sare bakoitzak bere protokoloa du, eta protokolo

horrek definitzen du nolakoak izan behar duten igorritako informazio-unitateek.

- Azkenean, A makinako sare-txarteleraino ailegatuko da bidali behar den trama bat (Ethernet txartela dela suposatuz). Txartelak sarean ipiniko du trama, bitez bit, kablean edo uhinetan, eta sarea arduratuko da trama hori lehenengo bideratzaileraino heltzeaz (1.5. irudian A makinaren eskuinean agertzen den konputagailua).
- Lehenengo bideratzailearen sarbide-mailak (driverrak) trama jasoko du, Ethernet goiburukoa kenduko dio (deskapsulatu), eta bere IP entitateari emango dio.
- Lehenengo bideratzailearen IP entitateak datagramaren goiburukoa aztertuko du, eta zein den bideko hurrengo bideratzailea ebatziko du. Gero, bideratzaile horretaraino ailegatzeko bere sare-txartelen artean (bideratzaile bat sare batera baino gehiagotara egongo baita konektatuta) datagrama zeinetik birbidali behar duen erabakiko du, eta txartel horren driverrari emango dio datagrama, zein bideratzaileri igorri behar dion adieraziz. Ikusten duzunez, bideratzailean ez da garraio-mailarik edota aplikazio-mailarik ezertarako behar[4].
- Driverrak sarbide-mailako informazio-unitate berri bat sortuko du (agian beste Ethernet trama bat), eta txartelari pasatuko dio. Horrek bidaliko dio bigarren bideratzaileari.
- Bigarren bideratzailean, azken bi urratsak errepikatuko dira, eta, horrela, bideratzailez bideratzaile, datagrama bidaiako azken bideratzaileraino ailegatuko da. Horrek B konputagailua zuzenean atzigarria duela atzemango du, eta dagokion txartelaren bidez igorriko dio datagrama.
- B makinako sarbide-entitateak, bideko bideratzaile guztietan gertatu den bezala, bere sarbide-mailako informazio-unitatetik datagrama erauziko du, eta IP entitateari emango dio. IP entitateak, datagramaren goiburukoa aztertuz, jakingo du nori eman behar dion datagramaren barruan dagoena (TCPri edo UDPri[5]). Azkenik, garraio-mailako entitateak, segmentua edo mezuaren goiburukoa miatuta, argituko du zein zerbitzariri eman behar dion barrukoa, eta, dagokion interfazea erabiliz (gehienetan, *socket* interfazea), emango dio.

4. Hau ez da guztiz egia. Birbidali behar duen IP datagrama bat prozesatzeko, bideratzaileak ez du aplikazio- edo garraio-mailarik behar. Baina bere zereginak betetzeko, bideratzaileak aplikazio-mailako protokolo batzuk erabiltzen ditu, eta, berez, garraio-mailakoak ere. Adibidez, bideratze-taula dinamikoki eguneratzeko erabiltzen diren protokolo batzuek TCP erabiltzen dute.

5. Errealitatean, aukera gehiago daude.

- Orain, zerbitzariak jasotako komandoa aztertuko du, eta dagokion erantzuna prestatuko du. Erantzun hori A makinan dagoen bezeroraino helarazteko, prozesu guztia errepikatuko da, baina orain kontrako noranzkoan.

Azaldutako TCP/IP arkitekturaren ezaugarrien laburpena hurrengo orrialdean dagoen taulan duzu.

Liburuaren hurrengo kapituluetan zehatzago adieraziko dugu zer egiten den arkitekturako goiko hiru mailetan: 2. kapituluan IP maila aztertuko dugu; 3. kapituluan garraio-maila, eta 4. kapituluan TCP/IP arkitekturan funtsezkoak diren aplikazio batzuk.

LABURPENA

Internet erabiltzen dugunean, hiru sare mota agertzen dira informazioak egingo duen bidean: Interneti lotutako bertako sareak (jatorrizkoa eta helburukoa), Interneterako sarbide-sareak, eta Internet hornitzaileen sareak (ISPren sareak). Hauek handizkariak edo txikizkariak izan daitezke. Txikizkariek erabiltzaileei ematen diete Interneterako sarrera, eta Internet osorako konexioa handizkariei kontratatzen diete. Handizkariek txikizkarien sareen arteko interkomunikazioa bermatzen dute. Bi motatako handizkariak bereizten dira, Tier2 eta Tier1. Tier2 erakoek txikizkariei ematen diete zerbitzua, eta konexio bat dute, gutxienez, Tier1 batekin. Tier1 sareek Interneten ardatza, edo *backbone*, osatzen dute.

Sarearte batean zehar informazioa konputagailu batetik bestera mugitzeko mota desberdinetako arazo asko konpondu behar dira. Horregatik, *divide et impera* printzipioa erabiltzen da, eta arazo txikiagotan zatitzen da komunikazio hori gauzatzea den arazo konplexua. Horrela eginda, komunikazioa maila desberdinetan egituratzen da, non maila bakoitzeko arazoa konpontzeko azpiko mailak ematen duen zerbitzua erabiltzen den. Maila bakoitzeko entitateek elkarrekin komunikatzen dira maila horren protokoloak erabiliz, goiko mailari eman behar dioten zerbitzua gauzatzeko. Maila bateko entitate batek beheko mailako entitateen zerbitzuak eskuratzen ditu mailen arteko interfazea erabiliz. Mailen definizioak, maila bakoitzaren zerbitzuen definizioak, eta maila bakoitzaren protokoloek osatzen dute sare-arkitektura baten definizioa.

Interneten erabiltzen den sare-arkitektura TCP/IP protokolo multzoa erabiltzen duena da. Ondoko lau mailek, goitik behera, osatzen dute arkitektura hori: aplikazioa, garraioa, sareartea (edo IP maila), eta sarbide-maila. Liburu honetan aurreko hirurak aztertuko ditugu, azkena ez baita sartzen TCP/IP protokolo multzoan.

Maila	*Zerbitzua*	*Protokoloa*	*Informazio-unitatea*	*Goiburukoaren eduki nagusia*	*Entitatearen inplementazioa*	*Zerbitzurako interfazea*
Aplikazioa	Aplikazioaren araberakoa (informazioa jasotzea, mezuak trukatzea, izen-helbide itzulpena…)	Aplikazioaren araberakoa (HTTP, SMTP, POP, DNS…)	Protokoloaren araberakoa (komandoa eta erantzuna gehienetan)	Aplikazioaren araberakoa (erabiltzailearen identifikadorea, mezuaren edukia…	Erabiltzailearen programak: bezeroak eta zerbitzariak, gehienetan C-z edo bere eratorrietako batez (Java, Perl…) idatzirik	Inplementazioaren eta erabiltzailearen (gizakiak ala programak) araberakoa
Garraioa	Komandoak eta erantzunak aplikazio-entitate batetik bestera eramatea	TCP/UDP	TCP segmentuak, UDP datagramak	TCP: konexioa kontrolatzeko informazioa eta portuak UDP: portuak	SEren zatia. Gehienetan, C-z idatzirik	SEren araberakoa. Gehienetan, socket-ak
Sareartea	Sarearteko makina batetik bestera informazioa eramatea	IP	Datagramak	@IP	SEren zatia. Gehienetan, C-z idatzirik	SEren inplementazioaren araberakoa
Sarbidea	Sare bereko makina batetik bestera informazioa eramatea	Sarbide-sarearen araberakoa: PPP, Ethernet, ATM, WiFi…	Tramak, zelulak, paketeak…	@fisikoa, errroreak kontrolatzeko bitak	Hardwarea/ firmwarea	Sare-teknologia eta inplementazioaren araberakoa

1.1. taula. TCP/IP arkitekturaren laburpena.

2. IP sareak

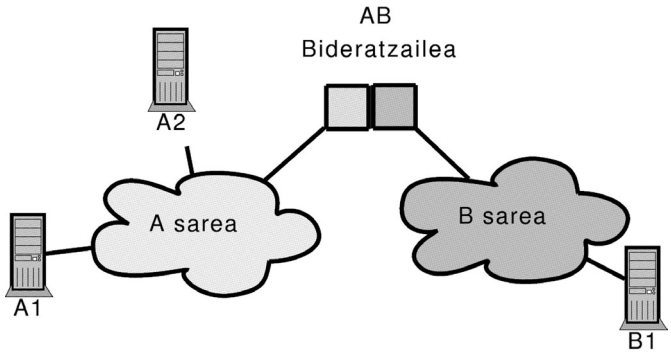
Kapitulu honetan honako gai hauek aztertuko dira:

- Nolakoak diren IP zerbitzua eta protokoloa.
- Nolakoak diren eta nola kudeatzen diren IPv4 helbideak.
- Nola bideratzen diren datagramak sareartean zehar: nolakoak diren bideratze-taulak eta ARP taulak, nola betetzen diren, nola eguneratzen diren, eta nola erabiltzen diren.
- Zeintzuk diren Interneten dauden bideratzeko arazo nagusiak.
- Zer den IPv6, eta zertan hobetzen duen IPv4.

Bi konputagailu sare bakar baten bidez komunikatzen ditugunean (adibidez, bi konputagailu Ethernet sare berean daudenean), nahikoa da sarbide-mailak ematen digun zerbitzua informazioa konputagailu batetik bestera eramateko. Baina Internet guztiz desberdinak izan daitezkeen sareak elkarturik dago eginda. Horrelako sare heterogeneoen arteko komunikazioak arazoak sortzen ditu. Kapitulu honetan ikusiko dugu zeintzuk diren arazo horiek, eta nola konpondu diren TCP/IP arkitekturan.

2.1. SAREARTE-MAILAREN BEHARRA

Demagun komando bat bidali nahi dugula 2.1. irudiko A1 konputagailutik A2 konputagailura. Biak A motako sare berean daudenez, egin behar dena, oso laburrean, komandoa A motako trama baten informaziorako eremuan jarri, trama horren goiburuko helburuko helbidean A2 konputagailuarena idatzi, eta trama hori sareari eman. Sareak eramango du trama hori A2 konputagailuraino.



2.1. irudia. Bi sarek osatutako sarearte xume baten adibidea. Sare bakoitza teknologia desberdinekoa da. Horregatik, tartean dagoen bideratzaileak bi *nortasun* izango ditu, edo, hobeto esanda, bi sarbide-maila izango ditu. Hau da, bi sare-txartel eta dagozkion driverrak izango ditu.

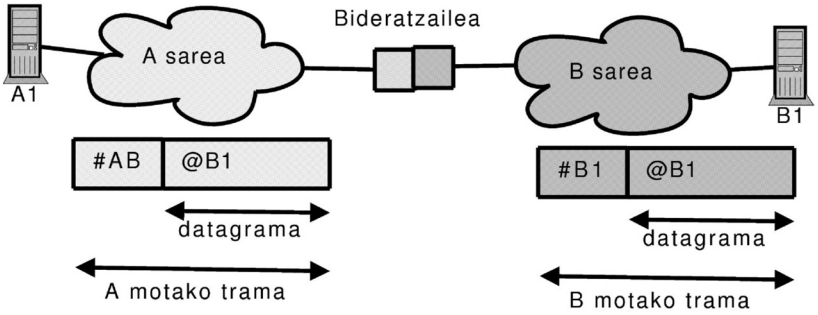
Demagun orain A2 konputagailuaren ordez, B1 konputagailua dela komandoaren helburua. Orain igorlea eta hartzailea ez daude sare berean, baina bai interkonektatuta dauden bi sareetan. Horrela izanik, bidalketa zertxobait konplikatuagoa da. Honako hauek dira agertuko diren arazoak eta, oso laburrean, beraien irtenbideak:

- A1-ek ezin dio ezer bidali B1-i zuzenean, bere sarean ez dagoelako. A1-ek ebatzi beharko du zein den sareartean zehar jarraitu behar den bidea B1-eraino heltzeko. Horri **bideratzea** deitzen diogu. Arazo hau sare baten barnean ere agertzen da, sare hori kommutatua baldin bada. Adibidez, irudiko A edo B sareak kommutagailu zentral batean oinarritutako Ethernet sareak badira, kommutagailu (edo *switch*) horrek ere ebatzi behar du zein lineatatik birbidali behar duen jasotzen duen trama bakoitza. Baina, gehienetan, sarbide-mailan egiten den sare barruko bideratze horri kommutazioa deitzen zaio (edo, batzuetan, 2. mailako bideratzea, OSI ereduko arkitektura erabiliz), bideratzea terminoa sarearterako utziz.
- Informazioa bideratzeko abiapuntua helburuko helbidea da. Helbide hori aztertuta, igorleak ebatziko du nondik joan behar duen informazioa. Konputagailu guztiek badute helbide bat esleituta sare batean konektatzen ditugunean; hori da A1 eta A2-ren arteko aurreko komunikazioan erabili dugun A2-ren helbidearen kasua, hain zuzen ere. Helbide horiek konputagailua identifikatzeko balio dute, baina bere sarean bakarrik. Hau da, gure sare berean dagoen konputagailu batek bidal diezaguke trama bat, trama horretan gure helbidea ipiniz helburu gisa. Baina B1 eta A1 sare berean ez daudenez, zein helburuko helbide jarriko du A1-ek igorri behar duen A motako traman? Kontura gaitezen A motako tramen helburuko eremuan ezin direla idatzi B motako helbideak. Sareko helbideak lokalak dira, eta guk helbide

globalak, sareartean erabiltzekoak, behar ditugu. Beraz, konputagailuak sareartean identifikatzeko **helbideratze-eskema globala** behar dugu. Sarearte batean kokaturiko konputagailuek bi helbide izango dituzte: helbide lokala eta helbide globala. Gaur egun, Interneten, helbide lokalari **helbide fisikoa** deitzen zaio, eta helbide globalari sarearteko helbidea, edo, gero ikusiko dugunez, IP helbidea. Helbide fisikoak adierazteko, # karakterea erabiliko dugu testu honetan (beste hainbat testutan egiten den moduan), eta sarearteko helbideak @ karakterearekin bereiziko ditugu (idem). Beraz, B1 konputagailuaren helbide fisikoa #B1 idatziko dugu, eta bere sarearteko helbidea, @B1.

- Argi dabilen irakurlea ohartuko zenez, A motako trametan B motako helbideak jartzerik ez dagoen arrazoi beragatik, helbide globalak ere ezin dira trama horietan idatzi. Irtenbide bakarra trametarako ere formatu globala erabiltzea da, eta formatu horren goiburukoan helbide globalak jartzea. Sare-teknologia globalik ez dagoenez, trama global horiek tokiko sareformatuko trametan sartuko ditugu, hau da, trama lokalen informaziorako eremuan garraiatuko ditugu trama globalak. Azken finean, trama eta helbide globalak definitzean, protokolo global bat, edo **sarearteko protokolo bat** ari gara definitzen. Protokolo horrek ahalbidetuko digu konputagailuak sarearteen bidez komunikatzea, sarbideko protokoloek sare bereko konputagailuekin komunikatzea ahalbidetzen diguten era berean. Interneten, sarearteko protokoloaren informazio-unitateei ez zaie trama deitzen, **datagrama** baizik.
- Azkenik, helbide fisikoen eta globalen arteko erlazioa ezarri behar da. Bestela, nola jakingo du AB bideratzaileak zein helburuko helbide fisiko jarri behar duen A1-ek B1-i bidalitako datagrama garraiatuko duen B motako traman? Nolabait, bideratzaile horrek **helbide-itzulpena** egin beharko du, datagraman ikusi duen @B1 helbidetik #B1 helbidea lortzeko. 2.2. irudian tramen eta datagramen arteko erlazioa azaltzen da, baita helbide fisikoen eta globalen artekoa ere.

Beraz, sarearteko protokolo bat eta sarearteko zerbitzu bat definitu ditugu, eta, konturatu gabe, sarearte izeneko maila sortu dugu. Sarearte-mailak ematen digun zerbitzua datagramak konputagailu batetik bestera *sareartean zehar* eramatea da. Horretarako, sarearte-mailako entitateek bideratzea eta helbide-itzulpena egin beharko dute, eta sarearteko protokoloa erabiliko dute bidalketak egiteko.



2.2. irudia. Datagramak, tramak, helbide globalak eta helbide fisikoak A1-etik B1-era egindako bidalketa batean.

2.2. INTERNETEKO SAREARTE-MAILA: IP PROTOKOLOA

TCP/IP arkitekturako sarearte-mailan erabiltzen den protokoloa IP da (Internet Protocol). Atal honetan IP protokoloaren ezaugarri nagusiak ezagutuko ditugu. Hurrengoetan protokoloak definitzen dituen helbideak eta datagramak nola bideratzen diren aztertuko dugu.

2.2.1. IP zerbitzua

Sarearte-mailak garraio-mailari ematen dion zerbitzua informazioa sarearte batean zehar garraiatzea da. Hori da IP protokoloaren lana. Oro har, sare-arkitekturako maila guztietan, protokolo batek gauzatutako zerbitzua konexioaren bidezkoa ala konexiorik gabekoa izango da. Izaera horrek asko baldintza ditzake zerbitzuaren ezaugarriak, eta, IPren kasuan, baldintzatzen ditu. IP zerbitzua konexioduna ala konexiorik gabekoa den argitu baino lehen, ikus dezagun zertan datzan aukera bakoitza.

Konexioen bidezko zerbitzua bada, bi sarearteko entitateen artean datagramak[6] bidali baino lehen, bien arteko konexioa ezarri behar da. Konexioa ezartzea komunikazio horri dagozkion ezaugarriak bi muturren artean hitzartzea da. Adibidez, datagramek jarraituko duten bidea ezartzeko, datagrama horien tamaina maximoa mugatzeko, edo beste aldeari komunikazio-eskaerari uko egiteko aukera emateko besterik ez.

Konexioen bidezko zerbitzuak duen abantailarik behinena hau da: sareartean gertatzen dena kontrola daiteke eta, ondorioz, komunikazioaren kalitatea errazago berma daiteke. Honi, ingelesez, QoS deitzen diote (Quality of Service). Zehatzmehatz, konexioen bidezko komunikazioetan eragiketa hauek egin daitezke:

6. Teorian, *datagrama* hitza konexiorik gabeko zerbitzuekin soilik erabiltzen da. Konexio bidezko zerbitzuetarako *pakete* hitza erabili izan da, baina hemen ez dugu erabiliko, terminologia gehiago ez nahastearren.

- Aurrez ezar daiteke komunikazioari dagozkion datagrama guztiek sareartean zehar egingo duten ibilbidea. Horrek sareartearen trafiko-kudeaketa asko errazten du, kongestioak aurreikusi eta bideratzaileen arteko trafiko-zama orekatu daitekeelako.
- Datagramak zein bideratzailetan ibiliko diren jakinda, posible da bideratzaile horietan baliabide-erreserbak egitea (lineak eta tokia ilaretan) eta, horrekin batera, datagramek beren bidaian zenbat denbora emango duten aurreikustea.
- Konexio bati dagokion datagramaren baten galera atzeman daiteke, eta datagrama birtransmititu. Sareartean dagoen sareren batek errore-zuzenketa egiten ez badu (Ethernet sareek, adibidez, ez dute egiten), sarearte-mailak egingo du. Edota kongestioak ezin badira guztiz saihestu, bideratzaileen pilaketetan galdutako datagramak sarearte-mailak errekupera ditzake, berriro ere birtransmisioen bidez.

Horiek dira alde onak. Konexioen bidezko komunikazioaren kontrako aldeak honako hauek dira:

- Errore- eta kongestio-kontrolak egiten badira, bideratzailearen lana zaildu egiten da. Datagrama bakoitzaren prozesamendua luzeagoa eta konplexuagoa izango da, bideratzaileak denbora gehiago emango du datagrama bakoitzarekin, eta, birtransmititzea badago, bufferra luzeagoan hartuko du datagramak (bere transmisioaren onespena jaso arte). Gaur egungo Interneten baliabide kritikoa bideratzaileak baldin badira, horien lana zailagoa egiteak ez dirudi oso ideia ona.
- Gainera, bideratzaileetan kontrol batzuk ezartzeak lanak bikoiztea edo behar ez diren lanak egitea ekar dezake. Adibidez, bidean zeharkatzen ditugun sare guztiek errore-kontrola egiten badute. Edo datagramek egiten duten ibilbidean galera-tasa oso txikia suertatzen bada, beharbada hobe litzateke gainean dagoen mailan errore-kontrol xumea egitea. Halaber, aplikazio batzuek ez dute inongo QoSrik behar, edo behar duten kalitatea beste mota batekoa da. Denbora errealekoek, hain zuzen ere, behar dutena ez da datagramak bere helburura ondo ailegatu direla bermatzea, datagramak ahal bezain laster ailegatuko direla segurtatzea baizik, nahiz eta arintasun horren truke bidean datagrama batzuk galdu.
- Zirkuitu-kommutazioan gertatzen zen bezala, konexioen bidezko komunikazioetan baliabideak erreserbatzen badira, baliabide horien ustiaketa ez da hobezina izango. Konexio batzuen kalitatea bermatzeko beste konexio batzuei uko egin beharko zaie, nahiz eta, benetan, bideratzaileetan denentzako tokia egon une horretan.

Konexiorik gabeko zerbitzuen[7] ezaugarriak kontrakoak dira. Konputagailu batek beste batera informazioa bidali nahi duenean, ez du berarekin inongo kontakturik egin behar aurretik: zuzenean datagramak sortu eta bidali. Datagrama bakoitza bere sarearteko zeharbidea egiten saiatuko da, komunikazio berari dagozkion beste datagramekiko inongo erlaziorik gabe. Nolabait, datagramek beren burua aski dute beren sareartean zeharreko bidaian.

Datagrama-sarearteen arazoak hauek dira:

- Ez dago QoS bermatzerik. Ezin da jakin datagrama bat bere helburura ailegatuko den ala ez, ezta horretarako zenbat denbora beharko duen ere. Sare hauei *best effort* sareak ere deitzen zaie (euskaraz, *ahal den hoberena* sareak). Ez da ezer bermatzen, baina horrek ez du esan nahi kalitate eskaseko zerbitzua emango denik. Berez, zerbitzuaren kalitatea sareartearen egoeraren eta egin behar den ibilbidearen araberakoa da.
- Sareartearen kontrola askoz zailagoa da. Ez dago kongestioak aurreikusterik ezta trafiko-zama orekatzerik ere.

Eta abantailak honako hauek dira:

- Sare-baliabideak hobeto ustiatzen dira, kasu hobezinetik gertuago, erreserba egiterik ez dagoelako.
- Bideratzaileen lana bideratzea besterik ez denez, hori azkarrago egitea badute. Ekipo sinpleagoak izango dira.

2.1. taulak laburbiltzen du bi zerbitzu horiek sarearteetan zer duten alde eta zer kontra:

	Konexioen bidezko zerbitzua	**Konexiorik gabeko zerbitzua**
Alde	Sareartearen kudeaketa errazten da Badago QoS bermatzea	Baliabideen ustiaketa hobea da Bideratzaileen lana errazten da
Kontra	Lana alferrik egin daiteke bideratzaileetan Bideratzaileen lana zailtzen da Baliabideen ustiaketa txarra izan daiteke	Ez dago QoS bermatzerik. Zailagoa da sareartearen kudeaketa

2.1. taula. Konexioen bidezko zerbitzua eta datagrama-zerbitzua.

IP protokoloak gauzatzen duen zerbitzua konexiorik gabekoa da. Horrek honako ondorio hauek dakartza:

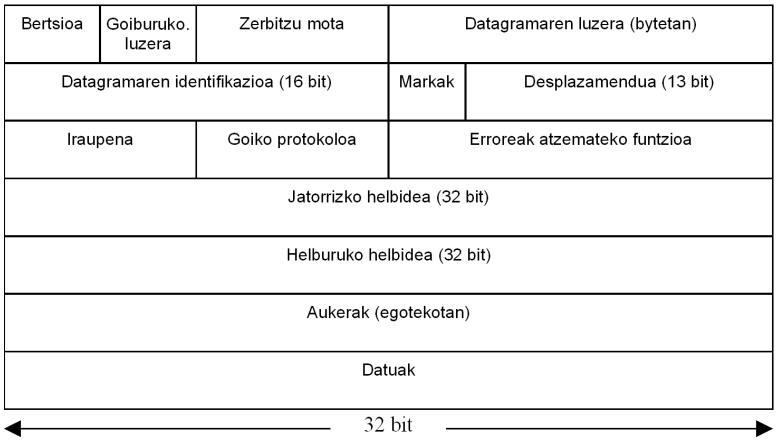
7. Askotan, *datagrama-zerbitzua* ere deitzen zaie konexiorik gabeko zerbitzuei.

1. IPk ez die ezer exijitzen sareartean konekta daitezkeen sareei. IPk goiko mailari ezer bermatu behar ez dionez, bere azpian egongo diren sarbide-mailei (sare-teknologiei, alegia) datagramak sare baten konputagailu batetik bestera eramatea besterik ez die eskatuko. Hau izan da TCP/IP protokolo multzoaren arrakastaren gako nagusietako bat: edozein sare-teknologia erabil daiteke TCP/IP sarearte batean. Horregatik Interneten ez da inongo estandarrik definitzen sarbide-mailarako, eta horregatik testu honetan ez dago sare-teknologiez diharduen kapitulurik.
2. Bideratzaileen lana errazten da. Datagrama bakoitzaren tratamendua arinagoa da eta, beraz, datagrama horrek denbora gutxiago emango du bideratzailearen bufferretan.
3. IP sareek (eta, beraz, Internetek) ez dute QoS bermatzen.

2.2.2. IPv4 datagramaren formatua

Gaur egungo Interneten gehien erabiltzen den IP protokoloaren bertsioa laugarrena da. Horretan jarriko dugu arreta hurrengo ataletan. Gero, zabaltzen ari den 6. bertsioa aztertuko dugu.

Datagrama-zerbitzu batean ez da errore-kontrolik egin behar. Horrek asko sinplifikatzen ditu transmititzeko prozedurak, baita datu-unitatearen formatua ere (datagramarena, alegia). IPrena bezain sinplea den zerbitzua emateko, datagramaren goiburukoan behar dugun eremu bakarra helburuko helbidea dela dirudi, datagramen prozesamendua bideratzaileetan bideratzea besterik ez denez. Haatik, IPren goiburukoan askoz eremu gehiago agertzen dira, haren formatua adierazten duen 2.3. irudian ikus daitekeen legez.



2.3. irudia. IPv4 datagramaren formatua. Zerbitzu-mota (Type of Service) izeneko eremua bi zatitan (Differentiated Services eta Explicit Congestion Notification eremuak) banatzeko proposamena ez da agertzen, proposamen hori ez baita oraindik estandarra estatuseraino ailegatu.

Formatuaren azterketarekin batera, protokoloaren beraren ezaugarriak ezagutuko ditugu. Irudian agertzen diren eremu batzuk beharrezkoak dira; beste batzuk, aldiz, ez. IP protokoloa diseinatu zenean guztiz esperimentala zen (70eko hamarkadaren hasieran), eta horregatik agertzen dira beharrezkoak ez diren eremu batzuk. Gero, protokoloaren erabilera hedatu egin da harik eta sareen arteko elkarrekiko konexio unibertsalerako tresna bihurtu arte, baina bere «haurtzaroaren» arrastoak hor gelditu dira. IP datagramaren formatuaren azterketa eremuen garrantziaren araberako ordenan egingo dugu.

Helburuko helbidea

Eremu hau agertzea ezinbestekoa da edozein sarearte-mailako protokolotan. Ematen duten informazioa nahitaezkoa da sarearte-mailako zerbitzua betetzeko, hau da, datagramak sareartearen mutur batetik beste bateraino helarazteko. Jatorrizko makinan eta bideko bideratzaileetan egindako datagramaren prozesamendua helbide honetan datza. Haren balioa aztertuta ebatziko dute konputagailu horiek nondik bideratu behar duten datagrama. Geroko atal batean aztertuko dugu sakonean nolakoak diren IP helbideak.

Beharrezkoak diren eremuak

Honako eremu hauek ez daude zuzenean lotuta datagramak bideratzearekin, baina, hala eta guztiz ere, daramaten informazioa beharrezkoa da zerbitzua osatzeko.

- Goiko protokoloa. Eremu hau helmugako konputagailuak behar du, eta ez bideko bideratzaileek. Helmugako IP entitateak datagrama nori eman behar dion jakiteko ezinbestekoa da. Hasiera batean, IP mailaren erabiltzailea garraio-maila denez, badirudi argi dagoela nori eman behar zaion: helburuko konputagailuko garraio-mailako entitateari. Baina hurrengo kapituluan ikusiko dugunez, TCP/IP arkitekturako garraio-mailako entitate bat baino gehiago aurkituko ditugu helburuko konputagailuan. Gainera, gerta daiteke IP mailaren erabiltzailea garraio-mailako entitate bat ez izatea. Adibidez, aplikazio batzuek IP maila erabiltzen dute zuzenean, garraio-mailako zerbitzuak erabili gabe. Baita IP maila berean kokatzen diren beste entitate batzuek ere IP zerbitzuak erabiliko dituzte (adibidez, kapitulu honetan aztertuko dugun ICMP protokoloak IP erabiltzen du bere bidalketetarako). Erabiltzaile posible horien guztien artean datagramaren helburua zein den asmatzeko behar dugu *protocol* izeneko eremu hau IP goiburukoan.
- Bertsioa. Honek datagramak nolako goiburukoa duen adierazten du, bertsio ezberdinek datagrama ezberdinak erabiltzen baitituzte. Hau da, eremu honek goiburukoa irakurtzeko gakoa ematen die datagrama aztertu behar duten programei. Horregatik da goiburukoaren lehenengo eremua.

- Jatorrizko helbidea. Irudian ikusten denez, helburuko helbidea ez ezik, jatorrizkoa ere datagramaren goiburukoan dago. Bideratzaileek ez dute jatorrizko helbide hori bideratzeko behar, nahikoa baitute helburukoarekin. Baina datagrama jasoko duenak, normalki, erantzuna eman beharko dio datagramaren igorleari. Horretarako behar da jatorrizko makinaren helbidea igorritako datagrametan.
- Goiburukoaren luzera. Datagrama prozesatu behar duten IP entitateek (bideko bideratzaileenak eta helburuko konputagailuarena) jakin behar dute goiburukoa non bukatzen den eta garraiatutako datuak non hasten diren. Goiburukoan aukerazko eremu batzuk daudenez, haren luzera ez da finkoa, eta eremu hau beharrezkoa suertatzen da.
- Datagramaren luzera. Goiburukoaren luzera ez ezik, datu-eremuarena ere ez dago finkatuta. Hasiera batean, datagrama osoaren luzera jakitea arkitekturako mailen arteko interfazearen kontua da, eta ez luke goiburukoan agertu behar. Hau da, sarbide-mailako entitateak kontrolatzen du zenbat byte erauzten duen tramatik, eta datu hori ematen dio IP mailako entitateari datagramarekin batera (nolabait «tori datagrama hau, hainbat bytetakoa» esango dio). Hala ere, sarbideko protokolo batzuek zaborra gehitzen diote transmititzeko ematen dieten datagramari. Hori da jatorrizko difusioko Ethernet sareen kasua, non talkak atzemango direla bermatzearren, tramek luzera minimo bat izan behar duten. Traman sartu behar den datagramak luzera minimo hori ez badu, zabor betegarria sartzen da tramaren informazio-eremuan. Gero, helburuan, datagramari itsatsita datorren betegarria bereizteko, datagramaren goiburukoan dugun *luzera* izeneko eremua erabiliko du IP entitateak.

  Luzera eremuan 16 bit daudenez, eta bytetan neurtzen denez, datagramarik handiena 65.536 bytekoa izan daiteke (datuak gehi goiburukoa). Dena dela, oso arraroa da 1.500 byte baina handiagoa den datagrama bat aurkitzea (hori da Ethernet sare batean sartzen den datagramarik handiena), eta sistema askok 576 bytera mugatzen dute datagramaren tamaina (eremu zabaleko sare askok onartzen duten tamaina maximoa).
- Iraupena edo TTL (ingelesetik: Time To Live). Eremu honi balio bat ematen zaio jatorrizko konputagailuan, eta bideko bideratzaile bakoitzak 1 kentzen dio, gutxienez; eremuaren balioa 0-raino heltzen bada, bideratzaileak datagrama ezabatuko du, inora birbidali gabe. Mekanismo honen helburua da datagrama galduak edo oso atzeratuak saretik kentzea (adibidez, bideratze-errore bat badago eta datagramak begizta batean harrapatuta gelditzen badira). Beraz, sareko garbiketarako behar da eremu hau.

Datagrama berreraikitzeko eremuak

IP datagrama bat sareko pakete batean sartzeko handiegia baldin bada, zatitu egin behar da. Datu-zatirik txikiena 8 bytekoa izan daiteke; beraz, IP sareartean sar daitezkeen sare guztiek 28 byteko datagramak onartzeko gai izan behar dute (IP goiburukorik txikiena 20 bytekoa da). Praktikan, honek ez du batere murrizten konekta daitezkeen sareen multzoa. Zatiketa bideratzaileek egiten dute, baina helburuko konputagailuak berreraiki beharko du jatorrizko datagrama, zati guztiak jaso ondoren. Zatiketa-lana zama estra bat da bideratzaileentzat; horregatik, aukera hori kendu egin da IPren bertsio berrian, non zatiketa egitekotan jatorrizko konputagailuak egin behar duen, ez bideratzaileek. IPv4 goiburukoari honako 3 eremu hauek erantsi behar izan zaizkio zatiketagatik, helburuko konputagailuari datagrama berregiteko behar den informazio guztia emateko:

- Datagramaren identifikazioa. Zati guztiek jatorrizko datagramaren identifikazioa eramango dute. Horrela helburuko konputagailuak zati guztiak bil ditzake.
- Desplazamendua. Eremu honek zati honen kokapena jatorrizko datagraman adierazten du. Zatirik txikiena 8 bytekoa denez, horrela adierazten da desplazamendua, zortzi byteka. Eta horregatik eremu honek 13 bit behar ditu:

$$\frac{Datagramaren\ tamina\ maximoa(2^{16})}{Zatirik\ txikiena(2^{3})} = 2^{13}$$

- Bit-markak (edo *flagak*). Hiru dira, baina aurrenekoa ez da erabiltzen. Besteak *Ez zatitu* bita eta *Zati gehiago* bita dira —ingelesez, *Don't Fragment* (DF) eta *More Fragments* (MF)—. Batak bideratzaileei datagrama hori ezin dela zatitu jakinarazteko balio du (aplikazio batzuek horrela beharko dute). Besteak hori ez dela jatorrizko datagramari dagokion azkeneko zatia adierazten dio helburuko IP entitateari.

Beharrezkoak ez diren eremuak

Honako hauek dira:

- Erroreak atzemateko funtzioa. Goiburukoari bakarrik ezartzen zaion funtzio matematiko sinplea da. Datagramak bere bidean bisitatuko dituen bideratzaile guztiek TTL eremuaren balioa aldatuko dutenez, birkalkulatu beharko dute eremu hau. Praktikan, bideratzaileek ez liokete inongo kasurik egin behar eremu honi, zeren gaur egungo sare gehienek IPrena baino askoz indartsuagoak diren erroreak atzemateko funtzioak erabiltzen baitituzte (CRC funtzioak gehienetan) beren trametan, eta, gainera, datagramaren eremu guztiei aplikatzen zaizkie funtzio horiek (ez bakarrik goiburukoari). Beraz, eremu honi kasu egitea denbora galtzea da: txartelak ez lioke IP mailari matxuratuta dagoen datagrama bat pasatuko.

- Sailkapena. Eremu honek datagramen artean lehentasunak ezartzeko balio du. Erabiliz gero, bideratzaile batek badaki kongestio bat sortzen denean zeintzuk diren lehen ezabatu behar dituen datagramak. Egungo Interneten IP zerbitzua berdintasunezkoa da, hau da, ez da bereizten datagramen artean, eta, berez, eremu hau ez dute erabiltzen bideratzaile gehienek. Hala ere, denbora errealeko aplikazio interaktiboentzako (IP telefonia, kasu) oso baliagarria denez, eremu honen erabilera bultzatzen dutenak gero eta gehiago dira. Bere definizioa maiz aldatu da urteetan zehar. Hasiera batean (RFC 791), *Type Of Service* izeneko 8 biteko eremua definitu zen. Egun indarrean dagoen definizioan (RFC 2474), byte horren hasierako 6 bitak erabiltzen dira datagramak sailkatzeko, orain *Differentiated Services Code Point* izenpean.
- Buxadura-oharra (*Explicit Congestion Notification*). Garai bateko *Type Of Service* izeneko eremuko azkeneko 2 bitek osatzen dute 2001. urtean gehitutako eremu hau (RFC 3168). Erabiltzea hautazkoa da (oraingoz, behintzat). Bideratzaileek kasu egiten badiote, buxadura dagoela jakinarazteko balio du. Ohar honekin zer egin, garraio-mailako entitateek erabakiko dute (buxaduren arazoa hurrengo kapituluan, garraio-mailan, aztertuko dugu).
- Aukerak. Bideratzaile askok ez diote kasurik egiten. Protokoloaren ezaugarri berriak frogatzeko sartu zen eremu hau goiburukoan. Gaur egun aukera batzuk daude definituta. Adibidez, eremu honetan datagramak jarraitutako bidea adieraz daiteke (bideko bideratzaileak hori grabatzeko prest baldin badaude, noski).

2.3. ICMP PROTOKOLOA

IP protokoloak ematen duen zerbitzua datagrama erakoa denez, sareartearen kudeaketa zaila da, trafikoaren kontrol zehatza egiterik ez baitago. Hala eta guztiz ere, sareartean gertatzen denaren monitorizazioren bat egitea badago. Hau da, sarearteko bideratzaileetan gertatzen denaren berri jaso dezakegu, baldin eta bideratzaile horiek informazio hori emateko prest badaude. Adibidez, bideratzaile batek datagrama baten TTL balioa agortuta dagoela atzematen badu, zakarrontzira botako du; bideratzailea «jatorra» bada, datagrama igorri duenari bere bidaiaren amaieraren berri emango dio.

Horrelako komunikazioak gauzatzeko behinola ICMP protokoloa definitu zen (Internet Control Message Protocol, RFC 792). IPren protokolo laguntzaile bat da[8], sarearte-mailan kokatua. Honekin nahaste-borraste teorikoa sor daiteke, zeren, ICMP mezuak IP datagramen barruan, informazioaren eremuan sartzen baitira eta,

8. Berez, ICMP protokoloa IPren zatia dela ezartzen da RFC1812 eta RFC1122 agirietan.

ondorioz, ICMPk sarearte-mailatik gorako maila baten protokoloa dirudi. Hala ere, normalki, IPrekin batera inplementatu ohi da ICMP.

ICMP oso protokolo sinplea da: definitzen duen gauza bakarra mezuen formatua eta erabilera da (noiz bidali). Ez da inongo prozedurarik definitu behar mezu horiek bidaltzeko, datagrama batean sartu eta datagrama hori bidali besterik ez delako. Bideratzaile batzuek ez diete hartutako ICMP mezuei jaramonik egiten, eta haiek ere ez dute ICMP mezurik bidaltzen, baina portaera hori ez da ohikoena.

Bere erabileraren adibide bat *ping* programa ezaguna da, ICMPn oinarrituta baitago. Konputagailu bat sareartean zehar atzigarri dagoen ala ez jakiteko egiten dugun lehenengo gauza berari *ping* egitea da. *Ping*-ek adierazitako konputagailuari ICMP oihartzun-eskaera bat (*echo request* izeneko ICMP mezua) bidaltzen dio eta horren erantzunaren zain (*echo reply* ICMP mezua dena) gelditzen da. ICMPn oinarritutako beste programa ezagun bat *traceroute* da, konputagailu batetik bestera joateko datagramek egiten duten bidea ezagutzeko erabiltzen dena. Horrek «TTL agortuta» izeneko ICMP mezua erabiltzen du (*TTL expired*, ingelesez; ICMP mezuen izen «ofizialak» ingelesezkoak dira).

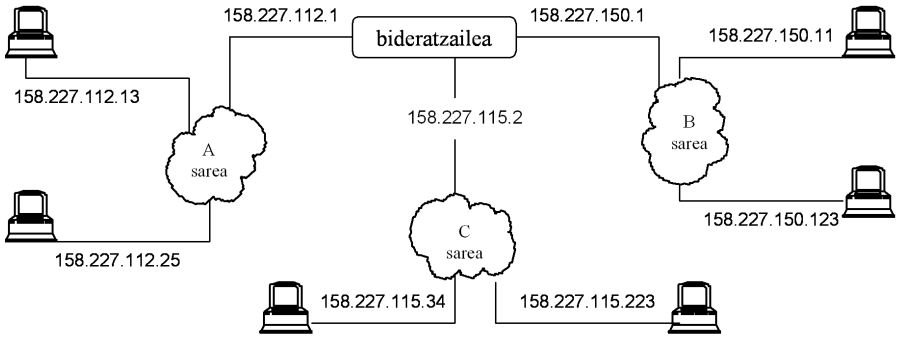
2.4. IPv4 HELBIDEAK

IP helbide guztiek 32 biteko luzera dute (hau da, 4 byte), beraz, 2^{32} IP helbide posible daude. IP helbideak **notazio hamartar puntudun**ez idazten dira. Formatu horretan, 4 byteetako bakoitza notazio hamartarrez idazten da, 0tik 255era. Adibidez, IP helbide tipiko bat 192.33.217.137 da, notazio horretan idatzita. 192 zenbakia helbidearen lehenengo 8 biten adierazpena da, era hamartarrean; 33, helbidearen bigarren 8ko bit sortaren adierazpena da era hamartarrean, etab. Horrela, 192.33.217.137 helbidearen idazkera bitarra ondoko hau da (hutsune batzuk sartu ditugu byteak ondo bereizteko):

11000000 00100001 11011001 10001001

Sare-txartel bakoitzak (IP hizkeran, **sare-interfaze** bakoitzak) bere IP helbidea behar du. Beraz, konputagailu batek dituen sare-lotura bezainbeste IP helbide izango ditu. Horregatik, bideratzaileek IP helbide bat baino gehiago izaten dituzte, eta erabiltzaileen makinek IP helbide bakarra izaten dute. Adibidez, 2.4. irudian erabiltzailearen makina batzuk eta bideratzaile bat agertzen dira, irudiko hiru sareak lotzen dituena.

Irudiari buruzko ohar batzuk aipatu behar dira. Lehenengoz, erabiltzailearen makina bakoitzak sare-interfaze bakarra du, eta irudian IP helbide bakarra esleitu zaio. Bideratzaileak, aldiz, hiru sare-interfaze ditu, bakoitza bere helbide propioarekin.



2.4. irudia. IP helbideak eta sare-interfazeak.

2.4.1. Helbideen egitura

Bigarrengoz, A sareari konektatuta dauden sare-txartel guztiek, bideratzailearenak barne, 158.227.112.xxx erako IP helbidea dute. Era berean, B sareari eta C sareari konektatuta dauden interfaze guztiek 158.227.150.xxx eta 158.227.115.xxx erako IP helbideak dituzte, hurrenez hurren. Horrek adierazten digu helbide bakoitzak bi zati dituela. Lehenengoak (aurreneko 3 byteak adibide honetan) sarea identifikatzen du; bigarrenak (azkeneko bytea adibide honetan) sareari konektatuta dagoen konputagailu bat helbideratzen du, edo, hobeto esanda, sare-interfaze bat helbideratzen du (gogoan izan konputagailu batek IP helbide asko izan ditzakeela, sare-interfaze beste). Hedatuta dagoen IP hizkeran sarearen identifikazioari **sare-helbidea** deitzen zaio, eta interfazeari dagokion zatiari **makinaren identifikazioa**.

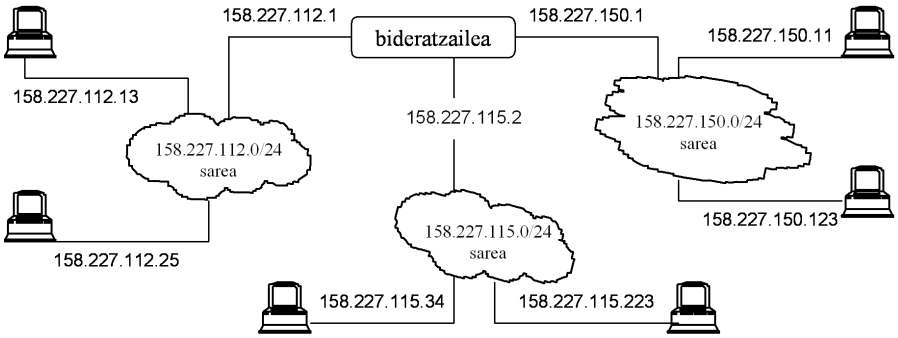
Beraz, IP helbide baten ezkerreko bitek sarea identifikatzen dute, eta eskuinekoek konputagailua (hobeto, sare-interfazea). Baina, zenbat bit esleitzen zaizkio zati bakoitzari? IP sare bakoitzak IP helbideen bit-banaketa berezkoa du. Banaketa hori **sare-maskara**ren bidez adierazten da. Sare-maskarak bi era ezberdinetan idazten dira. Lehen gehien erabili izan denak IP helbideen sintaxia du, non sareari dagozkion bitei 1 balioa esleitzen zaien eta interfazeari dagozkionak 0 diren. Adibidez, 2.4. irudiko C sareko maskara 11111111 11111111 11111111 00000000 da. IP helbideak bezala, era honetan adierazitako sare-maskarak notazio hamartar puntudunez idazten dira, eta aurrekoa 255.255.255.0 idatziko genuke.

Sare-maskarak adierazteko bigarren era laburragoa da: sarea identifikatzeko erabilitako bit kopurua sare-helbideari eransten zaio. Adibidez, irudiko C sarearen kasuan 158.227.115.0/24 idatziko dugu, sarea identifikatzeko ezkerreko 24 bit erabiltzen direla adieraziz. Bigarren notazio hau gero eta gehiago erabiltzen da, erosoagoa eta interpretatzeko errazagoa baita. Bigarren notazio honetan ere, helbideak sinplifika daitezke 0koak kenduz. Adibidez, aurreko helbidea

158.227.115/24 ere idatz daiteke. Notazio honekin batera, IP helbideen egituraketa izendatzeko beste terminologia ere zabaldu da, eta oso ohikoa da sare-helbidea **sare-aurrezenbakia** deitzea, eta sare-maskararen ordez, **aurrezenbakiaren luzera** aipatzea.

Kontuan izan sarea eta interfazea bereizteko ezinbestekoa dela sare-maskara ezagutzea. Oso akats arrunta da sarearen eta interfazearen arteko bit banaketa zortzinaka egin behar dela uste izatea eta, beraz, sare-helbidea ezagutzea nahikoa dela banaketa hori zein den ondorioztatzeko. Horrela izanik, C sarearen helbidea 158.227.115.0 dela jakinez gero, ez genuke ezertarako maskara erabili behar. Baina sare-helbidea 158.227.115.0 izanda ere, gerta liteke interfazearenak eskuineko 7 bit bakarrik izatea (edo 6, edo 5... edo bakar bat). Anbiguotasun hori desegiteko erabili behar dira sare-maskarak.

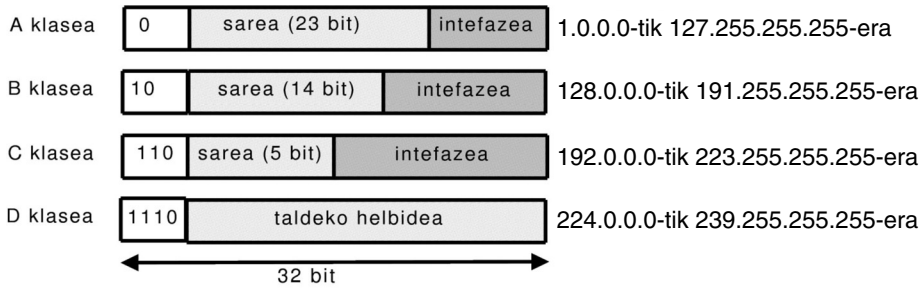
2.4. irudiko sareartea sare-helbideak erabiliz berregiten badugu, 2.5. irudia izango dugu. Irudi honetan agertzen diren konputagailuen interfazeen helbideei dagokien maskara bere sarearena izango da.



2.5. irudia. Sare-helbideak eta maskarak.

Helbide-klaseak

Ikusi dugun helbideratze-eskema, maskaretan oinarrituta, 1993. urtean definitu zen, CIDR izenarekin (Classless InterDomain Routing, RFC 4632), eta geroztik erabiltzen dena da. Baina 1981ean definitu zen IPv4 helbideen antolaketa beste era batekoa zen, sinpleagoa. Lau helbide-klase bereizten ziren (A, B, C eta D klaseak). Helbide bat zein klaseri zegokion jakiteko, helbidearen aurreneko 4 bitak erabiltzen ziren. Helbide barruko sarearen eta interfazearen arteko bit-banaketa finkatuta zegoen klase bakoitzeko. Banaketa hori 2.6. irudian ikus daiteke.



2.6. irudia. IPv4 jatorrizko helbide-klase zaharkituak.

Bosgarren klasea ere, 11110 bitekin hasten zena, definitu zen, baina ez zen erabiltzen: etorkizunerako gorde zen.

Klaseetan oinarritutako helbideratzea ez da jadanik IP helbideratze-sistema. Interneten hazkundeak eta IP helbideen eskaera handiak azaleratu zuten egituraketa hori itxiegia zela, eta ez zeukala inongo malgutasunik eskaera horri aurre egiteko. Helbide gehiegi xahutzen ziren sistema horretan: erakunde gehienek C klaseko sare-helbide batek ematen dituen 254 IP helbideak[9] baino gehiago behar zituzten, eta horregatik B klaseko sare-helbide bat esleitu behar zitzaien. Baina, beste alde batetik, kasu gehienetan erakundeek B klaseko sare-helbide batek ematen dituen 65.534 IP helbideak baino askoz gutxiago behar zituzten, eta, ondorioz, IP helbide gehienak gordeta gelditzen ziren, erabili gabe.

Hala eta guztiz ere, sistema eragileen bertsio zaharretan ez da zaila klase-helbideen nomenklatura topatzea konputagailuaren sare-interfazea konfiguratzean. Batzuetan, bi sistema, klaseena eta maskararena, nahastuta agertzen dira, helbideratze zaharretik berrirako trantsizioan nolabaiteko bateragarritasuna lortzeko asmoz. Baina gaur egun ekoitzitako sistema eragileek eta aplikazioek klaserik gabeko helbideratzea, maskaretan oinarrituta, erabiltzen dute.

Helbide bereziak

IP helbide baten 32 bitek ahalbidetzen dituzten helbide guztiak ez dira erabilgarriak sarearteko konputagailuak identifikatzeko (edo, hobeto esanda, konputagailu horien interfazeak identifikatzeko). Horietako batzuk erabilera berezietarako daude gorderik. Ondoren laburbiltzen ditugu helbide horiek, honako notazio hau erabiliz notazio estandarra egokia ez denean:

{sare-aurrezenbakia, makinaren identifikazioa}

Aurreko eremuetako bit guztiak zerokoak direnean, <0> idatziko dugu, eta guztiak batekoak direnean, <1>.

9. Laster ikusiko dugu zergatik ez diren $2^8 = 256$ helbide.

- *Loopback* helbideak: 127/8 sorta.

  Bere lehenengo byteak 127 balio duen helbideak ezin dira erabili konputagailutik at. Helbide sorta hau (2^{24} helbide!!) gordeta dago konputagailuaren barruko entitateak identifikatzeko. Haatik, bat besterik ez da erabiltzen, konputagailuak bere burua identifikatzeko erabiltzen duena. Bere burua izendatzeko, *localhost* izena erabiltzen du sistema eragileak, eta horrekin lotzen du 127.0.0.1/8 helbidea.

- Sareko difusio-helbidea (*broadcast*): {sare-aurrezenbakia, <1>}.

  IP helbide batzuk gordeta daude difusioa egiteko: beren interfazearen identifikazioaren bit guztiak 1ekoak dituztenak. Horrelako helbideek balio dute datagramak sareko konputagailu guztiei igortzeko. Adibidez, 155.233.0.0/16 sareko makina guztiei bidalitako datagrama batek 155.233.255.255 helburuko helbidea izango du.

- Difusio mugatuko helbidea: {<1>,<1>}.

  Beste aldetik, bere 32 bitak, sarekoak eta interfazekoak, 1ekoak dituen helbidea ere gordeta dago (255.255.255.255 helbidea, alegia): hori difusio mugatuko helbidea da. IP konfigurazio dinamikoa erabiltzen denean sortzen da egoera xelebre hori, gero ikusiko dugun DHCP zerbitzua erabiltzen denean, adibidez. Kasu horietan, beren nortasuna zein den ez dakiten konputagailuek 255.255.255.255 helbidea erabiltzen dute beren sareko konputagailu guztiei laguntza-deia helarazteko. Espero da hartzaileetako batek erantzungo diola mezuaren bidaltzaileari bere nortasunaren berri emanez.

  Pentsa daiteke 255.255.255.255 eta sareko difusio-helbidea baliokideak direla, baina ez da hala. Aurreko adibideari jarraituz, 155.233.255.255 eta 255.255.255.255 helburuko helbideek ez dute tratamendu bera izango bideratzaileetan. Suposatzekoa da 155.233.0.0/16 sarea beste azpisare askok osatuko dutela, eta haien artean bideratzaileak egongo direla. Bideratzaile horietan dago aldea: 155.233.255.255 helburuko datagramak birbidaliko dituzte, 155.233.0.0/16 sareko makina guztiek jaso ditzaten, baina 255.255.255.255 helburukoak iragaziko dituzte. Beraz, azken horiek, igortzailearen sare-segmentuko konputagailuek soilik jasoko dituzte, eta ez sareko guztiek.

- Sare-helbideak: {sare-aurrezenbakia, <0>}.

  Interfazearen identifikazioko bitak ezin dira denak 0koak izan, horrela sortutako helbidea sare-helbidea delako. Nahiz eta maskara erabiliz teknikoki posible izan bereiztea sare-helbidea (adibidez, 155.233.0.0/16) eta interfazearen bit guztiak 0 baliokoak dituen makina baten helbidea (hau da, gure adibidean, 155.233.0.0/32 helbidea), nahasgarria litzateke eta ez egitea gomendatzen da.

- Autokonfigurazioko helbide lokalak: 169.254.0.0/16 sorta.

  Helbide hauek gordeta daude interfaze bati dagokion helbidea zein den lortzerik ez dagoenean erabiltzeko. Konfigurazio automatikoan soilik erabil daitezke. Gehienbat erabiltzen dituzte DHCP bezeroek, DHCP zerbitzariarekin komunikatzea lortzen ez dutenean. Orduan sorta honen edozein helbidetako bat aukeratzen dute, eta interfazeari esleitzen diote. Helbide hauek zuzenean konektatuta dauden konputagailuekin komunikatzeko besterik ez dute balio; bideratzaileek ez dituzte birbidaltzen. RFC3330 agirian gordetzen dira, eta RFC3927an arautzen da haien erabilera.

- Helbide pribatuak: 10/8, 172.16/12, eta 192.168/16 sortak.

  Helbide hauek norberaren sare pribatuaren barnean soilik erabiltzekoak dira. Hau da, ez dago Internetera bidaltzerik bere helburuko helbide gisan hauetako bat daraman datagramarik. Izan ere, Internet osatzen duten sareak interkonektatzen dituzten bideratzaileek ez dituzte helbide hauek daramatzaten datagramak birbidaltzen.

Helbideak	Sorta	Erabilera	RFC	Kopurua
10.0.0.0-10.255.255.255	10/8	Sare pribatuak	1918	16 777 216
127.0.0.0-127.255.255.255	127/8	Loopback	1700	16 777 216
169.254.0.0-169.254.255.255	169.254/16	Autokonfigurazioa	3330	65 536
172.16.0.0-172.31.255.255	172.16/12	Sare pribatuak	1918	1 048 576
192.0.2.0-192.0.2.255	192.0.2/24	Dokumentazioa eta adibideak	3330	256
192.88.99.0-192.88.99.255	192.88.99/24	IPv6 migraziorako	3068	256
192.168.0.0-192.168.255.255	192.168/16	Sare pribatuak	1918	65 536
224.0.0.0-239.255.255.255	224/4	Multicast	3171	268 435 456
240.0.0.0-255.255.255.255	240/4	Gordeta	1700	268 435 456

2.2. taula. IPv4 helbide bereziak.

Goiko taulan adierazitako helbide bereziak dauzkagu, gehi beste batzuk. Erabilera bereziko IPv4 helbide guztien zerrenda RFC3330 agirian dugu (*Special-Use IPv4 Addresses*).

2.4.2. Azpisareak

Sareak era hierarkikoan egituratzen dira. Sareen sarea den Internet, adibidez, **sistema autonomo** izeneko sareetan egituratzen da, non sistema autonomo bat

erakunde administratibo bereko sarea baita[10]. Era berean, sistema autonomo bakoitza beste sare askok osa dezakete, sare-hierarkian beste maila bat gehituz. Adibidez, Euskal Herriko Unibertsitateko sarea sistema autonomo bat da, 158.227.0.0/16 sare-helbidea duena. Sare horren barruan, beste sare asko daude elkarren artean konektaturik. Oso komenigarria izango da, bereziki gero ikusiko ditugun bideratze-lanak errazteko, sistema autonomo, edo oro har, IP sare baten barruan dauden beste azpisareak identifikatzea. Horretarako maskarak erabiltzen dira. Beraz, maskarak sare-helbidea eta interfazearen identifikazioa bereizteaz gain, sarea azpisaretan banatzeko ere balio du.

Azpisareak identifikatzeko, helbideko interfazearen identifikadorearen bitak erabiltzen dira, eta, hala, maskara luzatu egiten da. Har dezagun berriro 155.233.0.0/16 sarea adibide gisa. Demagun sare horren barnean 10 azpisare daudela. Azpisare horiek identifikatzeko, 4 bit behar ditugu gutxienez. Bit horiek kenduko dizkiogu interfazearen identifikadoreari; hau da, helbidearen hirugarren bytearen hasierako lau bitak izango dira. Sare-helbidea era bitarrean adierazten badugu, horiek dira ondoan nabarmenduta dauden lau bitak:

10011011 11101001 **0000**0000 00000000

Eta, horrela, honako 16 azpisare-helbide hauek lortuko ditugu, notazio hamartar puntudunez adierazita:

155.233.0.0/20	155.233.96.0/20	155.233.192.0/20
155.233.16.0/20	155.233.112.0/20	155.233.208.0/20
155.233.32.0/20	155.233.128.0/20	155.233.224.0/20
155.233.48.0/20	155.233.144.0/20	155.233.240.0/20
155.233.64.0/20	155.233.160.0/20	
155.233.80.0/20	155.233.176.0/20	

Horietako edozein 10 esleitu diezazkiekegu gure 10 azpisareei. Adibidean 4 bit hartu ditugu azpisareak identifikatzeko, baina bit gehiago hartzea ere bazegoen (gutxiago, aldiz, ez). Azpisarea identifikatzeko, zenbat bit beharko ditugun zehazteko bi datu hartu behar ditugu kontuan:

- Bata, noski, zenbat azpisare identifikatu behar ditugun. Horrek bit kopuru minimoa ezartzen du.

10. Zehatza izanda, Interneteko sistema autonomo bat sarearteko bideratzerako unitate bat da. Izan ere, gerta daiteke entitate administratibo bakar batek kudeatutako sareek sistema autonomo batean baino gehiagotan banatuta egotea. Sistema autonomoaren definizioa RFC1930 agirian dago.

- Bestea, zenbat interfaze identifikatu behar diren azpisare bakoitzean. Horrek maskararen bit kopuru maximoa ezartzen du. Gure adibidean azpisare bakoitzean asko jota 100 interfaze egongo balira, 7 bit utzi beharko genituzke interfazea identifikatzeko. Hala, 128 identifikadore izango genituzke, soberan alegia, baina ezin da gutxiago hartu (6 bitekin 64 identifikadore besterik ez genuke lortuko eta). Beraz, adibidean, maskarak 9 bit izan ditzake asko jota.

Irakurleak suposatuko duen bezala, ez dago inongo trabarik azpisareak ere beste azpisare batzuetan banatzeko. Argiago ikusteko, demagun adibideko 155.233.0.0/16 sarea erakunde batena dela, eta erakunde horrek 10 egoitza dituela toki desberdinetan. Horregatik sortu behar izan ditugu goiko 16 azpisare-helbideak. Baina gerta liteke, halaber, 155.233.16.0/20 azpisarea kudeatzen duen egokitzako arduradunak beste azpisare batzuetan egituratu nahi izatea helbideratze-eremu hori, bere sail bakoitzeko azpisarea bereizteko. Demagun 3 sail desberdin daudela eta aurreikusten dela 400 konputagailu izatea, asko jota, horietako sail bakoitzak. Orduan, 2 eta 3 bit bitartean erabil ditzakegu sailen azpisareak bereizteko. Demagun 3 bit erabiltzen ditugula; kasu horretan, 8 azpisare-helbide lortuko ditugu 155.233.16.0/20 helbide-espazioan, bakoitza 512 konputagailu hartzeko ahalmenarekin. Ondoan, helbidearen hirugarren bytearen 8 balio bitar posibleak ditugu (bigarren azpisare-maskararenak nabarmenduta), baita horietatik sortzen diren 8 azpisare-helbideak ere (hauek, notazio hamartar puntudunez idatzirik):

0001 **000**0 → 155.233.16.0/23
0001 **001**0 → 155.233.18.0/23
0001 **010**0 → 155.233.20.0/23
0001 **011**0 → 155.233.22.0/23
0001 **100**0 → 155.233.24.0/23
0001 **101**0 → 155.233.26.0/23
0001 **110**0 → 155.233.28.0/23
0001 **111**0 → 155.233.30.0/23

Horietako edozein 3 hartuko genituzke egoitza horren sailak identifikatzeko.

Luzera aldakorreko maskarak erabiltzea deitzen zaio era errekurtsibo horretan azpisareak definitzeari (*variable-length subnetting* edo *variable-length mask subnetting*). Teknika horri esker, sare bat azpisaretan banatzean ez dugu erabili behar azpisare-tamaina bera azpisare guztietarako. Hori oso garrantzitsua da helbide-eremuaren kudeaketa eraginkorra lortzeko. Gure adibidearekin jarraituz, gerta liteke azpisareetako baten 6 sailek oso behar desberdinak edukitzea, eta, nahiz eta horietako batek 400 helbide behar izatea posible izan, 50 helbide nahikoa izatea beste 5 sailetako bakoitzerako. Hala, helbideak xahutzea litzateke 50 helbide behar dituen azpisare bati 512 esleitzea; nahikoa baita interfazeak identifikatzeko 6 bit besterik ez uztea 5 azpisare horietan. Horretarako, 26 biteko sare-maskara erabiliko genuke azpisare horietan. Adibide batean argiago ikusteko, demagun

155.233.16.0/23 helbidea esleitzen diogula 400 konputagailu beharko dituen sailari. Beste 5ek beharko dituzten 250 identifikadore lortzeko, 155.233.18.0/23 helbide-eremua hartuko dugu eta honela azpibanatuko dugu:

0001 001**0** **00**00 0000 → 155.233.18.0/26
0001 001**0** **01**00 0000 → 155.233.18.64/26
0001 001**0** **10**00 0000 → 155.233.18.128/26
0001 001**0** **11**00 0000 → 155.233.18.192/26
0001 001**1** **00**00 0000 → 155.233.19.0/26
0001 001**1** **01**00 0000 → 155.233.19.64/26
0001 001**1** **10**00 0000 → 155.233.19.128/26
0001 001**1** **11**00 0000 → 155.233.19.192/26

Horietako helbide multzo bakoitzak 64 interfaze identifikatzeko ahalmena du, nahikoa 5 sail horien beharrak asetzeko, horietako sail bakoitzari goiko sare-helbideetako bat esleituta. Beraz, gure erakundeko azpisare batzuek 20 biteko maskara erabiliko lukete (erakundeko egoitza bakoitzeko sareek); beste batzuek, berriz, 23 bitekoa (400 interfaze-identifikadore behar dituen sailekoek), eta beste batzuek, 26 bitekoa (50 identifikadore besterik behar ez duen sailekoek).

Aurreko adibideetan sortutako azpisare-helbideen artean, badaude azpisarearen identifikadorearen bit guztiak 0koak dituztenak (155.233.0.0/20, 155.233.16.0/23, eta 155.233.18.0/26) eta azpisarearen identifikadorearen bit guztiak 1ekoak dituztenak (155.233.240.0/20, 155.233.30.0/23, eta 155.233.19.192/26). Hasiera batean, horrelako helbideak erabiltzea eragozten zuen RFC 950 agiriak, helbide-klaseak erabiltzen zituzten sistemetan sortzen zituzten honako arazo hauengatik:

- Azpisarea identifikatzeko, nahasketak sortzen ziren bit guztiak 0koak zituzten helbideen artean. Adibidez, ez zegoen 155.233.0.0/20 eta 155.233.0.0/23 sare-helbideak bereizterik; bideratzaileentzako helbide bera ziren.
- Sare baten difusio-helbidearen eta sare horren azpisare baten difusio-helbidearen artean ere, nahasketak sortzen ziren. Esaterako, gure adibideetako 155.233.31.255/20 eta 155.233.31.255/23 helbideen artean ez zegoen bereizterik bideratzaile batentzat.

Hala ere, bi arazo horiek desagertu ziren maskararen erabilerarekin, eta, gaur egun, badago horrelako sare-helbideak erabiltzea (RFC 1812 agiriak baimentzen ditu). Dena dela, kontuz ibiltzea gomendatzen da, oraindik gerta baitaiteke helbide horiek onartzen ez dituen sistemaren bat topatzea.

2.4.3. Helbideen esleipena

Nola lortzen du konputagailu batek IP helbide bat? Berriro ere, helbideen bi zatiak bereizi behar ditugu. Konputagailuak lortu nahi duen IP helbidea bere sare-interfaze bati (sare-txartelari) esleitzeko izango da, baina interfaze hori sare konkretu batekin konektatzeko izango da. Sare horrek bere sare-helbidea eta maskara izango ditu, non adierazten den sare horretan zenbat sare-txartel konekta dezakegun, eta zeintzuk diren sare-txartel horiei eman dakizkiekeen IP helbideak. Adibidez, 2.5. irudiko sare baten helbidea eta maskara 158.227.112.0/24 da, eta, beraz, sare horrekin konekta daitezkeen sare-txartelen IP helbideak 158.227.112.1-etik 158.227.112.254-ra dauden 254 horiek dira. Kontuan izan helbide sorta guztietan gordeta gelditzen direla interfazearen identifikazioaren bit guztiak 0koak eta 1ekoak dituzten bi helbide; bata sarearen beraren identifikazioa delako, eta bestea sare-difusiorako helbidea delako. Horrenbestez, helbidearen esleipenak bi urrats ditu:

1. Sareak bere helbide sorta lortu behar du, bere sare-helbidea edo sare-aurrezenbakia, alegia.
2. Sareko konputagailuek aurrezenbakiak zehazten duen helbide sortatik bere IP helbidea lortu behar dute.

2.7. irudia. RIRen mapa. http://www.ripe.net/ripencc/about/infosheet.pdf URLtik hartua.

Gure TCP/IP sarea Internetekin konektatuta egotea nahi izanez gero, aurrezenbakiarekin batera, sarea kudeatzen duen erakundeak dagokion domeinua ere lortu beharko du (aurrerago ikusiko dugu zer diren *domeinuak* Interneten). Hori guztia ICANNek kudeatzen du (Internet Corporation for Assigned Names and Numbers), RFC 2050ean agertzen diren gidalerroei jarraituz. Sarearen

kudeatzaileak lortu nahi duen domeinu motaren arabera, ICANNek izendatutako erregistratzaile batzuekin edo besteekin hitz egin dezake (eta ordaindu) sare-helbidea eta domeinua lortzeko. Erregistratzaileak hierarkikoki eta geografikoki daude antolaturik. ICANNek mundu mailako ardura du, eta, munduko eskualde bakoitzeko, RIR (Regional Internet Register) mailako beste erakunde baten eskuan utzi du ardura hori. Gaur egun, bost RIR dabiltza (ikusi 2.7. irudia): RIPE NCC (Europa, Ekialde Ertaina, eta Asiako eskualde batzuk), APNIC (Asiako Pazifikoa), ARIN (Ipar Amerika), LACNIC (Latinoamerika), eta AFRINIC (Afrika). RIR horiek, berriz, eskualde txikiagoak hartzen dituzten LIR (Local Internet Register) izeneko beste erregistratzaileei uzten dizkiete IP helbide sortak, beraiek banatzeko. Hego Euskal Herrian beren zerbitzuak eskaintzen dituzten LIRak zein diren ikusteko, jo http://www.ripe.net/membership/indices/ES.html helbidera; Iparraldekoak, berriz, http://www.ripe.net/membership/indices/FR.html URLan dituzu.

Gure sarea isolaturik baldin badago, hau da, Internetekin konektatuta ez badago, ez dugu inongo oztoporik izango nahi ditugun sare-helbideak erabiltzeko, eta ez diogu inori inongo baimenik eskatu behar nahi dugun IP helbide sorta erabiltzeko. Hala eta guztiz ere, gogoratu RFC 1918 agirian kasu isolatu hauetan erabiltzeko sare-helbide batzuk erreserbatzen direla (10/8, 172.16/12, eta 192.168/16 sortak).

Behin sareak bere helbide sorta lortuz gero, sareko makina bakoitzak berea lortu behar du. Horretarako bi dira mekanismoak:

- Eskuzko konfigurazioa edo konfigurazio estatikoa. Konputagailuaren kudeatzaileak egin behar du, horretarako sistema eragileak izango duen interfazea erabiliz.
- Konfigurazio dinamikoa, sarearen bidez DHCP zerbitzari bat erabiliz (Dynamic Host Configuration Protocol). Konputagailuak DHCP zerbitzariari IP helbide bat eskatuko dio, eta horrek eman egingo dio. DHCPren konfigurazioaren arabera, konputagailu bati esleitzen zaion IP helbidea beti izango da berdina, edo aldatu egin daiteke.

2.5. IP DATAGRAMAK BIDERATZEA

2.5.1. Bideratze-taulak

Kapitulu honen hasieran ikusi dugunez, IP entitatearen lana datagramak bideratzea izango da. Horretarako **bideratze-taula**[11] erabiliko du. Bideratze-

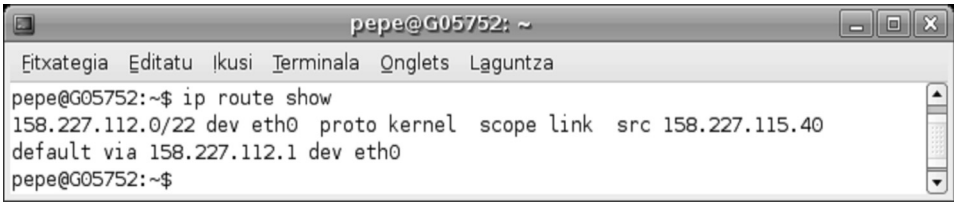
11. Taula hau izendatzeko adostasunik ez dago Interneten. Testu askotan (ikusi RFC 1812), **birbidaltze-taula** deitzen zaio, (*forwarding table* edo FIB-*Forwarding Information Base*). Beste askotan, aldiz, bideratze-taula terminoa erabiltzen da (*routing table*, ikusi RFC 4271).

taulako lerro bakoitza bide bat da (*route*). Honako informazio hau izango dugu, gutxienez, bide bakoitzeko:

- Sare-aurrezenbakia. Hau da taulan bilatzeko gako nagusia. Helbide sorta bat da (helbide bakarra izatea ere posible da, 32 maskara erabiliz), bide honetatik atzigarriak ditugun helbideak biltzen dituena. Datagrama bat bideratzean, IP entitateak bilatuko du bere taulan zein helbide sortari dagokion datagramak daraman helburu-helbidea, eta taulako lerro horretan aurkituko duen informazioa erabiliko du datagrama bideratzeko.
- Hurrengo urratsa. Datagramak bere bidean bisitatu behar duen sarearteko hurrengo bideratzailearen IP helbidea da. Gure konputagailua eta helburukoa sare berean badaude, ez dago hurrengo urratsik bide honetan.
- Interfazea. Makinak duen interfazeen artean, nondik transmititu behar den datagrama. Taulako zutabe honek IP entitateari adieraziko dio ea zein sarbide-entitateri eman behar dion datagrama.

IP inplementazioaren arabera, informazio gehiago egon daiteke (eta egoten da) bideratze-tauletan, baina hauek dira daturik garrantzitsuenak. Helburu batera iristeko bide bat baino gehiago daudenean, erabilgarria da bide bakoitzari balio bat esleitzen dion beste parametro bat izatea taulan. Hori da metrikarena:

- Metrika. Helburura iristeko kostua. Bideratzaileek metrika eremu hau erabiltzen dute taulan agertzen diren aukeren artean bat hartzeko. Gehienetan, bidean zeharkatu beharko den bideratzaile-kopurua (ingelesez, *hop*) adierazten du metrikak, baina beste irizpideak erabiltzea ere badago.



2.8. irudia. Bideratze-taula xume bat, erabiltzaile baten konputagailu batena (eta ez bideratzaile batena).

2.8. irudian Linux sistema baten bideratze-taula ikus daiteke, *ip route* komandoaren bidez lortuta[12]. Irudiko taula interfaze bakarra duen makinaren bi bideko taula tipikoa da. Taulako lehenengo lerroan agertzen da lehenengo bidea, 158.227.112.0/22 helbideetara doana, taula duen konputagailuko sare berean dauden helburuetara joateko bidea, alegia. Bertako saretik atera behar ez denez, ez

12. Badago beste komando batzuekin informazio bera lortzea, adibidez, *route* edo *netstat* komandoekin, baina beste aukera horiek baztertzen ari dira berriagoa den *ip* komandoaren mesederako.

da agertzen hurrengo urratsik, eta zuzenean `eth0` interfazetik birbidaliko dira datagramak. Bigarren lerroa besterik ezeko bidea da (*default* ingelesez), hau da, beste edozein tokitara joateko bidea. Hor bai agertzen dela hurrengo urratsa zein den (`via 158.227.112.1`), erabili behar den interfazeaz gain. Erabiltzaile baten konputagailua denez, interfaze bakarra du, `eth0` izenekoa, eta hortik igorri behar dira, halabeharrez, datagrama guztiak.

Bideratze-taulen erabilera

IP entitate batek datagrama bat bidali (edo bideratzaileen kasuan, birbidali) behar duenean, honako urrats hauek ematen ditu datagramari dagokion bidea bideratze-taulan bilatzeko[13]:

1. *Basic match* araua. Taulako bide bakoitzean, egiaztatu ea bat datorren datagramaren helburuko helbidea bideko helburu-helbidearekin. Bat datozen bideek balizko bideen multzoa osatzen dute.
2. *Longest match* araua. Balizko bideen artean, aukeratu maskara luzeena dutenak. Horiek dira gure datagramari gehienbat dagozkion bideak.
3. *Best metric* araua. Aurreko urratsean aukeratutako bideen artean, metrika hoberena dutenak hautatu.
4. *Vendor policy* araua. Oraindik bide bat baino gehiago baldin badaude balizko bideen multzoan, kudeatzaileak definitutako berezko irizpideak erabili (horrelako irizpideak badaude) beraien artean aukeratzeko.
5. Hautatutako bideak bat baino gehiago baldin badira, edozein aukeratu.

Adibidez, demagun 155.233.18.78 helburua duen datagrama bideratu behar dugula, eta lehenengo urratsa beteta, taulan helbide horrekin bat datozen honako bost bide hauek ditugula:

Destination	*Next hop*	*Metric*	*Interface*
155.233.0.0/16	191.166.12.1	3	Geth0
155.233.16.0/20	191.166.12.2	4	Geth0
155.233.18.0/23	180.96.138.2	4	sdh0
155.233.18.64/26	191.166.12.1	3	Geth0
155.233.18.64/26	191.166.14.3	1	FastEth0

2.3. taula. 155.233.18.78 helbidearekin bat datozen bideratze-taulako bideak.

Bigarren urratsa betez gero, 2.3. taulako azkeneko bi bide besterik ez dugu izango. Hirugarren urratsean horietako bat bakarrik aukeratuko dugu, datagrama

13. Algoritmo hau ez dago estandarizatuta era formalean. RFC 1812 agirian deskribatzen da, baina agiri berak algoritmo hau «Interneten folklorearen zati bat» dela adierazten du.

`FastEth0` interfazetik bidaliko duena. Laugarren eta bosgarren urratsak ez ditugu bete behar izango.

Helbiderik gabeko interfazeak

Batzuetan, konputagailu batetik bestera joateko ez da erabiltzen sare kommutatu bat, baizik eta konexio zuzen bat. Horrelako konexioak erabiltzen dira, askotan, bideratzaileen artean. Adibidez, bi sare lotzen dituzten bi bideratzaileak zuntz optiko baten bidez lotuko dira askotan. Kasu berean daude sare telefonikoaren bidez konektatutako konputagailuak (PPP loturak, alegia), zuntz baten lambdaz (edo kanala) egindako konexioak, edo zirkuitu birtual baten bidezko konexioak (adibidez, ATM eta FR konexioak). Kasu horietan guztietan egoera berezi bat sortzen da bideratze-taula betetzean. Azter dezagun arazoa eta bere irtenbidea.

Hasiera batean, konputagailu baten interfaze bakoitzari esleitu behar zaio IP helbide bat. IP helbide hori interfazearen bidez atzitzen den sareari dagokion helbide sortatik erauzitako helbide bat izango da. Baina konexio zuzenen kasuan ez dago horrelako sarerik, linea bat besterik ez baitugu. Badago linea hori minisare bat bezala hartzea, muturreko bi konputagailuak besterik ez duena, eta sare-aurrezenbaki bat esleitzea lineari. Baina hori IP helbideak xahutzea litzateke, eta, horregatik, helbiderik gabeko lineen kontzeptua sortu dute (*unnumbered lines*). Horrelako linea bati ez zaio inongo sare-aurrezenbakirik esleitzen, eta, ondorioz, linearekin lotuta dauden sare-interfazeek ere ez dute IP helbiderik.

Helbiderik gabeko interfaze hauek zenbait arazo berezi sortzen dituzte. Horietako bat datagrama baten bidean zein den hurrengo urratsa adierazten duen bideratze-taulako parametroaren balioa da. Egoera normal batean, hau da, bere IP helbidea duen interfazearen kasuan, konexioaren beste muturrean dagoen konputagailuaren bideratze-taulan agertuko litzatekeen hurrengo urratseko IP helbidea, interfazearena litzateke. Konexio zuzenen kasuan, ordea, interfazeak ez du IP helbiderik eta konexioak ez du aurrezenbakirik esleituta. Horrela izanik, zein IP helbide ipini behar dugu bideratze-taulako «hurrengo urratsa» eremu horretan? Egia esan, arazoa hutsa da, ez baitago horren IP helbidearen inongo beharrik, interfaze horretatik bidaltzen den guztia toki berera joaten delako (konexioaren beste muturrean dagoen makinara, alegia). Baina bideratze-taulan zerbait jarri behar denez, zenbait trikimailu asmatu dira. Beharbada gehien erabiltzen dena helbiderik gabeko interfazea duen konputagailuaren beste IP helbideren bat (konputagailu batek beti behar du IP helbide bat gutxienez) ipintzea da. Hainbat interfazetarako erabiltzen den helbide horri *router-id* deitzen zaio testu batzuetan (RFC 1812 agirian, adibidez).

2.5.2. Helbide-itzulpena

Bideratze-taulatik IP prozesuak honako bi datu hauek aterako ditu: datagrama bidaltzeko interfazea eta IP helbide bat, datagramak bisitatu behar duen hurrengo makinarena dena (bideratzaile batena edo bere helburuko konputagailuarena). Berriro kapitulu honen hasieran erabili dugun adibidea erabiltzen badugu, 2.1. irudiko A1 konputagailuak B1 makinari datagrama bat bidali nahi dionean, bere bideratze-taulak adieraziko dio IP entitateari datagrama bidali behar duela @AB helbidera, hau da, AB bideratzailea dela datagrama horren bideko hurrengo urratsa. Orduan, IP entitateak datagrama eman behar dio dagokion interfazea kontrolatzen duen beheko mailako entitateari, sarbide-mailarena egiten duenari, berak trama bat eraiki dezan eta sarean sar dezan. Baina sarbide-mailako entitate horri, datagrama emateaz gain, datagrama hori nori bidali behar dion ere esan behar zaio. Hau da, A sarearekin AB bideratzailea lotzen duen sare-interfazearen helbide fisikoa (#AB idazten duguna) eman behar zaio. Horretarako, nolabait, IP entitateak bideratze-taulatik lortu duen @AB IP helbidetik #AB helbide fisikoa lortu beharko du. Gero, datagrama AB bideratzailean dagoenean, makina horren IP entitateak errepikatzen du prozesua: bere bideratze-taulan bilatuta, datagramaren helburua, @B1, zuzenean konektatuta dituen sareetako baten helbide bat dela ebatziko du, @B1 helbide horretatik #B1 helbide fisikoa lortuko du, eta bideratze-taulak agindutako sarbide-entitateari (sare-interfazeari) emango dio datagrama eta dagokion helbide fisikoa.

IP helbideetatik helbide fisikoak lortzeko modu bakarra makinak duen sare-interfaze bakoitzeko itzulpen-taula[14] bat izatea da, non interfaze horren bidez atzigarria dugun sarean dauden makina guztien IP helbideei dagozkien helbide fisikoak aurkituko ditugun. Nola eraiki itzulpen-taula horiek? Eskuz egitea aukera bat da, sare-interfazearen bidez irisgarriak diren makina kopurua txikia bada. Baina hori ez da ohiko egoera. Sare lokaletan, adibidez, hamarnaka edo ehunka makina izan daitezke konektatuta. Gainera, maiz konektatzen eta deskonektatzen dira makinak sarean, eta gerta daiteke makina berak konexio bakoitzean IP helbide desberdina erabiltzea (gero ikusiko dugun DHCP protokoloa erabiliz). Kasu horietan, sareko konputagailu guztien itzulpen-taulak eskuz eguneratzea ez da bideragarria. Hobe dugu lan hori automatikoki egitea.

Automatizazio hori sare bakoitzean ezberdina izango da, sare bakoitzak barneko funtzionamendua eta helbide-egitura berezkoa dituelako. Hau da, bideratzen ari garen datagramarako hurrengo urratseko IP helbidearen eta bere helbide fisikoaren arteko itzulpena datagrama birbidaltzeko erabiliko den interfazearen araberakoa izango da. Oro har, honako bi era hauetako sare-interfazeak izango ditugu:

14. Itzulpen taula izendatzeko beste aukera bat *bizilagunen taula* da, ingelesetik *neighbor table* terminotik itzulita.

- Zuzenean beste konputagailu batekin konektatzen gaituen sare-interfazea. Puntutik punturako lotura (*point to point*) edo konexio zuzen deitzen diegu. Izan daiteke linea fisiko bat (garai bateko linea alokatuak, edo gero eta gehiago erabiltzen diren zuntz ilunak), baina arruntagoa da izatea sare publiko baten linea kommutatua (etxean dugun ISParekiko konexioa, xDSL/sare telefonikoaren bidezkoa), zirkuitu birtual bat (garai bateko X.25 zirkuituak, edo gero eta gutxiago erabiltzen diren FR eta ATM zirkuituak), edo gero eta gehiago erabiltzen diren WDM kanalak (zuntz ilunetako *lambdak* edo koloreak deiturikoak).
- Beste konputagailu askorekin konektatzen gaituen sare-interfazea. Hori da hain ohikoa den Ethernet txartelaren kasua. Hauei puntu anitzeko konexioak esaten zaie.

Konexio zuzenen kasuan ez dago itzulpen-taularen beharrik, beraren sare-interfazetik zuzenean atzigarria dagoen konputagailu bakarra baitago. Puntu anitzeko konexioetan beharko da, bai, itzulpen-taula sortzeko eta eguneratzeko era automatikoa.

Kasu berezi bat zirkuitu birtual edo kanal bat baino gehiagorako irteera den sare-txartelena da (adibidez, bi ATM zirkuituko txartel bat, edo bi lambda dituen SDH txartel bat). Hasiera batean itzulpena behar da, datagrama txartelean dauden bideetako zeinetik bidali behar den jakiteko. Baina horrelako kasuetan, sistema eragileak sare-interfaze berezitutzat hartzen du bide bakoitza. Adibidez, SDH sare-txartel baten bidez erabiltzen dugun zuntzean bi lambda badaude, sistemak `sdh0` eta `sdh1` edo antzeko izeneko bi sare-interfaze bezala hartuko ditu. Era berean, PPP protokoloarekin ezarritako loturak interfaze fisiko berezituak balira bezala hartzen dituzte sistema eragileek. Linux testuinguruan, `ppp0, ppp1, ppp2...` izeneko interfaze gisa ikusiko ditugu horrelako loturak.

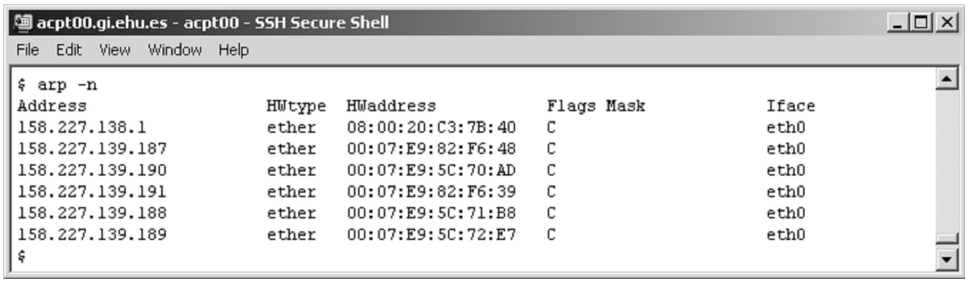
ARP protokoloa

IP helbidearen eta helbide fisikoen arteko itzulpen automatikoa egiteko ARP protokoloa sortu da (Address Resolution Protocol). Protokolo hori erabiliz, sareko konputagailuak komunikatzen dira, beraien helbide fisikoa eta IP helbidea elkarri jakinarazteko eta norberaren itzulpen-taula osatzeko. Ethernet moduko difusio-sareetan, protokoloaren funtsezko funtzionamendua honako hau da:

- Pizten den konputagailuak, ARP difusioko mezu baten bidez, bere helbide fisikoa eta IP helbidea sarean zehar iragartzen ditu. Horrela, dagoeneko sarean dauden beste konputagailu guztiek etorri berriaren berri izango dute, eta beren itzulpen-tauletan sartuko dute (**ARP taula** izenekoan).
- Dena dela, ARP taula cache moduan kudeatzen da, eta sarrera bakoitzak iraupen mugatua du. Horrela izanik, gerta daiteke interfaze baten itzulpena

ez aurkitzea gure ARP cachean, interfaze hori sarean egon arren. Horregatik, ARP taulan IP helbide baten itzulpena aurkitzen ez badugu, sarean zehar ARP eskaera bat hedatu beharko da, berriro ere sareak duen difusio-ahalmena erabiliz.

Ondoko irudi honetan konputagailu baten ARP taula ikus dezakegu, Linuxeko `arp` komandoa erabiliz lortua[15].



2.9. irudia. ARP taula.

Benetan interesatzen zaizkigun irudiko taulako zutabeak lehenengoa (*Address*) eta hirugarrena dira (*HWaddress*). Lehenengoan IP helbide bat agertuko da beti, eta bestean helbide horri dagokion helbide fisikoa.

Hasiera batean ARP protokoloa Ethernet sareetarako definitu zen. Izan ere, argitaratu zenean (RFC 826 agirian) emandako izena *Ethernet Address Resolution Protocol* izan zen. Gaur egun, alde handiz gehien erabiltzen diren sare-konexioak Ethernet direnez, bere erabilera nagusiak IP helbideetatik Ethernet helbideak lortzea izaten jarraitzen du. Baina definitu dira ARP aldaerak beste mota bateko sareetan ere erabiltzeko. Beraien jatorrizko definizioan ARP mezuak Ethernet tramen informazio-eremuan bidaltzen dira. Beste teknologiako sareetan erabiltzeko, teknologia horien trametan ARP mezuak nola sartu definitu behar izan da.

2.5.3. Interneten bideratzea

Orain arte bideratze-taulak nolakoak diren eta nola erabiltzen diren ikusi dugu. Baina, nola eraikitzen dira? Galderaren erantzuna taula erabili behar den makinaren araberakoa da. Dauden kasuak bereizteko, datagrama bati lagunduko diogu haren Internet zeharkako bidaian. Gogoratu 1. kapituluaren hasieran deskribatutako Interneten egitura fisikoa, 1.1. eta 1.2. irudietan azaltzen dena. 1.1. irudiko Z erabiltzailearen makinatik A sarean dagoen X zerbitzariraino heltzeko, zenbait makinatatik igaroko da gure datagrama: jatorrizko eta helburuko konputagailuak (irudiko Z eta X), erabiltzaileen sareetako bideratzaileak, baita ISP txikizkarien, Tier2-ren, eta Tier1-en bideratzaileak ere (irudian agertzen ez

15. *ip neigh show* komandoa erabiliz ere lor dezakegu taula hau.

direnak). Gure bidaian, makina horien bideratze-taulak nola eraikitzen diren ikusteaz gain, nolakoak diren eta nola erabiltzen diren ere berrikusiko dugu.

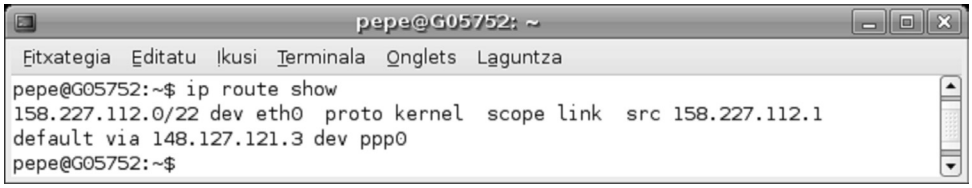
Bertako sarean

Datagramaren bidaia konputagailu igorlean hasiko da. Normalki, horrelako konputagailu batek sare-txartela bakarra izango du. Datagrama txartela horretatik aterako da halabeharrez, baina zein da bere bideko hurrengo urratsa? Hori jakiteko Z konputagailuaren bideratze-taula kontsultatu behar da. Erabiltzailearen konputagailu baten taula oso xumea izango da, 2.8. irudian agertzen denaren antzekoa. Irudi horretan agertzen den taulak bi bide besterik ez du:

- Lehenengoa konputagailuaren sare bereko makinetara joatekoa da (158.227.112.0/22 helbide sortarekin identifikatuta). Bide honetan ez dago hurrengo urratsa den IP helbiderik, hau da, helburura ailegatzeko ez da inongo bideratzailetatik igaro behar.
- Bigarrena konputagailuaren bertako saretik at dagoen edozein helburutara joateko bidea da. Hori da besterik ezeko bidea (*default*). Bide honetan igaro behar da, bai, bideratzaile batetik (taulan, 158.227.112.1 helbidekoa). Bideratzaile horri *sareko irteera, pasabide, atebide,* edo *besterik ezeko bide* deituko diogu[16].

Nahiz eta taula txikia izan, nahikoa da Interneteko edozein konputagailutara doan datagrama bat bideratzeko. Horrelakoak izango dira erabiltzaileen konputagailu gehienetan aurkituko ditugun taulak. Erabiltzailearen sarean irteera bat baino gehiago baldin badaude, beste bide batzuk agertuko dira taulan. Edonola ere, taula xumeak izango dira, eta, gainera, oso taula egonkorrak dira: beraien edukia oso gutxitan aldatzen da denboraren poderioz. Kasu hauetan taulak eskuz betetzea badago, sistema konfiguratzen denean, eta eguneratzea ere, egin behar denean, eskuz egiten da askotan. Linux sistemetan, adibidez, *ip route* komandoa erabiliz egiten da. Hala ere, gero eta gehiagotan, erabiltzaileen bideratze-taula automatikoki konfiguratzen da DHCPren bidez, zeren taulak betetzeko behar den informazioa interfazearen IP helbidea (bere maskararekin) gehi sarearen irteeraren IP helbidea besterik ez baita. Datu horiek DHCP zerbitzari batek eman ditzake.

16. Ez dago denok erabiltzen dugun izen bat honetarako, ez euskaraz, ezta beste hizkuntzetan ere. Ingelesez, *default router* edo *gateway* deitzen dute, eta gaztelaniaz, *encaminador de salida, encaminador por defecto,* edo, Windows sistemetan, *puerta de enlace* (nahiko gaizki aukeratutako izena, nire iritziz).



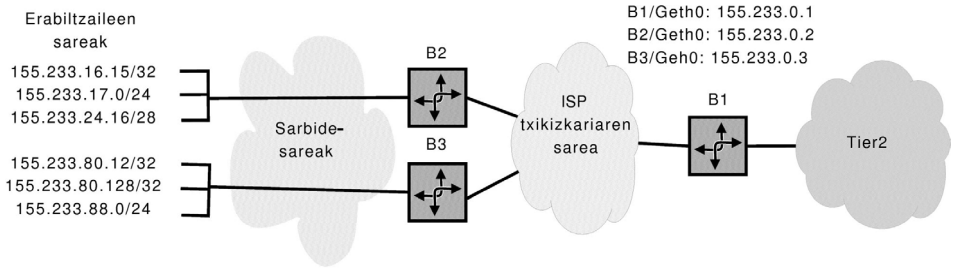
2.10. irudia. Irteera bakarreko sare baten atebidearen bideratze-taula.

Bere bidaiako lehen urratsa eman ondoren, gure datagrama 1.1. irudiko B sareko atebidea den bideratzailean egongo da (makina hori ez dago irudian). Hurrengo urratsa emateko, atebidearen bideratze-taulak dioenari jarraitu behar diogu. 1.1. irudiko A edo B sareetako irteerako bideratzailearen taula nahiko xumea izango da, 2.10. irudiaren antzekoa. Irudi horren taularen eta 2.8. irudian dugun erabiltzaile baten konputagailuaren taularen arteko alde bakarra erabilitako interfazeetan datza (bideratzaileak bi dituelako), baina bi taulek bi bide besterik ez dute: bertako sarera doan trafikorako bidea, eta kanpora doan beste edozein trafikorako bide. Erabiltzailearen konputagailuaren kasuan bi bide horiek makinak duen interfaze bakarretik igarotzen dira (`eth0`), eta bideratzailearen kasuan, aldiz, bide bakoitza interfaze fisiko desberdinetik doa (bertako sarera doana, `eth0`, eta kanpora doana, `ppp0`). Hain konexio gutxiko bideratzaileen taulak eskuz betetzen dira, DHCP ez baita erabiltzen bideratzaileak konfiguratzeko.

ISParen sarean

Gure datagramak kanpora, hau da, Internetera doan bidea, hartuko du, eta B sarearen ISParen bideratzaile batera helduko da. ISP hori 1.2. irudiko txikizkariren bat izango da. Berriro ere, makina honen bideratze-taulari begiratuko diogu bidaiari jarraitzeko.

1.2. irudiko ISP txikizkari baten bideratzaile baten taula erabiltzailearen sarean aurkitutakoak baino handiagoa izango da. Bideratzaile horrek ISPa bere bezeroen sareekin konektatzen duenez, bide bat agertuko da bere taulan bezero bakoitzeko. Besterik ezeko bidea, edo *goranzko bidea*, ISP txikizkariaren sarea Tier2 baten sarearekin lotzen duena izango da. Txikizkariak bideratzaile bat baino gehiago baldin baditu bere bezeroei sarrera emateko, beraien taulak ez dira hain sinpleak izango. Har dezagun 2.11. irudiko ISParen sarea kasu hau aztertzeko. Irudiko ISPak bi bideratzaile erabiltzen ditu, B2 eta B3 izenekoak, bere bezeroei harrera emateko, eta hirugarren bat, B1 izenekoa, Tier2 batekiko konexiorako. Azken hori da, beraz, ISParen atebidea edo goranzko bidea. Irudian 6 bezero agertzen dira ISParekin konektatuta, hiru B2-ren bidez, eta beste hiru B3-ren bidez.



2.11. irudia. ISP txikizkari baten sarea. Errealitatean ISPen sareak konplexuagoak izaten dira, baina azalpenaren garbitasunari eusteko hobe dugu horrenbesteko sare xumea hartzea.

ISPak 155.233/16 helbideratze-espazioa kudeatzen du: B2 bideratzailearen bidez konektatzen direnei 155.233.16/20 azpisortako helbideak esleitzen dizkie; B3-ren bidez sartzen direnentzako 155.233.80/20 helbideak ditu gordeta, eta beste guztiak ez ditu oraindik erabiltzen. Irudiko egoerari dagokion B3 bideratzailearen taula ondoan duzu (2.4. taula), `ip route` komandoa erabiliz lortutako formatuan baino argiagoa den era batean. B2-ren taula oso antzekoa izango da.

Helburua	*Bideratzailea*	*Interfazea*
155.233.80.12/32	-	ppp0
155.233.80.128/32	-	ppp1
155.233.88.0/24	155.233.88.1	ppp2
155.233.80.0/20	-	null
155.233.16.0/20	155.233.0.2	Geth0
default	155.233.0.1	Geth0

2.4. taula. 2.11. irudiko B3 bideratzailearen bideratze-taula.

B3 bideratzailea ISParen sarearekin lotzen duen interfazeak `Geth0` izena du. Bere beze-roekin komunikatzeko PPP lotura bat eratzen du bezero bakoitzeko, eta `ppp0`, `ppp1`, `ppp2`... izeneko interfazeak balira bezala identifikatzen ditu horrela eratutako loturak. Interfaze horiek erabiltzen dira 2.4. taulan agertzen diren hasierako hiru bideetan. Bide horiek beraien bezero bakoitzari datagramak bidaltzeko erabili behar direnak dira. Lehenengo bi bezeroek 32 biteko maskara erabiltzen dute: horrek esan nahi du konputagailu bakarreko sareak direla, edo gero ikusiko dugun NAT protokoloa erabiltzen dutela beren sarean. Bi helbide horiekiko komunikazioa zuzena denez, hurrengo bideratzaileari dagokion zutabean ez da ezer agertzen, helbide horietara heltzeko beste inongo bideratzaileetatik ez dela igaro behar adieraziz. B3-ren bidez ISPraino heltzen den hirugarren bezeroaren kasua desberdina da: bezero horrek 254 IPv4 helbide ditu esleituta (155.233.88.0/24 sorta). Helbide horiek dituzten konputagailuetaraino heltzeko, datagramek ISParen hirugarren bezero horrek duen bideratzailetik igaro beharko dute. Irudian ez da agertzen, baina taularen arabera, 155.233.88.1 helbidea izango du esleituta bideratzaile horrek. Hau da lehen azaldu dugun helbiderik gabeko

interfaze baten adibidea: ohartu B3 bideratzailearekiko lotura egiten duen bezeroaren bideratzailearen interfazeak ezin duela 158.233.88.1 helbidea eduki, 155.233.88.0/24 sareak ez baititu elkartzen bi bideratzaileak. Horregatik bezeroaren atebidearen beste interfaze baten helbidea (155.233.88.1) erabiltzen da B3-ren bideratze-taulan.

Taulan agertzen den laugarren bidea berezia da: haren interfazea *null* da. Horrek esan nahi du datagrama ez dela inora birbidaliko, hau da, datagrama baztertuko duela bideratzaileak. Bide berezi horrek **baliogabeko bidea** du izena (ingelesez, *null route* edo *blackhole route*). Taula honetan zertarako erabiltzen den ulertzeko, har dezagun kontuan B3 bideratzailearen bidez konektatzen diren bezeroentzako 155.233.80/20 helbide sorta duela erreserbatuta ISPak. Bezero bat konektatzen den bakoitzean, sorta horretatik hartutako helbide bat (edo azpisorta bat) esleituko zaio, `pppn` izeneko interfaze berri bat sortuko da, eta esleitutako sortari dagokion bide berria erantsiko zaio bideratze-taulari, orain `ppp0, ppp1` eta `ppp2` interfazeei dagozkienak bezalakoa. Beraz, 155.233.80/20 helbideetako batera doan datagrama bat B3 bideratzaileari helduko zaio. Baina helbide hori erabiltzen duen bezeroa ez badago une horretan konektatuta, datagrama horrek ez du beste inora joaterik, eta baztertu behar da. Horretarako erabiltzen da baliogabeko bidea. Ohartu ezen, datagrama hori taulako lehenengo hiru bideetako batera joatekoa bada, maskara luzeeneko arauari jarraituko diola bideratzaileak datagrama baliogabeko bidean ez deuseztatzeko.

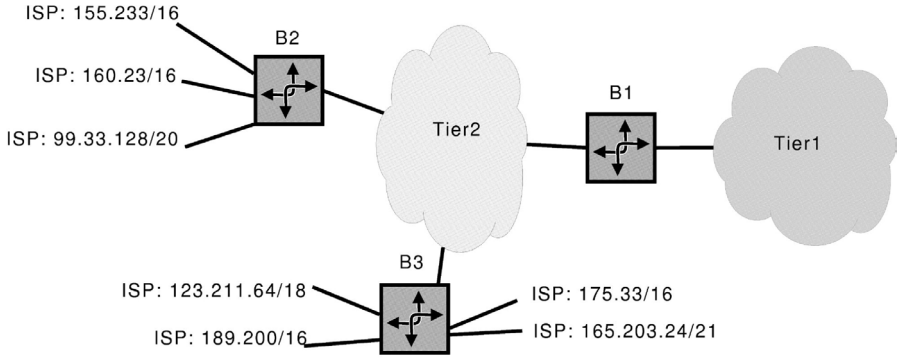
B3-ren hurrengo bidea B2 bideratzailearen bidez konektatuta dauden sareetara garamatzana da. Horregatik agertzen da B2-ren IP helbidea hurrengo bideratzailearen zutabean. Azkeneko bidea Interneteko beste edozein helbidetara joatekoa da. Horretarako B1-etik atera behar da ISParen saretik, taulan agertzen den moduan. Irudiko B1 eta B2 bideratzaileen taulak B3-renaren antzekoak izango dira, baina B1-enaren kasuan, goranzko bidean agertuko den bideratzailea ISPak kontratatuta duen Tier2-ren sareko bideratzaile bat izango da.

2.8. eta 2.10. irudietan agertzen diren taulekin alderatuta, bi dira 2.4. taulan atzematen ditugun aldeak: bata, bide gehiago agertzen direla, eta bestea, bide horiek ez direla beste tauletakoak bezain iraunkorrak. Iraunkortasunik ez horrek taularen eguneratze erdi-automatikoa ekarriko du. Zuzenean kudeatzen dituen bideak (ppp loturak) era automatikoan erantsiko ditu, eta ezabatuko ditu bideratzailearen IP entitateak bezero bat konektatzen eta deskonektatzen den bakoitzean. Bere sareko beste bideratzaileetara doazen bideak, aldiz, iraunkorrak dira, eta eskuz gehituko dira taulan, bideratzailea konfiguratzean.

Tier2-ren sarean

1.1. irudiko Z makinatik atera zen datagrama, 1.2. irudiko Tier2-ren baten sareko bideratzaile batean egongo da dagoeneko. Bideratzaile horren taularen

egitura lehen ikusi dugun ISP baten bideratzaile baten taularen antzekoa izango da, baina, kasu honetan, berriro, bide gehiago agertuko dira. Horretaz jabetzeko, ikusi 2.12. irudiko B2 bideratzailearen taula sinplifikatuta, 2.5. taula dena. Taula horretan agertzen diren *atmX* izeneko interfazeak B2 bideratzailearekin lotura zuzena duten ISPekiko konexioak dira. *sdhX* izenekoak B3 eta B1 bideratzaileekiko konexioak dira (konexio fisiko bakar batean eratutako bi interfaze desberdinak dira).



2.12. irudia. Tier2 baten sare sinplifikatua. Aurreko irudian bezala, errealitatea konplexuagoa da. Tier baten sarearen benetako egitura ezagutzeko, ikusi [Medhi & Karthikeyan, 2007, 9. kap.].

Helburua	*Bideratzailea*	*Interfazea*
155.233/16	-	atm0
160.23/16	-	atm1
99.33.128/20	-	atm2
123.211.64/18	@B3	sdh1
189.200/16	@B3	sdh1
175.33/16	@B3	sdh1
165.203.24/21	@B3	sdh1
default	@B1	sdh0

2.5. taula. 2.12. irudiko B2 bideratzailearen bideratze-taula. Beste bideratzaileen interfazeen IP helbideak era sinbolikoan adierazi ditugu (@B1 eta @B3) taularen irakurgarritasuna hobetzearren.

2.5. taulak 8 bide ditu. Ez da lehen ikusi dugun txikizkariaren sareko bideratzailearen taula baino askoz handiagoa, baina gauzak oso sinplifikatuta daude irudian. Errealitatean, askoz gehiago izaten dira Tier2 sare bateko bideratzaile baten taulan agertuko diren bideak, honako hauek kontuan hartzen baditugu:

– Irudian eta 2.5. taulan, Tier2 horren sareko hiru bideratzaile besterik ez dira agertzen, baina askoz gehiago izan daitezke beren bezeroarekiko konexioak gauzatzeko behar diren bideratzaileak. Bideratzaile horietako bakoitzetik,

ISP txikizkari asko egon daitezke *zintzilik*, eta txikizkari horietako bakoitzera joateko bideak agertu beharko dira Tier2-ren bideratzaileen tauletan. Adibidez, Tier2 sarean 10 bideratzaile baldin badaude txikizkariei sarrera emateko, eta bideratzaile bakoitzean batez beste 15 ISP konektatzen badira, orduan 10 x 15 = 150 beheranzko bide agertuko dira Tier2 sareko bideratzaileen tauletan.

– Askotan, txikizkari baten bezeroek esleituta izango dituzten helbide guztiak ez dira izaten helbide sorta berekoak. Kasu horietan, Tier2-rekin konektatuta dagoen ISP bakoitzeko bide bat baino gehiago agertu behar dira tauletan. Aurreko adibideko txikizkari bakoitzak batez beste 4 IP helbide sortan baldin baditu bere bezeroak, 150 x 4 = 600 beheranzko bide agertuko dira Tier2 sareko bideratzaileen tauletan.

– Aurreko bideei goranzko atebidea gehitu beharko diegu. Baina baliteke gure Tier2-k goranzko konexio bat baino gehiago kontratatzea (ikusi 1.2. irudia), eta ez edukitzea 2.12. irudikoan bezala, Interneten ardatzerako atebide bakarra (B1, 2.12. irudian). Horri *multihoming* deitzen zaio, eta Interneteko sare-hierarkiaren edozein mailatan erabiltzen da (oso ohikoa da ISPen artean). Horren helburua Interneterako *back-up* atebideak edukitzea edo atebide desberdinen artean trafikoa banatzea izaten da. Gure Tier2-k horrelako *multihomed* sare bat badu, besterik ezeko bide gehiago agertuko dira Tier2-ren bideratzaileen tauletan. Gure adibideko sarean 2 goranzko konexio badaude, 602 bide agertuko dira dagoeneko tauletan.

– Gainera, 1.2. irudian agertzen den bezala, Tier2 batek beste Tier2 batzuekin izan ditzake konexio zuzenak, trafikoa elkarrekin trukatzeko Interneten ardatzetik igaro gabe. Horri *peering agreement* esaten zaio ingelesez. Beraien bezeroei zerbitzu hobea emateko egiten dira horrelako konexioak, zeren datagramak bidezidorra hartuta lehenago iritsiko baitira beren helburura. Kontura gaitezen zer suposatzen duen horrelako bidezidor bakoitzak Tier2-ren bideratze-tauletan: gurekin konektatzen den beste Tier2-ren bideratzaileetan agertzen diren beheranzko bideak erantsi behar dizkiegu gure taulei. Demagun adibideko Tier2 sareak horrelako konexio zuzenak dituela bere tamaina bereko beste hiru Tier2 sareekin, hau da, horietako bakoitzak 600 beheranzko bide zuzenak dituela. Orduan gure Tier2 sareko bideratzaileen tauletan agertuko dira beraien 602 bide propioak gehi beste 600 x 3 = 1800 bide, zeinak Interneten ardatz-saretik igaro gabe beste Tier2 sareen bidez zuzenean atzigarri dauden. Osoan, 2402 bideko taula izango luke B2 bideratzaileak.

Beraz, ez da harritzekoa ehunka edo milaka bide dituen taula aurkitzea Tier2 baten bideratzaileetan. Horrenbeste bide dituen taula nekez beteko dugu eskuz. Gainera, sareko konexioetan suertatzen den aldaketa bakoitzak (adibidez, ISP

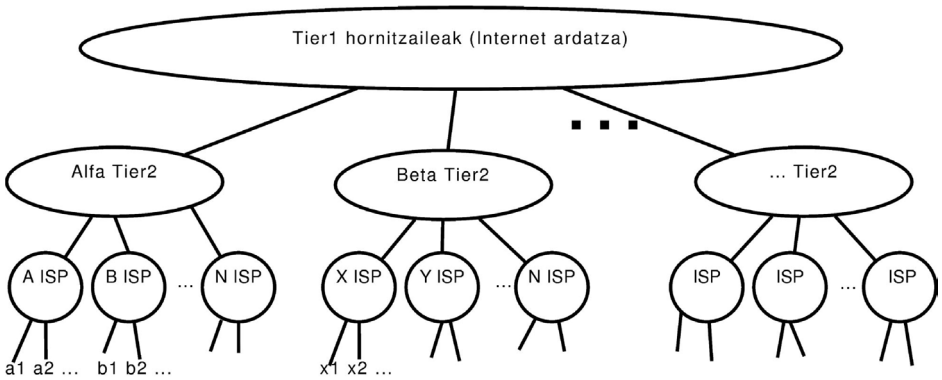
batekiko linea bertan behera gelditzen denean) eragina izan dezake bideratzaile askoren tauletan. Horrek guztiak taulen kudeaketa automatikoa behartzen du, hau da, taulak era automatikoan bete eta eguneratu behar dira hainbeste konexio dituen sare batean. Horretarako erabiltzen dira laster aztertuko ditugun bideratze-algoritmoak eta protokoloak.

Ardatz-sarean

Tier2 mailan topatu dugun egoera areagotzen da Tier1 mailan: bideratzaileen taulen tamaina izugarria da. Tier1 batekiko konexioa duen Tier2 bakoitzaren beheranzko bide guztiak agertu behar dira Tier1 horren tauletan, gehi beste Tier1 mailako sareen bidez atzigarri dauden beheranzko bide guztiak (gogoratu Tier1 mailako sare batek konexio zuzena duela beste Tier1 mailako sare guztiekin). Hau da, Tier1 sare baten tauletan Interneteko bide guztiak agertuko dira. Horregatik maila honen bideratze-taulei **Interneten bideratze-taula** deitzen zaie (*Internet routing table*, ingelesez). Testu hau idaztean (2008ko uztailean) 275.000 bide baino gehiago zituen Interneten taulak. Datu horren egungo balioa ezagutzeko, jo http://www.cidr-report.org URLra. Bideratze-algoritmoak eta protokoloak beharrezkoak genituen Tier2 mailan, eta are beharrezkoagoak Tier1 mailan.

Trafiko-trukaguneak

Gure datagrama Interneten bihotzeraino heldu da, ardatz-sareraino, alegia. Bere helburura heltzeko, Interneten bazterrerako bidea hartu beharko du orain, eta, Tier maila bakoitzean beheranzko bideak hartu 1.1. irudiko A sarean dagoen zerbitzariraino ailegatzeko. Bidaltzen diren datagrama guztiek ez dute, hala ere, Interneten ardatzeraino igo behar. ISP txikizkari bereko bi bezeroen artean komunikazioari dagozkion datagramak ez dira Tier2 mailara igaroko. Hori da 2.13. irudiko a1 eta a2 sareko bi konputagailuen arteko komunikazioari dagozkion datagramen kasua. Era berean, komunikazioaren bi muturrak ISP desberdinetan kokatuta badaude, baina ISP horiek Tier2 bereko sarearen bidez badute goranzko bidea kontratatuta, bidalitako datagramak ez dira Tier1 mailatik pasatuko. Hori da 2.13. irudiko a1 eta b1 sareen arteko komunikazioetan gertatzen dena. Baina a1 eta x1 sareen artekoetan saihestezina dirudi Interneten ardatzetik igarotzeak, eta horrela izango litzateke 2.13. irudian agertzen den zuhaitzean zirkuitulaburrak ez baleude. Zirkuitulabur horiek hornitzaileen arteko konexio zuzenak dira, 1.2. irudian agertzen direnak bezalakoak. 2.13. irudiko A eta B ISPen artean horrelako konexio zuzen bat balego, a1 eta b1 arteko datagramak konexio horretatik bidaliko lirateke, hierarkiako Tier2 mailara igo gabe. A eta B ISPen bezeroen arteko trafikoa handia baldin bada, oso onuragarria litzateke horrelako bidezidor bat sortzea bi hornitzaileen artean: alde batetik, datagramak lehenago helduko lirateke beren helburura, eta, beste alde batetik, ISP horiek gutxitzen dute beren goranzko konexioetan txertatzen duten trafikoa (eta agian konexio merkeagoa kontrata dezakete beren Tier2 hornitzailearekin).



2.13. irudia. Internet hornitzaileen hierarkia, zuhaitz moduan adierazita eta trukagunerik gabe.

Oro har, horrelako zirkuitulaburrak sortzea komenigarria da bezeroak merkatu berean dituzten ISPen artean. Konpetentzia direnen arteko lankidetza honi *koopetizioa* deitzen zaio (*coopetition* ingelesez, *coopetencia* gaztelaniaz). Merkatu berean ISP pare bat baino gehiago egoten direnez, horrelako konexio zuzen asko egin beharko ditu ISP bakoitzak inguruko beste ISP guztiekin konektatuta egon nahi badu. Horrelako konexio bakoitzak bere kostua duenez, ziur asko gutxi batzuekin egiteak besterik ez du ekonomikoki mereziko, ISP horien bezeroen zoritxarrerako. Zirkuitulabur horien kostuak merkatzeko eta ISPen arteko konektibitatea hobetzeko asmoarekin sortu dira Interneteko trafiko-trukaguneak (ingelesez, IXP edo IX siglekin ezagututa, hau da, *Internet eXchange Point*). Trafiko-trukagune bat sare lokal bat da; gehienetan, abiadura handiko Ethernet bat, non ISP asko konektatuta dauden. Hala, beren arteko trafikoa zuzenean truka dezakete, maila goragoko hornitzaileen sareetatik igaro gabe. Honako abantaila hauek dakartza horrek:

- Trafiko-trukagunean parte hartzen duten ISPen bezeroen datagramak azkarrago heltzen dira beren helmugara, trukagunean dauden beste ISPetara doazenean baino. Beraz, bezeroek zerbitzu hobea jasoko dute.
- Hori lortzeko, linea bakarra behar dute ISPek, trukagunearekin lotzen dituena. Hau da, ez dute linea bat ezarri behar trafikoa trukatu nahi duen beste ISP bakoitzarekin, eta, beraz, irtenbide horren bideragarritasun ekonomikoa askoz eskuragarriagoa bihurtzen da.
- Are gehiago: ISP baten gorako trafikoa asko murriztu daiteke, trukagunean zehar bideratuko delako, eta ez bere Internet handizkariaren bidez. Kasu batzuetan, posible izango da handizkariarekin kontratatutako trafikoa jaistea, eta horrekin batera, handizkariari ordaintzekoa.

- Interneten ardatzeraino trafiko gutxiago heltzen denez, benetan hortik igaro behar duten datagramek bideak garbiagoak topatuko dituzte. Oro har, hobeto ibiliko da Internet.

Hala ere, trukaguneetan arazoak sor daitezke parte-hartzaileek elkarri bideratutako trafikoa kontrolatzen ez bada. Horrelako kontrolik ez badago, gerta daiteke A izeneko parte-hartzaile maltzur batek bere gorako trafiko guztia B parte-hartzaileari bidaltzea, nahiz eta datagramen helburua ez egon han. Hala balitz, A-k ez luke kontratatu beharko inongo handizkarirekin bere gorako trafikoaren irteera, B-ren lepotik bideratuko luke eta.

Horrelako bizkarroikeria saihesteko, trafikoa elkarren artean trukatzeko hitzarmenak (lehen aipatutako *peering* hitzarmenak) behar-beharrezkoak dira trukagunean. Hitzarmen horiek trukaguneko bideratze-tauletan irudikatuko dira: hitzartutako trafikoa soilik bideratuko da parte-hartzaile baten sarean zehar.

Euskal Herrian, oraingoz, trafiko-trukagune bakarra dugu, Euskonix izenekoa (www.euskonix.net). Europan asko daude, gehienak Euro-IX (European Internet Exchange Association, www.euro-ix.net) elkartearen inguruan bilduta.

2.5.4. Bide-elkarketa

Taulen kudeaketa ez da arazo bakarra bideratze-taulen tamaina handiegia suertatzen denean. Gainera, bideratzailearen lana asko mantsotu daiteke: gogoratu datagrama bakoitzaren helburuko helbidea taulako bide guztiekin alderatu behar dela. Taulak oso handiak direnean, bideratzea luzatzen da, eta, horrekin batera, bere helburura ailegatzeko datagramak hartuko duen denbora. Gainera, datagrama bakoitza prozesatzeko denbora luzea denez, bideratzaileen ilaretan pilatuko dira datagramak beren txandaren zain, kongestioa sortuz. Kongestioa larria denean, okerrena gertatzen da: heldu berriko datagramak baztertuak izango dira beraientzako tokirik ez badago bideratzailearen ilaretan. Horregatik guztiarengatik taulen tamaina ahal den txikiena mantendu behar da. Horretarako oinarrizkoa da maskaren erabilera ahalbidetzen duen bide-elkarketa.

Bide-elkarketa zer den adierazteko, har dezagun berriro 2.11. irudiko ISP txikizkariaren sarea. ISP horrek 155.233.0.0/16 helbide sorta kudeatzen du. Bezero berri bat lortzen duenean, sorta horretatik ateratako helbide bat (edo azpisorta bat) esleituko dio. Irudian 6 bezero agertzen dira, hiru B2 bideratzaileari lotuta, eta beste hiru B3 bideratzailearekin konektatuta. Beraz, B2 bideratzailearen bidez atzigarri dauden hiru sare horietara joateko bideak agertu beharko lirateke B3 bideratzailearen taulan, baina bakarra agertzen da, 155.233.16.0/20 bidea (2.4. taula ikusi). Hor dago bide-elkarketa baten adibidea: ohartu bide bakar hori beste hirurak (eta ez horiek bakarrik) biltzen dituen helbide sorta bat dela. Era berean,

B2 bideratzailearen taulan ez dugu B3 bideratzailearen bidez atzigarri ditugun hiru sareetarako bideak (155.233.80.12/32, 155.233.80.128/32, eta 155.233.88.0/24 bideak) adierazi behar, nahikoa baita horiek guztiak biltzen dituen 155.233.80.0/20 bide bakarra adieraztea.

Bideak elkartuz asko gutxitu daiteke bideratze-taulen tamaina. Baina askotan bideratzaile baten bidez atzigarri ditugun sare guztiak ezin dira elkartu bide bakar batean, 2.4. taulan egiten den bezala. 2.11. irudiko egoera nahiko ideala da, ISP horren bezero guztiek beren IP helbidea lortu dutelako ISP horren bitartez, eta, beraz, sorta berekoak dira. Ondorioz, helbide horiek berriro elkar daitezke bideratze-tauletan, ISPak ondo egin duelako bere bezeroen arteko helbideen esleipena, irizpide topologiko bati jarraituz. Hau da, ISP horrek ez dio inolaz ere 155.233.16.0/20 sortatik kanpo dagoen helbide bat esleituko B2 bideratzailearekin konektatuko den bezero bati.

Baina demagun bezero berri batek baduela bere IP helbidea dagoeneko esleituta. Helbide hori 155.233.0.0/16 eremutik kanpokoa izango da, eta, beraz, ezin izango da beste bideekin elkartu ISParen tauletan, ezta gero ISP horri goranzko bidea ematen dion Tier2 sarearen tauletan ere.

Hutsuneak tauletan

Arazoa bezero berri hori irabazi duen ISParentzat ez ezik, galdu duenarentzat ere izan daiteke. Demagun 155.233.88.0/24 helbide sorta esleituta duen sarea ISPz aldatzen dela, eta, beraz, jadanik ez dagoela atzigarri 2.11. irudiko txikizkariaren bidez. Orduan, irudiko ISParen taulak aldatu beharko dira. 2.4. taula ez da ia aldatuko, 155.233.88.0/24 sarera joateko bidea moldatu besterik ez da egin behar, ondoko taulan agertzen den bezala:

Helburua	*Bideratzailea*	*Interfazea*
155.233.80.12/32	-	ppp0
155.233.80.128/32	-	ppp1
155.233.80.0/20	-	null
155.233.16.0/20	155.233.0.2	Geth0
155.233.88.0/24	155.233.0.1	Geth0
default	155.233.0.1	Geth0

2.6. taula. 2.11. irudiko B3 bideratzailearen bideratze-taula, 155.233.88.0/24 sarea ISPz aldatzen denean.

Maskara luzeeneko arauak eragotziko du 155.233.88.0/24 sarera doan datagrama bat baliogabeko bidean deuseztatzea. B2 taulan, aldiz, bide berri bat agertuko da. Bideratzaile horren taula ondokoa izango da, 155.233.88.0/24 sareak alde egin eta gero:

Helburua	*Bideratzailea*	*Interfazea*
155.233.16.15/32	-	ppp0
155.233.17.0/24	-	ppp1
155.233.24.16/28	-	ppp2
155.233.16.0/20	-	null
155.233.80.0/20	155.233.0.3	Geth0
155.233.88.0/24	155.233.0.1	Geth0
default	155.233.0.1	Geth0

2.7. taula. 2.11. irudiko B2 bideratzailearen bideratze-taula, 155.233.88.0/24 sarea ISPz aldatzen denean.

Ikusi 155.233.88.0/24 sareak alde egiteak hutsunea sortu duela 155.233.80.0/20 bidean, sorta horretan dauden helbide guztietara joateko bidea ez baita berdina. Gehienetara ailegatzeko B3 bideratzailetik igaro behar da (2.7. taulako bosgarren lerroa), baina 155.233.88.0/24 azpisorta salbuespena da: ISParen saretik atera behar da, Internetera eraman duen goranzko bidea hartuta (B1 bideratzailea, taularen seigarren lerroan agertzen den moduan). Salbuespena den lerro berri hori Interneten ardatzeraino heldu arte dauden sare guztien tauletara hedatuko da. Oro har, ugariak dira horrelako salbuespenak tauletan, Internet hornitzailea aldatzea maiz gertatzen delako.

IPv4 helbideen esleipena

Bide-elkarketaren ahalmen osoaz baliatzeko, irizpide topologikoak hartu behar dira kontuan IP helbideak esleitzeko, gure aurreko adibideko ISPak egin duen bezala. Horrela egingo balitz, eta hutsuneak sortuko ez balira, hornitzaile bakoitzaren sarearen bidez atzigarriak diren sare guztiak bide bakar batean elkartuko lirateke. Egoera ideal horretan, Tier1 sare baten tauletan ez lirateke agertuko ehunka bide baino gehiago. Horretarako, RIR bakoitzak kontrolatzen dituen IP helbideek jarraituak izango beharko lukete, hutsunerik gabekoak, edo, beste era batean esanda, maskara bakar batekin bateragarriak. Orduan posible litzateke RIPE, ARIN, LACNIC, AFRINIC edo APNIC eremu bakoitzeko erakunde guztien sareak elkartzea maskara egokia erabiliz. Internet hasiera-hasieratik mundu mailako sarearte gisa planifikatu izan balitz, IP helbideak irizpide topologiko horrekin banatuko ziren hastapenetik, eta RIR bakoitzeko sare guztiak bide bakar batekin identifikatuko ziren beste eremuetako ardatz-sareko bideratzaileetan. Ez zen hala gertatu, eta helbide sorta asko logika topologiko horretatik at daude; hala ere, CIDR helbideratzearekin hasi zenetik, irizpide topologikoa erabili izan da, bide-elkarketa ahalbidetzeko eta bideratze-taulen tamaina gutxitzeko asmoz. Harrezkero, ICANNek jarraituak diren 2^{24} helbide sortak (hau da /8 maskara erabiliz) uzten dizkie RIRei, eta hauek irizpide berari jarraitzen diote LIRei eta ISPei helbideak uzten dizkietenean. Hau da, sorta jarraituak esleitzen dizkie RIR batek bere LIRei, /8 baino laburragoak diren maskarak badira ere.

CIDR-ri esker 2001. urteko bukaera aldean lortu zen Interneten bideratze-taulen hazkundea moteltzea. Ordu arte, hazkundea esponentziala zen. Geroztik, CIDR eta bide-elkarketaren erabilerak lineal bihurtu zuen hazkunde hori. Hala ere, 2004ko erdialdean, taulen hazkundearen erritmoa berriro handitu zen, gero eta gehiagotan erabiltzaileen sareak ISP batekin baino gehiagorekin konektatzen direlako (*multihoming*), bideak ugarituz.

2.5.5. Bideratze-algoritmoak

Bideratze-taulak automatikoki konfiguratzeko eta eguneratzeko, honako hauek behar ditugu:

1. Helburu bakoitzera joateko zein den bide egokia ebazteko algoritmoa. Hau da, bideratze-taula betetzeko algoritmoa.

2. Algoritmo hori burutzeko behar den informazioa trukatzeko protokoloak.

Algoritmoei dagokienez, globalak eta banatuak bereiziko ditugu. Bietan sareartea grafo baten moduan irudikatzen da, non adabegiak bideratzaileak diren eta adabegien arteko loturak bideratzaileak lotzen dituen sareak diren, eta lotura bakoitzaren kostua parametro desberdinak izan daitezke (sarea zeharkatzeko kostu ekonomikoa, horretarako eman behar den denbora, sare horrekiko interfazearen banda zabalera…). Bi algoritmo moten arteko aldea grafoa eraikitzeko erabiltzen den informazioan eta grafo horretan bideak aukeratzeko metodoan datza.

Algoritmo globalak

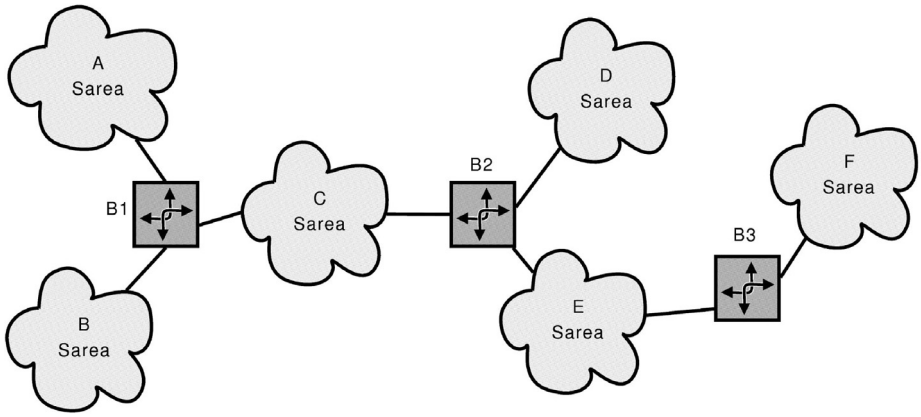
Sarearteko grafo osoa eraiki behar da, gero grafo horretan bide hoberenak bilatzeko. Grafoa eraikitzeko, bideratzaile guztien informazio topologikoa behar da: zer lotura dituen, eta, horietako lotura bakoitzeko, norekin lotzen duen eta zer kostu duen. Lan hori guztia, grafoa eraikitzea eta gero grafo horretan bide hoberenak bilatzea, gune bakar batean egin daiteke (eta orduan algoritmo global zentralizatua izango genuke), edo gune askotan. TCP/IP sarearteetan algoritmo globalak erabiltzen direnean, bideratzaile guztiek jaso behar dute sarearteko beste bideratzaile guztien informazio topologikoa, grafoa eraikitzeko eta bertan bideak bilatzeko. Grafoa eraikita, zenbait aukera daude grafo horretan bideak aurkitzeko. Horien artean, ezagunena grafo batean biderik laburrena bilatzeko Dijkstra-ren algoritmoa da.

Bideratzaile bakoitzak bere loturei buruzko informazioa bidali behar duenez, *link-state* erako algoritmoak izena dute ingelesez (hau da, loturen egoerako algoritmoak).

Algoritmo banatuak (taulen trukaketa)

Kasu honetan bideratzaile bakoitzak periodikoki bidaltzen die bere bideratze-taula berarekiko konexio zuzena duten beste bideratzaileei soilik. Hau da,

bideratzaile bakoitzak bere bizilagunekin bakarrik trukatzen du informazioa, eta ez sareko beste bideratzaile guztiekin, algoritmo globaletan egiten den bezala. Hasiera batean, bideratzaile bakoitzak *bere* sareetara joateko bideak bakarrik ezagutzen ditu, hau da, zuzenean konektatuta dituen sareetarakoak, edo, beste era batean esanda, 0 distantzian dauden sareetara joateko bideak. 2.14. irudiko sareartea hartzen badugu, B1 bideratzaileak, hasieran, A, B, eta C sareetara joateko bideak besterik ez du ezagutuko (hiru bide horiek soilik agertuko dira haren bideratze-taulan). B2 bideratzaileak, aldiz, C, D eta E sareetara joateko bideak izango ditu, eta B3 bideratzaileak, E eta F sareetarakoak. B1-en eta B3-ren bizilagun bakarra B2 da, eta B2-k, beste biak ditu bizilagun. Taulak lehenengo aldiz trukatu eta gero, 1 distantzian dauden sareetara joateko bideak ikasiko dituzte bideratzaileek. Irudiko kasuan, B1-ek D eta E sareetara joateko bideak ikasiko ditu, B2-k bidalitako taulan. B2-k bere aldetik, lehenengo trukaketa egin eta gero sarearteko edozein konputagailutara joateko bidea ezagutuko du, urrutien dituen sare guztiak (A, B eta F sareak) 1 distantzian baitaude. B3-k D eta C sareetarako bideak ikasiko ditu. Taulak bigarren aldiz trukatu eta gero, B1-ek 2 distantzian duen F sarerako bidea ikasiko du, eta B3-k A eta B sareetarako bideak bereganatuko ditu. Bigarren trukaketa hori egin eta gero, bideratzaile guztiek osatu dituzte beren bideratze-taulak.



2.14. irudia. Bideratze-algoritmo banatua (*Distance Vector*).

Deskribatutako algoritmoa *Distance Vector* izena du ingelesez (‘distantzien bektorea’), bideratzaileek trukatzen dutena bektoreak (taulak izendatzeko beste termino bat) direlako, eta taula horietan gordetzen dena sareetara joateko distantziak direlako. Algoritmo globalekin alderatuta, honako hauek dira aldeak:

- Taulak trukatzen dituzten algoritmoek trafiko gutxiago sortzen dute: bizilagunei ez beste inori bidaltzen zaie informazioa. Algoritmo globalaren kasuan, beste bideratzaile guztiei helarazi behar zaie informazio topologikoa.

- Bideratze-taulak lehenago osatzen dira algoritmo globalak erabilita. Algoritmoaren egikaritzapen bakoitzean bide guztiak bideratzaile guztietan birkalkulatzen direnez, bideren bat aldatzen denean (linea bat bertan behera gelditzen delako, adibidez), hurrengo informazio-bidalketan izango dute aldaketa horren berri sarearteko bideratzaile guztiek. Taulak trukatzen direnean, aldiz, aldaketaren berri pixkanaka hedatzen da sareartean. Sareartearen diametroa N baldin bada (hau da, gehienez N sare daude edozein bi bideratzaileren arteko bidean), kasurik okerrenean N trukaketa gauzatu behar dira taulak eguneratzeko aldaketa bat suertatu eta gero.
- Algoritmo globalak sendoagoak dira, hau da, errore bat izatekotan, eragin txikiagoa izango du, bideratzaile bakoitzak jatorrizko informazioa erabiltzen duelako bere taula osatzeko. Taulak trukatzen dituzten algoritmoen kasuan, bideratzaile batek akats bat egiten badu bere taula osatzean, akats hori zabalduko du sareartean.

2.5.6. Bideratze-protokoloak

Sistema autonomoak, barrurako bideratzea eta kanporako bideratzea

Bideratze-algoritmoak aztertu ditugunean, suposatu dugu sarearteko bideratzaile guztiek algoritmo bera egikarituko dutela. Suposizio hori zuzena izan daiteke sarearte txiki batean, baina ez Interneten, honako bi arrazoi hauengatik:

1. Interneten tamainagatik. Gaurko Interneten ehunka milioi konputagailu daude, ehunka mila saretan kokaturik. Sarearte erraldoi horren grafoa eraikitzea eta horretan bideak bilatzea ez da bideragarria, milioika bideratzaile baitaude tartean. Algoritmo globala erabiliz, bideratzaile bakoitzak bidali beharko lukeen informazio topologikoen zaparradarako (beste bideratzaile guztiei beren bidalketa egin behar zaie) ez genuke izango banda-zabalera nahikorik. Taulen trukaketa egiten duen algoritmoa erabiliz gero, ez genituzke inoiz taulak osatuko: Interneten diametroa hain handia izanik, hasierako egoerari dagozkion taulak osatu baino lehen, aldaketak suertatuko lirateke topologian. Ondorioz, taula inkoherenteak ibiliko ziren bueltaka sareartean, eta bideratzea eromen bihurtuko.

2. Interneten egitura administratiboagatik. Ez dago Internet kudeatzen duen entitaterik. Asko jota, Internet osatzen duten milaka sare horien koordinaziolana egiten duten entitateak daude. Baina sare (edo sarearte) bakoitzaren kudeaketa administratiboa eta teknikoa bere jabeari dagokio. Kudeatzaile horrek erabakiko du, besteak beste, zein den erabili behar den bideratze-algoritmoa, bere sarearen ezaugarrien eta interesen arabera.

Bigarren arazo hori aldi berean irtenbidea da: bideratze-taulak ezin dira eraiki eta eguneratuta mantendu Internet osorako, beraz, bideratzeko arazoa zatika

konpondu behar da, maneiukorra den tamainako sare bakoitzean. Horrelako bideratze-entitate bakoitzari **sistema autonomo** izena eman zaio Interneten (azpisareak azaldu ditugunean aipatu dugu Interneten egituraketa hau). Askotan, Internet hornitzaile bakoitzak sistema autonomo bat osatzen du, baina beti ez da horrela. Adibidez, Tier1 batzuek zenbait sistema autonomotan zatitu dute beren sarea. Erabiltzaileen sare handiak ere sistema autonomoak izaten dira. Adibidez, Euskal Herriko Unibertsitateko sarea AS15488 identifikadorea duen sistema autonomoa da. Sistema autonomoak identifikatzeko zenbakiak, IP helbideak bezala, ICANNek esleitzen ditu.

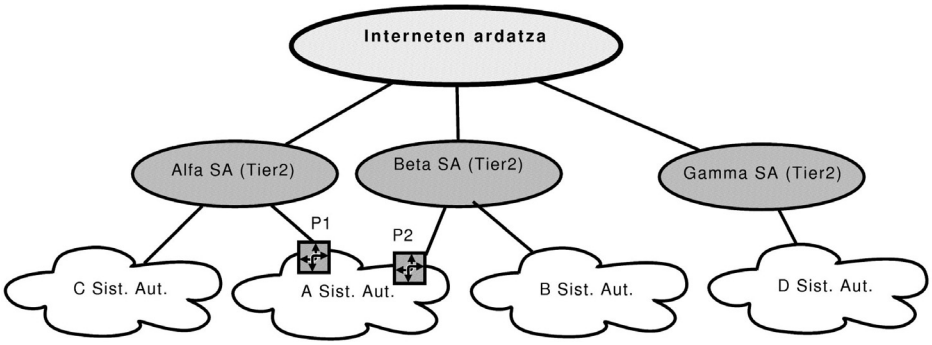
Sistema autonomo baten barruan betetzen da, bai, bideratze-algoritmoak aztertzean egindako suposizioa: bideratzaile guztiek algoritmo bera egikaritzen dute, eta, tamaina mugatuko sareak direnez, bideragarriak dira bi erako algoritmoak. Horrela, konponduta izango dugu sistema autonomoko barneko helburua duten datagramak bideratzeko arazoa. Bideratze horri, sistema autonomoaren barnean egiten denari, **barrurako bideratze** deituko diogu. Bideratze hori **barruko bideratzaile**ek gauzatzen dute, eta horretarako elkarri bidaltzen diote bideratze-informazioa, **barrurako bideratze-protokolo** bat (*intra-AS routing protocol*) erabiliz. Informazio horri esker, dagokion bideratze-algoritmoa egikaritu eta bideratze-taula osatuko dute.

Datagramak sistema autonomotik kanpoko helburua dutenean bideratu ahal izateko, sistema autonomoen arteko lotura egiten duten bideratzaileak behar ditugu. Bideratzaile berezi horiek nortasun bikoitza dute: alde batetik, besteak bezalako barruko bideratzaileak dira, baina beste alde batetik, **kanporako bideratzaileak** dira. Izen bat baino gehiago ematen zaizkie: **pasabideak** (*gateway*) edo kanporako bideratzaileak dira, beharbada, egokienak. Hemen pasabide terminoa erabiliko dugu. Kontuz ibili *pasabide* eta lehen erabili dugun *atebide* terminoa ez nahasteko: batzuetan, atebide bat sistema autonomoen arteko pasabidea izango da, baina ez beti. Pasabide batek beti izango du, gutxienez, beste sistema autonomo baten pasabide batekiko lotura bat. Bere bizilagunak diren pasabideekin trukatuko du bideratze-informazioa, bere bidez atzigarri dauden sareen berri emanez (gutxienez, bere sistema autonomoan daudenak), eta bere bizilagunen bidez atzigarri dauden sareak jasoz. Horretarako **kanporako bideratze-protokoloak** (*inter-AS routing protocol*) erabiltzen dira, eta, noski, bizilagunak diren bi pasabideek protokolo bera erabiltzea adostu behar dute (Interneten denek erabiltzen dute protokolo bera, gero ikusiko dugun BGP4 izenekoa). Jasotako informazioarekin beren bideratze-taulak osatuko dituzte.

Laburbilduz, konputagailu batek bere sistema autonomotik kanpora doan datagrama bat igortzen duenean, barruko bideratzaileek eramango dute datagrama hori sistema autonomotik ateratzeko pasabideraino, barrurako bideratze-protokoloari esker eraikitako taulak horretarako erabiliz. Gero, pasabideak aterako du

datagrama behar den bidetik, kanporako bideratze-protokoloak emandako informazioarekin eraikitako taula horretarako erabiliz. Eskema honek primeran funtzionatzen du sistema autonomo batetik ateratzeko pasabide bakarra dagoenean, baina ez da nahikoa zenbait pasabide badaude. Kasu horretan, barrurako pasabideek ere kanporako bideratze-informazioa behar dute, aukera egokia egiteko dauden pasabideen artean. Informazio hori pasabideetatik lortzeko kanporako bideratze-protokoloak erabili beharko dituzte barrurako bideratzaileek ere. Adibidez, 2.15. irudiko A sistema autonomoko barruko bideratzaile batek B sistemako konputagailu batera doan datagrama bat bideratu behar badu,

1. kanporako bideratze-protokoloari esker jakingo du B sistema autonomoko sareetara joateko bidea P2 pasabidea dela, eta
2. barrurako bideratze-protokoloari esker jakingo du zein den A sistema autonomoan zehar jarraitu behar den bidea P2 pasabideraino ailegatzeko.



2.15. irudia. Barrurako eta kanporako bideratze-protokoloen arteko erlazioa.

Irudiko C sistema autonomora baldin badoa igorritako datagrama, P1 pasabidera bideratuko dute era berean. Baina, datagrama D sistema autonomora baldin badoa, nondik bideratu behar da, P1 ala P2 pasabidetik? Kanporako protokoloak adierazten badu zein den bide bakoitzaren luzera (Alfa sistema autonomotik edo Beta sistema autonomotik), erabakia horren arabera har daiteke. Baina, askotan, bidearen luzera ez da irizpide egokia izaten kanpoko bideak aukeratzean. Oro har, bideak aukeratzeko (eta baztertzeko) irizpideak konplexuagoak izaten dira kanpoko bideratzean (politika komertzialak, kostu ekonomikoak, segurtasun maila…) barruko bideratzean baino, eta, horregatik, algoritmo eta protokolo desberdinak erabiltzen dira barruko eta kanpoko bideratzean.

Barrurako bideratze-protokoloak

Interneten sistema autonomoen barne bideratzerako protokolo batzuk definitu dira. Protokolo horiek betetzea ez da nahitaezkoa; gomendioak dira. Finean, hau barruko arazoa da, eta, beraz, sistema autonomoaren kudeatzaileari bakarrik

dagokio hori antolatzea. Interneteko gomendioez bestelako protokoloak erabiltzea badago, horrek ez baitie kanpoko sareei inongo trabarik egiten. Baina Interneteko gomendioak erabiltzea da errazena, dagoeneko software garatuta eta frogatuta dagoelako.

Honako hauek dira sistema autonomoaren barruko bideratzerako gehienbat erabiltzen diren protokoloak:

- RIP (Routing Information Protocol, RFC 1723/2453/4822).

  Protokolo hau dagoeneko zahartuta dago, baina, hala eta guztiz ere, oso erabilia izaten jarraitzen du. Taulak eraikitzeko eta eguneratzeko, bideratzaile bizilagunek bideratze-informazioa elkarri bidaltzen diote periodikoki (30 segundorik behin, gutxi gorabehera). Hau da, erabilitako algoritmoa banatua da (*distance vector routing*). Distantziarako neurria zeharkatu behar diren azpisareen kopurua da (ingelesezko *hop*-ak, hau da, 'jauziak'). Ezin da RIP erabili sistema autonomo handiegietan: sistema autonomoaren diametro maximoa 15 *hop* da, eta bidal daitekeen taularik handiena 25 bidekoa da. Protokolo hau oso erabilia da erabiltzaileen bertako sareetan eta ISP txikien sareetan.

- OSPF (Open Shortest Path First, RFC 2328/5340).

  RIP zaharra ordezkatzeko diseinatutako protokoloa da. Berriro ere bideratze-informazioaren trukaketa periodikoan datza, baina orain sistema autonomoko barruko bideratzaile guztiek elkarri egiten diote bidalketa. Hau da, bideratze-algoritmo globala erabiltzen da. Grafoan bideak bilatzeko erabiltzen den algoritmoa ibilbiderik laburrenarena da (Dijkstra-rena). RIPrekin alderatuta, honako hauek dira hobekuntzak:

  – Bideratze-informazioaren trukaketak segurtatzeko neurriak hartzen ditu.
  – Helburu bakoitzeko bide bat baino gehiago kalkulatzea badu.
  – Talde-bideratzea egiten du (multicast).
  – OSPFk barruko bideratzaileak hierarkikoki sailkatzen ditu, 4 kategoriatan. Honek asko zailtzen du protokoloa.

  Erabiltzaileen sare handietan eta ISPen sareetan erabiltzen da OSPF protokoloa. Izan ere, bideratzaileen betebeharrak definitzen dituen Interneterako proposatutako estandarrak (RFC 1812) OSPF inplementatzea (baina ez erabiltzea) behartzen du, eta beste barruko bideratzerako protokoloak, aldiz, hautazkotzat jotzen ditu. Hala ere, hornitzaile handi askok oso antzekoa den IS-IS izeneko protokoloa nahiago dute. Azken hori OSI sarearkitekturarako definitu zen. Interneterako egindako IS-IS protokoloaren inplementazioak *Integrated IS-IS* edo *Dual IS-IS* izena du *(*RFC 1195).

- EIGRP (Enhanced Internal Gateway Routing Protocol).

  Hau ere RIP ordezkatzeko diseinatutako protokoloa da, baina ez da Interneteko gomendio *ofiziala* (RFCn argitaratua), enpresa batek egindako proposamena baizik (Cisco Systems). Enpresa horrek egiten dituen bideratzaileak erabilienak direnez, bere proposamenak badu garrantzia. OSPF protokoloaren hasierako 'O' hizkia (*open* hitzari dagokiona) protokolo honek eragin du; EIGRP enpresa batena izanez, proposamen *itxia* da. OSPF, aldiz, *irekia* da. Hauek dira protokolo honen ezaugarri nabarienak:

  – Bideratze-algoritmo banatua erabiltzen da, RIPn bezala, eta bideratze-informazioa bideratze-bizilagunek bakarrik bidaltzen diote elkarri.

  – Trukaketa ez da periodikoki egiten, bideratzaile baten taulan aldaketaren bat sortzen denean baizik. Horrek asko murrizten du bideratze-protokoloak sortutako trafikoa.

  – Bideratze-erabakiak hartzeko zenbait neurri hartzen dira kontuan (atzerapena, transmisio-abiadura, fidagarritasuna, trafiko-zama...), eta sare-kudeatzaileak ezartzen du neurri bakoitzaren pisua erabakia hartzean.

  – Bideratze-informazioa garraiatzen duten bidalketen segurtasunerako neurriak hartzen dira.

Kanporako bideratze-protokoloa: BGP

Kanpoko bideratzaileen lana eta barrukoena oso bestelakoa da. Bion kasuan arazoa datagramak ahal den biderik onenetik bidaltzea da, baina «onena» zer den erabakitzeko irizpideak oso izaera ezberdinekoak dira batean eta bestean. Barruko bideratzerako irizpideak teknikoak dira, hau da, «onena» azkarrena edo laburrena izaten da. Kanpoko bideratzean teknikoak ez diren beste irizpide batzuk ere kontuan hartu behar dira; irizpide ekonomikoak, batez ere, baina politikoak edo segurtasunezkoak ere bai. Adibidez, hornitzaile batek ez ditu garraiatu nahi izango bere bezeroa ez den ISP baten datagramak, bere sarea datagrama horien biderik motzenean egon arren. Horrek taulak betetzeko eta eguneratzeko behar den informazio mota eta mekanismoak guztiz aldatzen ditu.

Barruko bideratzean bezala, Interneten RFC batzuk argitaratu dira kanpoko bideratzaileen arteko komunikazioak estandarizatzeko. Berriro ere, estandar horiek gomendioak besterik ez dira, baina *de facto* estandarrarena egiten dute.

Sistema autonomoen arteko bideratze-protokolo aitzindaria EGP da (Exterior Gateway Protocol), baina nahiko baztertuta dago gaur egun. Erabiltzen dena haren ordezkoa den BGP da (Border Gateway Protocol, RFC 4271). Gaur egun 4. bertsioa dugunez, askotan BGP4 izena erabiltzen da. Hauexek dira haren ezaugarri nagusiak:

- Bizilagunak diren bideratzaileek bakarrik bidaltzen diote informazioa elkarri, algoritmo banatuaren eran. Horrelako bizilagunek BGP bikoteak osatzen dituzte (*BGP peers*), eta beraien arteko komunikazioa BGP saioa da.
- Bi motako BGP saioak daude: kanpoko BGP saioak (*eBGP session*) eta barrukoak (*iBGP session*). Kanpoko saioak bi sistema autonomo desberdinetako pasabideen artekoak dira. Kanpoko saioen bidez ikasiko du BGP bideratzaile batek zein helburu dituen atzigarri berarekin zuzenean konektatuta dauden sistema autonomoen bidez. Informazio hori sistema autonomo bereko beste kanpoko bideratzaileei helaraziko die BGP barruko saioak erabiliz. Normalki, BGP barruko saio bana ezartzen da sistema autonomoan dagoen kanpoko bideratzaile bakoitzeko. Kanpoko bideratzaileen kopurua hazten denean agertzen diren eskalagarritasun-arazoak saihesteko, BGP bideratzaileen arteko barruko saioen kopurua murrizteko honako bi proposamenak daude: bide-birbidaltzaileak (*route reflectors*, RFC 4456), eta konfederazioak (RFC 5065).
- Bidaltzen dena ez da bideari buruzko informazioa (distantzien bektorea), bidea bera baizik (helburura heltzeko zeharkatu behar den sistema autonomoen zerrenda). Horregatik BGP bideen bektoreko protokoloa dela esaten da (*path vector protocol*). Trukatutako bideei atzigarritasun-informazioa deitzen zaie (*reachability information*) BGP hizkeran.
- Gerta daiteke bideratzaile batek helburu baterako bide baten baino gehiagoren berri jasotzea BGP saio desberdinetan. Kasu horretan, aukeratu behar da horietako zein gehitu bideratzailearen bideratze-taulari. Horretarako honako irizpide hauek erabiltzen dira:
  - Sistema autonomoaren kudeatzaileak lehentasunak esleitzen dizkie bideei. Lehentasun handiena duen bidea aukeratuko da. Batek baino gehiagok baldin badute lehentasun handiena, bigarren irizpidea erabiltzen da horien artean aukeratzeko.
  - Bide motzena aukeratu, hau da, sistema autonomoen zerrendarik laburrena duena.
  - Oraindik bide bat baino gehiago baditugu aukeratzeko, eta gure sistema autonomotik pasabide desberdinetatik ateratzen badira bide horiek, pasabideraino barruko biderik motzena duen bidea aukeratu. Horri patata beroaren algoritmoa deitzen zaio (*hot-potato*).
  - Hala ere, bide bat baino gehiago baldintza berean gelditzen bazaizkigu, sasi-zorizko mekanismo bat erabiltzen da bide bat aukeratzeko.

BGPren garrantzia egundokoa da Interneterako. IP protokoloak teknikoki desberdinak diren sareen artean datagramak mugitzea ahalbidetzen duen bezala, BGP protokoloak administratiboki desberdinak diren sareen arteko trafikoa ahalbidetzen

du. Sare telefonikoen artean deiak egiteko SS7 protokoloaren baliokidea dugu BGP Interneten.

Bideratzeari buruzko atal hau amaitzeko, ohartu bideratze-taula osatzeko eta eguneratzeko hainbat iturri daudela. Taula osatzen dute zuzenean lotutako sareetarako bideek, eskuz sartutako bide estatikoek, barruko bideratze-protokoloen bidez ikasitako bideek (horrelako protokoloren bat erabiltzen bada taulako makinan), eta BGPren bidez ikasitako bideek (BGP bideratzaileen kasuan). Gerta daiteke iturri desberdinek bide bera ekartzea. Adibidez, kanpoko bideratzaile batek jaso dezake helburu batera ailegatzeko bide bat bere BGP saio batetik, eta helburu berera joateko beste bide bat barruko bideratze-protokoloaren bidez. Bata edo bestea hartzea IP entitatearen inplementazioaren araberakoa da (hau da, sistema eragilearen araberakoa).

2.6. IP KONFIGURAZIO DINAMIKOA: DHCP ETA NAT

Konputagailu bat TCP/IP sare batean konektatzeko bere IP maila konfiguratu behar dugu. Konfigurazio horren atal nagusiak dira sarearekiko konexioa gauzatuko duen sare-interfazeari IP helbide bat esleitzea eta konputagailuaren bideratze-taula abiatzea. Erabiltzaileen konputagailuen kasuan, lan horiek eskuz edo automatikoki, konfiguraziorako zerbitzari bat erabiliz, egin daitezke. Konputagailu asko dituzten sareen kudeaketa-lana asko errazten du konfigurazio automatikoak, eta berdin gertatzen da konputagailuak sarritan konektatzen eta deskonektatzen direnean sarera, gero eta hedatuagoak dauden WiFi sare lokaletan gertatzen den moduan. Erabiltzaileen konputagailuen konfigurazio automatikoa ahalbidetzeko erabiltzen da DHCP protokoloa (bideratzaileentzat ez da erabiltzen).

Beste alde batetik, Interneten erabilitako IP helbideen kopurua murrizteko asmoz, RFC 1918 agirian definitutako helbide pribatuen erabilera bultzatu da. Horren ondorioz, datagramek garraiatzen dituzten helbideak dinamikoki aldatu behar dira, Internet publikoan helbide pribatuko helburua duten datagramak (bideraezinak direnak) ez txertatzeko. Hori NAT izeneko teknika erabiliz lortzen da.

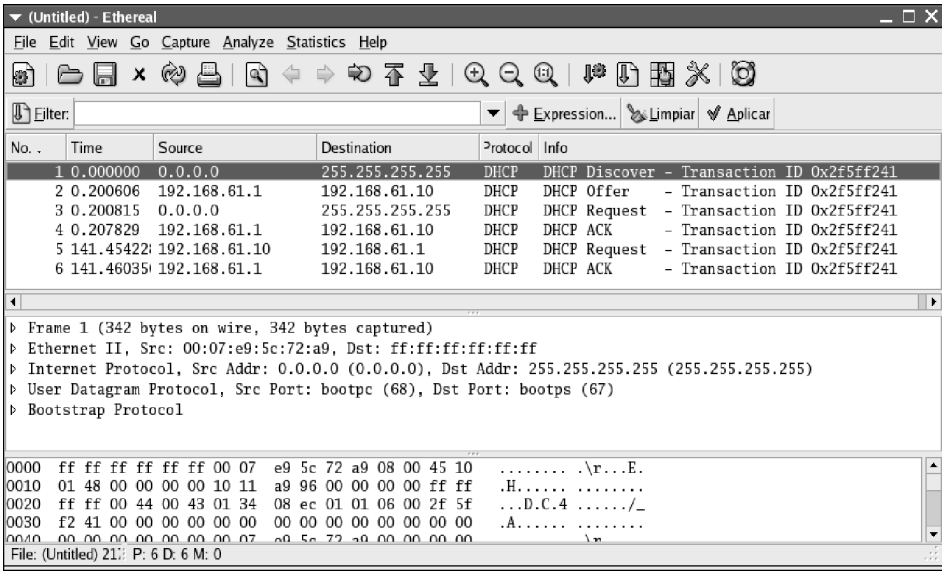
2.6.1. DHCP protokoloa

Sare-konfigurazioa automatikoki egiteko bezero-zerbitzari ereduari jarraitzen zaio. Hau da, badago konfigurazioa egiten duen zerbitzari bat, eta zerbitzari horri eskatu behar dizkiote erabiltzaileen konputagailuek (bezeroek) beren konfiguraziorako datuak. Bezeroen eta zerbitzarien arteko komunikaziorako protokolo bat behar da; hori da DHCP (RFC 2131).

DHCP zerbitzuak IP helbide sorta baten kudeaketa dinamikoa ahalbidetzen du. Hau da, interfaze batek jasoko duen IP helbidea estatikoa (beti berdina) edo dinamikoa (aldakorra) izan daiteke. Esleipen dinamikoa erabilgarria da, adibidez,

gure erakundeak kontrolatzen duen IP helbide sorta sarean dauden konputagailu kopurua baino txikiagoa denean, baina konputagailu guztiak ez badaude aldi berean konektatuta. Kasu horretan, konputagailu bat konektatzen denean IP helbide bat mailegatzen dio DHCP zerbitzariak. Deskonektatzen denean, erabilitako IP helbide hori askatuko da, eta beste konputagailu bati esleitu dakioke. Hori da ISPek egiten dutena IP dinamiko bat ematen digutenean. Horrela, ez dugu behar IP helbide bat sarean egon daitekeen konputagailu bakoitzeko.

DHCP izan da IP helbideen eskasiari aurre egiteko tresna bat, baina haren erabilera bultzatu duen beste arrazoi bat informatika higikorraren etorrera izan da. Gero eta maizago ikusten ditugu konputagailu eramangarria toki batetik bestera besapean daramatenak. Ikasle batek etxean erabiltzen duen makina bera eraman dezake ikastetxera, eta bietan Internetekiko konexioa beharko du. Gune bakoitzeko sare-konfigurazioa eskuz egitea ez da egokia, ezta, askotan, bideragarria ere. Behin-behineko erabiltzaile asko duten sareen baldintzak idealak dira DHCP erabiltzeko: konfigurazio-lanak maiz egin behar dira, konputagailuak etengabe konektatzen eta deskonektatzen direlako, eta benetan behar den IP helbide kopurua sareko erabiltzaile kopurua baino askoz txikiagoa da.



2.16. irudia. DHCP elkarrekintza, sniffer batek hartutako traza batean.

Oinarrizko jarduera

DHCP konfigurazioa egiteko urratsak 2.16. irudian ditugu. Honako hauek dira:

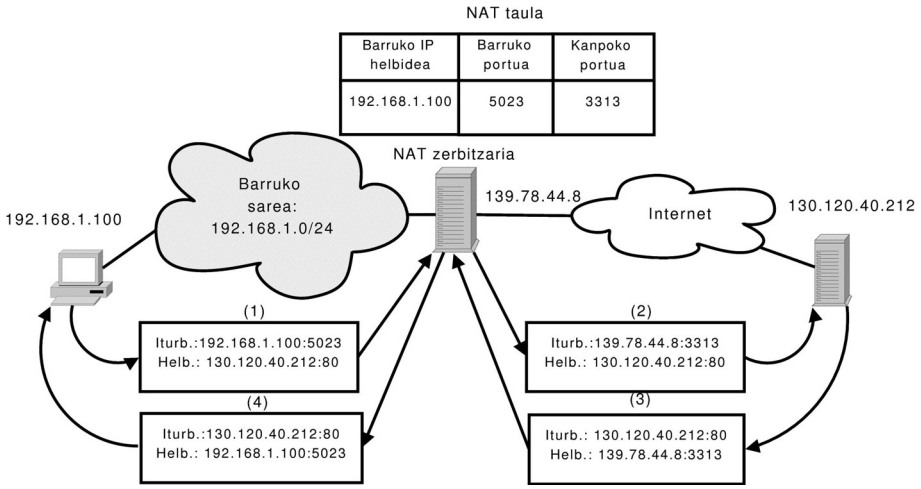
1. Aurkitu DHCP zerbitzaria. Horretarako, bezeroak *DHCP discover* mezu bat bidaltzen du, UDP segmentu batean sartuta (ikusi irudiko 1. bidalketa). UDP segmentua IP datagrama batean sartu behar da, baina, zein IP helbide —jatorrizkoa eta helburukoa— izango ditu datagrama horrek baldin bidaltzen duenak ez badu IP helbiderik esleituta oraindik eta datagramaren helburuaren DHCP zerbitzariaren IP helbidea ez badu ezagutzen? Jatorrizko helbide gisa 0.0.0.0 jarriko du bezeroak, eta helburuko IP helbiderako difusio mugatuko helbide berezia erabiliko du (255.255.255.255). Helburuko helbide hori erabiltzeak esan nahi du erabilitako sareak difusiorako ahalmena duela. Gaur egun sare-txartelean Ethernet teknologia erabiltzea ia unibertsala denez (erabiltzaileen konputagailuetan, behintzat), protokoloak ezartzen duen baldintza honek ez du inor baztertzen. Datagrama difusiorako helbide fisikoa izango duen trama batean (FF-FF-FF-FF-FF-FF helbidea, Etherneten kasuan) sartu eta bidaliko da. Trama hori sareko segmentu berean dauden konputagailu guztiek jasoko dute, eta, horien artean, DHCP zerbitzariak.
2. DHCP zerbitzariak eskainiko dio helbide bat bezeroari, *DHCP offer* mezu bat bidaliz. Hori da 2.16. irudiko 2. bidalketa. Protokoloak DHCP zerbitzari bat baino gehiago sare berean egotea onartzen duenez, agian zerbitzari batek baino gehiagok erantzungo diote bezeroak bidalitako *discover* mezuari. Hala bada, bezeroak aukeratuko du zein zerbitzari erabili. Erantzun horiek IP helbiderik ez duen konputagailuari helarazteko, *discover* mezuak ekarri duen jatorrizko helbide fisikoa erabiliko da. DHCP *offer* mezua UDP segmentu batean sartuko da, segmentu hori IP datagrama batean, eta datagrama hori bezeroaren helbide fisikora bidalitako trama batean.
3. Bezeroak eskainitako helbideen artean bat aukeratuko du, eta dagokion zerbitzariari eskainitako helbidea esleitzea eskatuko du, *DHCP request* mezu bat bidaliz. Eskaera hori eramango duen datagramaren jatorrizko IP helbidea 0.0.0.0 izango da, bezeroak ez baitu oraindik inongo IP helbiderik esleituta. Ikusi 2.16. irudiko hirugarren lerroa.
4. DHCP zerbitzariak helbidea esleituko dio bezeroari, DHCP *ACK* mezu baten bidez (irudiko 4. lerroa). Hemendik aurrera, eta zerbitzariak ezarritako epean, bezeroak badu esleitutako IP helbidea erabiltzea.

Egindako esleipena ez da betiko, iraungitze-epea baitu. Epe hori baino luzerago erabili nahi badu bezeroak «bere» IP helbidea, zerbitzariari eskatu behar dio esleipen hori luzatzea. Horretarako beste DHCP *request* mezua bidaliko dio. Luzapena DHCP *ACK* mezu baten bidez emango du zerbitzariak. Irudiko 5. eta 6. bidalketetan horrelako berritzea egiten da. Ohartu irudiko ezkerreko zutabeaz, *time* izenekoaz. Hor ikus daiteke egindako bi DHCP eragiketen artean emandako denbora. Hasierako 4 bidalketak, konfigurazioari dagozkionak, oso epe laburrean daude eginda (0,207 segundotan). Hurrengo bidalketa egin arte, berritzeari ekin diona, 141 segundo igaro dira: epe hori zen emandako konfigurazioa erabiltzeko muga.

DHCP ez da erabiltzen IP helbideak esleitzeko soilik. Beste sare-konfigurazioko parametroak ere jaso ditzake konputagailu batek DHCP zerbitzari batetik. Horien artean ohikoena sareko atebidearen IP helbidea izaten da. Datu horrekin, eta esleitutako IP helbidearekin, bezeroak bere bideratze-taula eraiki ahal izango du.

2.6.2. NAT

NAT (Network Address Translation, RFC 2663/3022) helbide-itzulpen sistema bat da, baina ez ARP bezalakoa, IP helbidearen eta helbide fisikoen arteko itzulpena egiteko, baizik eta IP helbide pribatu eta publikoen artekoa. Ikusi barruko komunikazioetarako ez direla behar urriak eta ordaintzekoak diren IP helbide publikoak, nahikoa baita RFC 1918-k erabilera pribaturako gordetzen dituen helbide sortak erabiltzea. Baina, beste alde batetik, helbide pribatu horiek Interneten dauden konputagailuekin komunikatzeko ez dute balio. NATek testuinguru bakoitzean helbide mota desberdina erabiltzea ahalbidetzen du, hau da, bertako komunikazioetarako helbide pribatuak erabiltzen dituzte konputagailuek, eta Interneten ibiltzeko, publikoak. Abantaila erabilitako IP publiko kopuruan datza: nahikoa da helbide publiko bakarra sare oso bat Interneten ordezkatzeko.



2.17. irudia. NAT zerbitzua.

NATen funtzionamendua 2.17. irudian adierazten da. Hor bertako sare bat agertzen da, 192.168.1.0/24 helbide pribatuen sorta erabiltzen duena. Sare horren Interneterako atebidean NAT zerbitzari bat dago kokatuta, saretik ateratzen diren edo sarera sartzen diren datagrama guztiek zerbitzari hori zeharkatu behar dutela ziurtatuz. Zerbitzariak helbide publiko bat (139.78.44.8) erabiltzen du bere Interneterako konexioan, eta helbide pribatu bat barrurako konexioan (irudian ez da agertzen). Ikus dezagun zerbitzari horren lana urratsez urrats:

1. Barruko sareko konputagailu batek bidaltzen du datagrama bat kanpora, 130.120.40.212 IP helbidera. Gogoratu 1. ikasgaian ikusi dugula garraiomailak (TCP/UDP protokoloak erabiltzen dituena) identifikatuko duela zein den, helburuko konputagailuan egikaritzen ari diren aplikazioen artean, datagramak daraman informazioaren helburua. Identifikazio hori hurrengo kapituluan sakonago aztertuko dugun portu-zenbakiak egiten du (1. kapituluko 1.1. taulan ere aurkituko dituzu portuak). Irudiko lehenengo bidalketa 80 portura doa, web zerbitzariek erabiltzen dutena. NATek portuekin lan egiten duenez (bere erabilera nagusian, behintzat), portuen kontu hau aurreratu behar izan dugu orain.
2. Datagrama horrek NAT zerbitzaritik igaro behar du kanpora ateratzeko. Kanpora birbidali baino lehenago, datagramaren jatorrizko helbidea ordezkatuko du NAT zerbitzariak, helburua den web zerbitzariak erantzuna inori eman ahal izateko (helbide pribatu bati ezin zaio erantzun, bideratzaileek ez dutelako prozesatuko). Jatorrizko helbide pribatu bakoitza beste helbide publiko batekin ordezkatzen badu NAT zerbitzariak, oinarrizko NAT (*Basic NAT*) dugu. Baina, kasu gehienetan, jatorrizko IP helbide publiko bera esleitzen zaie kanpora doazen datagrama guztiei, NAT zerbitzariaren kanpoko IP helbidea, hain zuzen. Horri **NAPT** (Network Address and Port Translation) edo **IP estalketa** (*IP masquerading*) deitzen zaio, barruko sare osoa NAT zerbitzariaren IP publikoak ezkutatzen baitu. Baina, nola bereiziko du NAT zerbitzariak nori dagokion Internetetik datorren datagrama bat, sareko konputagailu guztiek erabili badute jatorrizko IP helbide publiko bera kanpora bidali dituzten datagrama guztietan? Irakurleak asmatuko duenez, horretarako erabiltzen da portua. Kanpora bidalitako datagrametan, jatorrizko IP helbidea ez ezik, jatorrizko portua ere ordezkatuko du NAT zerbitzariak, irudiko (2) datagraman agertzen den moduan: jatorrizko 5023 portuaren ordez, 3313 ipini du NAT zerbitzariak. Egindako bidalketari dagokion erantzuna gero identifikatzeko, eta kontrako ordezkapena egin ahal izateko, [barruko helbidea + jatorrizko portua, esleitutako portua] bikotea gordeko du NAT zerbitzariak bere itzulpen-taulan. Irudian, 192.168.1.100:5023 bikotea 3313 portuarekin lotuta agertzen da itzulpen-taulan.
3. Bidalitako datagramaren erantzuna NAT zerbitzariari helduko zaio, bere kanpoko IP helbideari bidalita izango baita.
4. NAT zerbitzariak bere taula erabiliko du kontrako itzulpena egiteko: helburuko portuaren arabera, benetako helburuko IP helbidea eta portua eskuratuko ditu, eta datagrama barruko sarean birbidaltzeko balio horiek erabiliko ditu.

NAT teknologia IP helbideen eskasiari aurre egiteko sortu zen, baina segurtasuna ere bultzatzailea izan du, NAT zerbitzari batek barruko sarea ezkutatzen

baitu. Oso ohikoa da NAT eta DHCP batera erabiltzea, sareko atebidea den bideratzailean bi zerbitzariak kokatuta.

2.7. IPv6

90eko hamarkadaren hasieran hasi ziren IPv4 protokoloaren ordezkoa sortzeko ekimenak. Horren ondorioa da IPv6.

Zergatik IPv6

80ko hamarkadaren bukaera aldean, ordu arte ia unibertsitateen mundura soilik mugatuta zegoen Internetek komertzializatze eta zabaltzeari ekin zion. Erakunde askok Internetera konektatu zituzten beren sareak, A/B/C klaseetan egituratutako IP helbideak horretarako erabiliz. Interneten hazkundeak era esponentziala hartu zuen, eta hazkunde hori asetzeko IP helbideen ahalmenak kezka sortu zuen. Interneten ezaugarri teknikoak definitzen dituen IETF erakundeak (Internet Engineering Task Force) IP helbideratze-sistemarekin lotutako honako arazo hauek identifikatu zituen 1992. urtean (RFC 4632):

1. B klaseko helbideak agortzeko arriskua. Beren sarea Interneten sartu nahi zuten erakunde gehien-gehienek B klaseko helbide sorta eskatzen zuten, A eta C klaseen tamaina guztiz desegokia baita sare gehienerako.
2. Interneteko ardatz-sareko bideratze-taulen gehiegizko hazkundea, garaiko hardwareak eta softwareak kudea zezaketen tamaina gainditzeko mehatxua sortuz. Tamaina horrek honako bi arazo hauek sortzen ditu: taulen eguneraketa oztopatzen du, eta kongestioak sorrarazten ditu. Alde batetik, bideratzaileek trukatu behar duten informazio kopurua ikaragarria denez, bideratzaileek datagramak bideratzen baino denbora gehiago eman behar dute trukaketa horiek egiten. Beste alde batetik, datagrama bakoitza prozesatzeko taularen bide guztiak miatu behar direnez (maskara luzeenaren araua erabiltzearren), datagrama bakoitzaren prozesatzeko denbora luzatzen da, horrekin batera prozesatzeko zain dauden datagrama-ilarak handituz. Ilara horien gehienezko luzera gainditua, kongestioa dugu.
3. IPv4 helbide guztien agorpena.

Argi zegoen lehenengo bi arazoak kritikoak bihurtuko zirela 93-95. urteetarako. Irtenbide azkar baten bila, CIDR helbideratze-sistema berria definitu zuten 1993. urtean. CIDRk helbideen egitura klaseetatik askatu zuen, askoz eraginkorragoa den helbideen esleipena ahalbidetuz, eta bideratze-taulen hazkundea moteldu zuen, bide-elkarketaren bitartez. Honek guztiak hasierako bi arazoek sortzen zuten larrialdia gainditzeko balio izan zuen, baina hirugarren arazoa, IP helbide guztien agorpena alegia, konpontzeke gelditzen zen oraindik. Irtenbide bakarra IP helbi-

deen bit kopurua handitzea zen, eta horretarako nahitaezkoa zen IPren bertsio berri bat definitzea. Horra hor IPv6.

Zertan da hobea IPv6?

Hasierako eta helburu nagusia IP helbide kopurua handitzea izan arren, hori ez da IPv6k dakarren hobespen bakarra. Ondoko hauek dira protokolo berriaren hobespen nagusiak:

- IP helbide kopurua ikaragarria da: 2^{128}. Kopuru horrekin badago 7×10^{23} helbide esleitzea Lurreko metro karratu bakoitzean (itsasoak barne); badirudi nahikoa dela.
- Interneten ardatz-sareko bideratze-taulak txikiak izango dira. Hori bermatzeko honako baldintza hauek ezarri dira:
  - IPv6 helbideak zenbaki telefonikoak bezala egituratzen dira, bideratzeko informazio topologikoa aurrezenbakian sartuz. Hau da, helbideko aurrezenbakiak identifikatutako interfazeraino heltzeko bidea adierazten du. Ikusi IPv4ren maskarekiko aldea: maskarak bidea aurkitzeko informazioa ematen du, baina ez du adierazten zein den helbideak identifikatzen duen konputagailuraino heltzeko bide hori.
  - Aurrekoa ahalbidetzeko, IPv6 helbideak ez dira *esleitzen*, *uzten* baizik. Hau da, gure sarea Internetera konektatzen dugunean, gure ISPak utziko dizkigu horretarako beharko ditugun IPv6 helbideak. Helbide horien aurrezenbakia hornitzaile horrekin egongo da lotuta, bide-elkarketaren optimizazioa ahalbidetuz: hornitzaile baten bidez Interneten sartzen diren sare guztien helbideak bide bakar batean elkartu ahal izango dira tauletan. Internet hornitzailez aldatzen badugu, IPv6 helbideak horrekin batera aldatu beharko dugu, tauletan zuloak ez sorrarazteko. Sareen IP birzenbakitze hori era automatikoan egiten da. Horrek DNS-renganako eragina ere izango du, helbidea aldatu arren, konputagailuaren izenak ez baitu aldatu behar.
- IPv6 goiburukoaren egiturak dezente arintzen du bideratzaileek egin behar duten datagrama bakoitzaren prozesamendua, ondokoengatik:
  - IPv4 datagramak prozesatzeko, bideratzaileek aztertu behar dute lehenago non bukatzen den goiburukoa (goiburukoaren hasieran aurkituko duten *goiburukoaren luzera* eremuari begiratuz), bere luzera aldakorra baita. IPv6 bertsioan ez dute hori egin behar: datagrama guztien goiburukoek luzera bera dute (40 byte).
  - Bideratzaileetan ez dago datagramak zatitzerik. Zatiketak IPv4 goiburukoa konplexuagoa egiten du (3 eremu sartu behar dira zatiketak egiteko eta datagramak berreraikitzeko) eta, berriro ere, bideratzaileen lana

zailtzen du. IPv6n, datagrama bat handiegia baldin bada, bideratzaileak baztertuko du, eta datagramaren igorleari kontrol-mezu bat bidaliko dio (ICMPv6 protokoloaren bidez) horren berri emanez. Igorleak berak jatorrizko datagraman sartu nahi zuen informazioa datagrama txikiagoetan banatu eta bidali beharko du.

– Bideratzaileek ez dute inongo errore-kontrolik egin behar. Lehenago ikusi dugunez, IPv4k goiburukoan egiten duen errore-kontrolak ez du asko balio eta, gainera, bideratzaile bakoitzean datagramaren goiburukoa berregitera behartzen du (gogoan izan TTL aldatzen dela bideratzaile bakoitzean eta, beraz, errore-kontrolaren eremua birkalkulatu egin behar dela).

2.7.1. IPv6 helbideak

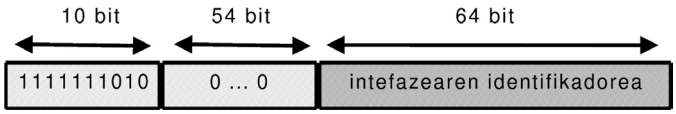
IPv6 helbideratze-sistema RFC 4291 agirian dago deskribatuta. IPv6 helbideak interfazeak edo interfaze sortak identifikatzeko 128 bitez osatutako identifikadoreak dira. Bit horien barneko egitura helbide motaren araberakoa da.

IPv6 helbide motak

Honako hiru IPv6 helbide mota hauek daude:

• Unicast helbideak, edo helburu bakarreko helbideak. Interfaze bakar bat identifikatzen dute[17]. Bi azpimota aurkituko ditugu unicast helbideetan:

– Bertako unicast helbideak (*Link-Local unicast*): izenak adierazten duen bezala, helbide hauek esanahi lokala besterik ez dute. IPv4 helbide pribatuak bezala, bideratzaileek ez dituzte prozesatzen IPv6 bertako unicast helbideak. Beraz, zuzenean lotuta dauden beste interfazeekin komunikatzeko balio dute soilik (interfazearen DHCP bidezko konfigurazioan, adibidez). Beraien formatua ondoko hau da:



2.18. irudia. IPv6 bertako unicast helbideen egitura.

– Unicast helbide globalak (*Global unicast*): IPv4 helbide publikoen baliokideak dira helbide hauek. Hau da, Interneten bidezko komunikazioak egiteko, mota honetako helbideak erabili behar dira. Datagramaren helbururaino heltzeko bideratzaileren bat baldin badago, helbide mota hau behar da helburuko interfazea identifikatzeko. IPv4n bezala, helbidearen

17. Badago salbuespen bat, RFC 4291 agirian definituta.

bitak bi zatitan daude egituratuta: hasierakoek, «aurrezen-bakia» izendatuta, sarea identifikatzen dute, eta besteek interfazea identifikatzen dute. Maskarak zehazten du zenbat bitek osatzen duten aurrezenbakia.

RFC 4291 argitaratu baino lehen (2006ko otsaila) bazegoen hirugarren azpimota bat unicast helbideetarako (*site-local unicast* izenekoa) baina dagoeneko baztertuta dago.

- Multicast helbideak, edo taldeko helbideak. Interfaze asko identifikatzen dituzte. Multicast helbide batera datagrama bat bidaltzen denean, datagrama horren kopia bana eman behar zaie helbideak identifikatutako interfaze guztiei.
- Anycast helbideak. Hauek ere interfaze asko identifikatzen dituzte, baina horietako bakar bati emango zaio anycast helbide batera bidalitako datagramaren kopia bat.

Interfaze batek behar du, gutxienez, bertako unicast helbide bat. Horrez gain, helbide gehiago ere izan ditzake interfazeak, edozein motatakoak (unicast, multicast, anycast).

Ikusi difusio-helbiderik (broadcast) ez dagoela IPv6 bertsioan, haren eginkizuna taldeko helbideek betetzen baitute. Etorkizunean helbide mota edo azpimota gehiago definitzea badago.

Bi helbide berezi hauek definitu dira:

- Zehaztu gabeko helbidea: 128 bitak zerokoak dira. Helbide hau ez zaio esleitu behar inongo interfazeri, helbiderik ez dagoela adierazten baitu, hain zuzen ere. Beronen erabilera tipikoa DHCP bidezko konfigurazioa egitekoa da. Ezin da erabili helburuko helbide gisa. Bideratzaileek ez dituzte birbidaltzen jatorrizko helbidean zehaztu gabekoa daramaten datagramak.
- *Loopback* helbidea: hasierako 127 bitak zerokoak eta azkenekoa batekoa dituen helbidea dugu hau. *Loopback* izeneko interfazea birtuala da, ez fisikoa. Bere buruari datagramak bidaltzeko erabiltzen dute makinek interfaze hori. Ezin zaio inongo interfaze fisikori *loopback* helbidea esleitu, eta ezin dira makinatik kanpora bidali helbide hau daramaten datagramak (ez jatorrizko helbide gisa, ezta helburukoa ere).

IPv6 helbide motak helbidearen hasierako bitek adierazten dute, ondoko taulan agertzen denaren arabera:

Helbide mota	*Hasierako bitak*
Zehaztu gabekoa	000...0 (128 bitak)
Loopback	000...1 (128 bitak)
Multicast	11111111
Bertako unicast	1111111010
Unicast globala	Beste guztiak

2.8. taula. IPv6 helbide moten identifikazioa

Anycast helbideak ez dira agertzen goiko taulan unicast helbideen espaziotik hartzen direlako (bertakoak edota globalak), eta ezin dira sintaktikoki bereizi.

IPv6 helbideen idazkera

Helbide baten 128 digitu bitarrak idaztea ez da batere erosoa. Horregatik, IPv4 helbideen 32 bitekin egiten den bezala, IPv6n ere beste notazio bat definitu da, idazteko eta ulertzeko errazagoa. Notazio horretan hiru modu daude IPv6 helbideak idazteko:

- Oinarrizko modua: helbidearen 128 bitak 16 biteko 8 taldetan banatzen dira, eta talde bakoitza notazio hamaseitarrean idazten da. Sortzen diren 8 zenbaki hamaseitarrak, bakoitza 4 digituk osaturik, : karakterearekin banatzen dira. Adibideak:
  - ABCD:EF01:2345:6789:ABCD:EF01:2345:6789
  - 2001:0DB8:0000:0000:0008:0800:200C:417A

  Beste notazioetan bezala, hamaseitarrean ere ez ohi dira ezkerreko zerokoak idazten. Horregatik bigarren adibidea arrotza da. Gehienetan, ondoko beste era honetan idatziko dugu helbide hori:
  - 2001:DB8:0:0:8:800:200C:417A
- Modu trinkotua: askotan topatuko dugu zerokoek osatutako segida luzeak dituzten IPv6 helbideak. Segida horiek :: karaktere-bikotearekin ordeztu daitezke. Anbiguotasunak ekiditeko, behin bakarrik ager daiteke :: karaktere-bikotea helbide batean. Adibideak:

Oinarrizko modua	*Trinkotua*
2001:DB8:0:0:8:800:200C:417A	2001:DB8::8:800:200C:417A
FF01:0:0:0:0:0:0:101	FF01::101
0:0:0:0:0:0:0:1 (*loopback* helbidea)	::1
0:0:0:0:0:0:0:0 (zehaztu gabeko helbidea)	::

- IPv4 eta IPv6 bertsioen arteko trantsizioan zehar, bi helbideratze-sistemak elkarrekin biziko dira. Epe horretarako IPv4 helbideak IPv6 helbideetan mapeatzeko espazioa gorde da, eta notazio bat ere definitu da horretarako (*IPv4-mapped IPv6 address* izenekoa). Mapeatze hori egiteko gordetako IPv6 helbide sorta unicast helbide globalen azpitalde bat da, hasierako 80 bitak zerokoak eta hurrengo 16ak batekoak dituena. Gelditzen diren azkeneko 32 bitak IPv4 helbidea adierazteko erabiltzen dira. Mapeatze honetarako definitutako notazioan, helbidearen aurreneko 96 bitak lehenago definitu dugun era hamaseitarrean idazten dira (gehienetan trinkotuta), eta azkeneko 32ak, aldiz, IPv4 era hamartar puntudunean. Beraz, horrelako helbide misto baten itxura ondoko adibideetakoa da:

Oinarrizko modua	*Trinkotua*
0:0:0:0:0:FFFF:129.144.52.38	::FFFF:129.144.52.38
0:0:0:0:0:FFFF:158.227.112.1	::FFFF:158.227.112.1

  RFC 4291 argitaratu baino lehen, bazegoen beste era bat IPv4 helbideak IPv6 helbideetan integratzeko (*IPv4-compatible IPv6 address* izenekoa), baina dagoeneko baztertuta dago.

Unicast global edo anycast helbide baten aurrezenbakiaren luzera IPv4ren era berean adierazten da, hau da, ondokoa:

IPv6_helbidea/aurrezenbakiaren_luzera

non

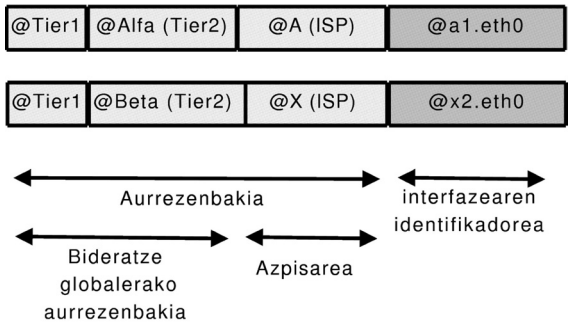
- IPv6_helbidea aurreko edozein notaziotan idatz daitekeen. Helbide osoa izan daiteke edo aurrezenbakia bakarrik. Bigarren kasu honetan idatzitakoa sare baten helbidea izango da.
- Aurrezenbakiaren_luzera hasierako zenbat bitek osatzen duten aurrezenbakia adierazten duen zenbaki hamartarra den.

Adibideak:

2001:0DB8:0:CD30:123:4567:89AB:CDEF/64	Interfaze baten helbidea, aurrezenbakia adierazita
2001:0DB8:0:CD30::/64	Aurreko interfazearen sarearen helbidea
2001:0DB8:0:CD30::/60	Aurreko sarea barne hartzen duen beste sare baten helbidea

Unicast helbide globalak

Unicast helbide globalen sarearen identifikadoreak barneko egitura hierarkikoa du. Zenbait eremuk osatuko dute identifikadore hori, eta horietako eremu bakoitzak Interneten barruti topologiko bat zehazten du. Adibide gisa, har dezagun 2.13. irudiko a1 eta x2 konputagailuen bi interfazen helbidea, a1.eth0 eta x2.eth0 izenekoak. Beraien unicast helbide globalen egitura ondoko irudian ageri da:



2.19. irudia. IPv6 unicast helbide globalen aurrezenbakiaren barneko egitura hierarkikoa (RFC 3587).

Zenbat eremuk osatuko duten helbidearen aurrezenbakia ez dago finkaturik, baina bi dira gutxienez: bideratze globalerako aurrezenbakia (*global routing prefix*) aurreneko n bit dira, eta hurrengo m bitek azpisarearen identifikadorea osatzen dute (*subnet ID*). Aurrezenbakiaren bi eremu hauek azpieremutan egituratzen dira, hauek ere era hierarkikoan. Bideratze globalerako aurrezenbakiaren egitura RIRek (gogoratu, Regional Internet Register) eta ISPek definitzen dute. Azpisarearen egitura sare bakoitzaren kudeatzaileek definituko dute.

Aurreko irudian, @A eta @X dira azpisareen identifikadoreak. Interneten hierarkia topologikoaren goialdean dauden interfazeen helbideek eremu gutxiago izango dituzte aurrezenbakian, eta hierarkia horren behealdean daudenek, gehiago. Erabiltzaileen konputagailuek ez dute ezagutu behar beren interfazeen helbideen

aurrezenbakiaren barneko egitura, hori bideratzeko informazioa baita. Nahikoa dute interfazearen identifikadorearen eta sarearen identifikadorearen arteko muga non dagoen jakitea. Bideratzaileen kasua desberdina da. Datagramak bideratu behar dituztenez, aurrezenbakiaren barneko egitura ezagutu eta erabili behar dute. Hala ere, bideratzaile guztiek ez dituzte ezagutu behar aurrezenbakiaren eremu guztiak. Datagrama bideratzeko zenbat aurrezenbakiko eremu ezagutu beharko dituen bideratzaileak, bideratzaile horrek sarean duen kokapenaren araberakoa izango da.

Aurrezenbakiaren luzera 64 bit da, hasierako 3 bitak 000 direnean izan ezik. Hiru zerokoekin hasten diren helbide globalen barneko egiturarako ez dago inongo mugarik definituta. Helbide berezi horiek erabiltzen dira, oraingoz, IPv4 helbideak IPv6 helbideetan mapeatzeko (ikusi gorago). Erabilera gehiago defini daitezke etorkizunean.

Interfazearen identifikadorea

Unicast helbideen (globalenak eta bertakoenak) interfazearen identifikadorea azkeneko 64 bitek osatzen dute, oraintxe aipatu dugun salbuespenean izan ezik. Interfaze baten identifikadoreak bakarra izan behar du aurrezenbaki bera duten interfazeen artean. Beraz, nodo baten interfaze desberdinek interfazearen identifikadore bera izan dezakete, betiere sare desberdin batera lotuta baldin badago nodoaren interfaze horietako bakoitza. Kasu horretan bi interfazeen IPv6 helbideak aurrezenbakiak bereiziko ditu.

Normalki, interfazearen identifikadorearen bitak sare-txartelaren helbide fisikotik abiatuta definitzen dira. Helbide fisiko bakoitzeko definitu behar da IPv6 interfazearen identifikadorea sortzeko prozedura. Ethernet helbideetarako, adibidez, RFC 2464 agiriak zehazten du prozedura hori.

Anycast helbideak

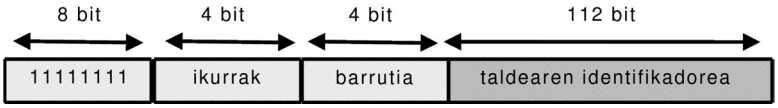
Datagrama baten helburuko helbidea anycast erako baldin bada, bideratzaileek erabaki behar dute nori bideratu datagrama hori, anycast helbide horrek identifikatzen dituen interfaze guztien artean egokiena aukeratuta. Gaur arte, ondoko bi eratako erabilerak definitu dira anycast helbide hauetarako:

- Zerbitzu bat ematen duten zerbitzari guztiak identifikatzea. Horrela eginez gero, gertuen dugun zerbitzariak erantzungo digu zerbitzu horri eskaera bat bidaltzen diogunean.
- Bideratzaile talde bat identifikatzea. Adibidez, ISP baterako sarrera diren bideratzaile guztiak identifikatuz, ISP horien bezeroek ez dute ISParen bideratzaile konkretu bat aukeratu behar. Era berean, sare baten bideratzaile guztiek konpartitu dezakete anycast helbidea.

Izan ere, azken erabilera horretarako anycast helbide bat dago aurredefinituta: azpisare baten bideratzaileen anycast helbidea (*subnet-router anycast address*) onartu behar dute sare baten bideratzaile guztiek. Helbide hori sare-helbidea bera da, hau da, sarearen identifikazioaren eskuinean dituen bit guztiak (interfazearenak barne) zerokoak dituen helbidea.

Helburu anitzeko helbideak (multicast)

IPv6 helburu anitzeko helbide batek interfaze talde bat identifikatzen du. Interfaze bera hainbat multicast taldetan egon daiteke. Multicast helbide batera igorritako datagramaren kopia bana eraman behar diote bideratzaileek taldeko interfaze bakoitzari. Beste alde batetik, ez dago multicast helbide bat agertzea datagrama baten jatorrizko helbidearen eremuan.



2.20. irudia. IPv6 multicast helbideen egitura.

Goiko irudian adierazten den moduan, helburu anitzeko helbide guztiak FF digituekin hasten dira. Geroko ikur-bitek bereizten dute ea helbidea aldi baterako (hau da, dinamikoki esleituta taldeari) edo betiko den, besteak beste. Betiko multicast helbideak IANAk gordetako helbide berezi batzuk dira. Adibide batzuk ondoko taulan dituzu:

Helbidea	*Identifikatutako taldea*
FF02::1	Sare-segmentu baten nodo guztiak
FF02::2	Sare-segmentu baten bideratzaile guztiak
FF05::2	Sare baten bideratzaile guztiak
FF0E::101	Internet osoan dauden NTP zerbitzari guztiak

2.9. taula. Gordetako IPv6 helburu anitzeko helbide batzuk.

Taldearen identifikadorearen aurrean dauden 4 bitek adierazten dute zein barruti topologikotan balio duen multicast helbideak. Bit horiek erabiliz, IPv4 helbideen difusiorako eta difusio mugaturako helbideez gain, beste barruti desberdinetarako ere difusio-helbideak defini daitezke. Hau da, topologiaren hierarkia bakoitzeko azpisarerako defini ditzakegu difusio-helbideak. Horren adibidea 2.9. taulako bigarren eta hirugarren lerroak dira.

Nodo baten helbideak eta multihoming

IPv6 nodo batek ezagutu beharko ditu beren burua identifikatzen duten helbide guztiak, interfazeei dagozkienak eta aurredefinituta daudenak. Nodo hori erabiltzailearen konputagailu bat baldin bada (hau da, ez da bideratzaile bat), honako hauek dira «bere» IPv6 helbideak:

- Bere interfaze bakoitzaren bertako unicast helbidea.
- Bere interfazeetan konfiguratutako beste edozein unicast (adibidez, globalak) eta anycast helbide.
- Loopback helbidea.
- Gordeta dauden «nodo guztietarako» multicast helbideak: nodo baten interfaze guztiak identifikatzen dituena (FF01::1) eta sare-segmentu baten nodo guztiak identifikatzen dituena (FF02::1).
- Bere unicast eta anycast helbide bakoitzetik sortzen den multicast helbide berezi bat, *Solicited-node multicast* izenekoa. Helbide hori autokonfigurazioan erabiltzen da.
- Nodoa taldekidea duten multicast talde guztien helbideak.

Bideratzaileen kasuan, aurrekoei honako hauek gehitu behar zaizkie:

- Bideratzaileari lotutako sare bakoitzeko, sare horren bideratzaileen anycast helbidea.
- Gordeta dauden «bideratzaile guztietarako» multicast helbideak: bideratzailea sareekin lotzen duten interfaze guztiak identifikatzen dituena (FF01::2), sare segmentu baten bideratzaile guztiak identifikatzen dituena (FF02::2), eta sare baten bideratzaile guztiak identifikatzen dituena (FF05::2).

Gogoratu gero eta arruntagoa dela sare batek goranzko konexio ugari edukitzea (*multihoming*). Kasu horretan, sare horren nodoek unicast helbide global ugari izango dituzte, bakoitza aurrezenbaki desberdinekin. Horrek arazo berriak sortzen dizkigu datagrama bat bidaltzean:

- Alde batetik, zein jatorrizko helbide emango diogu igorri behar dugun datagrama bati, baldin interfazeak helbide global bat baino gehiago baditu? Kontuan hartu behar da aukeratuko dugun helbide horrek definituko duela zein izango den datagramaren erantzunak jarraituko duen bidea guregana itzultzeko.
- Beste alde batetik, datagramaren helburua ere sare multikonektatu batean baldin badago, zein helbide grabatu behar dugu datagramaren helburuko helbidearen eremuan?

Aukeraketa horiek egiteko algoritmoak proposatu dira dagoeneko (RFC 3484). Hala ere, sare multikonektatuena arazo irekia da Interneten. IPv6 sareetan arazoa bideratzeko lantaldea badago Interneten (ikusi http://www.ietf.org/html.charters/shim6-charter.html).

Helbidearen idazkera	*Helbide mota*
::/128	Zehaztu gabeko helbidea
::1/128	Loopback
FFxx:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh	Multicast
FE80::hhhh:hhhh:hhhh:hhhh/64	Bertako unicast
::FFFF:d.d.d.d	IPv4-mapeatutakoak
Beste guztiak*/64	Unicast globalak

2.10. taula. IPv6 helbideen laburpena. ‘h’ karaktereak edozein 4 bit biltzen dituen digitu hamaseitarra adierazten du. ‘x’ karaktereak aurredefinituta dauden 4 bit biltzen dituen digitu hamaseitarra adierazten du. ‘d’ karaktereak 8 bit biltzen dituen digitu hamartarra adierazten du. (*) Hasierako hiru bit 000 direnak izan ezik.

2.7.2. IPv6 bideratzea

IPv4 bertsioan bezala, sistema autonomoen barneko bideratzea eta kanpoko bideratzea bereizten dira IPv6n. Bideratze-taulak osatzeko eta eguneratzeko informazioa trukatzeko erabiltzen diren protokoloak IPv4n erabiltzen diren protokolo berak dira, behar diren egokitzapenak gehituta: barruko bideratzerako RIPng (RIP *new generation*) eta IPv6rako OSPF (RFC 5340), eta BGP4 kanpoko bideratzerako. Hala ere, IPv6 trafikoa bideratzea askoz azkarragoa da IPv4koa baino, atal honen hasieran aipatutako ondoko arrazoiengatik:

- IPv6 datagramen goiburukoak sinpleagoak direnez, datagrama bakoitzaren analisia laburragoa da. Horrek datagrama-ilarak arintzen laguntzen du.
- IPv6 helbideen egitura hierarkikoak bide-elkarketa optimoa egitea ahalbidetzen du. Horrek Interneten ardatzaren taulen tamaina txikiari eusten dio, eta, beraz, datagrama bakoitzaren bideratzea ekintza azkarra izango da.

Hala eta guztiz ere, taulen tamainaren arazoa ez da guztiz argitzen bideratzezuloen kontua konpontzen ez bada. Gogoratu IPv4 sare batek aldatzen duenean bere Internet hornitzailea (bere goranzko konexioa, alegia), zuloak sortzen direla bere hornitzaile zaharrak egindako bide-elkarketan, eta bide berri bat gehitu behar dela tauletan aldaketa egin duen sare horretarako. Hau izan da azken urte hauetan ardatz-sareko taulen hazkundearen arrazoi nagusietako bat. IPv6 bertsioan arazoa berriro gerta ez dadin, helbideen kudeaketan aldaketa handia egin da: unicast helbide globala ez zaie esleitzen nodoei, uzten baizik. Hau da, Internet hornitzailea

aldatzen badugu, gure sareko helbide globalak ere aldatu beharko ditugu. Horri sarea **birzenbakitzea** deitzen zaio. Eskuz egitea ez da bideragarria; horregatik definitu dira prozedurak era automatikoan egiteko.

IPv4n bezala, bideratze-taulak agindutako interfazetik datagrama bat bidaltzeko, trama baten barruan sartu behar dugu datagrama hori, eta, noski, trama horri helburuko helbide fisikoren bat eman beharko diogu. Helbide fisiko hori lortzeko ARP protokoloa erabiltzen dugu IPv4n; IPv6n, aldiz, ez dago ARPv6 izeneko protokolorik. Haren lana ND izeneko protokoloak betetzen du (Neighbor Discovery, RFC 4861). Protokolo hori IPv6 helbideen eta helbide fisikoen arteko itzulpena egiteko ez ezik, sare-segmentu bereko bizilagunak ezagutzeko ere erabiltzen da. Beharrezko laguntzailea izango dugu ND protokoloa gure saretik ateratzeko atebideak zein bideratzaile diren jakiteko, baita sare-konfigurazioa ezagutzeko eta, oro har, bideratze eta autokonfigurazioarekin zerikusirik duten hainbat lan betetzeko ere.

Autokonfigurazioa eta birzenbakitzea

IPv4 sareetan erabiltzen den autokonfigurazioa DHCP zerbitzarietan oinarritzen da. Zerbitzari horrek gordetzen ditu egindako konfigurazioak, bi nodok IP berdina erabiltzen ez dutela bermatzearren. IPv6 sareetan ere badago era bereko autokonfigurazioa egitea, DHCPv6 protokoloa erabiliz (RFC 3315). Horrez gain, zerbitzaririk gabeko autokonfigurazioa egitea ere badago IPv6 sareetan. Horren oinarria unicast bertako helbideak dira, helbide horiek era automatikoan sortzen direlako helbide fisikotik abiatuta, beste inongo konputagailurekin elkarrekintzarik gabe, eta ahalbidetzen dutelako sare-segmentu bereko beste konputagailuekin komunikatzea (gure *bizilagunekin*, alegia). Konfigurazio osoa egiteko (helbide globalak, atebideak, DNS zerbitzariak) arestian aipatutako ND protokoloa erabiltzen da, behin unicast bertako helbidea sortu eta gero. Autokonfigurazioa RFC 4862 agiriak deskribatzen du. Laburtuta, honako hauek dira eman beharreko urratsak:

1. Sortu unicast bertako helbidea, interfaze mota horretarako definitutako prozedurari jarraituz.
2. Egiaztatu beste inor ez dela erabiltzen ari bertako helbide bera gure sare-segmentuan (gerta baitaiteke). Horretarako ND protokoloa erabiltzen da.
3. Entzun segmentuko bideratzaileek ND protokoloa erabiliz zabaltzen duten konfiguraziorako informazioa, eta, hortik abiatuta, osatu konfigurazioa. Horretarako ND, ICMPv6 eta DHCPv6 protokoloak erabiltzen dira.

Ikusten denez, segmentuko bideratzaileek hartzen dute, neurri handi batean, IPv4n DHCP zerbitzariek egiten duten papera. Horrela, nodo baten konfigurazio globala bere atebideek zuzentzen dute, helbide globalen hierarkizazioa bermatuz.

Autokonfiguraziorako mekanismo honek ahalbidetzen du sarearen birzenbakitzea gure goranzko konexioa aldatzen denean. Horrelako aldaketa bat

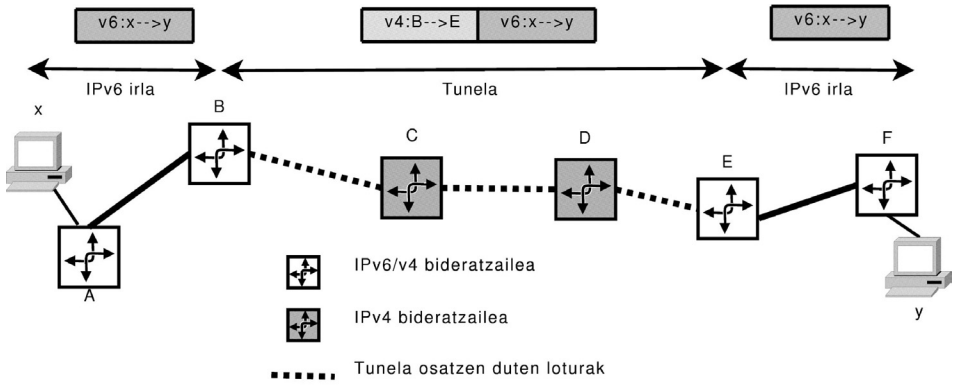
bideratzaileren baten konexioak aldatzearekin batera egiten da. Konexioa egiten duen interfazea automatikoki birkonfiguratuko da, bere helbide globala berria, hau da, bere aurrezenbaki berria, konexioaren beste aldean dagoen bideratzaileak emandako informazioa erabiliz. Gero, konfigurazio berri hori zabaldu behar da lehenbailehen bere azpisarean, nodoek ere birkonfiguratzeko. Prozedura RFC 2894 agirian dago deskribatuta.

2.7.3. IPv4-IPv6 trantsizioa

Urte luzeetan elkarrekin bizi beharko dute IPv4k eta IPv6k Interneten, eta, bitartean, bi mundu horien arteko komunikazioa bermatu behar da. Trantsizio horretan hiru motatako makinak aurkituko ditugu:

- IPv6 soilik ulertzen dutenak. Beraien artean komunika daitezke.
- IPv4 soilik ulertzen dutenak. Hauek ere beraien artean soilik hitz egin dezakete.
- IPv4/v6 ulertzen dutenak. Beste guztiekin hitz egin dezakete.

Gaur egun sortzen diren sistema eragileen bertsio berriek bi inplementazioak, v4 eta v6, dauzkate. Hala ere, oraindik IPv4 da gehienbat erabiltzen den inplementazioa, aplikazioek hori aukeratzen dutelako. Gaur egun, IPv6 sareak irla txikiak dira IPv4 itsasoan. Irla horien arteko IPv6 datagramak trukatzeko, tunelak erabiltzen dira, ondoko irudian adierazten den moduan.



2.21. irudia. IPv6 tunelak IPv4 Interneten.

Tunelen ideia honako hau da: IPv4 itsasoan ibiltzeko, sartu IPv6 datagrama bat IPv4 beste datagrama baten barruan, zeinak txaluparena egingo duen. Hori da irudian x eta y konputagailuen arteko komunikazioan egindakoa. Horietako konputagailu bakoitza IPv6 irla batean dago kokatuta, eta bi irlen arteko bideak, nahitaez, IPv4 mundua zeharkatu behar du. Irudian, bide hori B eta E IPv6 bideratzaileen

artekoa da. Bi bideratzaile horien arteko komunikazioak, nahitaez, IPv4 datagramek garraiatu behar dituzte, C eta D IPv4 bideratzaileak zeharkatu behar baitira. Jatorrizko datagramak, x konputagailuak igorritakoak, IPv6 formatukoak dira, eta beraien jatorrizko eta helburuko interfazeak x eta y konputagailuenak dira, hurrenez hurren. Horietako datagrama bat B bideratzaileak C-rantz birbidali baino lehenago, IPv4 datagrama baten informaziorako eremuan sartuko du. IPv4 datagrama horren jatorrizko eta helburuko helbideak B-rena eta E-rena izango dira, hurrenez hurren. C-tik D-raino helduko da IPv4 datagrama, tunelean zehar, eta D-k E bideratzaileari birbidaliko dio. E bideratzaileak jasoko du IPv4 datagrama hori, baina ez du birbidaliko; erauziko du bere barnean dagoen IPv6 datagrama eta hori birbidaliko du F bideratzailerantz.

IPv6 zabaltzearen atzerapena

Igaro dira urteak IPv6 definitu zenetik, eta oraindik oso gutxi zabalduta dago. Ondoko hauek dira atzerapen horren arrazoi nagusiak:

- Haren sorrera bultzatu zuen lehenengo arazoa, IPv4 helbideak agortzea, ez da gertatu, CIDR, NAT eta DHCPren erabilerari esker. Denek diote hauek behin-behineko irtenbideak direla, epe mugatu baterako, baina badirudi epe hori ez dela oraindik amaitu.
- Bigarren arazo nagusia, Interneten ardatz-sareko bideratze-taulen ikaragarrizko hazkundea eta tamaina, hor dago. Baina bideratzaileen teknologiaren aurrerapenak eutsi egin dio arazoari eta, oraingoz, Internet badabil. Beste alde batetik, Interneten topologiaren beheko mailetan (Tier2 eta ISP mailetan) trukaguneen erabileraren zabaltzeak ere, Tier1 sareetaraino ailegatzen den trafiko kopurua murrizten du, arazoa horrela arinduz.
- IPv6 estandarren definizioa ez da izan behar bezain egonkorra, eta mesfidantza sortu du. Lehenengo agiriak 1995. urtean argitaratu ziren, baina geroztik berrikuspen ugari egon dira. Adibidez, hain garrantzitsua den helbideratze-sistemaren definizioa 1995eko abenduan definitu zen (RFC 1884), baina geroztik 3 aldiz aldatu egin da, 1998, 2003 eta 2006 urtetan. Aldakortasun horrek atzeratu egin du inplementazio sendoak agertzea. Oraindik ere badaude guztiz argituta edo definituta ez dauden IPv6 inguruko gai batzuk (adibidez, multihoming tratamendua).

Oro har, IPv4 dabilen bitartean, inork ez du arriskurik hartu nahi. IPv6rako migrazioak kostuak ditu (sistema eragileak eta aplikazioak eguneratzea erabiltzaileen ekipoetan eta zerbitzarietan, bideratzaileak aldatzea, eta testuinguru berria erabiltzen ikastea, batez ere sare-kudeatzaileek), eta aldaketa gauzak dauden bezala uztea baino garestiagoa den bitartean, inor ez da mugitzen. Hala ere, instituzio publikoak ari dira arriskua bere gainean hartzen, eta migrazioa bultzatzen ari dira beraien sare informatikoetan IPv6 erabilera behartuz.

LABURPENA

Sare desberdinak elkarrekin konektatzeko sarearte horretan parte hartzen duten guztiek onartzen eta erabiltzen duten protokolo komun bat behar da. Protokolo hori IP da, Interneten bihotz teknologikoa dena.

Sarearte-mailako protokoloa da IP. Bere informazio-unitatea datagrama da, sarbide-mailako trametan sartzen direnak. Garraio-mailari zerbitzu bakarra eta bakuna eskaintzen dio IP sarearte-mailak: datagramak sareko mutur batetik bestera eramatea. Konexiorik gabeko zerbitzua da, inongo bermerik ematen ez duena. Hau da, gerta daiteke datagramak ez ailegatzea beren helburura, errepikatuta ailegatzea, atzeratuta, edo ordena aldatuta. ICMP protokoloa IPren laguntzaileetako bat da, IP entitateen arteko kontrol-mezuak elkarri bidaltzeko erabiltzen dena.

IP protokoloaren definizioaren barne IP helbideen definizioa dago. Helbide hauek sare-interfazeak, hau da, sare-txartelak, identifikatzen dituzte. Bi zati ditu IP helbide batek, maskarak bereizita: sare-identifikazioa eta interfazearen identifikazioa. Sarearen identifikazioa egituratuta dago, azpisareak identifikatzeko. IP helbideak publikoak edo pribatuak izan daitezke. ICANNek banatzen ditu IP helbide publikoak, baina ez du zuzenean lan hori egiten. Tokiko erakundeei, Interneteko erregistratzaileei (RIR eta LIR), uzten die lan hori. Hala ere, gero eta gehiago erabiltzen dira IP helbide pribatuak, NAT eta DHCPrekin batera. Horrek Internet publikoan ibiltzea sareko makina guztiei ahalbidetzen die, baina IP publiko gutxi batzuk, agian bakarra, erabiliz.

Datagramak bideratzeko bideratze-taulak erabiltzen dituzte IP entitateek. Taula horietan bilatzen dute datagramak duen helburuko helbideari dagokion bidea, hau da, zein bideratzaileri birbidali behar dioten datagrama. Gero, hurrengo bideratzaile horren IP helbideari dagokion helbide fisikoa lortuko dute ARP erabiliz, eta helbide fisiko horretara birbidaliko dute datagrama garraiatzen duen trama.

Bideratze-taulak oso bestelakoak dira Interneten topologiako hierarkia-maila desberdinetan. Erabiltzaileen sareetan eta ISPetan, beste sareekiko konexio gutxi batzuk besterik ez daude, eta taula txikiak dira. Baina Tier1 mailako bideratzaileetan, ehun milaka bide gordetzen dira tauletan, datagramen bideratzea motelduz eta kongestioak sortuz. Internet zeharreko datagramen bideratzea arintzeko trukaguneak sortu egin dira, hau da, topologiako beheko mailetako sareen (Tier2 eta ISPak) arteko zirkuitulabur topologikoak.

Bideratze-taulak txikiak direnean, eskuz edo automatikoki (DHCP erabiliz) betetzen dira. Handiak eta aldakorrak direnean, automatikoki bete eta eguneratu behar dira, bideratze-protokoloak erabiliz. Protokolo horiek IP entitateen artean bideratzeko informazioa trukatzeko erabiltzen dira. Informazioa trukatu nahi duten bi IP entitateak sistema autonomo berean badaude, barruko bideratze-protokoloak

erabiltzen dira (RIP eta OSPF). Sistema autonomo desberdinetakoak badira, BGP kanpoko bideratze-protokoloa erabili behar da.

Gaur egun erabiltzen den IP protokoloaren bertsioa laugarrena da (IPv4). Bertsio horrek dituen helbideratze- eta bideratze-arazoak konpontzearren, IPv6 ondorengo bertsioa definitu egin da. Helbideratze-ahalmen ikaragarria du IPv6k, eta helbide horien egituraketa guztiz hierarkikoa da, zenbaki telefonikoen egituraren antzekoa. Horrela, helbidearen aurrezenbakiak helburura heltzeko bidea adierazten du. Helbideen egitura horrek, gehi sareen autokonfigurazio eta birzenbakitze automatikoak, asko errazten du bideratzea, taulen tamaina mugatua mantenduz. IPv6 ez dago oraindik oso zabalduta, baina badirudi etorkizuneko Internet IPv6 Internet izango dela.

3. Garraio-zerbitzuak eta protokoloak

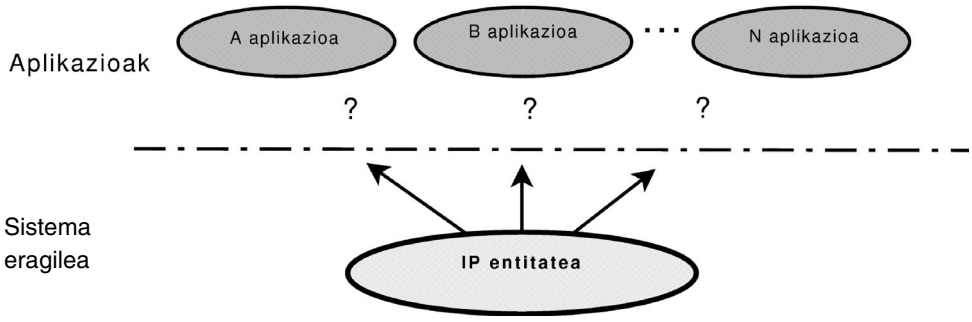
Kapitulu hau irakurri eta gero, irakurleak jakingo du:

- Zer diren portuak eta nola erabiltzen diren.
- Nolako zerbitzuak lor ditzakeen aplikazio-mailak garraio-mailatik.
- Nolakoa den UDP protokoloa.
- Nolakoa den TCP protokoloa: haren errore-kontrola, fluxu-kontrola, konexioak kudeatzeko modua, kongestio-kontrola eta abar.

3.1. GARRAIO-ZERBITZUAK

Aplikazioen identifikazioa

Demagun konputagailu batean sare-aplikazio bat baino gehiago abiatu ditugula. Adibidez, web orri handi bat jaisten den bitartean, aprobetxatu dugula gure posta-zerbitzarira atzitzeko, mezu batzuk bidaltzeko eta jasotzeke daudenak jaisteko. Sareko bi ekintza horiek martxan daudela, datagrama bat gure konputagailura heltzen denean, nola jakingo du IP mailak zein aplikaziori eman behar dion datagramak garraiatzen duen informazioa (ikusi 3.1. irudia)?



3.1. irudia. Nola jakin zein aplikazioari eman behar zaion jasotako datagrama?

Itxuraz, IP datagramaren *goiko protokoloa* izeneko eremua horretarako erabiltzen da (ikusi 2. kapituluaren 2.2. atala). Baina hori ez da nahikoa, arrazoi hauengatik:

1. IP datagramaren eremu horrek 8 bit besterik ez du. Horrekin 256 aplikazio identifika daitezke, besterik ez. Gaur egun askoz aplikazio gehiago erabiltzen dira sarean.

2. Zenbaki bat ez da nahikoa helburuko aplikazio-entitatea zein den identifikatzeko. Demagun ez garela web orri bat jaisten ari, baizik eta bi web orri, une berean, bi arakatzaile desberdin erabiliz. Web aplikazioaren identifikadorea daraman datagrama heltzen denean, zein web entitateri emango zaio (zein arakatzaileri, alegia)? Eta bi orri horiek jaisteko arakatzaile bakarra erabiltzen dugunean, nola jakingo du zein orriri dagokion jasotako datagramaren informazioa?

3. Helburuko aplikazioa identifikatzea ez da nahikoa. Askotan, datagrama bat hartzen duen aplikazioak datagrama hori nork bidali dion jakin beharko du (erantzuteko, adibidez). Horretarako jatorrizko aplikazioko entitatea zein den ere identifikatu beharko dugu.

IP mailan bertan helburuko aplikazio-entitatea identifikatzea bazegoen, baina IP diseinatu zutenean, IPren eta aplikazioaren artean beste maila bat txertatu beharko zela argi zegoenez, beste maila horren goiburukoetan identifikazio hori egitea erabaki zuten. Hau da, garraio-mailan egingo da aplikazioaren identifikazioa, zeren, maila honek, eta ez IP mailak, emango baitio zerbitzua aplikazio-mailari. Garraio-mailako protokoloak TCP eta UDP direnez, beraien goiburukoetan aurkituko ditugu jatorrizko eta helburuko aplikazioen identifikadoreak. Identifikadore horiek **portuak** dira.

Portuak

16 biteko zenbakiak dira. UDPrako 2^{16} portu daude, eta beste hainbeste TCPrako. Ondoko hiru motatako portuak definitu dira (ikusi http://www.iana.org/assignments/port-numbers):

- Portu ezagunak (*well-known ports*). Hauek 0 eta 1023 artekoak dira (biak barne). IANAk oso zabalduta dauden aplikazioei esleitutako portuak dira. Eskaeren zain dauden aplikazio-entitateek erabiltzen dituzte, zerbitzariek alegia. Sistema eragile gehienek mugatzen dute zein prozesuk duten portu hauek erabiltzea: sistemako prozesuei (*root*) edo ahalmen handiko erabiltzaileei besterik ez zaie portu hauek erabiltzeko baimena ematen. Adibide bat HTTP zerbitzarientzako 80 portua da.

- Portu erregistratuak (*registered ports*). 1024-49151 artekoak dira. Aurrekoekiko aldea oso mehea da: oso erabiliak diren aplikazioen zerbitzariek erabiltzen dituzte portu hauek, baina IANAk ez ditu esleitzen, «erregistratzen» baizik. Portu ezagunekiko beste aldea honako hau da: edozein prozesuk, eta ez sistemak edo ahalmen handikoek soilik, erabil ditzakete. Adibide bat da SOCKS protokoloa erabiltzen duten zerbitzarientzat erregistratutako 1080 portua da.
- Portu dinamikoak edota pribatuak (*dinamic/private ports*). Beste guztiak dira: 49152-65535. Aplikazioen bezeroek erabiltzen dituzte, baita erregistratuta ez dauden aplikazioen zerbitzariek ere.

Definizio ofizialetik harago, lehenengo bi taldeek portu erreserbatuen kategoria osatzen dute, eta hirugarrenak, portu libreena. Oro har, portu erreserbatuak aplikazioen zerbitzariek erabiltzen dituzte, eta libreak, aldiz, bezeroek.

Portu erreserbatuen eta libreen arteko banaketaren zergatia ulertzeko, ikus dezagun aplikazio banatu baten bezero baten eta zerbitzari baten arteko elkarrekintza nolakoa den. Normalki, eskaera bezeroak egingo dio zerbitzariari. Horretarako erabili beharko du portu bat, jatorrizko portua. Hori edozein izan daiteke, portu libreen artean, eta alda daiteke saio batetik bestera. Gainera, aplikazioko bezeroak jakin beharko du zein den helburuko portua, zerbitzariak erabiltzen duena, alegia, eskaera hara bidaltzeko. Horretarako erabiltzen dira portu erreserbatuak: bezeroak jakingo du, inori galdetu gabe, helburuko konputagailuko zein porturi bidali behar dion bere eskaera, aplikazio horren zerbitzariak beti —edozein konputagailutan egikarituta ere— erabiltzen duelako berarentzat erreserbaturik dagoen portua.

Portu erreserbatuen zerrenda Linux konputagailu batean ikusteko, egin *more /etc/services*. Bestela, joan http://www.iana.org/assignments/port-numbers web orrira, eta hor ikusiko dituzu erreserbatutako portu guztien esleipena eta aurreko hiru kategorien definizioa (*well-known ports, registered ports,* eta *dynamic/private ports*).

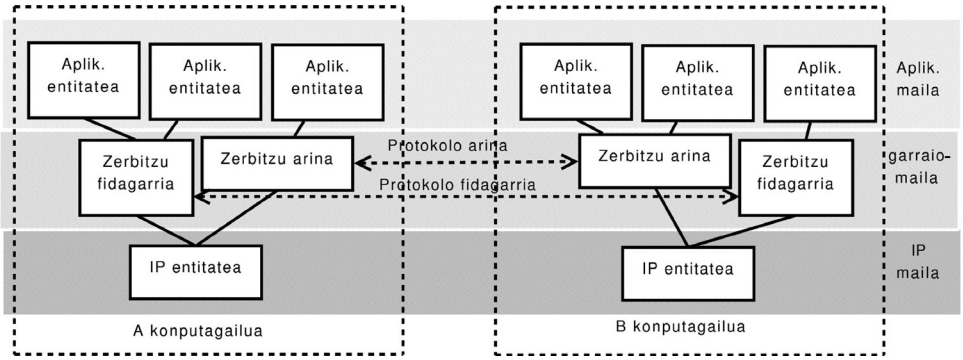
Garraio-mailako zerbitzuak

Mailakako komunikazio-arkitektura batean, maila bakoitzak goikoari ematen dion zerbitzuak beti gehitzen dio zerbait beheko mailatik jasotako zerbitzuari. Adibidez, aurreko kapituluan aztertu dugun TCP/IP arkitekturako sarearte-mailak ematen duen zerbitzuak sarbide-mailak ematen duen zerbitzua (sare berean dauden bi konputagailuren artean informazioa mugitzea) zabaltzen du, eta sare berean ez dauden konputagailuen artean datagramak mugitzea ahalbidetzen du. Hasiera batean, garraio-mailak zerbitzu honi gehituko diona aplikazioak bereiztea izango da: TCP/IP arkitekturako garraio-mailak ematen duen zerbitzua bi aplikazio-mailako entitateren artean (eta ez bi konputagailuren artean, besterik ez) informazioa mugitzea izango da. Horretarako, noski, IP zerbitzua erabiliko du, informazioa

sareartean zehar konputagailu batetik bestera eramateko. IP mailak datagrama helburuko konputagailura eramanda, bere lana bukatu du, eta garraio-mailako entitateari emango dio datagramaren barnean dagoen informazioa. Garraio-mailako entitateak bereiziko du bere konputagailuan egikaritzen ari diren aplikazioen artean zeini dagokion jasotakoa, eta helaraziko dio.

IP datagrama-zerbitzuari aplikazio-mailako entitatea bereiztea besterik gehitzen ez dion zerbitzua gauzatzeko, UDP protokoloa erabiltzen du garraio-mailak. Zerbitzu horrek IP zerbitzuaren ezaugarri nagusia heredatzen du: datagrama erako zerbitzua da, hau da, *best effort* erako zerbitzua. Baina aplikazio askorentzako zerbitzu hori ez da nahikoa. Oro har, bidalitako datuen osotasuna lehentasuna denean (web, posta elektronikoa…) hobe da fidagarritasuna ematen duen garraio-zerbitzua jasotzea. Aplikazio horietarako bigarren garraio-zerbitzu bat ere definitu egin da, IP zerbitzuari fidagarritasuna gehitzen diona. Zerbitzu fidagarri hori TCP protokoloa erabiliz gauzatzen da. Laburbilduz, TCP/IP arkitekturako garraio-mailak honako **bi zerbitzu** hauek eskaintzen dizkio aplikazio-mailari:

- Edozein konputagailutan kokatutako bi aplikazioko entitateren artean informazioa mugitzea, era arinean eta inongo bermerik gabe. Hau da UDP zerbitzua.
- Edozein konputagailutan kokatutako bi aplikazioko entitateren artean informazioa mugitzea, era fidagarrian. Hau da TCP zerbitzua.



3.2. irudia. Entitateak, zerbitzuak, eta protokoloak garraio-mailan. IP datagramaren *Goiko protokoloa*ren eremuak datagrama zein entitateri eman behar zaion adierazteko balio du. Irudi honetan sare-arkitekturako beheko mailak ez dira agertzen.

Erabiltzailearen konputagailu batean garraio-mailako entitate bat topatuko dugu zerbitzu bakoitza emateko. Fidagarritasuna behar duen aplikazio batek zerbitzu fidagarria eskaintzen dion entitateari emango dizkio bere bidalketak. Arintasuna behar duen aplikazioak, aldiz, datagrama-zerbitzua ematen dion

entitateari pasatuko dio bidali nahi duen informazioa. Garraio-entitate bakoitzak bere protokolo propioa erabiliko du bere zerbitzua gauzatzeko (batak TCP, besteak UDP), baina biek erabiltzen dute sarearte-mailako IP zerbitzua. Hori guztia 3.2. irudian ikus daiteke.

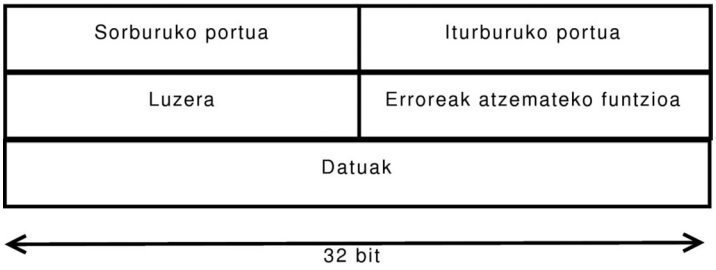
3.2. UDP

UDP zerbitzua

UDP protokoloaren bidez gauzatzen den garraio-zerbitzua datagrama erakoa da, eta, beraz, konexiorik gabekoa. Aplikazio-mailako entitate batek UDP entitateari datu multzo bat ematen dionean, UDP segmentu bakar bat eraikitzen da, eta segmentu hori IP datagrama bakar batean kapsulatu eta bidali egiten da. Datagrama horri ez zaio inolako jarraipenik egiten: bere helburura heltzen ez bada, UDP entitatea ez da jabetuko.

UDP protokoloa

Horrelako zerbitzu xumea emango duen protokoloa, halabeharrez, xumea izango da. RFC 768 agiri zahar eta laburrean dago definituta. Alde batetik bere goiburukoa definitzen da, eta beste aldetik prozedurak: aplikazioko informazioa bidaltzea eta hartzea, besterik ez. Aplikazioko mezu bat bidaltzeko, datuak jaso, UDP goiburukoa eraiki, eta IP mailari UDP datagrama ematea besterik ez da egin behar. Informazioa jasotzeko, beste hainbeste: UDP datagrama jaso, UDP goiburukoa kendu, eta dagokion aplikazioari datagramaren edukia eman. Finean, protokolo minimo bat da: bere betebehar bakarra jatorrizko eta helburuko aplikazio-entitatea identifikatzea da. Horretarako UDP portuak erabiltzen dira. Kontuan izan TCP eta UDP garraio-entitateak independenteak direnez, bakoitzak bere portu multzoa kudeatzen duela. Dena dela, aplikazio-zerbitzari bat bi zerbitzuren bidez atzigarri dagoenean, TCPn eta UDPn eskuarki portu-zenbaki bera erabiltzen du (kontuz: *portu-zenbaki* bera, baina ez *portu* bera).



3.3. irudia. UDP datagramaren egitura.

3.3. irudian dituzu UDP goiburukoaren eremuak. *Luzera* eremuak UDP goiburukoaren luzera gehi datuen luzera adierazten du, bytetan. Bere gutxieneko balioa 8 da, daturik gabeko UDP datagramak bidaltzea baitago. Eremu hau soberan dago, mailen arteko interfazeen kontua baita datu hori. Hau da, garraio-mailak aplikaziotik bidaltzeko byte multzo bat jasotzen duenean, aplikazio berak esan beharko dio zenbat byte dauden multzo horretan. Era berean, IP mailak UDP entitate bati datagrama batetik ateratako byte multzo bat (UDP datagrama bat) ematen dionean, multzo horren luzera zein den adieraziko dio.

Teorian, UDP datagrama baten gehienezko luzera IP datagramaren luzerak ezartzen du (65.535 byte). Horri IP goiburukoaren luzera minimoa (20 byte) eta UDP goiburukoaren luzera (8 byte) kentzen badizkiogu, erabiltzaileak UDP segmentu batean asko jota 65.507 datu-byte bidal ditzakeela aurkituko dugu. Baina errealitatean, sistema eragileak muga estuagoak ezartzen dizkie UDP bidalketei, UDP erabiltzen ari den aplikazio bakoitzarentzat 64 KB-eko bufferra erreserbatzea gehiegizkoa delako. Gehienetan, UDP segmentuentzat 8.192 byteko bufferra gordetzen da.

Erroreak atzemateko kalkulua IPn egiten denaren antzekoa da, baina UDPn haren erabilera hautazkoa da. Softwarearen bidez igorleak kalkulatutako balioa da, hartzaileak egiaztatuko duena. UDPren kasuan, hartzaileak kalkulatutakoa ez badator bat eremu horretan igorleak bidalitakoarekin, segmentua deuseztatzen da, aplikazioa ezertaz ohartarazi gabe. Kalkulu horretan goiburukoa eta datuak erabiltzen dira.

Errore-atzemate honen erabilgarritasuna kolokan dago. IP azpian dauden sare gehien-gehienetan hau baino fidagarriagoa den errore-atzematea egiten denez, ez du zentzu handirik garraio-mailan lan hori berriro egiteak. Aplikazioak UDP entitateari errore-kontrol hau gaitzeko edo desgaitzeko eska diezaioke. Interneteko estandarrek errore-kontrol hau besterik ezean gaituta egotea ezartzen dute.

3.3. TCP

3.3.1. TCP zerbitzua

Garraio-mailako zerbitzu fidagarria TCP protokoloaren bidez ematen da. Hau da, TCPk bi aplikazioko entitateren artean bidalitakoa ondo ailegatuko dela bermatzen du. «Ondo» horrek zer esan nahi duen ulertzeko, ikus dezagun zeintzuk diren datagrama batek topa ditzakeen arazoak:

- Datagramak galtzea. Datagramak sareartean zehar egindako bidaian gal daitezke ondoko uneetan:
  - Bideratzaileetan: bideratzaile bat gainezka dagoenean (kongestioa sortu dela, alegia), datagramak deuseztatzen ditu. Bideratzaile hori jatorra baldin bada, igorleari jakinaraziko dio ICMP mezu baten bidez, baina

askotan ez du ezta hori ere egingo. Gaur egungo Interneten beren helburura heltzen ez diren datagrama gehienak bideko makinetan sortutako kongestioetan galtzen dira.

Datagramaren TTLa agortzen denean ere, datagrama hori bideratzaile batek deuseztatuko du.

– Bideko lineetan: IPk ez dio ezer exijitzen bere azpian dagoen sarbide-mailari, eta, beraz, transmisio-erroreak maila horretan ez zuzentzea gerta daiteke. Bi bideratzaileren artean dagoen sareak urrats bakoitzean linea-kontrol zorrotzak egiten baditu, transmisio-erroreak zuzenduko dira. Baina errore-zuzenketak ez dira egiten sare eta linea guztietan; are gehiago, askotan ez dira egiten. Ethernet sareetan, adibidez, transmisio-erroreak atzematen dira, baina ez dira zuzentzen. Kasu horietan, trama batek erroreak baditu, trama horrek daraman datagrama ez zaio IP mailari emango.

– Hartzailearen konputagailuan: aplikazioak ez badu hartzeko bufferra husten igorleak betetzen duen baino arinago, buffer hori beteko da, eta heltzen den hurrengo datagramarako tokirik izango ez duenez, datagrama hori baztertua izango da. Azken finean, hau kongestioaren kasu partikularra da, hartzailearen konputagailuan suertatutakoa (eta ez bideratzaile batean). Hala ere, haren tratamendua bideratzaileetan gertatutako kongestioei ematen zaienaz bestelakoa da.

• Datagramen hurrenkeraren aldaketa: gerta daiteke datagramak iturburutik ateratzean zuten hurrenkeraz bestelako hurrenkera batean heltzea helburura. Adibidez, jarraian bidalitako bi datagramaren artean bideko bideratze-taularen batean aldaketaren bat suertatzen bada, gerta daiteke bigarren datagrama lehenengoa baino azkarrago heltzea, taularen aldaketak arinagoa den bide berri batetik bideratzen badu bigarren datagrama hori. Beste adibide bat: sare batetik irteteko bi goranzko linea daudenean, eta bi lineen artean trafikoa banatzen bada, posible da kanpora doazen eta hurrenkeran bidaltzen diren bi datagramek bide berdina ez hartzea. Horrela bada, lehenengoa atera zena bigarren heltzea ere gerta daiteke.

• Datagramen errepikapenak: gerta daiteke datagrama bakar bat igortzea, eta horren kopia bat baino gehiago hartzea bere helburuan. Fenomeno honen zergatia laster aztertuko ditugun TCP protokoloaren mekanismoetan datza normalki.

TCP zerbitzuak bermatzen du igorritako datagramak beren helburura ailegatuko direla, eta, halaber, datagramak beraien jatorrizko hurrenkeran eta errepikapenik gabe entregatuko zaizkiola TCP zerbitzuaren erabiltzaileari (aplikazioari, alegia). Berme horiek guztiak emateko, TCPk estuki zelatatu beharko du igorritako datagramen korrontea. Horretarako, ondoko teknika hauek erabiltzen ditu:

- Sekuentzia-zenbakiak erabili. Igorritako byte bakoitza identifikatuko du TCPk. Horrela, hartzaileak atzemango du noiz ez duen jaso igorritakoaren zatiren bat (hartutako byteen sekuentzia-zenbakien hurrenkeran hutsuneak agertuko dira), noiz jaso dituen bidalketaren baten kopia bat baino gehiago (sekuentzia-zenbaki bera duten byteak jasoko ditu), eta noiz aldatu den informazioaren hurrenkera bidean (jasotako byteen hurrenkera ez da zuzena izango).
- Birtransmititzeko tenporizadoreak, ACK (*Acknowledgment*, edo jaso-agiria), eta birbidalketak erabili. Igorritako byte segida bakoitzeko, TCPk tenporizadore bat abiatzen du. Tenporizadorea agortzen bada informazioaren helburuak itzulitako ACKrik jaso gabe, TCP igorleak galdutzat joko du bidalitakoa, eta byte segida birtransmitituko du. Transmisioaren eraginkortasuna hobetzeko, hau da, transmisio-abiadura fisikoa ahal den hoberena baliatzeko, TCPk ez du itxarongo igorritako byteen ACK jaso arte hurrengo byte segida igortzeko. Horri leiho mugikorra erabiltzea deitzen zaio.
- Fluxu-kontrola. Igorleak hartzaileak informazioa prozesatzeko abiadura baino azkarrago bidaltzen baditu byteak, hartzailearen jasotzeko bufferra beteko da, eta, heltzen den hurrengo bytetarako tokirik egongo ez denez, byte horiek galduko dira. Hau da, igorleak hartzailea itoko du. Hori ekiditeko, hartzailearen TCP entitateak izango du igorlea geldiaraztea, kreditu-sistema baten bidez. Haren funtzionamendua oso sinplea da: hartzaileak byte multzo bat jasotzen duen bakoitzean, dagokion ACKrekin batera bere bufferrean zenbat byterentzako tokia gelditzen zaion jakinaraziko dio igorleari. Hau da, hartzaileak igortzeko *kreditua* emango dio igorleari. Kreditu hori agortzen badu, igorleak bidalketa gelditu beharko du kreditu gehiago jaso arte.

Teknika horiek gauzatzeko, komunikazioa era kontrolatu batean antolatu beharko du TCPk. Ondokoak beharko ditu:

- Bi muturrek komunikazioaren ezaugarriak ezarri beharko dituzte transmititzen hasi baino lehen. Gutxienez, hasierako sekuentzia-zenbakia zein izango den adostu behar dute (beti berdina izatea ez da ideia ona izango, arrazoi desberdinengatik), baita hasierako kreditua eta bidalketa bakoitzaren tamaina maximoa ere. Tenporizadoreen balioa aldakorra izango da, bidalketa bakoitzari dagokion ACK itzuli arte emandako denboraren arabera. Beraz, denbora horren aurreko neurketa ere egin beharko da informazioa transmititzen hasi baino lehen. Gainera, konexioaren identifikadorea ezarri behar da, konexio honen bidez egingo diren bidalketa guztiak beste bidalketetatik bereizteko.

- Bidalitako datagrama bakoitzeko honakoak bermatu behar dira: datagrama bere helmugara ailegatzen dela, haren kopia bakarra onartzen dela, eta hurrenkera zuzenean hartzen dela.
- Komunikazioaren alde batek ezin du komunikazioa bukatutzat jo beste aldeak bere oniritzia eman arte. Horrela bermatuko dugu benetan jaso dela bidalitako informazio guztia.

Behar horiek asetzeko modurik egokiena komunikazioa konexioen bidez antolatzea da. Hau da, datuak bidali aurretik, bi muturren arteko konexioa ezarri behar da. Gero, konexio horren bidez bidaliko da informazioa. Bukaeran, bi muturren arteko adostasunarekin, konexioa amaituko da. Horregatik, TCPk emandako zerbitzua, IPrena eta UDPrena ez bezala, konexio bidezko zerbitzu bat da. Hau da, aplikazio batek, igorri nahi duen informazioa TCP entitateari eman baino lehenago, eskatu beharko dio TCP entitate horri konexio bat ezartzeko informazioaren hartzailearekin. Konexioa ondo ezartzen bada, orduan bidaltzen hasi ahal izango du igorleak. Bidalketa bukatu ondoren, TCP entitateari eskatu beharko dio konexioa bertan behera uztea. TCPk agindutakoa beteko du informazio guztia bere helburura heldu dela bermatu ostean.

3.3.2. TCP protokoloa

TCP/IP sareartearekin konektatutako erabiltzaileen konputagailu guztiek TCP entitate bat dute martxan. Entitate hori sistema eragilearen zati diren programa batzuek osatzen dute. Programa horiek egiten dutena da TCP protokoloa deskribatzen duten RFC agiriek definitzen dutena. Atal honetan RFC horien laburpen bat egingo dugu, hau da, deskribatuko ditugu TCP protokoloaren ezaugarriak.

TCPren ezaugarri nagusiak

Honako hauek dira:

- Zerbitzua konexio bidezkoa denez, protokoloak konexioaren bidez gauzatzen du[18] komunikazioa. Hau da, TCP entitate batek beste TCP entitate bati ezer bidali baino lehen, bere asmoa jakinaraziko dio, eta transmisio horren baldintzak negoziatuko dituzte. Horri konexioa ezartzea deitzen zaio. Behin konexioa ezarri eta gero, datuak elkarri bidaltzen dizkiote, eta, bidaltzeko besterik ez dagoenean, konexioa amaitu egiten dute.
- TCP konexioak duplex erakoak dira. Duplex izateak esan nahi du trafikoa aldi berean bi noranzkoetan joan daitekeela.

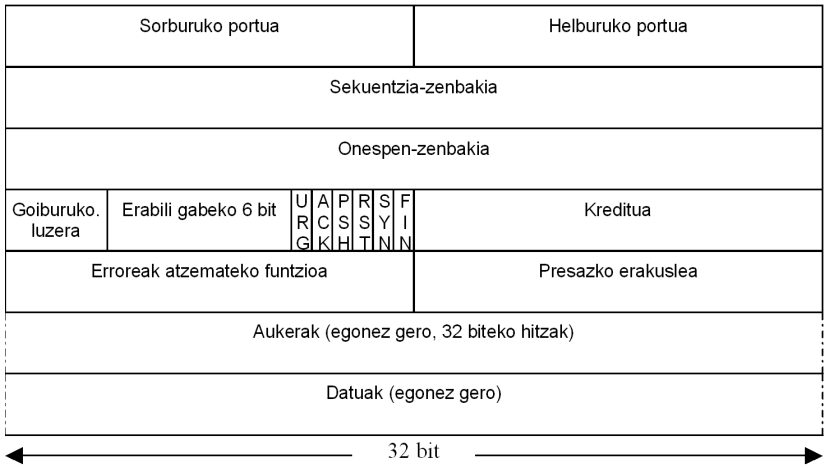
18. Badago konexio moduko zerbitzua ematea konexiorik gabeko protokolo baten bidez, baita alderantzizkoa ere, baina ez da normala. Errazena eta logikoena konexio moduko zerbitzua konexio bidezko protokolo bat erabiliz gauzatzea da.

- Konexioetan igorle bakarra eta hartzaile bakarra egon daitezke. Hau da, TCPk ez du talde-difusioa edo difusioa egiteko balio (*multicast* edo *broadcast*, hurrenez hurren, ingelesez).
- Konexio baten informazio-korrontea bytetan neurtzen da, noranzko bakoitzean. Hau TCPren bitxikeria da: nahiz eta informazioa bidaltzeko segmentutan elkartu, sekuentzia-zenbakiak eta erabilitako neurri guztiek ez dituzte segmentuak identifikatzen, byteak baizik. Hori dela eta, segmentu batek daraman sekuentzia-zenbakia ez dagokio segmentuari, baizik eta segmentuaren datu-eremuan dagoen lehenengo byteari.
- Sareartean gertatutako datu-galerak atzemateko (edo, hobeto esanda, datuak garraiatzen dituzten datagrama-galerak atzemateko), onespenak eta tenporizadoreak erabiltzen dira, aurreko atalean aipatu dugun eran. Galerak berreskuratzeko, byteak birtransmititzen dira leiho mugikorreko teknika erabiliz.
- Fluxu-kontrol esplizitua egiten du; hartzaileak muga dezake igorlearen transmisio-leihoaren tamaina, kreditu-sistema erabiliz.
- Sareartean gertatutako kongestioak atzematen ditu, eta egoera arintzeko neurriak hartzen ditu.

Komunikazio-protokolo batek bere informazio-unitateak (beren sintaxia eta semantika) eta informazio-unitate horien erabilera definitu behar ditu. 3.4. irudian dugu **segmentu** izena duen TCP entitateetako informazio-unitatearen formatua (sintaxia). Hurrengo ataletan argituko dugu zer diren eta zertarako erabiltzen diren irudi horretan agertzen diren eremuak (semantika eta erabilera).

Harrigarria bada ere, segmentu formatu bakarra dago TCPn. Normalki, TCP erako konexioaren bidezko protokolo konplexuetan formatu asko definitzen dira: formatu batzuk konexioen kudeaketarako, eta bat edo gehiago datuak garraiatzeko. TCPren kasuan, formatu bakarraren goiburukoaren 4. hitzean agertzen diren 6 bit erabiltzen dira segmentu motak bereizteko. Hurrengo ataletan ikusiko dugu ACK, SYN, FIN eta RST biten erabilera. PSH bita aplikazio-entitatearen eta TCP entitatearen arteko komunikazioa hobetzeko definitu zen, baina gaur egun ez da erabiltzen.

URG bita konexio baten barruan presazko datuak bereizteko erabiltzen da. Segmentu batean presazko datuak badaude, URG bitak 1 balio du, eta *presazko erakuslea*ren eremuak azkeneko presazko bytea zein den adieraziko du (non hasten diren presazko datuak aplikazio berak bereizi beharko du). TCP entitateak berehala bidaliko du presazko datuak daramatzan segmentu bat, nahiz eta krediturik ez izan horretarako (berehala ikusiko dugu kredituaren kontu hori). Estandarrak ez du definitu zer diren «presazko datuak»; aplikazioak berak erabakiko du hori.



3.4. irudia. TCP segmentuaren formatua.

Konexioaren identifikazioa

TCP entitate batek konexio asko izan ditzake ezarrita une berean. Jasotako segmentu bakoitza konexio horietako zeini dagokion bereizteko, nolabait identifikatu beharko ditu konexioak. Konexioari dagozkion segmentu guztiek konexio-identifikadore bera eramango dute. Hala ere, TCP konexioak ez dira identifikatzen zenbaki bakar baten bidez. Horren ordez, TCPk mutur igorlearen eta hartzailearen identifikadoreak biltzen dituen bikotea erabiltzen du. Hau da,

TCP konexioaren identifikazioa = [aplikazio igorlearen identifikadorea, aplikazio hartzailearen identifikazioa]

Aldi berean, mutur bakoitza ondoko pareak identifikatzen du:

Aplikazio entitate baten identifikazioa = [IP helbidea, portua]

Bi konexio desberdin identifikatzeko erabilitako 4 zenbakien artean, beharbada batzuk berdinak izango dira, baina bat gutxienez, desberdina izango da beti. Adibidez, web zerbitzari batek konexio bana badu ezarrita konputagailu berean dauden bi arakatzaileekin, bi konexio horien identifikadoreen 3 zenbakiren balioa (zerbitzariaren IP helbidea, zerbitzariaren portua, eta bezeroaren IP helbidea) berdina izango da, baina bezero bakoitzak portu desberdina erabiliko duenez, bi konexioei dagozkien segmentuak bereiztea badago.

Konexioak ezartzea

Demagun konputagailu batean egikaritzen ari den prozesu batek konexio bat hasi nahi duela beste konputagailu batean dagoen beste prozesu batekin. Komunikazioa hasieratu nahi duenari bezero deitzen ari gatzaizkio, eta besteari zerbitzari.

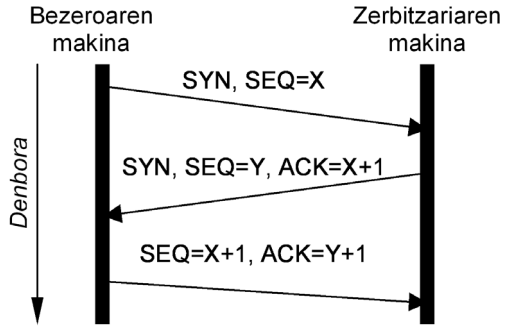
Bezeroak, aplikazio-mailaren eta garraio-mailaren arteko interfazea erabiliz, bere makinako TCP entitateari adierazitako IP helbidean eta portu-zenbakian dagoen zerbitzariarekin konexioa ezartzeko eskatuko dio. Bezeroak egindako eskaera horri **konexio-ezarpen aktiboa** deitzen zaio. Eskaera betetzearren, TCP mailan honako hiru urrats hauek egingo dira:

1. urratsa: bezeroaren TCP entitateak SYN bita gaituta daraman segmentu bat bidaliko dio zehaztutako zerbitzariaren TCP entitateari. Bidalitako segmentua **konexioa ezartzeko eskaera** bat da, edo, TCP hizkeran, SYN segmentua (ikusi 3.5. irudia). Segmentu horrek ez du inongo informaziorik garraiatzen bere datu-eremuan; kontrol-segmentu bat da. Hala ere, segmentu honetan hautazkoak diren eremu batzuk badira. Bezero-aldeak konexioari dagozkion baldintza batzuk kodifikatzen ditu hautazko eremu horietan. Adibidez, bezeroak ohartaraz dezake zerbitzaria zein den segmentu bakar batean bidal dezakeen byte kopuru maximoa. Hori da **MSS** parametroa (Maximun Segment Size, ingelesez), eta haren balio tipikoak 1460, 536 edo 512 byte dira.

2. urratsa: SYN segmentua hartu eta gero, zerbitzariaren TCP entitateak begiratuko du ea adierazitako helburuko portua bere zerbitzariren bati dagokion. Ordurako, inongo zerbitzarik ez badu TCP entitatea ohartarazi portu horretan konexioak jasotzeko prest dagoela, TCP entitateak uko egingo dio beste aldeak egindako eskaerari, eta RST bita gaituta daraman segmentu bat itzuliko du. Edozein eskaera jaso baino lehen, zerbitzariak bere TCP entitatea ohartaraztea **konexio-ezarpen pasiboa** egitea da. Demagun zerbitzari batek ezarpen pasiboa portu egokian egin duela, eta bezeroak egindako eskaera gustuko duela. Orduan, TCP entitateak SYN bita eta ACK bita gaituta dituen segmentu bat itzuliko dio bezeroaren entitateari. **SYNACK segmentu** horretan, zerbitzariaren TCPk berari bidalitako segmentuen MSS balioa muga dezake. SYNACK segmentuak ere ez du daturik garraiatzen; konexioa oraindik ezarrita ez dagoenez, ez dago aplikazio-mailako datuak bidaltzerik.

3. urratsa: zerbitzariaren TCPk bidalitako SYNACK hartzean, bezeroaren aldeak bezero-zerbitzari noranzkoan ezarritzat jotzen du konexioa. Baina zerbitzariak kontrako noranzkoan ezarritzat jo dezan, bezeroak ACK bita gaituta daraman beste segmentu bat bidali behar dio. Bidalitako hirugarren segmentu horrek datuak eraman ditzake. Zerbitzariak segmentu hori hartzean, bi noranzkoetan dago ezarrita konexioa, eta bi noranzkoetan bidal daitezke datu-segmentuak.

Konexioak ezartzeko prozedura horri **hiru urratseko akordioa** deitzen zaio (*three-way handshake*). Ondorengo irudian duzu.



3.5. irudia. Konexioak ezartzeko hiru urratseko akordia. Esanguratsuak diren segmentuaren eremuak bakarrik adierazten dira.

Aipatzekoa da segmentuetan dauden sekuentzia-zenbakien hasierako balioaren aukeraketa. Goiko irudian ikusten denez, balio hori konexioarekin batera ezartzen da, eta ez da 0, askok espero izango genukeen bezala. Mutur bakoitzak aukeratzen du zorizko balio bat, eta horren berri ematen dio beste aldeari hiru urratseko akordioaren lehenengo eta bigarren igorpenean (SEQ = X, SEQ = Y, irudian).

Konexioen hasierako sekuentzia-zenbakiaren balioa

Konexio guztietan hasierako zenbaki bera (adibidez, 0) ez erabiltzeko, bi konexioren artean datagramak nahasteko probabilitatea minimizatzea da jatorrizko arrazoia. Zehatzago adierazita, honako hau da arazoa:

1. Demagun segmentu bat atzeratzen dela sareartean, baina inongo bideratzailek ez duela deuseztatzen. Gogoan izan datagramaren TTL eremuak mugatzen duela datagramaren iraupena sareartean eta, beraz, datagramak daraman segmentuarena ere bai. Baina TTL urratsetan neurtzen da, ez segundotan. Teorian, gerta liteke datagrama batek ilara batean denbora asko ematea, eta bere TTLa bat gutxiago besterik ez izatea buxadura horretatik ateratzen denean.
2. Demagun segmentua buxaduran dagoen bitartean, beraren konexioa amaitu egiten dela, eta beste konexio berri bat ezartzen dela identifikazio berberarekin (sorburuko eta helburuko IP helbide eta portu berberekin, alegia).
3. Demagun segmentu zaharra askatzen dela eta bere helburura heltzen dela.
4. Segmentu zaharrak daraman sekuentzia-zenbakia konexio berriaren hartzaileak itxaroten dituenen artean badago, segmentuak daramatzan datuak onartuak izango dira, konexio berriari balegozkio bezala. Hori akats bat da.

Ikusten denez, oso zorte txarra izan behar da hori gertatzeko, baina, teorian, gerta daiteke, eta TCPk edozein gertaeraren aurrean prest agertu behar du. Dena dela, hau idazten duenaren iritziz, hori guztia teoria hutsa da, eta azaldutako arazo

horrengatik ez zen beharrezkoa hasierako sekuentzia-zenbakiak aldakorrak izatea, datagrama zaharrak ezin baitira konexio berrietan agertu. Kontuan hartu behar da datagrama batek ezin duela iraun sarean bere konexioa amaitzen den eta identifikazio bera duen beste bat ezartzen den bitartean, horretarako bi oztopo daudelako:

1. Bideratzaile gehienek, datagrama batek ilara batean denbora gehiegi ematen duenean, bat baino gehiago kentzen dizkiote datagrama horren TTLari. Ondorioz, datagramek ezin dute sarean «ezkutatuta» iraun.
2. TCPk berak ezinezkoa egiten du datagrama zahar bat konexio berri batean agertzea. Horretarako erabiltzen da gero azalduko dugun MSL izeneko tenporizadorea.

Hala eta guztiz ere, segurtasun-arazo bat hasierako zenbaki horiek aldakorrak izateko bestelako arrazoi, beharbada sendoagoa, bilakatu da. Lehen, hasierako sekuentzia-zenbakien balioak aurreikusteko modukoak ziren, mutur bakoitzeko konputagailuaren barruko erlojuaren araberakoak baitziren. Datu hori jakiteak eraso batzuk ahalbidetzen dituenez, gaur egun bi aldeek ezartzen duten hasierako sekuentzia-zenbakia (3.4. irudiko X eta Y balioak) zorizkoa da.

Konexioak amaitzea

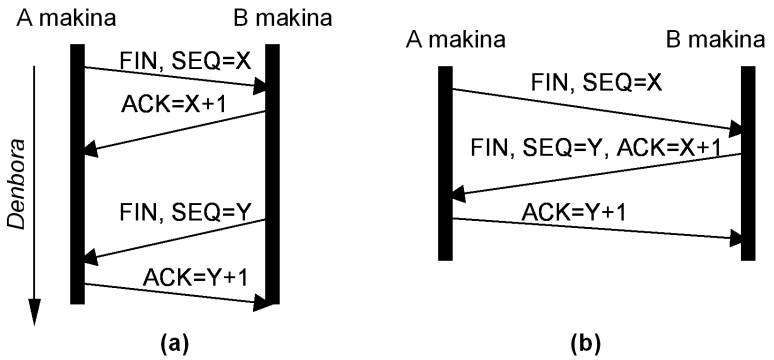
Konexioa ezartzeko hiru segmentu bidali behar badira, amaitzeko lau. Noranzko bakoitzaren amaiera independentea da. Hots, konexioa noranzko batean amaitu daiteke, eta beste noranzkoan ezarrita iraun. Konexioa egoera horretan dagoenean, erdi amaituta dagoela esaten da. Erdi amaituta dagoen konexio batean datu-segmentuak ezarrita irauten duen noranzkoan bakarrik agertuko dira. Kontrako noranzkoan kontrol-segmentuak bakarrik agertuko dira, gehienak ACK segmentuak.

Noranzko bakoitzean amaitzeko, bi urrats egiten dira. Bere datu-bidalketa amaitu duen muturrak FIN bit gaituta daraman segmentu bat bidaliko du (segmentu horrek noranzko horren azken datuak ere garraia ditzake). FIN segmentuari dagokion ACK jasotzen denean, konexioa noranzko horretan amaituta dago. Hau guztia 3.6a. irudian ikus daiteke.

Gerta daiteke (eta askotan horrela izaten da) FIN segmentu bati dagokion ACK erantzunean bertan beste noranzkoaren FIN eskaera ere bidaltzea, 3.6b. irudian agertzen den bezala. Orduan, hiru segmentu bidaltzea nahikoa da konexioa bi noranzkoetan ixteko.

Irudiko prozedurak balio du konexioak era ordenatuan amaitzeko. FIN segmentua edo dagokion ACK galtzen bada, birtransmititu egiten da, eta datuak ez dira galtzen inolaz ere. TCPk konexioak bat-batean ixteko beste era bat definitzen du, konexioak eteteko era, alegia. Mutur batek konexioa eten behar duenean, RST

bita gaituta daraman segmentu bat bidaltzen dio beste aldeari. Horrela egiten denean, RST bidali duenak ez dio inongo ACKri itxarongo, eta bere ilaretan zeuden datu guztiak deuseztatuko ditu; datu horiek galdu egingo dira. Beste muturrak ere, RST bat jasotzen duenean, konexioa bi noranzkoetan etenda dagoela, eta bidaltzeke zeuden datu guztiak galduta daudela jakingo du. Konexioak eteteko mekanismo hau larrialdietan bakarrik da erabiltzekoa. Konexioa eten egin badu, horren berri emango dio TCP entitateak aplikazioari.



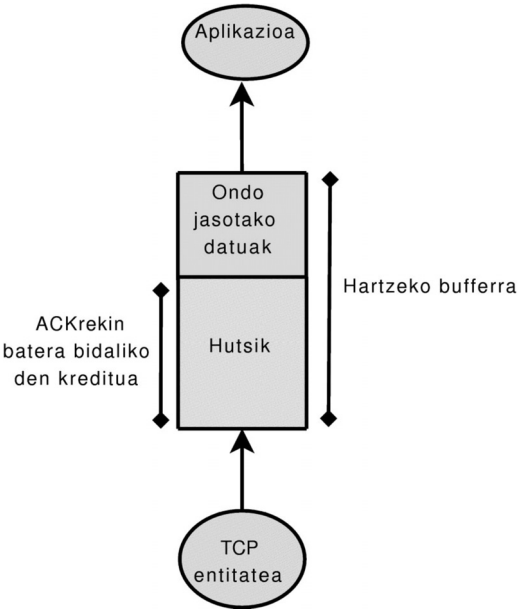
3.6. irudia. Konexioak amaitzeko prozedura. (a) Lau urratsetan (b) Hiru urratsetan.

Fluxu-kontrola TCP protokoloan

TCP entitate batek buffer pare bat esleitzen dio ezarritako konexio bakoitzari, bata heltzen diren datuak hor jartzeko (hartzeko bufferra), eta bestea bidalitako datuen kopiari eusteko (igortzeko bufferra). Konexio horretatik segmentu bat heltzen denean, dakarren informazioa hartzeko bufferrean kopiatuko du TCP entitateak, eta konexioa ezarri zuen aplikazio-mailako entitateari (bezeroa edo zerbitzaria, berdin da) jakinaraziko dio jasotzeko datu berriak dituela, bi mailen arteko interfazea erabiliz. Buffer hori, noski, mugatuta dago. Besterik ezean, TCP entitateak (batzuek sistema eragileak esango lukete, baina, finean, TCP entitatea sistema eragilearen zatia da) ezartzen du buffer horren tamaina (eskuarki 4096 edo 8192 byte), baina aplikazioaren eta garraio-mailaren arteko interfazeak aplikazioari buffer horren tamaina ezartzeko aukera eskaini diezaioke. Adibidez, *socket* izeneko interfazeak aukera hori ematen du.

Hartzeko bufferra edozein tamainatakoa izanda ere, gerta liteke datuak buffer horretatik erritmo motelagoan ateratzea sartzen direnean baino. Arrazoi askorengatik gerta daiteke hori. Adibidez, aplikazio hartzailea konputagailu motel batean egikaritzen bada, lehentasun gutxiagoko prozesua bada (eta, beraz, CPUa eskuratzeko aukera gutxi ditu), bera bezalako beste prozesu mordo bat ari bada lanean (eta, berez, are aukera gutxiago CPU hori eskuratzeko), bere sare-txartela 1 Gb/s-ko GigaEthernet bada (eta, berez, azkar sartzen dira datuak saretik), bidaltzen duena

GigaEthernet sare berean badago (eta, berez, azkar bidaltzea badu), eta askoz indartsuagoa den konputagailu batean egikaritzen den prozesu bakarra bada igorle hori (hau da, hartzailearena baino azkarragoa den CPUa erabiltzeko konpetentziarik ez du), ziur aski berehala beteko da hartzailearen bufferra, eta, TCPk ez badu konpontzen, datuak galdu egingo dira.



3.7. irudia. ACKren balioa TCPren fluxu-kontrolean.

Segmentu-galera horiek ekiditeko erabiltzen da segmentuaren goiburuko *kreditua* eremua (ikusi 3.4. irudia). Eremu horrek igorleari adierazten dio zenbat byte gelditzen zaizkion hartzaileari bere jasotzeko bufferrean erabili gabe (ikusi 3.7. irudia). Igorleak kontuan hartu behar du balio hori, zenbat byte gehiago bidal ditzakeen kalkulatzeko. Adibidez, demagun igorleak ACK segmentu bat jaso duela (hartzaileak bidalita), non ACK = 34 297 eta kreditua = 2048. Horrek jakinarazten dio igorleari hartzaileak 34 296garren byte arte ondo jaso dituela byte guztiak, eta 34 296garren byte hori prozesatu eta gero bere bufferrean beste 2048 byterako hutsunea baduela. Beste alde batetik, igorleak 34 296garren bytea transmititu eta gero, eta azken ACK hau jaso baino lehen, beste 1460 byte bidali baditu, hortik aurrera 2048 – 1460 = 588 byte gehiago besterik ezin du bidali, kreditu gehiago ematen dion beste segmentu bat jaso arte.

Errore-kontrola TCP protokoloan

TCPren ezaugarri nagusiak aztertzean aipatu dugu ezen segmentuen galerak berreskuratzeko onespenak eta tenporizadoreak erabiltzen direla. Sekzio honetan zehatzago deskribatuko dugu mekanismo hori.

Gogoan izan nola atzematen diren datagrama-galerak TCPn: datagrama bidali zenean abiatu zen tenporizadorea agortzen bada datagramari dagokion onespena jaso baino lehen, datagrama hori galdutzat joko du TCPk. Horrela esanda, ez bide da kontu zaila errore-kontrolarena. Hala ere, errealitatean gauzak ez dira hain argiak.

Lehenengo zailtasuna birtransmititzeko tenporizadorea kalkulatzea da: tenporizadorea laburregia baldin bada, behar ez diren birtransmisioak sortuko ditugu, eta luzeegia baldin bada, denbora gehiegi egongo da zain igorlea alfer-alferrik, zain duen hurrengo transmisioa egin barik. Birtransmititzeko tenporizadoreak hartu behar duen balioak honako elementu hauen batura izan behar du:

- Segmentu bat fisikoki transmititzeko denbora.
- Segmentu horrek bere helburura ailegatzeko behar duen denbora.
- Segmentu horri dagokion ACK transmititzeko behar den denbora.
- ACK horrek sarean zehar itzultzeko behar duen denbora.

Horren baturari **RTT** deitzen zaio TCP hizkeran (Round Trip Time). Transmisio fisikoari dagozkion batura horren osagaiak egonkorrak dira, baina sareartea zeharkatzeko denbora oso aldakorra izan daiteke. Horretan datza tenporizadore horiek kalkulatzeko zailtasuna.

Kalkulua dinamikoki egiten da, aurreko segmentuetan neurtutako RTT balioan oinarrituta (kalkulu zehatza zein den jakiteko, ikusi RFC 2988). Birtransmisio bat egiten denean sortzen da arazoa, aurreko RTTa neurtezin bihurtzen delako. Orduan, oso sinplea den **Karn-en algoritmoa** erabiltzen da: RTTa bikoizten da.

Tenporizadoreen erabileran oinarritzen diren birtransmisio-teknikak azaltzen direnean, esaten da transmititzen den segmentu bakoitzeko tenporizadore bat abiatzen dela. Teoria hutsa: hainbeste tenporizadoreren kudeaketa korapilatsuegia litzateke. TCPren kasuan, birtransmisio-tenporizadore bakarra dago konexio bakoitzean, igorrita eta onartzeke segmentu asko egonda ere. RFC 2988 agirian adierazten da nola kudeatzen den tenporizadore hori:

1. Datuak daraman segmentu bat bidaltzean, abiatu tenporizadorea, dagoeneko martxan ez badago. Kontuan izan onespen hutsak diren segmentuen kasuan ez dela abiatzen tenporizadorea.
2. Zain zeuden datuak onartzen dituen ACK bat jaso eta gero, onespenaren zain beste daturik ez badago, geldiarazi tenporizadorea. Datu gehiago gelditzen badira onespenaren zain, berrabiatu tenporizadorea.
3. Tenporizadorea agortzen denean, bikoiztu RTTa, berrabiatu tenporizadorea, eta transmititu behar den hurrengo bytearen sekuentzia-zenbakia eguneratu onartu gabe zegoen lehenengo bytearen sekuentzia-zenbakia-

rekin. Eguneratze horren ondorioz, segmentuak birtransmititzen hasiko da TCP.

Onespen-kudeaketaren kontua ere ez dago hain argi. Sinpleena hauxe litzateke: ordenan jasotzen den segmentu bakoitzeko ACK bat itzultzea, baina hori ez da eraginkorrena. Noranzko biko trafikoa badago (duplex), noranzko bati dagozkion onespenak kontrako noranzkoko trafikoarekin batera igorriko dira (*piggybacking* deitzen zaio teknika horri). Kontrako noranzkoko trafikorik ez badago segmentu berri bat jaso eta gero, TCPk itxarongo du hurrengo ACK bidaltzeko harik eta honako hauetako bat gertatu arte:

- Beste aldera bidali behar den datu-segmentu berri bat jasotzen du TCP entitateak aplikazio-mailatik. Kasu horretan, ACK bidaliko da segmentu horren barruan *piggybacking* eran.
- Konexioaren beste noranzkoan hurrengo datu-segmentua jaso da, datu berriekin. Orduan, bigarren segmentu horren ACK segidan itzuliko da, gehiago itxaron gabe. Ikusi ACK horrek bi segmentuak onartuko dituela. Horri **metatutako ACK** deitzen zaio (*cumulative ACK*).
- Segundo-erdi bat igaro da hasierako datu-segmentua jaso zenetik, eta aurreko bietako bat ere ez da gertatu. Kasu horretan, segmentu horri dagokion ACK berehala, gehiago itxaron gabe, bidaliko da. Onespen honi **atxikitako ACK** deitzen zaio (*delayed ACK*).

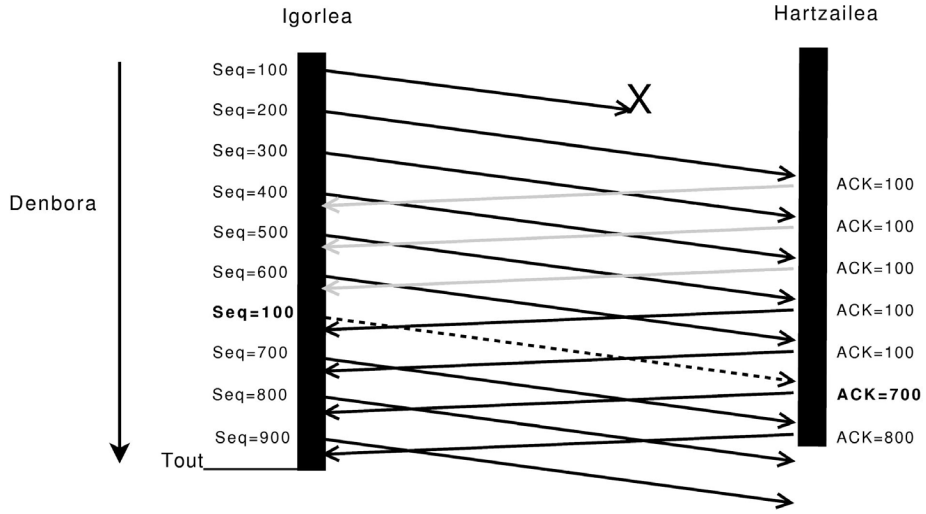
Eta zer egin behar da ordenan ez datorren segmentu bat jasotzen denean? Adibidez, demagun hartzailea 34297garren bytearen zain dagoela, baina sekuentzia-zenbakia 35797 duen segmentua jasotzen duela. Bi aukera daude: baztertu segmentu hori (blokekako birtransmisioa izeneko teknika da hori, *Go-Back-N* ingelesez), edo gorde segmentua, hutsunea beteko duten byteak heltzen diren bitartean (banakako birtransmisioa edo birtransmisio selektiboa). Argi dago bigarren aukera eraginkorragoa dela, baina TCPren kodea konplexuagoa egiten du. Harrigarria bada ere, RFCetan ez da zehazten zer egin behar den kasu horretan. Hau da, ez dago argi definituta ea TCPk blokekako ala banakako birtransmisioa erabili behar duen; inplementazioaren araberako kontua da hori. Hala ere, inplementazio gehienek egiten dutena honako hau da:

- Lekuz kanpo datorren segmentua jasotzen denean, TCPk gordetzen ditu segmentuak dakartzan datuak (aplikazioari eman gabe), eta espero zuen sekuentzia-zenbakia daraman ACK bat bidaltzen du segidan. Hau da, bidali zuen azken ACKren kopia bat (**errepikatutako ACK** deitzen zaio) bidaliko du. ACK horrek adierazten du zein den byte-korrontean sortutako hutsunearen beheko muga . Hutsunearen goiko muga lekuz kanpo zetorren segmentuaren sekuentzia-zenbakiak ezartzen du.
- Igorleak hiru errepikatutako ACK jasotzen baditu jarraian, hutsune bat dagoela ulertuko du, eta, tenporizadorea agortu baino lehen, hutsunearen

hasieran dagoen segmentua birtransmitituko du. Segmentu hori besterik ez du birtransmitituko. Horri **birtransmisio azkarra** deitzen zaio (RFC 2581), eta igorleak birtransmititzeko denbora murriztea du helburu.

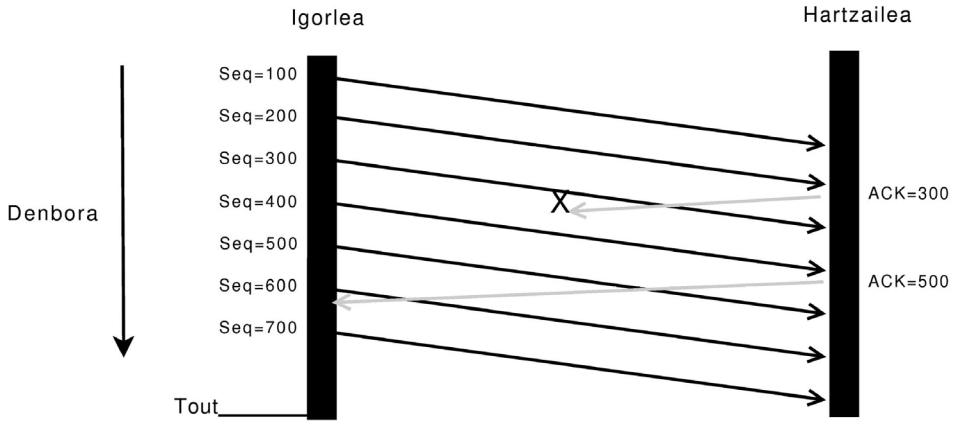
- Hartzaileak, hutsunearen beheko muga igotzen duen segmentua jasotzen badu, segidan itzuliko du dagokion ACK, atxiki gabe.

Ikusten denez, TCPk erabiltzen duena ez da blokekako birtransmisioa ezta banakakoa ere, bien arteko hibrido bat baizik. Batzuetan, TCPk galdutako segmentu bakarra birtransmitituko du, 3.8. irudian gertatzen den bezala. Irudi horretan, 100 byteko segmentuak bidaltzen dira, eta kreditu nahikoa dago etengabe transmititzeko. Lehenengo segmentua, 100 sekuentzia-zenbakia daramana, galduko da. Hartzaileak gordeko ditu lekuz kanpo datozen hurrengo hiru segmentuak (200, 300 eta 400 sekuentzia-zenbakidunak), baina ez dizkio aplikazioari helaraziko, eta errepikatutako 3 ACK bidaliko dizkio igorleari. Igorleak hirugarren ACK = 100 segmentua hartu bezain laster (SEQ = 600 daraman segmentua transmititzen ari den bitartean), segmentu *bat* galdu dela ulertuko du, 100 sekuentzia-zenbakian hasten zena, eta segmentu hori birtransmitituko du (irudian, segmentu etena), tenporizadorea agortu arte itxaron gabe. Segmentu hori birtransmititu eta gero, igorleak aurrera jarraituko du, SEQ = 700 duen segmentua transmitituz. Ohartu honako honetaz: igorleak kreditu nahikorik izan ez balu, segmentu bat baino gehiago birtransmitituko zituen, apika leiho osoa. Hartzaileak, birtransmititutako segmentua jaso arte, ACK errepikatuak bidaltzen jarraituko du (beste bi, irudian). Galdutako segmentuaren birtransmisioa ACK = 700 segmentuarekin erantzungo du, eta, gero, sekuentzia normala berreskuratuko du (ACK = 800, 900…).



3.8. irudia. Errore-kontrola TCPn. Kreditu nahikoa dagoenez, galdutako segmentua besterik ez da birtransmititzen irudiko adibidean, hiru errepikatutako ACK jaso eta gero.

3.9. irudiko kasua zeharo desberdina da. Horretan, galtzen dena ez da datu-segmentu bat, ACK bat baizik, zeina lehenengo bi segmentuei dagokien (metatutako ACK baita). Hala ere, ez da inongo segmenturik birtransmitituko, hurrengo metatutako ACK (400 segmentuari dagokiona, 500 zenbakia duena) 100 segmentuari dagokion tenporizadorea agortu baino lehenago heldu zaiolako igorleari.



3.9. irudia. Errore-kontrola TCPn. Igorritako bigarren ACK (500) hasierako segmentuari dagokion tenporizadorea agortu baino lehenago heldu denez, lehenengo ACKren galerak ez du inongo birtransmisiorik eragingo.

Kongestio-kontrola TCP protokoloan

Kongestioak sare barneko arazoa direnez, ez dirudi egokia garraio-mailan horretaz kezkatzeak. Sare bakoitzak bere sare-sarbide mailan definitzen du nola egin aurre kongestioei, eta sarearteko bideratzaileetan sarearte-mailaren ardura da arazo hori. Baina gogora dezagun Interneteko sarearte-mailan, IP mailan alegia, kongestioen aurrean ostrukarena egiten dela. Ezer ez, alegia. Kongestioak sortzen badira, datagramak galtzen dira eta kito. Garraio-mailak birtransmitituko du galdutako informazioa, TCP erabiltzen bada, behintzat. Beraz, TCPk ez duela buxaduraz arduratu behar ondoriozta dezakegu.

TCP kongestioez ez arduratzea teorikoki zuzena litzateke, baina ez praktikoa. TCPk kongestioei arreta eskaintzen ez badie, sareartetik lortuko duen zerbitzua okerragoa izango da: kongestioan segmentuak galtzen direnez, TCPk galduen kopiak birtransmitituko ditu, buxadura askatzen laguntzeko egin behar ez dena, hain zuzen ere. Egin behar dena da bidaltzeko erritmoa moteltzea, kongestioa desagertu arte. Hori da TCPk benetan egiten duena.

Eta nola asmatzen du TCP entitate igorle batek sareartean kongestioa dagoela eta, beraz, bere jarduera moteldu behar duela? TCP entitate batek ezin du bideratzaileen IP mailekin hitz egin, horien ilaren egoeraren berri jakiteko. TCPk egingo duena sarearteak ematen dituen sintomak aztertzea izango da. Berarentzat,

segmentu bat birtransmititzeko beharra kongestioaren sintoma bat da, eta horrekin batera bidaltzeko erritmoa apalduko du. Hau sinplifikazio bat da, birtransmititzeko tenporizadore bat agortzen denean ez baita beti kongestio batean segmentua desagertu delako gertatzen. Baina, hala eta guztiz ere, errealitatea ez da oso desberdina: egungo Interneten, birtransmisio gehienak kongestioei dagozkie, eta, beraz, TCPk gehienetan asmatzen du tenporizadore baten agorpena kongestioarekin identifikatzen duenean.

Slow-start eta kongestioak ekiditeko algoritmoa

TCP entitate igorle baten transmisio-leihoak ondoko bi muga hauek izango ditu:

- Mutur hartzaileak kredituen bidez adierazitako muga. Hau da, fluxu-kontrolak ezarritako mugak.
- Kongestioak ekiditeko bere buruak ezarritako muga. Muga horri kongestio-leihoa deitzen zaio.

Bi muga horien artean txikiena da une bakoitzean erabiliko den transmisio-leihoa. Kredituen erabilera ikusi dugu dagoeneko. Ikus dezagun orain nolakoa den kongestio-leihoaren kudeaketa.

Konexio berri bat ezartzen denean, dagokion kongestio-leihoa MSS aldagaiaren baliokoa izango da beti. Gero, ACK bat jasotzen den bakoitzean, leiho hori hazi egiten da. Hazitako kopurua ACK horretan onartzen den byte kopuru bera izango da. Datu-segmentu gehienetan MSS byte kopurua bidaltzen denez, ACKetan ere kopuru bera onartzen da. Ondoko hau izango da igorlearen jarduera (kredituaren muga beti kongestio-leihoa baino handiagoa dela suposatuko dugu):

- Lehenengo bidalketan, segmentu bakar batean MSS byte bidaltzen dira.
- Dena ondo joango dela suposatuz, RTT igarota (gutxi gorabehera) aurreko bidalketari dagokion ACK jasoko du igorleak, eta bere leihoa MSS byte igoko du, 2*MSS balioa hartuz. Orduan, bi segmentutan 2*MSS byte bidaliko ditu.
- Berriro RTT igaro eta gero, bidalitako 2*MSS byteei dagozkien onespenak etorriko dira (gehienetan, metatutako ACK bakar batean). Igorleak 4*MSS byte arte igoko du bere leihoa, eta lau segmentu berri bidaliko ditu.
- Dinamika hau errepikatzen da RTT segundoro: leihoa transmititu, dagozkion onespenak jaso, eta, horrekin batera, kongestio-leihoa bikoiztu.

Ikusten denez, kongestio-leihoaren hazkundea esponentziala da. Hazteko era honi ***slow-start*** (*hasiera motela*, ingelesez) algoritmoa esaten zaio, nahiz eta leihoaren hazkunde azkarra eragin.

Slow-start algoritmoak ezarritako hazkunde esponentziala ondo dago konexioaren hasieran, lehenbailehen kongestio-leihoak balio egokia har dezan, hau da, ahal den handiena baina kongestiorik sortu gabe. Baina balio egoki hori lortu eta gero, hazkunde-erritmoa apaldu egin behar da. TCPk «balio egokia»ri **kongestio-atalasea** deitzen dio. Balio hori zein den horrela kalkulatzen du:

- Konexio baten hasieran, kongestio-atalaseak transmisio-leihoaren balio maximoa hartzen du, hau da, kredituaren balio maximoa (65 536 byte).
- Birtransmisio bat egin behar denean, kongestio-atalaseak une horretan erabiltzen ari den transmisio-leihoaren balioaren erdia hartuko du. Gogoan izan transmisio-leihoa beti dela kredituaren eta kongestio-leihoaren arteko txikiena.

Konexioaren hasieran eta birtransmisio baten ondoren, slow-start algoritmoa erabiltzen da kongestio-leihoaren tamaina handitzeko, kongestio-atalaseraino heldu arte. Une horretan, bere hazkunde-erritmoa aldatzen du. Hortik aurrera kongestio-leiho oso baten onespenak jasotzen direnean MSS byte kopurua hazi egingo da. Adibidez, demagun atalasea 4*MSS dela eta kongestio-leihoak balio hori hartu duela. Orduan, hurrengo 4*MSS byteak bidaltzen dira, eta dagozkien ACKak itzultzen dira. Une horretan kongestio-leihoa 5*MSS izatera pasatuko da. Hazteko erritmo lineal horri **kongestioak ekiditeko algoritmoa** esaten zaio (*congestion avoidance*).

Orain arte kongestio-leihoa nola hazten den ikusi dugu, hasieran esponentzialki, eta gero, kongestio-atalasetik aurrera, linealki. Baina, nola gutxitzen da bere balioa kongestio bat atzematen denean? TCPren jatorrizko bertsioan (*Tahoe* bertsioan), birtransmisio bat egin behar zen bakoitzean, kongestio-leihoaren balioa konexioaren hasierakoa izatera itzultzen zen, MSS bytera alegia. Gaur egun gehien erabiltzen den TCPren bertsioan (*Reno* bertsioa), berriz, birtransmisio guztiek ez dute eragin bera kongestio-leihoaren balioan. Birtransmisioa tenporizadorearen agortzeak eragin badu, TCP Renoren portaera Tahoerena bera da; hau da, kongestio-leihoa MSS byte bilakatuko da. Baina birtransmisioa hiru errepikatutako ACK jarraian jasotzeak eragin badu (birtransmisio azkarra bada, alegia), kongestio-leihoak hartuko duen balio berria kongestio-atalaseak hartuko duen balio bera izango da, hau da, une horretan dagoen transmisio-leihoaren erdia. Horri **berreskuratze azkarra** deitzen zaio (*fast recovery*).

TCP Renoren hobekuntzak kontuan hartzen du ezen ACK errepikatuak jasotzen badira bi muturren arteko bidea guztiz moztuta ez egoteagatik izango dela. Hau da, datagrama galtzea eragin duen buxadura ez dela hain larria izan, eta hurrengo datagramek arazorik gabe zeharkatu dutela kongestionatutako bideratzailea. Egoera tenporizadorea agortzen denean bezain estua ez denez, kongestio-leihoa (eta, horrekin batera transmisio-leihoa) MSS bakar bateraino jaistea gehiegizko neurritzat jotzen du TCPk. Ohartu transmisio-leihoa jaisteak lortutako benetako transmisio-

abiadura neurri berean jaistea eragingo duela normalki, transmisio-leiho txiki batek igorlearen geldialdiak sorraraziko baititu.

TCP protokoloaren tenporizadoreak

TCPk tenporizadore asko kudeatu behar ditu zerbitzu fidagarria emateko. Dagoeneko ikusi dugun birtransmititzeko tenporizadorea alboan utzita, honako hauek dira TCPk erabiltzen dituen tenporizadore garrantzitsuenak:

- 2MSL tenporizadorea

Batzuetan koarentena-tenporizadorea deituta, 2MSL izeneko tenporizadorea (*Maximun Segment Life*) konexio bat amaitzen denean abiatzen da. Tenporizadore honek 30, 60 edo 120 segundo balio ohi du, hori baita, gutxi gorabehera, datagramen TTLren balioaren baliokidea segundotan. Konexio bat bi noranzkoetan amaitzen denean, aplikazio lokalak erabilitako portu-zenbakia koarentenan jartzen da 2*MSL segundotan. Beraz, identifikazio bera duen beste konexio bat ezartzea posible denerako, konexio zaharraren datagrama guztiak hilik eta lurpean egongo direla bermatzen da.

- Iraunkortasun-tenporizadorea (*persist timer*)

Demagun hartzaileak igorlea geldiarazten duela, kreditua 0 duen onespen bat bidaliz. Geroago, hartzaileak igorlea berpiztu nahi du, bere jasotzeko bufferrean tokia sortu baita. Horretarako kreditu berria (0 ez dena) daraman segmentu berri bat bidaltzen du, baina segmentu hori galdu egiten da. Orain bai igorlea, bai hartzailea, besteak zer egingo duen zain daude. Elkarren blokeatze hori saihesteko diseinatuta dago iraunkortasun-tenporizadorea.

Kreditua 0 daraman segmentu bat jasotzen denean abiatzen da iraunkortasun-tenporizadorea. Agortu baino lehen beste aldearen berririk ez badago, igorleak itaun-mezu bat bidaliko dio hartzaileari (*window probe*, TCPren hizkeran). Horren erantzunak kreditua adierazten du; oraindik 0 bada, ez da inongo elkarren blokeorik egon, itaun-mezua sobera zegoen, iraunkortasun-tenporizadorea berriro abiatzen da eta zikloa hasieratik hasten da. 0 ez bada, elkar blokeatuta zeuden bi aldeak, bata bestearen zain. Itaun-mezuak askatzen du blokeatze hori, eta datuak bidal daitezke berriro.

- Biziraute-tenporizadorea (*keepalive timer*)

TCP konexio bat ez da amaitzen muturrek hori eskatu arte, FIN segmentuaren bidez, edo RST segmentuaren bidez eten arte. Nahiz eta, adibidez, tartean dauden bideratzaileak edo lineak erori eta berriro abiatu, konexioa ez da galdu behar. Izan ere, gerta daiteke konexioaren mutur bat, bezeroa edo zerbitzaria, itzaltzea, eta beste muturrarentzat konexioa ez da amaituta egongo. Horren adibidea *telnet*-en kasua da, bezeroaren konputagailua itzaltzen denean *telnet* bezeroa amaitu gabe.

Kasu horretan, konexioa ezarrita dago zerbitzariarentzat, nahiz eta trafikorik ez egon. Horrelako telnet erabiltzaile asko zerbitzari honekin konektatuta baldin badaude, berehala agortuko da zerbitzariak ezarrita mantendu dezakeen konexio kopurua eta, beraz, zerbitzari hori K.O. teknikoan geldituko da: badabil inongo problemarik gabe, baina bere ahalmen guztia aktibo ez dauden konexioetan xahutzen du.

Erabili gabeko konexio horiek atzemateko erabiltzen da biziraute-tenporizadorea. Jasotako segmentu bakoitzarekin berrabiarazten da tenporizadorea. Inoiz agortzen bada, itaun-segmentu bat bidaltzen da (*keepalive probe*, TCPren hizkeran), ea beste aldea oraindik hor dagoen egiaztatzeko. Erantzunik ez badago, konexioa amaitu egingo da.

Tenporizadore honen inguruan eztabaida ugari egon da, jarduerarik ez hori aplikazio-mailako arazoa delako, eta aplikazioek atzeman behar zutelako, ez TCP entitateak. Gainera, gerta daiteke itaun-mezua galtzea eta, beraz, zerbitzariak konexioa amaitzea, nahiz eta bezeroak bizirik jarraitu. Horregatik biziraute-tenporizadorea ez dago TCP estandarrean onartuta. Baina, hala eta guztiz ere, TCP inplementazio gehienek erabiltzen dute tenporizadore hau.

Transmisio-abiadura TCP konexioetan

Konexioaren transmisio-abiadura erabilitako transmititzeko leihoaren eta RTTaren funtzioa da, beste edozein leiho moduko protokolotan bezala. Protokolo horietan, konexioak erabil dezakeen abiadura fisikoa % 100ean erabiltzeko, leihoak tamaina minimoa izan behar du. Tamaina horri **etengabeko transmisiorako leihoa** deitzen zaio, eta haren balioa **abiadura-atzerapena biderkadura** (*bandwith-delay product*) da:

$$Etengabeko\ transmisiorako\ lehioa(bit) = transmisio\text{-}abiadura\ (b/s) \times RTT\,(s)$$

Gure konexioan erabiltzen den transmititzeko leihoa horren tamainakoa edo handiagoa bada, TCP konexioaren transmititzeko abiadura abiadura fisikoa bera da. Hau da, 100 Mb/s-ko txartela badu igorleak, abiadura horretan transmitituko da fitxategia, beste inorekin (beste aplikazioak edo sistema eragilea) ez bada saretxartela konpartitu behar. Baina transmititzeko leihoaren tamaina minimo hori baino txikiagoa bada, orduan, TCP konexioaren abiadura abiadura fisikoa baino apalagoa izango da. Kasu horretan, honako hau da TCP mailako abiadura:

$$Abiadura = \frac{Lehian\ transmititutako\ \text{informazioa}}{RTT}$$

Zoritxarrez, kalkulu hauetarako erabilitako parametroak ez dira konstanteak TCP konexio batean, eta, beraz, ezer kalkulatu ahal izateko suposizio batzuk egin behar dira. Guk ondoko baldintza hauek suposatuko ditugu:

- Muturreko bi makinetan gure TCP konexioko entitateek beste inork ez du erabiltzen sare-txartela. Baldintza honek bermatzen digu sare-txartelaren abiadura fisikoa gure transmisiorako dela. Errealitatean, TCP konexio batek lor dezakeen abiadura fisikoa sare-txartelaren beste erabiltzaileen araberakoa da une bakoitzean.
- Protokolo guztietako goiburukoen transmisio-denbora oso txikia izango da informazioarenarekin alderatuz, eta, beraz, ez dugu kontuan hartuko.
- Halaber, informazioa garraiatzen ez duten segmentuak (ACK, SYN, SYNACK…) transmititzeko denbora ere aintzat ez hartzeko modukoa dela hartuko dugu.
- Kredituaren balioa konstante mantentzen da.
- RTTren balioa egonkorra dela suposatuko dugu. Horrek suposatzen du sarearen egoera egonkorra dela konexioak dirauen bitartetan. Hau da, gure konexioak sortutako datagramek zeharkatu beharko dituzten bideratzaileetan topatuko dituzten ilarak egonkorrak izango direla. Beraz, sarearen egoera alda dezakeen parametro bakarra gure konexioa izango da.

Honekin guztiarekin TCP igorlearen etengabeko leihoa zein den kalkula dezakegu. Baina ezin dugu suposatu igorlearen leihoa konstantea izango dela, TCPren konexioen ezaugarrietako bat leiho hori aldakorra izatea delako, hain zuzen ere. Gogoan izan fluxu-kontrolak eta kongestioen kontrolak mugak ezartzen dituztela, eta muga horiek aldakorrak direla. Hala ere, aldakortasun hori asko murriztu dugu gure baldintza-zerrendan: kreditua finkoa da. Beste alde batetik, ezin diogu muzin egin TCP konexioak eskaintzen duen transmisio-abiaduran kongestio-kontrolak duen eraginari. Azter dezagun eragin hori, ea nolabait aurreikus dezakegun TCPk lortuko duen abiadura.

Konexioaren hasieran, slow-start algoritmoa abiatzeak badu eragina: MSS bateko leihoarekin hasi eta leihoaren tamaina egonkorra lortu arte, ahal dena baino transmisio-abiadura txikiagoan transmitituko da. Hala ere, eragin hori esanguratsua izango da soilik fitxategia oso txikia eta RTTa oso altua direnean, edota leihoaren balio egonkorra oso handia denean (horren frogapena, Kurose eta Rose, 2008). Kontuan harturik gure konexioetan RTTa ez dela izango 150 m. baino altuagoa (oso litekeena egungo Interneten) eta kredituaren balioak oso leiho handiak erabiltzea eragotziko duela (horrela izaten da), bazter dezakegu slow-start algoritmoaren eragina konexioaren hasieran.

Hala eta guztiz ere, leihoaren tamaina aldatzen da kongestioak ekiditeko algoritmoak eraginda: birtransmisio bat dagoenean, transmisio-leihoa murriztu egiten da. Murrizketa horren eragina neurtzeko, honako baldintza hau gehituko diogu gure ereduari:

- Sortzen diren kongestio guztiak arinak dira, eta, beraz, birtransmisio guztiak hiru errepikatutako ACK jaso ondoren egiten dira (berreskuratze azkarra). Hau da, slow-start ez da abiatuko kongestioen ondorioz ere. Baldintza hau nahiko koherentea da aurrekoekin: RTT egonkorra bada, eta buxadura-sortzaile bakarra gure konexioa izango denez, buxadura horiek arinak izango dira, zaila baita konexio bakar batek buxadura larria eragitea.

Baldintza horietan, kongestioak daudenean leihoaren tamaina zein izango den kalkulatzeko, ikus dezagun nolako bilakaera izango duen balio horrek. Suposa dezagun leihoaren tamaina K byte denean sortzen dela buxadura sarean. Une horretan, konexioaren datagrama bat galduko da, eta berreskuratze azkarra abiatuko da. Beraz, transmisioaren leihoa K/2 izatera pasatuko da, eta, hortik aurrera, MSS bat haziko da leiho oso bat transmititzen den bakoitzean. Berriro K baliora heltzen denean, buxadura arina eragingo dugu sarean, datagrama galduko da, eta berreskuratze azkarra abiatuko da. Bilakaera zikliko hori behin eta berriro errepikatuko da fitxategi osoa transmititu arte. Hala, ondoriozta dezakegu leihoaren tamainaren batez besteko balioa 0,75 K izango dela.

Beraz, eredu sinplifikatu horri jarraituz, kalkula dezakegu TCP konexiotik aplikazioak lortuko duen transmisio-abiadura fitxategi bat bidaltzean. Kalkuluan erabili behar dugun leihoaren tamaina erabakitzea da arazo bakarra. Honako kasu hauek ditugu:

- A kasua: etengabeko transmisiorako leihoaren, kredituaren, eta K parametroen artean, etengabeko transmisiorako leihoa da txikiena. Kasurik errazena da hori: TCP konexioaren transmisio-abiadura eta transmisio-abiadura fisikoa bera da. Ohartu igorleak ez duela inoiz leiho horrek ahalbidetzen duena baino azkarrago transmitituko, eta, beraz, ezin izango du kongestiorik sortu, nahiz eta kongestio-leihoa K baino handiago bilakatu.
- B kasua: kreditua da aurreko hiruren artean txikiena. Kreditua da, orduan, TCP konexioaren abiadura kalkulatzeko erabili behar dugun leihoaren tamaina. Kasu honetan ere, ez da inoiz kongestiorik sortuko, leihoa ez baita inoiz K byte izatera helduko.
- C kasua: K baldin bada txikiena, orduan leihoaren batez besteko tamaina 0,75 K izango da.

Ikus ditzagun adibide batzuk. Honako taula honek hiru kasu hauek jasotzen ditu: baliorik txikiena etengabeko transmisiorako leihoa den kasua da lehenengo lerrokoa, kreditua minimoa da bigarrenean, eta kongestioa sortzen duen leihoaren tamaina balio txikiena da hirugarren lerroan.

Transmisio-abiadura fisikoa	Etengabeko transmisiorako leihoa	Kreditua	Buxadura sortzen duen leihoaren tamaina (K)	Benetako transmisio-abiadura
56 kb/s	700 byte	8192 byte	10 000 byte	56 kb/s
10 Mb/s	125 000 byte	8192 byte	10 000 byte	0,65 Mb/s
10 Mb/s	125 000 byte	8192 byte	5000 byte	0,3 Mb/s

3.1. taula. TCP konexioak eskaintzen duen benetako transmisio-abiaduraren kalkulua, testuan adierazten diren baldintzetan. Hiru kasuetan RTT = 0,1 segundo.

Transmisio-abiaduraren balioespen horiekin aurreikus dezakegu zenbat denbora beharko dugun fitxategi bat transmititzeko TCP erabiliz. Benetako transmisio-abiadura aplikatuz, TCP konexioa irekitzeko emandako denbora gehitu beharko genioke kalkulatutako denborari, konexio hori gure fitxategia transmititzeko esplizituki irekia izan bada, baita aplikazioaren komandoak bidaltzeko denbora ere (5. kapituluan ikusiko dugu nola egiten duten hori aplikazio batzuek). Demagun fitxategia jaisteko komando bakar bat bidali behar diola bezeroak zerbitzariari, eta horren erantzunarekin batera datuak bidaltzen hasiko dela zerbitzaria (horrela egiten da webaren kasuan). Kasu horretan, kalkulatutako denborari 2 RTT besterik ez genioke gehitu beharko: konexioaren eskaera egin (SYN segmentua bidali) eta RTT segundora bezeroak bere komandoa bidaliko du hiru urratseko akordioaren hirugarren bidalketarekin batera, eta, beste RTT segundo igaro eta gero, fitxategiaren hasierako byteak hartzen hasiko da. Askotan, gehitutako 2 RTT denbora hori ez da esanguratsua izango fitxategia transmititzeko denborarekin alderatuta. Edonola ere, kontuan hartu lortuko dugun balioespena abiadura maximo teorikoa izango dela, egindako suposizioengatik. Errealitatean TCP motelagoa izan daiteke (eta, gehienetan, izango da), baina inoiz ez da azkarragoa izango.

TCP ala UDP?

UDPren fidagarritasunik eza ikusita, gure aplikazio banatuetan UDP baztertu beharko genukeela dirudi, eta TCP beti erabili. Hala ere, askotan hobe da UDPren arintasuna TCPren fidagarritasuna baino. Erabaki hori aplikazioaren diseinatzaileari dagokio, eta horretarako faktore asko hartu behar ditu kontuan: aplikazioaren ezaugarriak eta beharrak, TCPren eta UDPren arteko aldeak, eta erabiliko diren sareen ezaugarriak. Ez dago inongo arau zehatzik erabaki hori hartzeko; batzuetan oso argia izango da aukera, eta beste batzuetan zaila. Izan ere, badaude bi protokoloak erabil ditzaketen aplikazioak, erabiltzailearen beharren edo sarearen arabera UDP edo TCP erabiltzeko aplikazioa konfiguratuz.

Ondoan erabaki hori hartzean kontuan hartu behar diren irizpide batzuk adierazten dira:

- Aplikazioa transakzionala denean, hobe ohi da UDP erabiltzea. Aplikazio transakzional batean bezeroak mezu motzak (datagrama bakar batean sartzen diren eskaerak) bidaltzen dizkio zerbitzariari, eta horrek datagrama bakar batean sartzen den erantzun laburra itzultzen du eskaera bakoitzeko.

  Aplikazio transakzionalak fidagarritasuna behar badu, aplikazio-mailan egin daiteke. Kontuan izan horrelako aplikazio transakzionaletan nahikoa dela egiaztatzea epe mugatu batean erantzuna hartu dela. Epe horren barruan zerbitzariaren erantzuna heltzen ez bada, aplikazio-mailako eskaera errepikatu egiten da.
- Denbora errealeko aplikazioetan hobe izaten da UDP erabiltzea, aplikazioarentzat datagrama batzuk galtzea jasangarria baldin bada. Kasu honetan TCPren fidagarritasuna lortzeko kostua (atzerapenak, leihoaren gora-beherak) kaltegarriagoa izaten da aplikazioarentzat, ekidin nahi den kaltea baino (datagrama batzuk galtzea edo hurrenkeraz aldatuta heltzea). Hori gertatzen da, oro har, denbora errealeko multimedia aplikazioetan. Hala ere, sareazpiegiturak hobetzen diren heinean, gero eta gehiago erabil daiteke TCP horrelako aplikazioetan, galerak gero eta urriagoak baitira, eta, berez, TCPren kongestioen eta galeren kontrolerako mekanismoen eragina txikia da.
- Mugitu behar den byte kopurua handia bada, eta byte guztiak ondo jasota izango direla beste aldean bermatu behar bada, ezinbestekoa da TCP erabiltzea. Hori da fitxategiak mugitzen dituzten aplikazioen kasua. Adibidez, web, ftp, edo posta elektronikoa.

LABURPENA

Garraio-mailak bi aplikazio-entitateren arteko komunikazioa gauzatzen du. Beraz, sareartetik jasotzen duen informazioa bere makinako zein aplikazio-entitateri eman behar zaion identifikatu behar du. Horretarako erabiltzen dira portuak, sarbide-mailan helbide fisikoak edo sarearte-mailan IP helbideak erabiltzen diren modu berean.

IP datagramak beren sareartean zehar egindako bidaian gal daitezke. Aplikazio askorentzat hori ez da onargarria: bidaltzen diren bit guztiek heldu behar dute beste muturreraino. IP zerbitzuak hori bermatzen ez duenez, eta aplikazio-mailan horretaz arduratzea zuzena ez denez, garraio-mailak egingo du lan hori.

Beste alde batetik, aplikazio guztiek ez dute behar fidagarritasunik. Horregatik, garraio-mailak bi zerbitzu desberdin eskainiko dizkio aplikazio-mailari, bata fidagarria eta bestea arina. Lehenengoa gauzatzeko TCP protokoloa erabiltzen du. Zerbitzu arina UDP protokoloaren bidez egiten du.

UDP oso protokolo sinplea da. Izan ere, IP zerbitzuari eransten dion gauza bakarra aplikazio-entitatearen identifikazioa da.

TCP protokoloa, aldiz, nahiko konplexua da. Konexioaren bidezko protokoloa da, horrela errazago baita behar den errore-kontrola eta fluxu-kontrola egitea. Datagrama-galerak berreskuratzeko birtransmisioak egiten ditu, eta fluxua kontrolatzeko kreditu-sistema bat erabiltzen du. TCPk kongestioei aurre egiten die, igorlearen transmisio-erritmoa motelduz.

4. Aplikazioak sarean

Kapitulu honetan aplikazio banatuen diseinua eta Interneten erabiltzen diren aplikazio garrantzitsuenak aztertuko ditugu. Kapitulua ikasi eta gero, ikasleak jakin beharko du:

- Zein diren aplikazio banatuen osagaiak, eta horrelako aplikazioak diseinatzeko urratsak.
- Zer den DNS, nolakoak diren izenak Interneten, zeintzuk diren DNSren osagaiak, eta zerbitzua emateko nolako elkarlana egiten duten.
- Zein diren webaren ezaugarriak: emandako zerbitzua, zer den hipertestua, zer den HTML formatua, HTTP protokoloaren oinarrizko ezaugarriak, URLren egitura, TCP konexioen kudeaketa, web cacheen erabilera, eta web aplikazioen osagaiak.
- Posta elektronikoko zerbitzuaren gorabeherak: osagaiak; elkarren arteko jarduera; SMTP, POP, eta IMAP protokoloen zergatia; RFC 822 formatuaren ezaugarriak; MIME zer den, eta nolakoak diren posta-helbideak.
- IP telefonia-sistemen oinarrizko funtzionamendua eta ezaugarriak.

4.1. APLIKAZIO BANATUEN DISEINUA

4.1.1. Sare-aplikazioen osagaiak

Edozein aplikazio informatiko osatzen duen softwareak ondoko 3 ataletan antolatzen da:

- Erabiltzailearekiko interfazea (*presentation tier)*. Atal honek aplikazioa erabiltzen duen erabiltzailearekiko komunikazioa ahalbidetzen du. Erabiltzaile hori gizaki bat izaten da, baina beste aplikazio bat ere izan daiteke. Lehenengo kasuan, interfaze horrek ahalik eta erosoena eta erabiltzen erraza izan behar du, eta aplikazioak eskaintzen dituen zerbitzu guztiak eskura jarri behar dizkio erabiltzaileari. Erabiltzailea beste programa bat

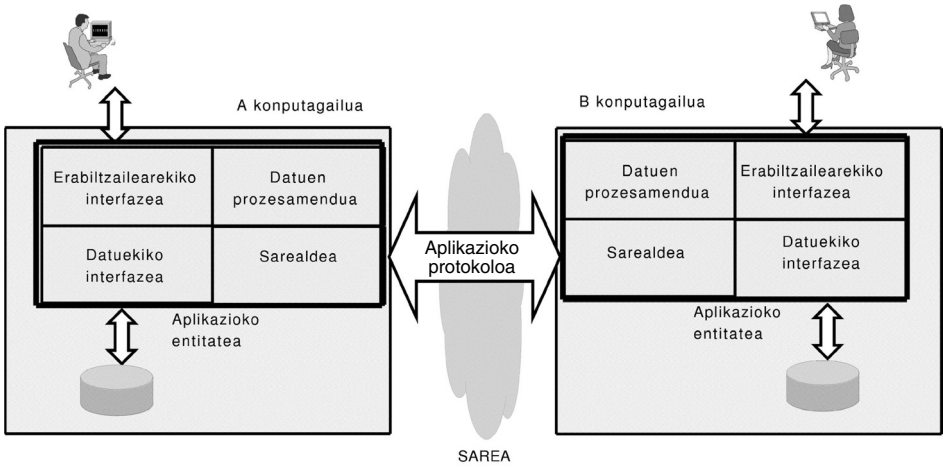
denean, interfaze hori sinpleagoa izaten da, zeren programen artean hobeto ulertzen baitute elkar, gizakiekin baino.

- Datuekiko interfazea (*data tier)*. Aplikazioaren zati honek aplikazioak behar dituen datuak lortzen ditu, konputagailuaren baliabideak erabiliz. Bere zeregin nagusia datuen biltegiak eta iturriak eskura jartzea da. Askotan zati honen gehiena datu-base bat atzitzeko prozedurek osatzen dute.
- Prozedurak (*logic tier)*. Hau da aplikazioaren «adimena», algoritmoak egikaritzen dituen softwarearen zatia, alegia. Algoritmo horiek abiatzeko behar diren datuak konputagailuaren baliabideetatik hartzen dira (datu-baseak, fitxategiak, kamara, sentsoreak…), edo zuzenean ematen ditu aplikazioaren erabiltzaileak. Algoritmoen emaitzak erabiltzaileari zuzenean helarazten zaizkio, edo datuen biltegian gordetzen dira.

Aplikazioak egituratzeko era honi *Three-tier architecture* deitzen zaio (hiru ataleko arkitektura). Konputagailu-sareak zabaldu ziren arte, aplikazioen hiru atal horiek konputagailu bakar batean kokatu eta egikaritu egiten ziren. Baina gaur egun oso ohikoa da aplikazioen atal desberdinak konputagailu desberdinetan egikaritzea, eta beraien artean sarearen bidez komunikatzea. Horrelako aplikazioei **aplikazio banatu** edo **sare-aplikazio** deritzegu. Gaur egiten diren aplikazio informatiko gehien-gehienak sare-aplikazioak dira, TCP/IP sareren bat erabiltzen baitute, bai publikoa (Internet), bai pribatua. Sare-aplikazioak monokonputagailukoak baino konplexuagoak dira, beren osagaien artean sare bat dagoelako, eta sare bidezko komunikazioa programa baten prozeduren artekoa edo konputagailu baten prozesuen artekoa baino konplexuagoa da. Horregatik, aurreko 3 atalei laugarren bat erantsi behar diegu sare-aplikazioen kasuan (ikusi 4.1. irudia):

- Sarealdea (*network tier*). Atal honek konputagailu desberdinetan egikaritzen diren aplikazio bereko osagaien arteko komunikazioa ahalbidetzen du. Horretarako ezinbestekoa izango da komunikazio hori arautuko duen protokoloaren definizioa. Protokolo hori gauzatzea da aplikazioaren sarealdearen betebeharra. Aplikazioko beste atalek sarealdea erabiltzen dute beren lanerako, adibidez, beste konputagailu batean dauden datuak atzitzeko, beste konputagailu horretan prozeduraren bat abiatzeko, edo prozedura horren emaitzak jasotzeko. Sarealdea aplikazio konkretu batentzako softwarea izan beharrean, edozein sare-aplikaziotan erabiltzekoa denean, batzuek *middleware*[19] izena ematen diote.

19. *Middleware* terminoaren esanahia eta erabilera ez dago adostuta.



4.1. irudia. Sare-aplikazioen egitura. Aplikazioa konputagailu desberdinetan egikaritzen diren aplikazioko entitateek osatzen dute. Horietako entitate bakoitzean aplikazio klasiko baten zatiak ager daitezke.

Lehenengo kapituluan ezagututako sare-arkitekturaren ikuspuntutik aztertzen badugu aplikazio banatu bat, konputagailu bakoitzean egikaritzen den zati bakoitza **aplikazio-mailako entitate** bat da (edo, laburrean, aplikazioko entitatea), eta beraien arteko komunikazioa antolatzen duen protokoloa, **aplikazio-mailako protokoloa** da (aplikazioko protokoloa).

Ikus dezagun hau guztia denontzako ezaguna den adibide batekin: weba. Weba sare-aplikazio bat da, konputagailu desberdinetan egikaritzen baitira bere hainbat zati, eta sare bat erabiltzen baitute beraien artean komunikatzeko eta erabiltzaileari emateko eskatutakoa (aplikazio-mailako zerbitzua, alegia). Webean bi motatako aplikazioko entitateak agertzen dira, bezeroak eta zerbitzariak. Bezeroarena egiten duen softwareari *arakatzailea* edo *nabigatzailea* esaten zaio. Lehen aipatutako lau ataletatik, gehienetan hiruk osatzen dute arakatzailea: erabiltzailearekiko interfazea, datuekiko interfazea, eta sarealdea. Laugarrena, datuen prozesamendua, zerbitzariek egiten dute normalki, baina gero eta gehiago agertzen ari dira zerbitzarietatik jasotako datuak prozesatzen dituzten arakatzaileentzako osagaiak. Entitate zerbitzariaren atalak datuen prozesamendua, sarealdea, eta datuekiko interfazea dira. Erabiltzailearekin zuzenean harremanik ez dutenez, zerbitzariek ez dute behar erabiltzailearekiko interfazerik. Aplikazioko entitateen arteko harremanetarako protokoloa HTTP da webaren kasuan. Webaren deskribapen osoa aurrerago egingo dugu, kapitulu honetan bertan.

Aplikazio-ereduak

Sare-aplikazio baten aplikazioko entitateen arteko harremana nolakoa den, honako eredu hauetako bati jarraituz diseinatzen dira aplikazioak:

- Bezero/zerbitzari eredua. Hau da, gehienez, zabalduena. Eredu honetan, bi motatako entitateen zeregina desberdina da:
  - Bezeroek erabiltzailearekiko interfazearena egiten dute, zerbitzariari helarazten dizkiote erabiltzailearen eskaerak, eta horren erantzunak jaso eta erabiltzaileari aurkezten dizkiote. Batzuetan, zerbitzariak emandakoa prozesatu ere egiten dute.
  - Zerbitzariek datuak gorde eta prozesatzen dituzte. Aplikazio batzuetan, zeregin desberdinetako zerbitzariak agertzen dira. Normalki, zerbitzariek ez dute erabiltzailearekiko interfazerik. Aplikazioak erabiltzaileari ematen dion zerbitzua zerbitzarietan datza. Zerbitzariak ez badaude atzigarri, jai dute bezeroek.

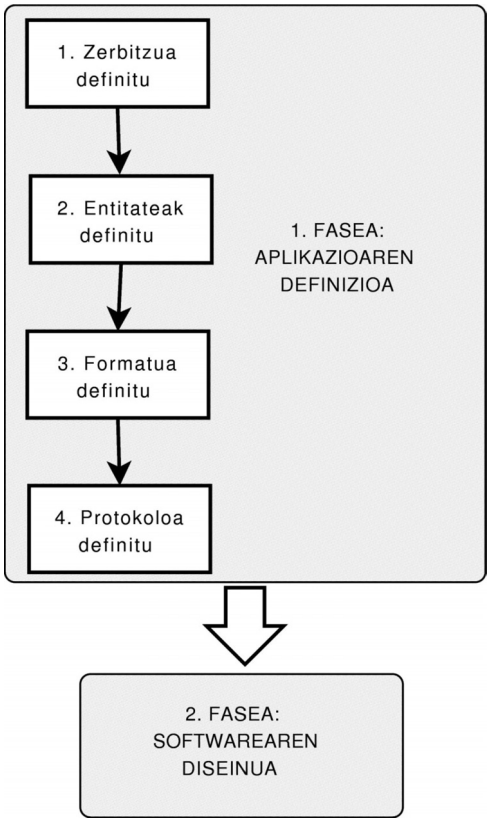
  Eredu honen adibideak, besteak beste, weba eta posta elektronikoa ditugu.
- P2P eredua (*Peer to Peer*). Eredu honetan, aplikazioko entitate guztiek zeregin berdinak dituzte. Hortik datorkio izena: berdinen artekoa da harremana. Hortaz, aplikazioko entitate guztiek betetzen dituzte edozein aplikaziotako atalak: erabiltzailearekiko eta datuekiko interfazeak, sarealdea, eta datu-prozesamendua. Ez dago aplikazioaren oinarri den eta beti atzigarri egon behar duen zerbitzaririk; harremanak aldizka konektatzen diren eragileen artekoak dira.

  P2P ereduaren erabileraren adibideak dira fitxategiak konpartitzeko erabiltzen diren hainbat aplikazio (*eMule*), IP telefonia-sistema batzuk (*Skype*), edo hainbat IP telebista-sistema dira.

Batzuek hirugarren eredu bat onartzen dute, aurreko bien artekoa. Eredu hibrido horretan komunikazioa era berdineko entitateen artekoa da (P2P eran, alegia), baina beraien artean elkarren berri izateko zerbitzariak erabiltzen dituzte. Zerbitzari horiek prest dauden entitateen katalogoa gordetzen dute. Entitate batek, edozein komunikazio hasteko, lehenago zerbitzariarekin komunikatzen du, prest dauden entitateen artean bere solaskidea(k) aurkitzeko. Fitxategiak partekatzeko aplikazio askok eredu misto honi jarraitzen diote.

4.1.2. Aplikazio banatu bat diseinatzeko urratsak

Sare-aplikazio baten diseinua bi fasetan antolatu behar da. Lehenengo fasean aplikazioaren egitura osoa definituko da, eta, bigarrenean, egitura horren softwarea diseinatuko da. Hemen lehengo fasea besterik ez dugu aztertuko. Bigarren fasea Softwarearen Ingeniaritzari dagokionez, Informatikaren arlo horri buruzko testuren batean aurkituko duzue fase horren deskribapena.



4.2. irudia. Sare-aplikazio baten diseinuaren faseak. Testu honetan lehenengo faseari besterik ez diogu ekingo.

Aplikazioa definitzeko urratsak honako hauek dira:

1. Aplikazioaren zerbitzua definitu.

   Aplikazioak erabiltzaileari zer eskainiko dion definitu behar da lehenengo urrats honetan. Horrekin batera, aplikazioaren erabiltzailea zein izango den ere definitzen da (kontuan izan aplikazio baten erabiltzailea batzuetan ez dela izango gizakia, baizik eta beste aplikazio bat).

2. Aplikazioko entitateak definitu.

   Aplikazioaren software-osagaiak zein izango diren definituko dugu. Gauza asko definitu behar dira urrats honetan, aldi berean:

   – Alde batetik, zeink izango diren aplikazioko entitateak.
   – Bestetik, zein izango den horietako entitate bakoitzaren zeregina, aplikazioaren zerbitzua gauzatzeko lanetan.

– Azkenik, nolakoa izango den entitateen arteko harremana. Gogoratu entitate horiek elkarlanean aritu behar dutela lehenengo urratsean definitutako zerbitzua betetzeko.

Diseinuko lehenengo fasearen bigarren urrats honetan aukeratu beharko dugu zein den aplikazioaren eredua, bezero/zerbitzaria ala P2P. Diseinuaen bigarren fasean, orain definitutako software zati hauetako bakoitzaren diseinua egin beharko da.

3. Informazioaren formatua definitu.

   Askotan, baina ez beti, aplikazioarekin batera formatu berri bat definitu behar da. Adibidez, posta elektronikoa asmatu zenean, horrekin batera posta elektronikoko mezuen formatua definitu zen. Era berean, webarekin batera HTML formatua sortu zen. Baina fitxategien transferentziarako FTP aplikazioa sortu zenean, ez zen beharrezkoa izan beste inongo formaturik sortzea. Hala ere, definitu zen, bai, zein formatutako fitxategietarako zegoen definituta aplikazioa: hasiera batean, ASCII formatuko fitxategiak besterik ez zegoen transmititzea. Hortik gutxira, aplikazioaren zerbitzuaren definizioa zabaldu zuten, edozein formatutako fitxategia transmititzeko ahalmena gehituz.

   Formatu berri bat (edo batzuk) definitu behar bad(ir)a aplikazioan, ez dago guztiz argi definizio hori noiz egin behar den. Batzuetan formatu horren definizioak ez du inongo eraginik izan behar 2. urratsean egindako entitateen definizioan. Horren adibidea posta elektronikoa da. Mezuen formatua bat edo bestea izateak ez du baldintzatzen aplikazioko entitateen lana. Baina kontrako adibidea DNS da: kapitulu honetan bertan ikusiko dugunez, aplikazio horren entitateen definizioa eta beraien arteko harremana zehazteko, aplikazioarekin batera definitzen den izen-sistema nolakoa den definituta izan behar dugu. Kasu horretan, izen-sistema horretan erabiltzen diren izen-formatuak aplikazioko entitateen definizioa baino lehenago zehaztu behar dira.

4. Aplikazioko protokoloa definitu.

   Behin aplikazioko entitateak definituta, eta erabili behar diren formatuak zehaztuta, aplikazioko entitateen arteko elkarrekintza arautuko duen protokoloa (edo protokoloak) definitu behar da. Azken urrats honetan sakonduko dugu hurrengo atalean.

4.1.3. Protokolo baten espezifikazioa

Aplikazioko protokoloa aplikazioaren araberakoa da guztiz. Aplikazio batzuetan aplikazioko entitateen arteko elkarrizketa oso sinplea da: eragiketa batzuk besterik ez dira egiten, trukatu behar diren datuek ez dute egiturarik, eta bezero/zerbitzari bikote bakarra dabil elkarrizketan. Beste askotan, aldiz, bezeroek zerbitzariei eragiketa ugari egiteko eska diezaiekete, datu mota asko erabil ditzakete

eragiketa horiek parametrizatzeko eta haien emaitzak jasotzeko eta, gainera, eragiketa bat gauzatzeko zerbitzari batek baino gehiagok har dezakete parte. Kasu horretan protokoloa konplexua izango da; izan ere, hobe izango da batzuetan protokolo bat baino gehiago definitzea. Konplexutasun horren ondorioz, erraza da protokoloaren definizioan akatsak, anbiguotasunak, edota gabeziak egotea, baina oso zaila izaten da akats, gabeziak eta anbiguotasun horiek atzematea protokoloa inplementatu eta aplikazioa abiatu baino lehenago. Horregatik, ikerlan asko egin da protokoloen definizioa eta egiaztapena era formal batean egiteko. Teknika matematikoetan oinarritutako formalizazio horien helburua bikoitza da: alde batetik, protokoloen definizioa argitzea eta erraztea, anbiguotasunak desagerrarazteko, eta, beste alde batetik, protokoloaren zuzentasuna matematikoki bermatzea, hau da, bere definizioan akatsik eta gabeziarik ez dagoela bermatzea. Gainera, horrelako formalizazio batek ahalbidetuko zuen protokoloaren zuzentasuna automatikoki egiaztatzea.

Zoritxarrez, formalizazio-ahalegin horiek ez dute fruitu handirik eman. Grafoen teorian eta automaten teorian oinarritutako teknika batzuk erabiltzen dira maiz protokoloen deskribapenetan, baina horrek ez du ahalbidetzen protokoloaren zuzentasuna formalki bermatzea. Asko jota, teknika horiek errazten eta argitzen dute protokoloaren deskribapena, eta, berez, haren analisia. Horregatik, protokoloak definitzeko mintzaira naturala erabiltzen da gehienetan, egoera-makinak eta antzeko teknika formalekin aberastuta batzuetan. Hori da behintzat Internet eta TCP/IP inguruko protokoloekin gertatzen dena. Protokolo horiek dagoeneko testu honetan askotan aipatu ditugun RFC (Request for Comments) izeneko agirietan definitzen dira, eta agiri horiek ingelesez idatzitako testuak dira.

Hortaz, ez dago protokoloak espezifikatzeko lengoaia formalik, edo, behintzat, ez dago erabilera handikoa den horrelakorik. Hala ere, mintzaira naturalaz egindako protokoloaren deskribapena ahal den zehatzena, osoena, eta anbiguotasunik gabekoa izan dadin, ezinbestekoa da espezifikazio hori minimoki egituratzea. Jarraian duzu egituraketa hori egiteko proposamen bat eta horren erabileraren adibide xume bat.

Protokoloen espezifikaziorako proposamena

Protokolo baten espezifikazioak honako hiru atal hauek izango ditu: mezuen sintaxiaren definizioa, mezuen semantikaren definizioa, eta mezuen erabileraren definizioa. Ikus ditzagun banan-banan.

- Mezuen sintaxiaren definizioa.

  Atal honetan zehazten da zein diren elkarrizketarako mezu posibleak. Definitutako arau sintaktikoak betetzen ez dituzten mezuak ulertezinak izango dira aplikazioko entitateentzat, eta baztertuak izango dira.

Mezuen sintaxiari dagokionez, honako bi talde hauetan sailkatzen dira protokoloak:

- Alde batetik, karakterezko protokoloak daude. Hau da, protokoloaren mezuak karaktereka kodetzen dira. Protokolo hauek lantzeko errazagoak egiten zaizkigu gizakioi, mezuak testu gisa idatz ditzakegulako. Kapitulu honetan bertan ikusiko dugunez, aplikazio-mailako protokolo gehienak horrelakoak dira, ASCII kodean oinarrituta.
- Beste alde batetik, bitezko protokoloak daude. Protokolo hauen mezuak interpretatzeko bitez bit aztertu behar dira. Karakterekoak baino eraginkorragoak dira, bit gutxiago behar dituztelako mezuak kodetzeko, baina gizakientzat askoz zailagoak dira lantzeko, hizkuntza bitarra mintzaira naturaletik urrutikoa delako. Aplikazio-mailatik beherako protokoloak bitekoak dira gehienetan, aurreko kapituluetan ikusi dugun bezala. Horien adibideak dira TCP, UDP eta IP protokoloen informazio-unitateen definizioak (segmentuak eta datagramak).

- Mezuen semantikaren definizioa. Mezu bakoitzaren esanahia argitu behar da, eta, horrekin batera, mezuak zertarako erabiliko diren definitzen da. Bi motatako mezuak egon ohi dira: informazioa garraiatzekoak eta kontrolerako mezuak. Beraien semantika definitzean, mezuetan agertzen den eremu bakoitzaren esanahia adierazi behar da.
- Prozeduren definizioa, edo mezuen erabileraren adierazpena. Prozeduren definizio honetan, aplikazioko entitateek elkarlanean burututako eragiketa bakoitzeko, zer mezu eta zer ordenatan trukatzen diren ezartzen da. Eragiketa bakoitzeko prozedura bat definitu behar da. Atal honetan erabiltzen dira maiz grafoak eta automatak. Horren adibide bat 4.3. irudian duzu, TCP protokoloaren definizioaren zatia dena.

  Prozeduren definizio honetan ezarriko da zein den protokoloak jarraitutako komunikazio-eredua. Komunikazio-eredua bitako bat izan daiteke: konexio bidezkoa ala konexiorik gabekoa. Bata zein bestea aukeratuta, protokoloak gauzatzen duen zerbitzuaren izaera ere horrelakoa izango da. Hau da, protokoloaren komunikazio-eredua eta zerbitzu mota berdinak izango dira. Adibidez, IP protokoloa konexiorik gabeko protokoloa denez, IP zerbitzua konexiorik gabeko zerbitzua edo datagrama-zerbitzua da. Gogora ditzagun bi eredu horien ezaugarriak:

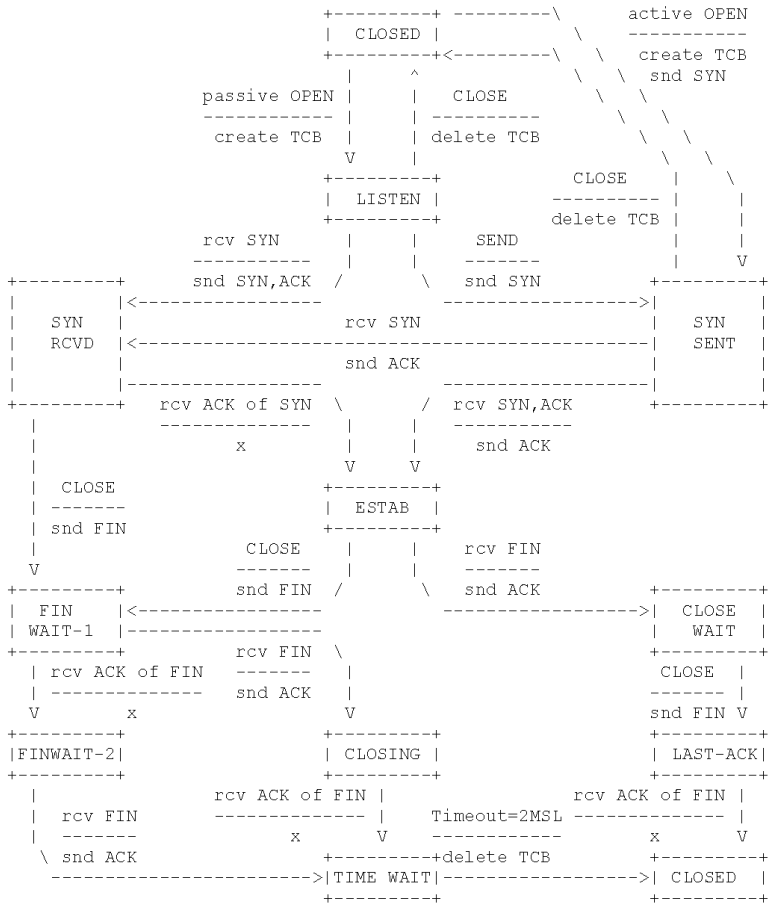
  - Konexio bidezko protokoloetan komunikazioa hiru fasetan antolatzen da:
    - Konexioa ezarri. Hau da, ezer egiten hasi baino lehen, komunikazioaren bi aldeak ados jartzen dira hitz egiteko, eta, beharrezkoa izatekotan, elkarrizketaren baldintzak ezartzen dituzte. Adibidez, aplikazio askotan, zerbitzariak bezero batekin lan egin baino lehen, bezero

horren atzean dagoen erabiltzailea identifikatzera behartuko du. Erabiltzaile horrek zerbitzari horrekin lan egiteko baimena duela egiaztatu eta gero, orduan hasiko dira elkarlanean aplikazioak eskaintzen dituen zerbitzuak erabiltzaile horri emateko.

- Komunikazioa gauzatu. Hau da, behar diren prozedurak bete erabiltzaileak eskatutako zerbitzuak gauzatzeko.
- Konexioa amaitu. Lan-saioaren amaieraren berri ematen diote elkarri bi solaskideek, eta biak ados badaude, bukatutzat joko dute elkarrizketa.

Ikusi dugun konexio bidezko protokolo baten adibidea TCP da.

– Konexiorik gabeko protokoloetan, aldiz, ez da faserik bereizten. Entitate batek beste batekin hitz egin behar duenean, zuzenean bidaltzen dizkio bere eskaerak, inongo agurrik edo aurreko negoziaziorik gabe. Honen adibideak dira UDP eta IP.



4.3. irudia. TCP egoera finituko makina, RFC 793 agirian agertzen den bezala.

Aplikazio-mailako protokoloen artean ez dago nagusia den komunikazio-eredurik. Asko konexio bidezkoak dira (SMTP, POP3…), eta beste asko konexiorik gabekoak (DNS, HTTP…). Eredu bat edo beste erabiltzea aplikazioaren araberakoa da.

Protokolo baten espezifikazioan protokolo horrek erabiliko dituen garraio-zerbitzuak ere definitzen dira. Hau da, protokoloak TCP ala UDP erabiliko duen zehazten da. Teorian, hau ez da espezifikazioaren zati bat, inplementatzailearen aukera baizik. Hala ere, praktikan, protokoloaren definizioan hori zehazten da. Izan ere, askotan, protokoloaren prozeduren definizioan eragin handia izaten du TCP ala UDP erabiltzeak. Aurreko kapituluaren bukaeran TCPren eta UDPren artean aukeratzeko irizpide batzuk dituzu.

Adibidea

Ondoan aplikazioko protokolo xume baten adibidea duzu. Aplikazioa fitxategiak banatzeko sistema bat da. FTP aplikazioa ezagutzen baduzu, antza hartuko diozu. Protokoloaren definizioa egin baino lehenago, diseinuko aurreko urratsak bete behar ditugu. Laburrean:

- Aplikazioaren zerbitzuaren definizioa.

  Aplikazioa fitxategiak banatzeko sistema xume bat da. Fitxategiak eskuratu ahal izateko, onartuta dagoen erabiltzaile-izen bat eman behar zaio aplikazioari. Baimendutako erabiltzaileak honako bi zerbitzu hauek besterik ez du jaso ahal izango:

  – Eskuragarri dauden fitxategien zerrenda ikusi.
  – Zerrendako fitxategi bat eskuratu.

- Aplikazioko entitateen definizioa.

  Bezero/zerbitzari moduko aplikazio bat izango da, non zerbitzariek gordeko dituzten fitxategiak, eta bezeroek zerbitzarietatik jaitsiko dituzten fitxategi horiek.

- Formatuaren definizioa.

  Ez dugu formatu berezirik definituko aplikazio honetarako. Trukatutako fitxategiak egituratu gabeko bit multzoak bezala tratatuko dira.

Aurrekoak definituta izanda, ekin diezaiogun protokoloaren espezifikazioari. Bitez *komandoak* bezeroak zerbitzariari bidali behar dizkion mezuak, eta *erantzunak* kontrako noranzkoan doazenak.

Sintaxia

Bezeroaren eta zerbitzariaren artean bidaltzen diren mezu guztiek honako egitura hau izango dute:

- 4 ASCII formatuko byte, bidaltzen ari den komandoa edo erantzuna adierazten dutenak. Hauek izan daitezke:
  - Komandoak: ERAB, ZERR, FITX, BIDA, BUKA.
  - Erantzunak: ADOS, KALE.
- 4 byte horien ondoren, hiru aukera daude:
  - Besterik ez izatea.
  - String bat ('\0'-z bukatutako ASCII karaktere-katea) agertzea, 80 karakterekoa asko jota.
  - Zehaztu gabeko datu sorta agertzea.

Semantika

Taula honetan adierazten da komando eta erantzun bakoitzaren semantika.

Komandoa/ erantzuna	Parametroak/ Datuak	Esanahia
ERAB	Erabiltzailearen izena (*string*)	Saioa irekitzeko eskaera
ZERR	(ezer ez)	Eskuragarri dauden fitxategien zerrendaren eskaera
FITX	Fitxategiaren izena (*string*)	Fitxategi jakin bat eskuratzeko eskaeraren lehenengo urratsa
BIDA	Fitxategiaren tamaina (*string*)	Fitxategi jakin bat eskuratzeko eskaeraren bigarren urratsa
BUKA	(ezer ez)	Saioa ixteko eskaera
ADOS	Fitxategi baten tamaina darama, fitxategi baten edukia (datu sorta), edo ezer ez	Komando baten onespena
KALE	Ezer ez edo ukapenaren zergatia	Komando baten ukapena

4.1. taula. Adibideko protokoloaren semantikaren definizioa.

Prozedurak

- Saio bat ezartzeko prozedura:

  Erabiltzailearen kautotze-mekanismoan datza. Hau urrats bakar batean egingo da, erabiltzailearen identifikazio onartuaren bidez (pasahitzak-eta alde batera utzita):

  – Bezeroak ERAB komando bat bidali behar du erabiltzailearen identifikazioarekin batera. Zerbitzariaren erantzuna hauetako bat izango da:
    - ADOS, saioa irekitzea onartzen badu.
    - KALE, bestela. Horrela bada, jakina, saioa itxita mantenduko da eta ezin izango da ezer egin.

- Eskuragarri dauden fitxategien zerrenda eskatzeko prozedura:

  ZERR izeneko komandoa, besterik gabe, bidali beharko du bezeroak. Zerbitzariak jasotakoan erantzungo du hauetako mezu batez:

  – ADOS eta jarraian kopuru zehaztugabeko karaktere sorta (ez string), asko jota 1496 karakterekoa. Ohar zaitez datu sorta Ethernet trama bakar batean kabitzen dela.
  – KALE, bestela. Horrela bada, saioa itxita geratuko da.

- Fitxategi jakin bat eskuratzeko prozedura:

  Bezeroak nahi duen fitxategiaren izena behin esanda, zerbitzariak, fitxategia bidaltzen hasi baino lehen, fitxategiaren tamaina jakinaraziko dio, byteak jasotzen noiz arte egon beharko den adierazteko. Beraz, eragiketa hau honako bi urrats hauetan egingo da:

  1. Lehenengoa: bezeroak FITX komandoa bidaliko du eskuratu nahi den fitxategiaren izenarekin batera, eta zerbitzariak itzuliko du:
     - ADOS, eta fitxategiaren tamaina (bera da hau ezagutzen duen bakarra), edo,
     - KALE, bestela. Kasu honetan, erabiltzaileari egoera jakinarazten zaio eta beste eragiketa bat egiteko aukera ematen zaio. Saioa ez da ixten.
  2. Bigarrena (ADOS jaso bada): bezeroak BIDA komandoa bidali behar du eta, berriro, fitxategiaren tamaina, zerbitzariaren ADOSaren onespen gisa. Orduan, zerbitzariak itzul dezake:
     - ADOS eta, jarraian, fitxategiaren edukia, edo,
     - KALE, bestela. Honetan, saioa ez da itxiko.

- Saioa bukatutzat emateko prozedura:

  Bezeroak BUKA komandoa bidaliko dio zerbitzariari, parametrorik gabe, eta zerbitzariak, ADOS erantzuna bidaliz, parametrorik gabe, saioa itxiko du.

Ohiko akatsak protokolo baten espezifikazioan

Honako hauek dira:

- Zerbitzuko hutsuneak.

  Hau da, protokoloak ez du ahalbidetzen aplikazioaren zerbitzu guztiak gauzatzea. Gehienetan prozedura bat definitu ez delako, edo prozeduraren batean zer edo zer falta delako gertatzen da hau.

- Blokeoak.

  Komunikazioaren bi aldeak beste aldeak zer edo zer bidaltzeko zain gelditzen direnean sortzen da blokeo bat. Prozedura baten definizioan balizko egoera guztiak ez aurreikusteagatik sortzen dira blokeo gehienak.

- Akats ezkutuak.

  Aurreko bi akatsak aplikazioa erabiltzean atzeman daitezke. Askoz arriskutsuagoak dira erabiltzaileak zuzenean atzematen ez dituen akatsak, gerta daitekeelako erabiltzaileak ontzat ematea aplikazioaren egikaritzapena, baina benetan lana gaizki eginda edo egin gabe egotea. Adibidez, larria litzateke adibideko aplikazioarekin programa bat deskargatzea, eta, nahiz eta aplikazioaren arabera dena ondo joan eta eskatutako fitxategia gure diskoan gorde, benetan deskarga osoa ez izatea, eta, beraz, jaitsitako programa ez ibiltzea.

- Erredundantziak.

  Hau akats arina ohi da. Ondokoan datza: protokoloaren espezifikazioa optimoa ez izatea, eta beraz, behar diren baino baliabide gehiago xahutzea zerbitzu bat emateko. Adibidez, espezifikazioan ager daitezke sobera dauden komandoak edo goiburuko eremuak. Alde horretatik ikusita, karakterezko protokoloak beti dira erredundanteak, bitekoak baino bit gehiago erabiltzen dituztelako informazio bera garraiatzeko. Larriagoak izaten dira prozeduretan agertzen diren erredundantziak, hau da, zerbitzu bat gauzatzeko solaskideek gehiegizko mezuak trukarazten dizkiotenean elkarri.

Protokoloa inplementatu baino lehenago erroreak atzeman ahal izateko, garrantzitsua da espezifikazio argia eta zehatza egitea. Errazagoa da akatsak inplementatze-lanetan ari garenean atzematea, baina askoz neketsuagoa da orduan egitea. Akatsak inplementatze-fasean agertzen badira, gerta liteke protokoloaren definizioa bera aldatu behar izatea akats horiek zuzentzeko. Protokoloaren birdefinitzeak aplikazioaren diseinu osoan eragin dezake, eta kostu handiko diseinatze-inplementatze-birdiseinatze ziklo batean sarrarazi. Beste alde batetik, okerrena aplikazioa erabiltzen ari denean akatsak atzematea litzateke. Hori ere ekiditeko, abiapunturik hoberena protokoloaren espezifikazio ona egitea da.

4.2. DNS

Aurreko kapitulu batean ikusi dugun legez, Interneten dauden konputagailuak IP helbideen bidez identifikatzen dira. Aplikazio gehienetan, erabiltzaileak adierazi behar dio bere bezeroari zerbitzaria zein konputagailutan dagoen kokatuta. Horretarako IP helbideak erabil daitezke, baina gizakientzat ez da batere erosoa, IP helbideak gogoratzeko zailak direlako. Horregatik, Interneten izen-sistema bat definitu da, erabiltzaileek makinak izendatzeko. Izenak, esaterako, `www.rfc-editor.org`, `jazzvitoria.com`, `www.konektazaitez.net`, `gaia.cs.umass.edu`, `mailin.sc.ehu.es` edo antzekoak dira; mnemoteknikoak dira eta, beraz, pertsonek estimatuak.

Beraz, honako bi bide hauek daude sareko konputagailuak identifikatzeko: izenak eta IP helbideak. Jendeak nahiago du izenak erabiltzea, baina konputagailuek IP helbideak behar dituzte. Bi nahi horiek adiskidetzeko, izenen eta IP helbideen arte-ko itzulpena egiten duen direktorio-zerbitzu bat behar dugu. Hau da Interneteko **Domain Name System** delakoaren **(DNS)** lan nagusia.

DNSaren funtsa izen-eskema hierarkiko bat eta izen-eskema hori gauzatzeko milaka konputagailutan banatutako datu-basea da. DNS RFC 1034 eta RFC 1035ean definitua dugu, eta eguneratua beste RFC batzuetan. Sistema konplexua da; hemen aplikazio honen oinarriak eta funtzionamendua aztertuko ditugu.

DNS zerbitzuaren definizioa

DNS direktorio-zerbitzu bat da. IP helbideak eta DNS izenak lotzen ditu. Hala ere, funtsezko zerbitzu horretatik harago, beste zerbitzu osagarri batzuk ere ematen ditu DNSk. Ondokoak dira nagusiak:

- Konputagailuak izen bat baino gehiago erabiltzeko aukera ematen du, goitizen-zerbitzua alegia (*alias*).
- Posta-zerbitzariak identifikatzeko zerbitzua. Geroxeago, kapitulu honetan bertan, aztertuko dugu zerbitzu hau.
- Oro har, beste aplikazioetako zerbitzariak identifikatzeko zerbitzua. Adibidez, IP telefoniarako zerbitzariak bilatzeko aukera ere ematen du DNSk.

DNS ez dago pentsatuta gizakiei zuzenean zerbitzua emateko. Gizakiok DNS izenak erabiltzen ditugu. Guri zerbitzuak zuzenean ematen dizkigun aplikazioei izen horiek ematen dizkiegu, adibidez, posta elektronikoko helbideetan edo web orri bat atzitzeko gure arakatzaileari eskatzen diogunean. Gero, aplikazio horiek eskatuko dizkiote DNSri izen horri dagozkion datuak (gutxienez, bere IP helbidea). Hala ere, baditugu gizakioi DNS erabiltzea ahalbidetzen diguten programak (adibidez, Unix munduko *host* komandoa).

4.2.1. Aplikazioaren osagaiak

DNS bezero/zerbitzari moduko sare-aplikazio bat da. Honako hauek dira haren osagaiak:

- Bezeroa.

  Beste bezero/zerbitzari aplikazio askotan ez bezala, DNS bezeroa ez da programa edo prozesu bat, errutina batzuk baizik. Linux konputagailuetan, oro har, aplikazioen bezeroek `gethostbyname()` liburutegiko funtzioa erabiltzen dute DNS zerbitzua atzitzeko. Beste aplikazioen bezeroei egiten zaien bezala, DNS bezeroari izen berezia ematen zaio: **ebazlea** (*resolver*). Beste programek (adibidez, arakatzaile batek) ebazlea osatzen duten errutinak erabiltzen dituzte DNS zerbitzuak erabili behar dituztenean.

- Zerbitzariak.

  Era desberdinetako zerbitzariak behar dira DNSren zerbitzua emateko. Mota bakoitzaren zeregina eta beraien arteko harremana oso lotuta daude DNS izenen eta izen horiek gordetzen dituen datu-basearen egiturarekin.

- DNS protokoloa.

  Hau da bezeroen eta zerbitzarien arteko harremanetarako erabilitako protokoloa, baita zerbitzarien arteko komunikazioetan ere.

- Aplikazioko informazioaren formatua.

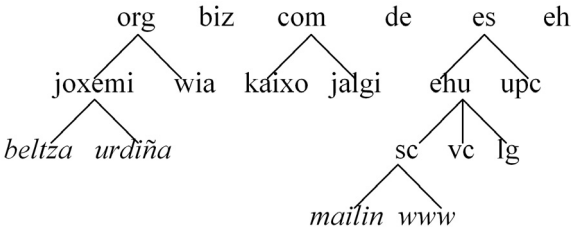
  DNS entitateek (bezeroak eta zerbitzariak) **DNS erregistroak** elkarri bidaltzen dizkiote. Erregistroetan **DNS izenak** eta **domeinuak** aurkituko ditugu. Haien egitura eta haiek osatutako DNS datu-basea dira DNSren oinarria. Beraien definizioak aplikazio osoaren diseinua eta funtzionamendua baldintzatzen du. Horregatik hasiko dugu hortik gure DNSren deskribapena.

4.2.2. DNS izenak eta domeinuak

Interneteko konputagailuak domeinuetan elkartzen dira. Domeinu batean administratiboki lotuta dauden konputagailuak elkartzen dira. Definizio honek IP mailan ikusitako sistema autonomoak ekarriko dizkigu burura. Oso antzekoak dira bi kontzeptuak; izan ere, oro har, sistema autonomo bateko konputagailu guztiak domeinu berean egoten dira. Baina horrek ez du beti horrela izan behar; sistema autonomoen eta domeinuen arteko erlazioa ez dago definituta: bata IP mailan erabiltzen da, bideratzeko lanetan, eta bestea TCP/IP sare-arkitekturatik at gelditzen den kontzeptu bat da, erabiltzaileek Interneten erabilera eta kudeaketa errazteko erabiltzen dutena.

Domeinu baten barruan azpidomeinu asko egon daitezke, eta, aldi berean, horietako bakoitza hainbat azpidomeinutan banaturik egon daiteke. Horrela,

domeinu-sistema 4.4. irudian ageri den zuhaitz baten moduan hierarkikoki antolatuta dago. Zuhaitzaren hostoak konputagailuak dira.



4.4. irudia. Interneteko izen-zuhaitzaren zati bat. Konputagailuak letra etzanean agertzen dira. Beste guztiak domeinuak eta azpidomeinuak dira.

Hierarkiaren goren puntuan dagoen domeinuaren izena azkena idazten da izen batean, eta bere aurretik azpidomeinu guztiak, hierarkian behera eta puntuz bereizita. Ezkerreko muturreko izena konputagailu batena denean, konputagailu hori izen osoarekin identifikatzen da; azpidomeinu batena denean, azpidomeinu horretan dagoen konputagailu multzoa erreferentziatzen ari gara. Aurrean aipatutako `mailin.sc.ehu.es` izenaren kasuan, adibidez, konputagailu jakin batez ari gara, `sc.ehu.es` domeinuan dagoen eta `mailin` izena hartu duen batez. Izenek ez dituzte letra larriak eta xeheak kontuan hartzen eta, beraz 'ehu' eta 'EHU' gauza bera dira. Azpidomeinuek 63 karaktereko luzera ere eduki dezakete, eta izen osoek ezin dute 255 karaktere gainditu.

Izenen kudeaketa

Goi-mailako domeinuak (TLD -*Top Level Domain*) mota honetakoak izan daitezke:

- Herrialdekoak (ccTLD – *country code TLD*). Hauek dira bi letra duten TLDak (.fr, .es, edo .de modukoak). *Herri* bat zer den ebaztea ez da lan erraza batzuetan, eta, horregatik, ICANNek onartzen dituen herrialdeetako domeinuak ISO 3166 agirian agertzen direnak dira. Honako esteka honetan daude zerrendatuta: www.iana.org/root-whois. ccTLDen kudeaketa dagokion herriaren gobernuak erabakitako erakundeak egiten du. ccTLDei buruzko informazio gehiago www.iana.org/cctld/cctld.htm estekan duzu.
- Orokorrak (gTLD – *general TLD*). Hauek dira bi letra baino gehiago dituztenak. Ondoko bi azpitaldetan banatzen dira:
  - Babeslerik gabekoak (*unsponsored TLDs – uTLDs*). Hauen kudeaketarako irizpideak ICANNek ezartzen ditu zuzenean. Honen adibidea .com, .net, eta .org domeinu historikoak dira. Gero beste asko gehitu dira (adibidez, .info edo .name), eta etorkizunean gehiago agertuko dira.

  - Babestuak (*sponsored TLDs - sTLDs*). Domeinu hauek Interneten erabiltzaile talde mugatuentzako sortzen dira, lurraldekoak bezalakoak, baina kasu honetan geografiko-politikoa ez den beste irizpideren baten arabera. Kategoria honetan sar daitezke hasierako beste domeinu historikoak (.gov, .mil, .edu, eta .int). Gero, beste asko sortu dira, erabiltzaile taldeek eskatuta. Horren adibideak dira .aero (industria aeroespaziala) eta .cat (kultura katalana). Domeinu babestuaren kudeaketa erakunde babeslearen esku gelditzen da. Domeinu babestuen zerrenda ere handitzen ari da denboraren ahala.

- .arpa domeinu berezia. Azpiegitura teknikoen beharretarako erabiltzen da. Bere erabilera ezagunena kontrako itzulpenak ahalbidetzea da (hau da, IP helbide batetik, dagokion izena eskuratzea). ICANNek kudeatzen du, IABk zuzenduta.

TLD guztiak, herrialdekoak eta orokorrak, beste esteka honetan dituzu: http://data.iana.org/TLD/tlds-alpha-by-domain.txt.

Bigarren mailako domeinu bat lortzeko, ICANNek (babeslerik gabeko TLDentzako) edo babesleak (beste guztietan) baimendutako erregistratzaileengana jo behar da, eskatu, eta, erabili gabe baldin badago eskatutako domeinua, kuota bat ordaindu domeinu horren kudeaketaren truke. Gehienetan, domeinuarekin batera IP helbide multzo bat lortuko da, hirugarren kapituluan aipatu dugun bezala.

Domeinu bakoitzaren kudeatzaileak bere azpiko domeinuak esleitzeko modua kontrolatzen du. Beste domeinu bat sortzeko, bere baitan hartuko duen domeinuaren baimena behar da. Adibidez, Euskal Herriko Unibertsitateko Arkitektura Eskolak Interneten ark.ehu.es domeinu propioa izan nahi badu, ehu domeinua kudeatzen duenaren baimena beharko du. Era berean, ark.ehu.es domeinuan bere konputagailua sartu nahi duen irakasleak izen bat esleitzeko eskatu beharko dio domeinu hori kontrolatzen duenari.

Izenek erakundeen mugak erakusten dituzte, ez sare fisikoak. Adibidez, Fisikako eta Astrofisikako sailak eraikin berean kokatuta baleude ere eta sare lokal bera konpartituko balute ere, domeinu desberdinak izan litzakete. Era berean, Astrofisikako departamentua elkarrengandik urrun dauden bi eraikinen artean banatuta balego ere, bi eraikinetako konputagailu guztiak domeinu berekoak lirateke normalki.

4.2.3. DNS erregistroak

DNSren informazioa baliabide-erregistroetan (*resource records* edo, askotan, RR bezala erreferentziatuak) biltzen da. Izen bakoitzari —konputagailu bakar batena edo domeinu batena izanik ere— gutxienez baliabide-erregistro bat dagokio. Baliabide-erregistroei **DNS erregistroak** ere deitzen zaie askotan. Erregistroak

DNS zerbitzarietan daude banatuta, datu-base bat osatuz. Hurrengo sekzioan aztertuko dugu nola aurkitzen den erregistro bat mundu osoan zehar barreiatutako datu-base horretan. Sekzio honetan datu-basearen edukia diren DNS erregistroen egitura adieraziko dugu.

DNS erregistroen lau eremu nagusiak honako hauek dira:

(*Name, Value, Type, TTL*)

Eraginkortasunagatik bitarrean kodetuta badaude ere, testuetan ASCII kodean adierazten dira. *TTL* eremuak erregistroaren iraupena cache memorian ezartzen du. Emandako adibideak argiagoak izateko, alde batera utziko dugu eremu hori. *Name* eta *Value* eremuen esanahia *Type* eremuaren menpean dago. Ondoan azaltzen dira *Type* eremuaren balio posibleen artean garrantzitsuenak direnak:

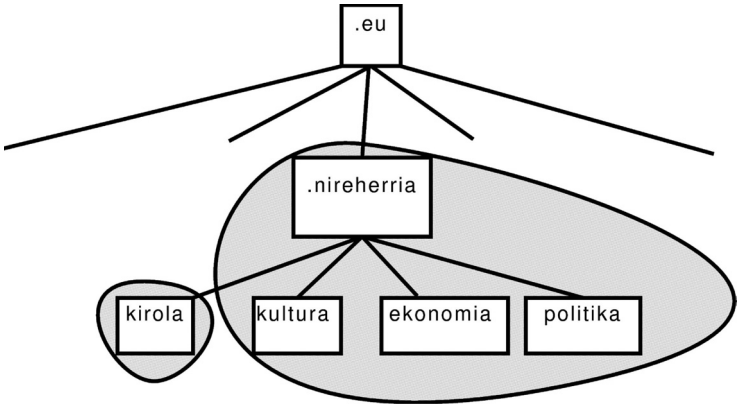
- *Type* = A bada, *Name* konputagailu baten izena izango da, eta ez domeinu batena. *Value* konputagailu horri dagokion IP helbide bat izango da. Adibidez (www.joxemi.org, 190.210.65.70, A) A motako erregistro bat da. Kontuan izan konputagailu batek IP helbide bat baino gehiago izan ditzakeela eta, beraz, A motako erregistro bat baino gehiago ere.
- *Type* = AAAA bada (*quad-A record*), aurrekoa bezalakoa da erregistroa, baina *Name* konputagailuaren IPv6 helbide bat izango dugu *Value* eremuan, eta ez IPv4 helbide bat.
- *Type* = NS bada, *Name* domeinu baten izena izango da (*joxemi.org* adibidez) eta *Value* domeinu horren konputagailuei buruzko erregistroak lortzeko galdetu behar zaion izen-zerbitzari baten izena da.
- *Type* = CNAME bada, *Value* konputagailu baten izen kanonikoa da, eta *Name* konputagailu horren goitizen bat. Adibidez, (*www.joxemi.org*, *zerbitzuak.joxemi.org*, CNAME) CNAME erregistro bat da, non *zerbitzuak.joxemi.org* izeneko konputagailuari *www.joxemi.org* goitizena lotzen zaion. Konputagailu batek goitizen bat baino gehiago baditu, CNAME erregistro bat baino gehiago ere izango ditu. CNAME erregistroen bidez ematen du DNSk goitizenen zerbitzua (*aliasing*). Zerbitzu hau oso erabilia da konputagailu batean aplikazio desberdinetako zerbitzariak kokatzen direnean. Goitizenak erabiliz, konputagailu bera erreferentziatzeko izen desberdinak erabiliko ditugu konputagailu horretan egikaritzen den aplikazio bakoitzean. Adibidez, *lgsx01.lg.ehu.eu* konputagailuak *www.ehu-leioa.eu* eta *mailin.ehu-leioa.eu* goitizenak eduki ditzake. Kasu honetan, *lgsx01.lg.ehu.eu* izena **izen kanonikoa** dela esaten da.
- *Type* = MX bada, *Name* eremuak domeinu baten izena du, eta *Value* eremuak domeinu horri dagokion posta-zerbitzari baten izena gordetzen du. Posta elektronikoa ikasten dugunean, kapitulu honetan bertan, MX erregistroak topatuko ditugu berriro.

4.2.4. DNS zerbitzariak eta ebazpenak egiteko prozedura

Domeinuak bezala, DNS erregistroak gordetzen dituzten zerbitzariak ere era hierarkikoan daude antolatuta. Zerbitzariak elkarlanean aritzen dira ebazle batek egindako DNS galdera bati dago(z)kion erregistroa(k) aurkitzeko. Hau da, nahiz eta ebazleak zerbitzari konkretu bati eskaera egin, zerbitzua DNS sistema osoak ematen du. Ikus dezagun nola egiten duen.

Domeinu-zuhaitza, DNS barrutiak, eta zerbitzari fidagarriak

DNS domeinu-zuhaitza barrutitan dago zatituta (***DNS zones***). Barruti batean DNS kudeatzaile bera duten zuhaitzaren nodoak (domeinuak) eta hostoak (konputagailuen izenak) elkartzen dira. Barruti batean dauden nodoei eta hostoei dagozkien DNS erregistroak barruti horretako DNS jatorrizko zerbitzariak (edo zerbitzari fidagarriak, ingelesez ***authoritative server***) gordetzen ditu. Zerbitzari hori da barrutiko DNS informazio-iturburua Internet osorako. Horregatik, barruti bakoitzeko egon behar du DNS zerbitzari nagusi batek (*primary server*), eta, gutxienez, haren kopia den laguntzaile batek (*secondary server*). Barruti guztietan egongo da domeinu-zuhaitzean altuena den nodo bat (izen bat, alegia). Nodo horren izena erabiltzen da normalki barrutia izendatzeko.



4.5. irudia. Domeinuen zuhaitza eta DNS barrutiak. Xehetasun gehiago RFC 1034 agiriko 4.2. atalean.

Domeinuen eta barrutien arteko erlazioa ez da bana-banakoa. Domeinu bakoitzaren kudeatzaileak erabakiko du noiz komeni den azpidomeinu bati (edo batzuei) dagokion barrutia bereiztea. Adibidez, demagun *nireherria.eu* domeinuaren kudeaketa lortzen dugula, eta hasiera batean, domeinu osoa, bere azpidomeinu guztiekin, barruti bakar batean elkartzen dugula. Hau da, zerbitzari bakar batek (eta bere laguntzaileek, noski) gordetzen ditu *nireherria.eu* domeinuari dagozkion jatorrizko DNS erregistroak. Demagun geroxeago, domeinuaren barruan dauden azpidomeinu eta konputagailuen (eta beraz, izenen) kopurua handiegia egiten dela,

eta segundoro jasotako galdera kopuruak gainezka eginarazten diola jatorrizko zerbitzariari. Bada unea bigarren barruti bat sortzeko. Demagun galdera kopururik handiena *kirola.nireherria.eu* azpidomeinuari dagokiola; domeinu hori kudeatzeko barruti berri bat sortuko dugu. Orduan, jatorrizko zerbitzariak izango ditugu *kirola.nireherria.eu* domeinuko informazioa gordetzeko, eta *nireherria.eu* domeinuko gainerako informazio guztia gordetzeko. Azken horren DNS zerbitzariak gordeko ditu *kirola.nireherria.eu* barrutiko DNS zerbitzariari dagozkion NS eta A motako erregistroak. Aurreko irudian adibideko domeinuen eta barrutien arteko erlazioa adierazten da.

Beraz, ebazle batek, erregistro baten bila dabilenean, dagokion jatorrizko zerbitzariari helarazi behar dio bere galdera. Interneten milaka direnez, nola aurkitu izen bati dagokion jatorrizko zerbitzaria?

TLD zerbitzariak eta erro-zerbitzariak

Jatorrizko zerbitzariak bilatzeko gidariarena egiten duten DNS zerbitzariak daude. Zerbitzari horiek NS erregistroak gordetzen dituzte barrutietako gidak osatzeko (beste mota bateko erregistroak ere badituzte, baina zerbitzari horien zeregina NS erregistroetan datza). Hasiera batean, ondoko bi mailako hierarkian daude antolaturik gida-zerbitzari hauek:

- TLD zerbitzariak.

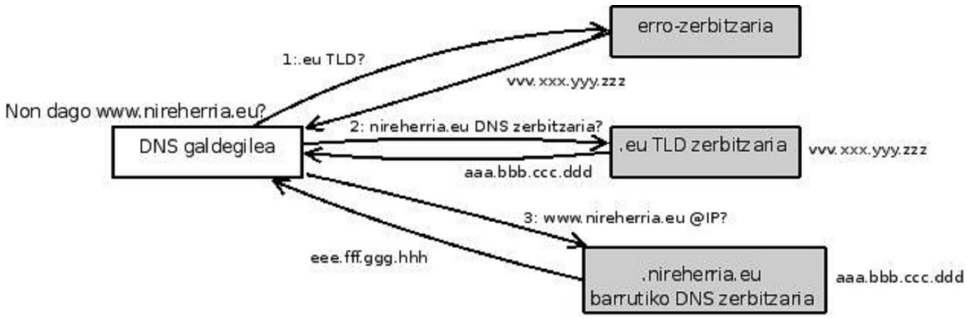
  Horrelako zerbitzari batek bere TLD azpian dagoen bigarren mailako domeinu bakoitzeko NS erregistro bat izango du domeinu horren goreneko barrutiari dagokion zerbitzaria zein den adieraziz. Adibidez, www.nireherria.eu izenari dagokion IP helbidea aurkitu nahi badugu, .eu TLDari dagokion TLD zerbitzariari galdetu beharko diogu zein den *nireherria.eu* domeinuaren informazioa duen zerbitzaria. Demagun zerbitzari hori *ns1.nireherria.eu* dela. Horri galdetuz topatuko dugu bilatutako IP helbidea.

- Baina TLDak 200etik gora direnez, guztien IP helbidea aurrekargatuta eta eguneratuta mantentzea ebazle guztietan ez da bideragarria. Horregatik behar da beste maila bat DNS gida-zerbitzarien hierarkian, TLD zerbitzarien IP helbideak emateko. Horiek dira **erro-zerbitzariak**. Izenez 13 dira, baina fisikoki askoz gehiago, bost kontinenteetan zehar barreiatuta. Denek gordetzen dute informazio beraren kopia, eta 13 horietako bati egindako galderak bere kopien artean banatzen dira anycast teknikak erabiliz (ikusi RFC 3258). Erro-zerbitzariei buruzko informazio zehatzagoa RFC 2826/2870 agirietan duzu.

Beraz, izen bati dagokion erregistroa aurkitzeko eman beharreko urratsak honako hauek dira:

1. Galdetu erro-zerbitzariren bati zein den izenari dagokion TLD zerbitzariaren IP helbidea. Aurreko adibidean, .eu domeinuari dagozkion TLD zerbitzariaren izena eta IP helbidea lortuko genituzke erro-zerbitzaritik.
2. TLD zerbitzari horri galdetu zein den izenari dagokion jatorrizko zerbitzariaren IP helbidea.
3. Jatorrizko zerbitzari horri eskatu DNS erregistroa.

Prozesu hori da ondoko irudian ageri dena.



4.6. irudia. Izen baten ebazpenerako oinarrizko prozedura.

Aurreko adibidearekin jarraituz, bilatutako helbidea *www.kirola.nireherria.eu* balitz, 4. urrats bat ere beharko genuke, 3. urratsean galdetutako *ns1.nireherria.eu* zerbitzariak ez baitu kudeatzen *kirola.nireherria.eu* domeinua duen barrutia. Haren erantzuna barruti hori kudeatzen duen DNS zerbitzariaren IP helbidea izango litzateke, eta, 4. urratsean, beste zerbitzari horrek emango liguke eskatutako IP helbidea, *www.kirola.nireherria.eu* izenari dagokiona, alegia. Hierarkian hainbat maila egon daitezkeenez, hainbat DNS zerbitzarik hartuko dute parte ebazpen batean, dagokion jatorrizko zerbitzariraino ailegatu arte.

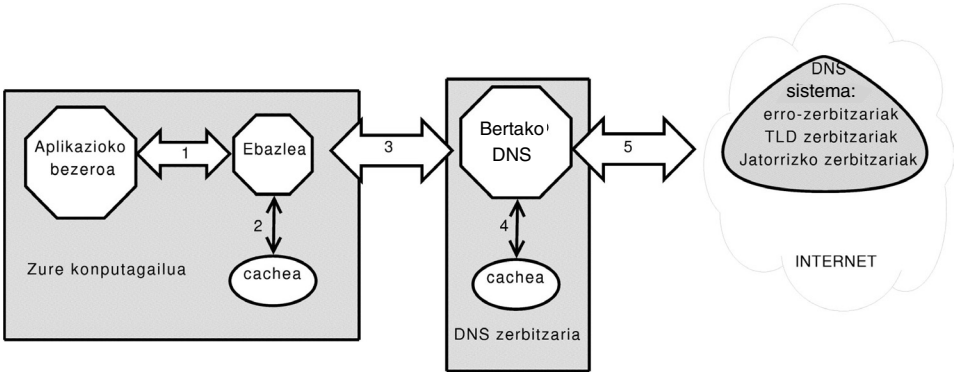
Bertako zerbitzariak eta cacheak

Deskribatutako oinarrizko prozedurak arazo bat du: Internet osoko izen-ebazpenak erro-zerbitzarietatik abiatu behar dira. Nahiz eta zerbitzari horiek ehunka izan, Interneten segundoro sortzen diren milioika DNS galderek erro-zerbitzariren batetik pasatu behar badute, kolapsoa sortuko da. Hori ekiditeko *caching* teknika erabiltzen da, hau da, egindako galderen erantzuna, eta erantzun hori lortzeko informazioa, epe batean gordetzen da (laster ikusiko dugu *nork* gordetzen duen), eta, epe horren barruan informazio hori behar bada beste galdera bat ebazteko, cachetik hartuko dugu, DNS zerbitzariei galdetu gabe. Horretan datza DNS erregistroen TTL eremuaren erabilgarritasuna. Aldi baterako gordetze-lan hori toki desberdinetan egiten da. Hori ikusteko, berregin ditzagun izen bat ebazteko emandako urratsak, oinarrizko prozeduratik harago, benetan egiten dena deskribatuz:

1. Aplikazio batek, adibidez arakatzaile batek, izen bati dagokion IP helbidea jakin behar badu, ebazleari eskatuko dio ebazpena egiteko. Ebazleak bere cache propioa badu, horretan begiratuko du ea eskatutakoa dagoen (agian beste aplikazio batek orain dela gutxi eskatuta). Hor badago, bilaketa amaitzen da. Bestela, galdera gure konputagailutik aterako da bigarren urratsean.
2. Konputagailu bat sarean konektatzen dugunean, konfiguratu behar diren parametroetako bat DNS zerbitzaria da. Hori da ezagutzeko falta zaigun azken DNS zerbitzari mota: **bertako DNS zerbitzaria**. Haren lana DNS ebazleen proxiarena egitea da, hau da, datu-basea gordetzen duten zerbitzarien (jatorrizko zerbitzariak, TLD zerbitzariak, eta erro-zerbitzariak) eta ebazleen arteko artekaria da. Sare bateko konputagailu guztien ebazleek beren galderak birbidaltzen dizkiote beren sareko bertako zerbitzariari[20] beren cachean ez badute erantzunik. Bertako zerbitzariak bere cachean begiratuko du ea jasotako galderaren erantzuna duen. Horrela ez bada, 3. urratsari ekingo dio, eta galdera bertako saretik aterako da Interneterantz.
3. Bertako zerbitzariak bigarrenez bere cachean bilatuko du, baina orain DNS jatorrizko zerbitzari baten bila. Bilatutako izenari dagokion jatorrizko DNS zerbitzariaren helbidea badu, zuzenean galdetuko dio zerbitzari horri, erro-zerbitzaritik, TLD zerbitzaritik, eta beste inongo zerbitzaritik pasatu gabe. Erantzuna jasota, bere cachean gorde, eta ebazleari bidaliko dio. Ebazleak arakatzaileari emango dio. Bertako DNS zerbitzariaren cachean ez badago jatorrizko zerbitzariaren helbidea, ebazpenak jarraitzen du 4. urratsean.
4. Bertako zerbitzariak goraka egingo du izen-hierarkian, ea izena ebazteko dagokion zerbitzari baten IP helbidea duen bere cachean, hortik bilaketa hasteko. Adibidez, *www.kirola.nireherria.eu* baldin bada egindako galdera, eta 3. urratsean *kirola.nireherria.eu* domeinuari dagokion NS erregistroa ez duela egiaztatuta, *nireherria.eu* domeinuari dagokion NS erregistroa cachean duen begiratuko du. Horrela bada, erregistroak dioen zerbitzariari galdetuko dio, eta, haren erantzunetik abiatuta, bilatutako IP helbidea lortuko du. Bere cachean ez badu *nireherria.eu* domeinuko NS erregistroa, .eu TLD zerbitzariaren helbidea beharko du. Cachean badu, hortik hasiko du bilaketa. Ez badu, erro-zerbitzari batera jo beharko du, oinarrizko prozeduran adierazitako moduan, eta, hortik abiatuta, bilatutako IP helbidea lortuko du. Argi dago bertako zerbitzarien konfigurazioko parametro bat izango dela erro-zerbitzari batzuen IP helbidea. Datu hori beharko dute haritik tiraka hasteko bere cachean ez badute nondik hasi. DNS bertako zerbitzariarena egiteko gehien erabiltzen den softwarearen kasuan (BIND izeneko softwarea), 13 erro-zerbitzariren IP helbideak *named.cache* izeneko fitxategian gordetzen dira. Fitxategi hori www.internic.net/zones/named.cache helbidetik jaitsi daiteke.

20. Sare batean bertako DNS zerbitzari bat baino gehiago egon daitezke, eta sareko konputagailu guztiek ez dute zerbitzari berberarekin lan egin behar.

DNSren cache-hierarkia hurrengo irudian adierazten da. Cache horiei esker, Internet osoko funtzionamendua hobetzen da, sortzen diren DNS galderen zati txiki batek besterik ez baitu pasatu behar erro-zerbitzarietatik.



4.7. irudia. DNSren cache-hierarkia.

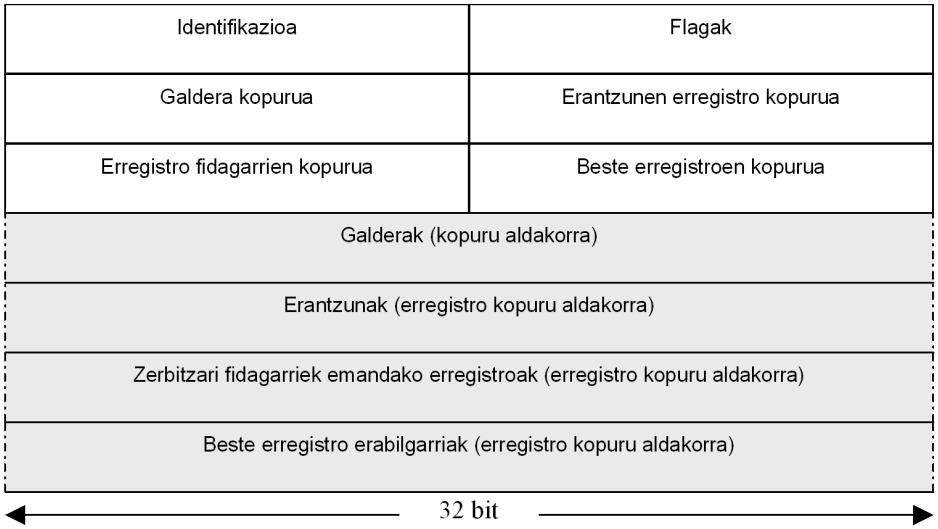
Zerbitzariek bi portaera izan ditzakete DNS galdera bat jasotzen dutenean, eta horren erantzun zuzena ez badute: bilaketa jarraitzeko hurrengo zerbitzariaren erreferentzia eman (*non-recursive question*), ala kargua hartu eta bilaketarekin segi erantzuna lortu arte, eta orduan erantzuna eman galdetu duenari (*recursive question*). Lehenengo portaera hartzen dute beti erro-zerbitzariek eta TLD zerbitzariek, eta bigarrena da bertako zerbitzariena. TLD zerbitzariaren eta erantzuna duen jatorrizko zerbitzariaren artean zerbitzari gehiago baldin badaude, zerbitzari horien kudeatzaileak erabakiko du (eta konfiguratuko du) nolakoa izango den zerbitzariaren portaera. Adibidez, *www.kirola.nireherria.eu* izena ebazteko, *nireherria.eu* domeinuko zerbitzariak bi aukera ditu bertako zerbitzari baten galdera jasotzen duenean:

- *kirola.nireherria.eu* barrutiari dagokion NS erregistroa itzuli, bertako zerbitzariak jarrai dezan bere bilaketa. Kasu honetan, erantzun iteratiboa edo ez-errekurtsiboa dela diogu. Zerbitzari lanpetuen aukera hau izaten da.
- *kirola.nireherria.eu* domeinuari dagokion jatorrizko zerbitzariari birbidali galdera, eta, erantzuna jasotzean, galdera egin zion bertako zerbitzariari birbidali. Kasu honetan, erantzuna errekurtsiboa dela diogu.

4.2.5. DNS protokoloa

DNS aplikazio-entitateen arteko harremanak oso errazak dira: batek erregistroak eskatzen ditu eta besteak erantzuten du. Eskaera egiten duena identifikatzeak ez du zentzu handirik, DNS zerbitzua zerbitzu irekia eta unibertsala baita. Egoera honetan, aplikazio-protokoloak konexiorik gabekoa izan behar du, eta definitu behar direnak eskaerak eta erantzunak besterik ez dira. Eskaera eta erantzun horiek dira DNS mezuak.

DNS mezu guztiek formatu bera dute, 4.7. irudikoa. Mezuen sintaxia mistoa da. Eremu batzuk karaktereka interpretatu behar dira, eta beste batzuk, aldiz, biteka. Ondoren adierazten ditugu eremu bakoitzaren sintaxia eta semantika.



4.8. irudia. DNS mezuen formatua.

Mezuek bi zati dituzte: aurrenekoa 12 byteko luzera finkoko goiburukoa da, eta bestea luzera aldakorreko 4 datu-eremuk osatzen dute. Ikus ditzagun banan-banan:

- Goiburukoa: 6 eremu daude, bakoitza 2 bytekoa.
  - Aurreneko 16 bitak eskaeraren identifikazioa dira. Identifikadore hau galderan eta beronen erantzunean grabatzen da; horrela, ebazleak (edota zerbitzariak) erantzunak dagozkien galderekin lotzen ditu (gogoan izan ez dagoela konexiorik aplikazio-mailan).
  - Hurrengo 16 bitak markak (flagak) dira. Batek mezua galdera (0) ala erantzuna (1) den adierazten du. Beste bat gaitzen da erantzuna jatorrizko zerbitzari batek ematen duenean. Beste bat galdera era errekurtsiboan ebazteko eskatzen denean gaitu behar du galdegileak (ebazlea edo DNS zerbitzaria delarik). Antzekoa da zerbitzariaren errekurtsibotasuna adierazten duena: beren erantzunetan zerbitzariek gaitzen dute, galdera errekurtsiboak onartzen dituztenean.
  - Goiburukoaren azkeneko 8 byteak 4 kopuru-eremu dira. Horietako bakoitza 2 byteko zenbaki bat da. Bere balioa datu-eremu batean dagoen ale kopurua da: galderen kopurua, erantzunen kopurua, jatorrizko erantzunen kopurua eta beste erregistroen kopurua.

- Datu-eremuak 4 dira, guztiak luzera aldakorrekoak.
  - Galdera-eremuak (*question section*) egindako galderei buruzko informazioa du. Bi atal daude galdera bakoitzeko: (i) izena (zer izeni buruzko erregistroak eskatzen diren), eta (ii) mota (zer motatako erregistroak eskatzen diren: A, MX, CNAME…).
  - Erantzun-eremuak (*answer section*) jasotako erantzunak gordetzen ditu, hainbat baliabide-erregistrok osatutako zerrenda moduan. Batzuetan zerrenda hutsa izango da, beste batzuetan RR bakarra aurkituko dugu zerrenda horretan, eta beste batzuetan hainbat RR egongo dira. Gogoratu izen bati mota bateko erregistro bat baino gehiago ezar dakizkiokeela. Adibidez, konputagailuak IP helbide bat baino gehiago baditu, konputagailuaren izenarekin A erregistro bat baino gehiago lotuko dira.
  - Jatorrizko zerbitzariak (*authoritative section*) izeneko eremua ere baliabide-erregistroek osatutako zerrenda bat da, baina kasu honetan NS erako erregistroak izango dira. Eremu hau erabiltzen dute zerbitzari iteratiboek galdegileari hurrengo urratsa adierazteko.
  - Azkeneko eremua (*additional section*) beste RR zerrenda bat da. Hemen agertzen diren erregistroak ez dira zuzenean galdetutakoaren erantzuna, baina erantzun horren osagarriak dira. Adibidez, erantzun iteratiboen *jatorrizko zerbitzari*en eremuan agertzen diren NS erregistroetako izenei dagozkien A motako erregistroak aurkituko ditugu hemen.

4.9. irudian bi DNS mezuren eskema dugu, Linux-eko `host` tresna erabiliz lortuta. Eskema horretan goian adierazitako eremuak agertzen dira, era irakurgarrian. Erabilitako `host` tresnak (eta dauden antzekoek) gizakiok DNS erabiltzeko balio du (gogoratu DNSren erabiltzaileak beste programak direla normalki). Irudiko lehenengo galderan —*www.joxemi.org* izenari dagokiona— ez da agertzen azkeneko datu-eremua (beste erregistro erabilgarriak). Bigarrenean, aldiz, bai.

Orain arte DNStik erregistroak nola atera ikusi dugu. Baina, nola sartzen dira? Nork eta nola eraikitzen du DNS den datu-base banatua? Hasiera batean, DNS zerbitzari bakoitzeko edukiak estatikoki konfiguratuta zeuden, hau da, zerbitzariaren kudeatzaileak sortu behar zuen DNS fitxategia eskuz. 1997ko apirilean proposamen bat egin zen lan hori dinamikoki egiteko, sarearen bidez. Proposamena RFC 2136 agirian dago. Hor UPDATE mezua gehitu zaio DNS protokoloari, erregistroak zerbitzari bati dinamikoki gehitu edo kentzeko.

```
$ host -v www.joxemi.org
Trying "www.joxemi.org."
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27684
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;www.joxemi.org.              IN      A

;; ANSWER SECTION:
www.joxemi.org.         73     IN      A       217.76.130.167

;; AUTHORITY SECTION:
joxemi.org.                          171973    IN             NS
dns2.servidoresdns.net.
joxemi.org.                          171973    IN             NS
dns1.servidoresdns.net.

$ host -v dns1.servidoresdns.net
Trying "dns1.servidoresdns.net."
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51726
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;dns1.servidoresdns.net.       IN      A

;; ANSWER SECTION:
dns1.servidoresdns.net. 60     IN      A       217.76.128.128

;; AUTHORITY SECTION:
servidoresdns.net.      60                IN                NS
atlante.servidoresdns.net.
servidoresdns.net.      60                IN                NS
prometeo.servidoresdns.net.

;; ADDITIONAL SECTION:
atlante.servidoresdns.net.  60  IN      A       217.76.128.4
prometeo.servidoresdns.net. 60  IN      A       217.76.129.4
```

4.9. irudia. Host komandoarekin lortutako bi DNS mezu.

4.3. WEB: INFORMAZIO-AMARAUNA

4.3.1. Web zerbitzua

Gizakien artean, oro har, informazioa bi eratan zabal daiteke:

- Bi lagunen artean. Kasu honetan bat igorlea da eta bestea hartzailea. Interneten, binakako komunikazio era hau gauzatzeko, FTP eta posta elektronikoa erabiltzen dira besteak beste. Mendeetan zehar beste sareen bidezko zerbitzu batzuk erabili izan dira helburu berarekin: posta-zerbitzuak aspalditik, eta telegrafoa, telefonoa eta faxa ez hain aspalditik.
- Igorle baten eta hartzaile askoren artean. Taldeko komunikazio era honek **difusioa** du izena. Antzina, herriko plazetan egiten zen batez ere, ahoz. Inprenta asmatu zenetik, berriz, liburu, aldizkari eta egunkarien bidez egin da. Gero, irratia eta gaur egun nagusia den telebista agertu ziren. Internet? Ikus dezagun.

Internet hasiera-hasieratik informazioa taldeetan banatzeko erabili izan da. Interneteko lehenengo erabiltzaile horiek unibertsitateko zientzialariak ziren, eta berehala hasi ziren Internet erabiltzen lanaren berri elkarri emateko. Horretarako, binakako komunikazioetan erabiltzen zituzten aplikazio berberak hartu zituzten: posta elektronikoa (eta bere horretan oinarritzen den berrien edo *news* sistema) eta FTP. Posta elektronikoa (edota berrien zerbitzua) batek taldeari informazioa helarazi nahi zionean erabiltzen zuten (*push*). Informazioa eskatzeko ere (*pop*) posta elektronikoa erabiltzen zuten hasieran, baina laster azaldu ziren arazoak: oso artikulu interesgarria idatzi zuenak ez zuen egun erdia pasatu nahi artikuluaren kopia eskatzen zuten kideen mezuei erantzuten. Hobe zuen artikulua sarearen erabiltzaile guztientzat atzigarri zegoen nonbait uztea, eta nahi zuenak, nahi zuenean, kopia bat eskura zezala. Horretarako zeuden FTP biltegiak; ez zegoen beste aplikazio bat asmatzeko beharrik…

Edo bai? FTP biltegiak ugaldu bezain laster, informazioa banatzeko sistema horren mugak agerian gelditu ziren. Honako hauek ditugu:

- FTP biltegi-sistemak ez du ezartzen inongo erlaziorik eskaintzen dituen agirien artean. Lan hori erabiltzaileak egin behar du. Adibidez, agiri batean beste agiri bati egindako erreferentzia agertzen bada, eta erabiltzaileak erreferentziatutako hori aztertu nahi badu, berak jakin beharko du non edo nola lortu.
- FTP biltegi bakoitzaren bidez lor daitekeen informazioa zerbitzari horrek gordetzen dituen agiriak besterik ez da. Aurreko adibideari jarraituz, demagun erabiltzaileak badakiela erreferentziatutako agiria atzigarri dagoela, baina beste FTP biltegi batean. Orduan, erabiltzaileak bertan behera utzi beharko du hasierako FTP zerbitzariarekiko lan-saioa, eta erreferentziatutako agiria duen beste FTP zerbitzariarekin lan-saio berri bat abiatu.

- FTP zerbitzariek gordetzen duten informazioa besterik ez dugu atzigarri FTPren bidez. Baina 80ko hamarkadan zehar, informazioa gordetzeko edota informazioa bilatzen laguntzeko beste sistema batzuk garatu ziren. *Gopher*, *archie*, *veronica* eta *wais* dira horiek, baina ez ditugu ikasiko, zaharkituta baitaude. Egoera hura ez zen batere erosoa erabiltzailearentzat, behar zuen informazioa non zegoen ez zekienean, bereziki. Informazioa aurkitu arte sistema desberdinak arakatu behar zituen, software desberdinak erabiliz. Eta, askotan, informazioa aurkitu eta gero, agiria jaisteko beste programa bat abiatu eta erabili behar zen (maiz, FTP).

Laburbilduz, difusioko informazio-atzipena FTP erabiliz ez zen batere arina, ezta osoa ere. Egoera konpontzeko, 90eko hamarkadako erdialdean weba agertu zen. Emandako zerbitzua informazio-difusioa da, honako ezaugarriak betez:

- Informazioa erlazionatuta dago, informazio-amarauna osatuz. Agiri desberdinen arteko erlazioa agirietan bertan ezarrita dago, eta erabiltzailea lan-saio berean agiri batetik bestera *ibil* daiteke. Horri *nabigatzea* esaten zaio. Agirien arteko erlazioa hipermedia kontzeptua erabiliz lortu da webean.
- Erabiltzaileak ohartu gabe lan-saio batean zerbitzari desberdinekin egin dezake lan. Erabiltzaileak ez du lan-saio berri bat ezarri behar web zerbitzari desberdin baten informazioa eskuratzen duen bakoitzean.
- Beste sistemetan gordetako informazioa ere araka daiteke. Web bezeroak FTP, berriak, gopher, archie eta abarren zerbitzariarekin egin baitezake lan.

Multimedia eta hipermedia

Informazioa adierazteko bide asko daude. Bide horiei, gero eta gehiagotan *media* deitzen zaie. Liburuetan, adibidez, testua erabiltzen da nagusiki, baina irudiak eta grafikoak ere agertzen dira, eta, beraz, liburuak **multimedia sistema** bat direla esan dezakegu. Dena dela, multimedia termino hori, oro har, testua, irudiak eta soinua elkartzen dituzten sistemei aplikatzen zaie. Adibidez, konputagailuak sistema multimedia dira.

Beste alde batetik, testu batean beste testuekiko loturak txerta daitezke. Esaterako, entziklopedia bateko termino baten azalpenean, termino horrekin erlazionatuta dauden beste terminoei egindako erreferentziak aurki ditzakegu. Erreferentzia, lotura edo **esteka** horiei jarraituz ateratzen da gai bati buruz entziklopediak duen informazioa. Testu batek horrelako estekak edo jauziak dituenean, **hipertestua** dela diogu. Gaur egungo agiri elektronikoetan oso ohikoak dira horrelako hipertestuko estekak. Adibidez, liburu honen bertsio elektronikoa erabiltzen ari bazara, atal honen hasierara zuzenean joateko hemen klikatu, eta hipertestuan jauzi bat egingo duzu esandako tokira joateko.

Esteken bidez testua ez ezik, beste edozein mediatako informazioa ere lotu dezakegu. Honela, irudi bat, abesti bat, grafiko bat, pelikula bat… txerta dezakegu testuarekin batera, betiere euskarriak onartzen badu. Adibidez, paperezko liburu batean, testuarekin batera, irudiak eta grafikoak erabil ditzakegu, baina ezin dugu liburuaren orrialdeetan film bat ikusi, edo grabazio bat entzun. Euskarri multimedia erabiliz, konputagailuak kasu, bai. Media guztietako informazioa esteken bidez lotzen dugunean, agiri **hipermedia** dugula diogu. Horrelako estekei, toki askotan, hiperestekak ere esaten diete.

4.3.2. Web osagaiak

Weba da gaur egun informazioa argitaratzeko eta eskuratzeko Interneten erabiltzen den aplikazioa. Bere oinarria hipermediaren kontzeptua da. Honako hauek dira aplikazioaren osagaiak:

- Informazioa duten fitxategiak egiteko eta erlazionatzeko formatu hipermedia bat. Hori **HTML** formatua da (HyperText Markup Language). Webean edozein formatutako agiriak aurkituko ditugu, baina informazio-sarearen oinarria HTML formatuko fitxategiak izango dira.
- Aplikazio-mailako entitateak. Bezero/zerbitzari ereduko aplikazio bat denez, aplikazio-mailako entitateak web bezeroak eta web zerbitzariak dira. Bakoitzaren zeregina honakoa da:
  - Web bezeroa. Euskaraz **arakatzailea** edota **nabigatzailea** deitutakoa. Arakatzaile bat pantailan HTML formatua adierazteko gai da. Gainera, beste formatu errazak ere ulertzen ditu. Gauza zailagoak egiteko (formatu konplexuak, *script*ak egikaritzeko, bideoak erreproduzitzeko…) aproposak diren programak egikariarazten ditu.
  - Web zerbitzariak, informazioa gordetzen du. TCPko 80 portua dute esleituta web zerbitzariek bezeroen eskaeren zain egoteko.

    Askotan, bezeroen eta zerbitzariaren artean bitartekari bat agertzen da. Bitartekariak **web proxi** izena jasotzen du. Bere zeregina cachearena egitea eta web trafikoa kontrolatzea da.
- Arakatzaileen, proxien, eta web zerbitzarien arteko komunikazioetarako protokoloa. Hori **HTTP** da (HyperText Transfer Protocol).

Normalki, aplikazio banatu baten helbideratze-sistema protokoloaren zati bat da, baina kasu honetan garrantzi propioa du. Webean, informazioa erreferentziatzeko helbideratze-sistema propioa garatu dute. Sistema horren helbideek izen berezia dute: **URL** (Universal Resource Locator[21]). URL batean honako hiru

21. Berez, gaur egun URI (Universal Resource Identifier) dute izena, eta URL deitzea zaharkituta dago. Hala ere, gehienok URL esaten jarraitzen dugunez, eutsiko diogu. URI eta URL kontu honetaz gehiago jakiteko, jo RFC 3986 agirira.

osagai hauek aurkituko ditugu: protokoloa://zerbitzaria/fitxategia. Web bezeroek onartzen duten edozein protokolo erabil dezakegu. Horrek ahalbidetzen du web bezero baten bidez web zerbitzaria ez den beste zerbitzari batek duen informazioa eskuratzea. Adibidez, ondoko URL honek FTP zerbitzari batek duen informazioa erreferentziatzen du: *ftp://ftp.rfc-editor.org/in-notes/rfc1738.txt*. Zerbitzariak identifikatzeko DNS izenak erabiltzen dira. URLren azken zatiak (*/in-notes/rfc1738.txt* aurreko adibidean) adierazten du zerbitzarian informazioak duen kokalekua.

4.3.3. HTTP protokoloa

HTTPk definitzen du nola eskatuko dion web bezeroak agiri bat web zerbitzariari, eta, orobat, nola bidaliko dion eskatutakoa zerbitzariak bezeroari. Erabiltzaileak agiri bat eskatzeko, URL bat ematen dio arakatzaileari. Hori HTML orri baten esteka batean klik baten bidez egin dezake, edo zuzenean URL hori tekleatuz. Arakatzaileak HTTP eskaera bidaliko dio URLak adierazten duen web zerbitzariari. Eta zerbitzariak HTTP erantzun baten bidez helaraziko dio eskatutakoa arakatzaileari.

HTTP protokoloak **TCP erabiltzen du** garraio-mailako protokolo gisa. Beraz, aurreko eskaera/erantzuna trukaketa gauzatu baino lehen, arakatzaileak eta zerbitzariak TCP konexio bat ezarri behar dute. Gogoratu garraio-mailan TCP erabiltzeak konfiantza ematen diola aplikazioari. Horrek inplikatzen du arakatzaileak bidaltzen dituen HTTP eskaerak oso-osorik helduko zaizkiola zerbitzariari; antzera, zerbitzariak bidaltzen dituen HTTP erantzunak iristea bermatuta dago. Aplikazioak ez du horretaz kezkatu behar.

HTTP egoera gabeko protokolo bat da. Horrek esan nahi du zerbitzariak ez duela arakatzaile batekin egindakoa gordetzen. Hau da, arakatzaileak tarte labur batean agiri bera bi aldiz eskatzen badu, zerbitzariak ez dio «dagoeneko emanda duzu» esanez erantzungo; ostera, zerbitzariak agiria birbidali egingo du, aurretik egindakoa zeharo ahaztua edukiko balu bezala. Izan ere, ahaztua du: egoera gabeko protokoloei memoriarik gabeko protokolo ere esaten zaie.

Arakatzailearen eta zerbitzarien arteko erlazioak ahalik eta arinena eta azkarrena izan behar du. Horregatik, HTTP eskaera/erantzuna moduko protokolo bat da eta, beraz, ez du aplikazio-mailan inongo konexiorik ezartzen. **Konexiorik gabeko** protokolo bat da, alegia, DNS bezalakoa eta laster ezagutuko dugun SMTP ez bezalakoa.

HTTP eskaeren eta erantzunen formatua

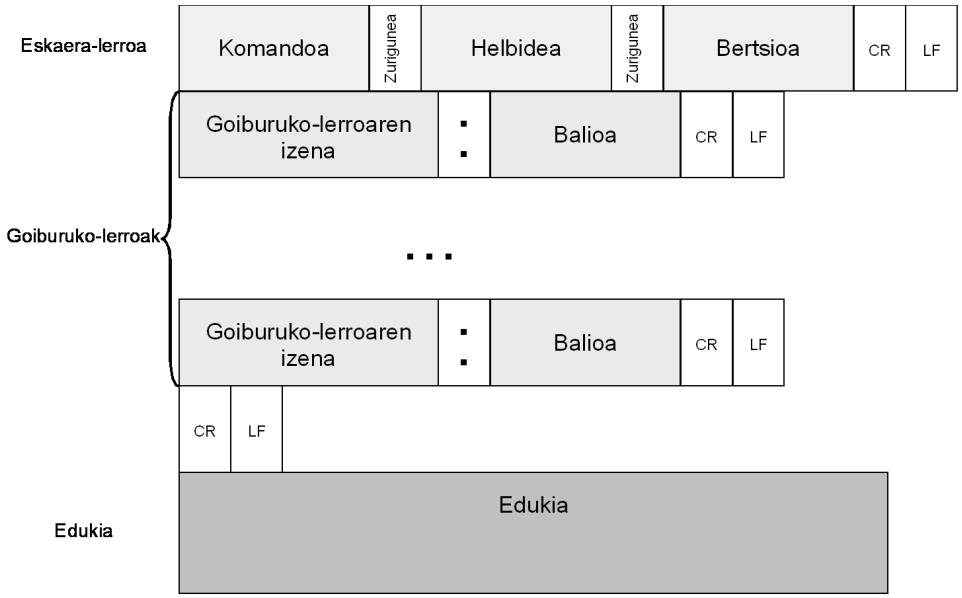
Bi motatako HTTP mezu daude, eskaerak eta erantzunak. ASCII formatua erabiltzen da HTTP mezuetan, eta lerroka antolatzen dira, beste aplikazio-protokoloetan egiten den era berean. HTTP eskaera eta erantzunak ASCII testua

besterik ez direnez, irakurgarriak dira gizakiontzat. Horrek protokoloa ulertzea eta ikastea errazten du .

4.10. irudian HTTP eskaeren egitura orokorra dugu. Hiru ataletan antolatzen da eskaera:

- Lehenengo lerroan, eskaera adierazten da. Lerro hau, gainera, hiru zatitan banatuta dago: komandoa*, helbidea eta HTTP bertsioaren eremua. Komandoak eskaera mota zehazten du. HTTP eskaera gehienek GET komandoa erabiltzen dute, hori baita agiriak eskatzeko erabiltzen dena. Beste balio arruntak POST edota HEAD dira. POST komandoa formularioak bidaltzeko erabiltzen da. Adibidez, erabiltzaileak bilatzaile bati hitz bat ematen dionean. POST mezuekin agiri bat eskatzen zaio zerbitzariari, baina agiri horren edukia erabiltzaileak formularioan emandako datuen araberakoa izango da. HEAD komandoa agiri baten ezaugarriak eskatzeko erabiltzen da; adibidez, HTML agiri bati egindako azken aldaketaren data jakiteko. GET eta POST komandoen kasuan ez bezala, HEAD baten erantzunean ez dago HEADren lehenengo lerroan adierazitako agiria. Horregatik erabiltzen da komando hau zerbitzarien funtzionamendua probaldietan egiaztatzeko, banda-zabalera gutxi kontsumitzearren.
- Bigarren zatia goiburuko-lerro batzuk dira, mezu elektronikoen RFC 822 goiburukoen oso antzekoak. Goiburuko hauek balio dute zerbitzariari eskaerari buruzko xehetasun batzuk emateko. Goiburuko-lerroaren kopurua aldakorra denez, eskaeraren bigarren zatiaren bukaera adierazteko lerro-bukaera marka (CR-LF karaktere bikotea) bi aldiz jartzen da.
- Hirugarren zatia eskaeraren edukia da. Eremu honetan arakatzaileak informazioa bidaltzen dio zerbitzariari. GET komandoetan eremu honek ez du zentzurik.

* HTTP protokoloaren komandoei objektu-programazioaren mundutik hartutako «metodo» izena ematen zaie protokoloa definitzen duen RFC 2616 agirian. Beste izen hori erabiltzeak ez du inolako ekarpenik egiten, nahasketa izan ezik. Horregatik, hemen, betiko «komando» hitzari eutsiko diogu, beste aplikazio-protokoloetan egiten den moduan.



4.10. irudia. HTTP eskaeren formatua. Zuriz dauden karaktereek balio finkoa dute. CR (Control Return) eta LF (Line Feed) karaktereak lerro-bereizleak dira.

Ondoan HTTP eskaera bat dugu:

```
GET /ibilbidea/orria.html HTTP/1.1
Connection: close
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language:ba
```

Eskaera hau GET bat da, ohikoena. Lau goiburuko ditu, non zerbitzariari esaten zaion TCP konexioa ixteko agiria bidali eta gero (`Connection: close`), zein diren erabilitako arakatzailearen modeloa eta bertsioa (`User-agent: Mozilla/4.0`), onartuko diren formatuak (`Accept: text/html, image/gif, image/jpeg`) eta testuetarako hizkuntza gustukoena (`Accept-language:ba`). Komandoa GET denez, ez dago edukirako eremurik.

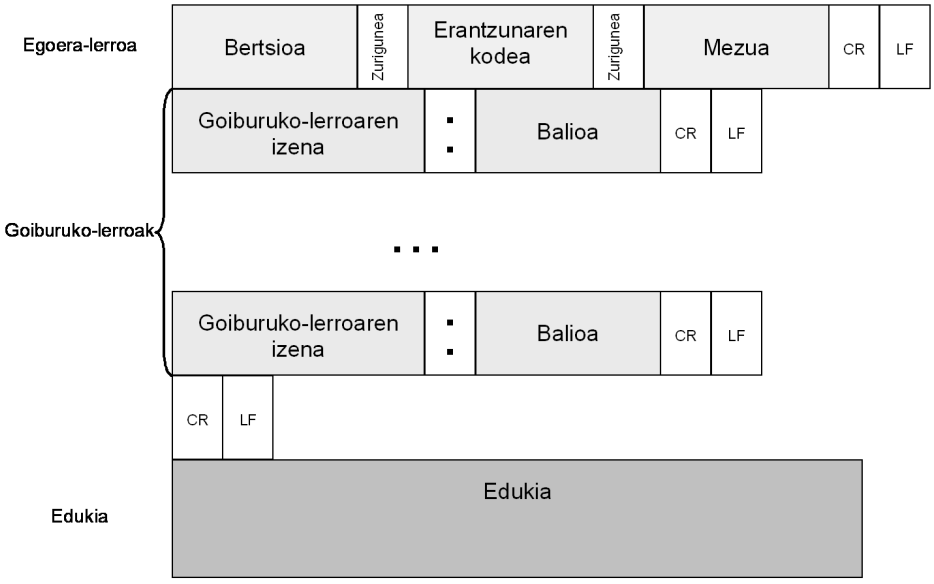
HTTP erantzunak ere hiru ataletan daude antolatuta, 4.11. irudian adierazten den moduan.

- Egoera-lerroak hiru eremu ditu: HTTP bertsioaren eremua, erantzunaren kodea eta kodeari uztartutako mezua. Azken biek eskaeraren emaitza adierazten dute. Ondoan kode eta uztartutako mezu batzuen esangura dugu:
  - `200 OK`. Dena ondo joan da. Eskatutako orria edukiaren aldean bidaltzen da.

- `301 Moved permanently`. Eskatutakoa tokiz aldatu da; URL berria `Location:` goiburukoan zehazten da. Arakatzaileak URL helbide berria automatikoki berreskuratuko du.
- `400 Bad Request`. Erroreren bat suertatu da eta zerbitzariak ezin izan du eskaera bete.
- `404 Not Found`. Eskatutako agiria ez dago zerbitzari honetan.
- `505 HTTP Version Not Supported`. Zerbitzariak ez du ulertzen eskaeran adierazitako HTTP protokoloaren bertsioa.

• Bigarren zatia goiburuko-lerroak dira, eskaeraren kasuan bezala. HTTPren zehaztapenak goiburuko-lerro asko eta asko definitzen ditu: zerbitzariak erabakiko du zeintzuk erabili, erantzunaren eta beste faktore askoren arabera.

• Hirugarren zatia erantzunaren mamia dugu: eskatutako agiria bidaltzen da hemen.



4.11. irudia. HTTP erantzunen formatua.

Behean HTTP erantzun tipiko bat dugu. Erantzun hau lehen emandako eskaeraren adibidearen erantzuna izan liteke.

```
HTTP/1.1 200 OK
Connection: close
Date: Fri, 27 Jun 2003 09:20:15 GMT
Server: Apache/1.3.27
Last-Modified: Mon, 23 Jun 2003 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html
```
Eskatutako agiriaren edukia ...

Azter ditzagun erantzun horren goiburukoak. Zerbitzariak `Connection: close` erabiltzen du arakatzaileari mezua bidali ondoren TCP konexioa itxiko duela esateko. `Date:` goiburuko-lerroak zerbitzariak HTTP erantzuna sortu eta bidali dueneko ordua eta data adierazten ditu. `Server:` goiburuko-lerroak web zerbitzariaren softwarea zein den adierazten du (Apache bat, kasu honetan); HTTP eskaeraren `User-agent:` goiburukoaren analogoa da. `Last-Modified:` goiburukoak agiria sortu edo azkeneko aldiz aldatu deneko ordua eta data jakinarazten ditu. `Content-Length:` goiburukoak erantzunaren edukiaren tamainaren berri ematen du. `Content-Type:` goiburukoak, azkenik, edukiaren formatua zein den adierazten du (HTML adibidean).

TCP konexioen kudeaketa

HTTP protokoloak TCP garraio-zerbitzua erabiltzen du, hau da, komandoak eta erantzunak TCP konexioen bidez bidaltzen dira. Baina HTTPren kasuan TCP konexioak kudeatzeko era bat baino gehiago daude, agiri bat osatzen duten fitxategi guztiak konexio beretik bidaltzen diren ala ez kontuan hartuta. Honako hauek dira:

- Konexio ez-iraunkorrak: arakatzaileak eta zerbitzariak agiriaren fitxategi bakoitzeko TCP konexio berri bat ezarriko dute (horrelakoa da 4.9. irudiko eskaeraren kasuan, `Connection: close` goiburukoak adierazten duenez). Konexio horiek une berean egon daitezke ezarrita.
- Konexio iraunkorrak: agiri bateko fitxategi guztiak, baita agiri desberdinetakoak ere, TCP konexio bakar baten bidez bidaltzen dira. Konexioa konfigura daitekeen aktibitaterik gabeko epe baten ostean amaituko da. Hau da HTTPren azkeneko bertsioen lan egiteko ohiko era. Konexio iraunkorrak ondoko bi aukera ditu:
  - Banan-banan: arakatzaileak ez du hurrengo fitxategiari dagokion GET komandoa bidaliko harik eta aurretik eskatutako fitxategi osoa jaso arte.
  - Jarraitua (*pipelining*): arakatzaileak GET komandoak isurtzen ditu oinarrizko HTML agirian URLak aurkitzen dituen heinean, jasotzen ari den fitxategi osoa bereganatu arte itxaron gabe. Hau da HTTPren azken bertsioen besterik ezeko lan modua.

Konexio ez-iraunkorrak ez badira une berean erabiltzen (paralelismorik ez), garestiagoak dira zenbait zentzutan:

- Eskaera bakoitzeko konexio berri bat ezarri eta mantendu behar da: TCP bufferrak esleitu eta aldagaiak gorde, bai zerbitzarian, bai arakatzailean. Horrek web zerbitzarian karga handia jar dezake, zerbitzariak aldi berean ehunka bezerori arreta jarri behar dienean.

- Gainera, fitxategi bakoitzeko bi RTT kontsumitzen dira (Round Trip Time, gogoratu), bata TCP konexioa ezartzeko eta bestea objektuaren eskaera eta erantzuna jasotzeko. Konexio iraunkorretan bi besterik ez dira kontsumitzen era jarraian, fitxategi kopurua zein den axola gabe.
- Azkenik, konexio ez-iraunkorretan, fitxategi guztiek beren *slow-start* (gogoratu 3. kapituluan ikusitakoa) fase propioa izango dute.

Beste alde batetik, agiriko fitxategiak ekartzeko TCP konexio bat baino gehiago aldi berean ezartzea, konexio iraunkor bakarra erabiltzea baino azkarragoa izaten da. Adibidez, HTML agiri nagusia jaisteko ezarritako konexioaz gain, arakatzaileak eska diezaioke zerbitzariari konexio berri bat ezartzeko HTML oinarri-agiri horretan topatzen duen esteka bakoitzeko (adibidez, irudiak kargatzeko). Zerbitzariek mugatzen dute arakatzaile bakar batekin aldi berean onartzen duten TCP konexio kopurua.

4.3.4. Web cachea eta proxiak

DNSn bezala, webean ere eskatutakoa azkarrago emateko eta sarean dabilen trafikoa arintzeko cache memoriak erabiltzen dira. Bi mailatako cacheak erabiltzen dira webean:

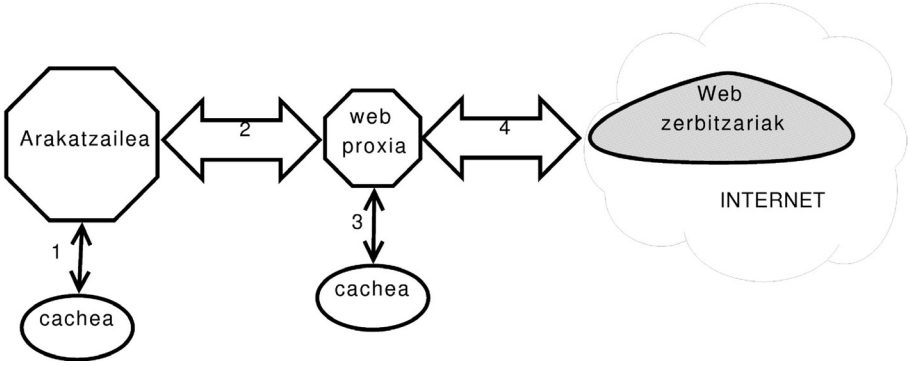
- Arakatzailean kokatutako cachea.

  Arakatzaileak bere memoria lokalean gordetzen ditu ikustatutako orriak. Memoria horren tamaina, erabilera eta iraunkortasuna taxutu daitezke. Badago arakatzailea sarerik gabe ere lan egiteko taxutzea, cachean gordeta dituen agiriak besterik ez erabiliz.

  Cacheak erabiltzearen arazo nagusia gordetako informazioaren gaurkotasunari eustea da. Gaurkotasun hori zaintzeko, HTTPk bere mekanismoak ditu. Hasteko, GET bati emandako erantzunean agiri bat jasotzen denean, goiburuko batek adieraziko du noizkoa den agiri hori. Hori da lehen ikusitako HTTP eskaera baten adibidean agertzen den `Last Modified` goiburukoa. Gero, arakatzaileak agiriaren gaurkotasuna egiaztatu behar duenean, data hori sartuko du, beste goiburuko batean, jatorrizko agiria duen zerbitzariari bidalitako GET eskaera batean. GET hau berezia da, zerbitzariak duen agiriaren bertsioa data berekoa baldin bada, erantzunean ez baitu agiririk bidaliko. Horregatik baldintzapeko GET dela esaten da. Egiaztapen hori noiz egin taxutu daitekeen arakatzailearen ezaugarri bat da: ezar dezakegu inoiz ez egitea (sarerik gabe lan egitea, alegia), automatikoki periodikoki egitea, arakatzailea abiatu dugunetik agiria eskatzen dugun lehenengo aldian soilik egitea, edo agiria eskatzen dugun bakoitzean.

- Web proxian kokatutako cachea.

  Web proxi batek sare edo makina talde batetik egindako HTTP eskaerak aztertu eta betetzen dituena da. Hau da, proxiaren atzean dauden arakatzaileen eskaerak ez dira zuzenean joaten eskatutako URLak adierazten duen web zerbitzarira, proxira baizik. Proxiak honako bi eginbehar hauek izaten ditu: alde batetik galbahearena egiten du, segurtasun-arrazoiengatik, eta bestetik, cachearena, eraginkortasun-arrazoiengatik. Eskatutako URLak segurtasun-arauak betetzen baditu, eta adierazitako agiria cachean gordeta badago, proxiak emango dio erantzuna arakatzaileari. Cachean ez badago, orduan proxiak eskatuko dio agiria zerbitzariari, jasotakoa eskaera egin zuen arakatzaileari helaraziko dio, eta bere cachean gordeko du agiria, hurrengo baterako.



4.12. irudia. Cache eta proxien erabilera webean.

Web proxiak hierarkikoki antola daitezke eraginkortasuna hobetzeko. Adibidez, hiru campus dituen unibertsitate batek campus bakoitzean web proxi bat izan dezake, eta horietako batek, gainera, unibertsitate osoko proxiarena egin. Campus batean egindako eskaerak campus horretako proxiari helduko zaizkio; horrek ez badu erantzuna bere cachean, unibertsitate osoko proxira joko du. Azken horrek erantzunik ez badu, orduan eskaera Interneten dagoen web zerbitzariari bidaliko dio. Beharren arabera (eta inplikatuen adosteko ahalmenaren arabera), proxien hierarkia handitu daiteke. Gure adibidearen kasuan, probetxugarria izan daiteke herrialde osoko unibertsitate guztietarako beste proxi bat abiatzea.

4.3.5. WEB APLIKAZIOAK

Aplikazio baten hiru zatiak (erabiltzailearekiko interfazea, datuekiko interfazea, eta datuen tratamendurako prozedurak) konputagailu desberdinetan kokatzea da gaur egungo aplikazioetan egin ohi dena. Ohikoa da datuak gordetzen dituzten fitxategiak eta datuak atzitzeko softwarea konputagailu batean kokatzea (zerbitzarian), eta erabiltzailearekiko interfazea beste konputagailu batean

(erabiltzaileen konputagailuetan). Datuak kontsumitzen edota sortzen dituzten prozedurak bi konputagailuetako batean, edo bietan, egoten dira. Kapitulu honen hasieran ikusi dugunez, izaera banatu honek behartzen du aplikazioko zati bakoitzari (aplikazioko entitateei) sarealdea izeneko laugarren atala gehitzera. Eta, berez, aplikazioa garatu behar duenak aplikazioko entitateen arteko komunikazioetarako protokoloa diseinatu eta inplementatu beharko du. Horrek asko zailtzen ditu aplikazioaren diseinua eta inplementazioa.

Gehitutako zailtasun hori arintzeko, edo guztiz ekiditeko, dagoeneko garatuta dagoen aplikazio banaturen baten softwarea erabiltzeko aukera laster mahaigaineratu zen. Hau da, beste aplikazio banatu batean definitutako eta garatutako komunikazioetarako osagaiak berrerabiltzea balego, gure aplikazioa era banatuan edo ez-banatuan garatzea oso antzekoa litzateke. Hala ere, aplikazio banatu baterako egindako sarealdea beste aplikazio batean erabiltzea ez da zuzenean bideragarria. Izan ere, berrerabili nahi den aplikazio banatuaren diseinua (bere protokoloa, bereziki) prest ez badago berrerabilera horretarako, jai dugu. Weba eta bere protokoloa, HTTP, prest daude beste aplikazio banatuen oinarria izateko. Webaren sarealdea erabiltzen duten aplikazio banatu horiei, oro har, web aplikazio esaten zaie.

Web aplikazioen osagaiak

Honako hauek dira:

- Arakatzailea.

  Arakatzaileak aplikazioko bezeroaren zeregina bereganatzen du. Hau da, web aplikazio bat garatzen dugunean, ez dugu bezerorik sortu behar, webarena erabiliko baitugu. Horretan datza web aplikazioen erakargarritasunetako bat, ahalbidetzen baitu sare-aplikazio berriak masiboki eskuragarri jartzea, erabiltzaileen konputagailuetan inongo software berririk instalatu beharrik gabe. Arakatzaileen lanak honako hauek dira:

  – Erabiltzaileekiko interfazea. Interfaze grafikoa da, HTML kodean oinarrituta. Erabiltzaileak egindako eskaeretarako parametroak formularioen bidez biltzen ditu arakatzaileak.

  – Zerbitzariarekiko komunikazioa. Horretarako HTTP protokoloa erabiltzen du arakatzaileak. Eskaeren parametroak zerbitzariari helarazteko POST komandoa erabiltzen da.

  Aurreko bi hauetaz gain arakatzaileak beste lanik egiten ez badu, bezero arina (*thin client*) dela esaten da. Hala ere, aplikazio batzuetan komenigarria eta bideragarria da arakatzaileak ahal den heinean zerbitzariaren lana arintzea. Adibidez, kasu askotan arakatzaileak eskatzen dizkio zerbitzariari erantzuna emateko behar diren datuak, baina arakatzaileak berak prozesatuko ditu datu horiek. Aukera honetan oso erabilia da AJAX (Asynchronous JavaScript and XML) teknologia multzoa.

- Web zerbitzaria.

  Aplikazioko zerbitzariaren lanen artean, bezeroarekiko komunikazioez arduratzen da web zerbitzaria. Hau da, web zerbitzariak jasoko ditu bezeroak egindako eskaerak, eta helaraziko dizkio erantzunak, baina ez ditu agindutako eragiketak beteko; horretaz prozeduretarako softwarea arduratuko da. Bezeroarekiko komunikazio horretarako HTTP erabiltzen du zerbitzariak. Web zerbitzaria aplikazioaren zerbitzariaren beste zatia den prozeduretarako softwarearekin ere komunikatuko da, bezeroak eskatutakoa emateko, eta eskaera horien emaitza jasotzeko. Konputagailu berean dauden zerbitzariaren bi software zatiak (web zerbitzaria eta prozeduretarako softwarea) komunikatzeko hainbat aukera daude, prozeduretarako softwarearen araberakoak. Zaharrena CGI interfazea da, baina horren mugak gainditzen dituzten beste aukerak erabiltzen dira gehienbat gaur egun.

- Prozeduretarako softwarea.

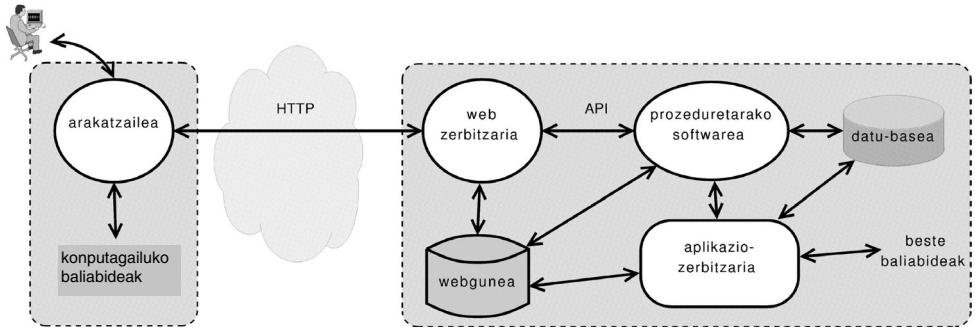
  Hau da aplikazioko zerbitzariaren bigarren zatia, eta web aplikazioetan diseinatu eta garatu behar den software bakarra[22] (zerbitzaria eta arakatzailea estandarrak dira). Berriki aipatu dugunez, software hau hainbat eratakoa izan daiteke: CGI bidez atzitutako *script* multzo bat, Java makina birtual baten bidez egikaritutako programak (*servlet*-ak), HTML kodean txertatutako *script*ak (*server side scripting*) edo errutina baterako deiak (ASP.NET eta *Java Server Pages*), eta, web aplikazio handietarako gehien erabiltzen dena, aplikazio-zerbitzari batekin egikaritzen diren *script*ak. Aplikazio-zerbitzariak (*application servers*) beste aplikazioek erabiltzen dituzten hainbat programa, errutina edo *script*ak biltzen dituen softwarea dira. Gainera, baliabide horiek atzitzeko APIa ere (Application Programming Interface) definitzen du aplikazio-zerbitzariak. Aplikazio-zerbitzariaren osagaiek asko errazten dute beste software- eta hardware-baliabideak erabiltzea. Adibidez, askotan, datu-base bat atzitzeko erabiltzen dute web aplikazioek aplikazio-zerbitzaria.

- Datuak.

  Beste edozein aplikaziok bezala, web aplikazioek datuak eskuratzen eta sortzen dituzte. Web aplikaziorako sortutako softwareak egin dezake datu horietarako atzipen-lana, baina gehienetan aplikazio-zerbitzari bat edo software-liburutegi bat erabiliko da horretarako. Oso ezagunak eta erabiliak diren bi aplikazio-zerbitzariak JavaEE eta .NET dira. Gehienetan, web aplikazioek erabiltzen dituzten datuak datu-base batean bilduta dauden datuak edota web zerbitzariak kudeatutako webgune baten orriak izaten dira.

22. Izan ere, oro har, «web aplikazio» terminoa aplikazioaren zati hau besterik ez izendatzeko erabiltzen da.

Web aplikazioen adibideak web bidezko posta-sistemak (ikusi hurrengo atala), web dendak, edo enkanteguneak dira. Ondoko irudian dugu web aplikazioen osagaien elkarren arteko harremana.



4.13. irudia. Web aplikazioen ohiko egitura. Gehienetan, prozedurei esaten zaie «web aplikazio», hori besterik ez baitu sortu behar aplikazio berria sortzen duenak. Aplikazio-zerbitzaria erabiltzen bada, ez da ohikoa prozeduretarako softwareak datuak zuzenean atzitzea.

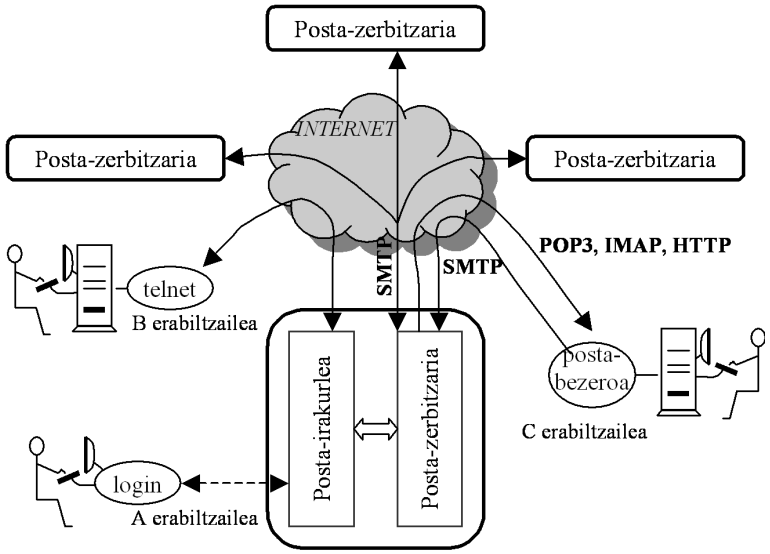
4.4. POSTA ELEKTRONIKOA

Bada bi hamarkadatik gora posta elektronikoa dugula *(email* ingelesez). Posta elektronikoaren lehen sistemak FTPren antzeko fitxategien transferentziarako sistemak besterik ez ziren, non mezu bakoitzaren (hau da, fitxategi bakoitzaren) lehen lerroan hartzailearen helbidea jartzen zen. Denbora aurrera joan ahala, planteamendu horren mugak nabariak egin ziren.

Esperientzia hartu ahala, posta elektronikoko sistema landuagoak proposatu ziren. 1982an, ARPANETen posta elektronikoko proposamenak RFC 821 gisa (transmisio-protokoloa) eta RFC 822 gisa (mezu-formatua) plazaratu ziren. Geroztik, Interneteko *de facto* estandar bilakatu dira. 2001eko apirilean, RFC 2821/2822 tandemak RFC 821/822 estandarra ordezkatzeko proposamena egin zen, baina testu hau idaztean, estandar ofiziala RFC 821/822 da oraindik. Internetena ez ezik, beste saio batzuk ere egin dira posta elektronikoa estandarizatzeko, baina gaur egun Interneteko RFC 821/822 da jaun eta jabe sistema eta sare guztietan.

4.4.1. Aplikazioaren osagaiak

4.14. irudiak Interneteko posta-sistemaren eskema azaltzen du. Honako osagai hauek ditugu: aplikazioko entitateak (posta-bezeroak eta zerbitzariak), aplikazio-mailako protokoloak eta bidalitako informazioaren formatuaren definizioa. Ondoren ditugu bakoitzaren ezaugarri nagusiak. Aplikazio-mailako protokoloek eta mezuen formatuak azterketa sakonagoa merezi dute; geroko ataletan egingo dugu azterketa hori.



4.14. irudia. Posta elektronikoko eragileen arteko harremanak.

- Bezeroek erabiltzaileak mezuak irakurtzeko, erantzuteko, osatzeko, igortzeko eta gordetzeko balio dute. Liburu batzuetan posta-bezeroei erabiltzaile-agenteak esaten zaie (OSI mundutik hartutako terminoa da), eta kalean, askotan, posta-irakurleak ere deitzen dituzte. Hasieran, eta urte askotan zehar, posta-bezeroen erabiltzailearekiko interfazea testu modukoa izan ohi zen, eta posta-zerbitzari baten konputagailuan bertan egikaritzen ziren; horregatik, hain zuzen ere, «posta-irakurle» deitzen zieten, eta ez «bezero», zerbitzariarekiko komunikazioak ez zirelako sarean zehar egiten (prozesuen arteko komunikazioetarako sistema eragilearen baliabideak erabiltzen ziren). Testuinguru horretan, posta-erabiltzaileek telneten bidez atzitzen zuten posta-zerbitzua. Baina 1990aren bukaerarako posta-erabiltzaile gehienek konputagailu pertsonaletan egiten zuten lan, eta GUI (Graphical User Interface) erako posta-bezeroak nagusitu ziren, zeinak erabiltzailearen PCan bertan egikaritzen diren, eta zerbitzariarekiko harremana sarean bidez egiten duten. Horien adibideak dira Thunderbird, Eudora, edo Microsoft-en Outlook Express, baina web posta sistema bat erabiltzen bada, arakatzaileak berak egiten digu posta-bezeroarena. Hala eta guztiz ere, badaude oraindik *mail*, *pine* eta *elm* testu moduko bezeroak erabiltzen dituztenak.
- Posta-zerbitzariek sistemaren ardatza osatzen dute. Erabiltzaile guztiak posta-zerbitzari bati lotuta daude, non erabiltzaile bakoitzak bere postontzia duen (*mailbox*). Postontzia zerbitzariak kudeatzen duen datu-egitura bat besterik ez da. Neurri handi batean, postontzi bat katalogo bat da, non

fitxategiak postako mezuak diren. Mezuek bezeroan hasten dute beren bidaia, hortik igorlearen posta-zerbitzarira joaten dira, ondoren, askotan, hartzailearen posta-zerbitzarira helduko dira eta, bukatzeko, hartzailearen bezeroak bere postontzitik hartuko du mezua. Beste batzuetan, aldiz, mezua ez da zuzenean igaroko igorlearen zerbitzaritik hartzailearen zerbitzarira, baizik eta tartean dauden beste zerbitzari batzuetatik ere. Adibidez, gerta daiteke patxi@ehu.es erabiltzaileari igorritako mezua, Euskal Herriko Unibertsitateko posta-zerbitzari orokor bati ematea igorlearen zerbitzariak, eta, gero, EHUko posta-zerbitzari orokor horrek «patxi» izeneko erabiltzailearen buzoia duen zerbitzariari birbidaltzea mezua (adibidez, Ibaetako campuseko posta-zerbitzari bati). Beraz, DNSn gertatzen den bezala, zerbitzarien artean ere komunikazio zuzena dago, eta ez bakarrik bezeroen eta zerbitzarien artean. Posta-zerbitzarien arteko komunikazio horiek ere bezero/zerbitzari ereduari jarraitzen diote: igorlearen posta-zerbitzariak bezeroarena egiten du, eta hartzailearen posta-zerbitzariak zerbitzariarena.

- Posta elektronikoak badu bere bitxikeria: ez du aplikazio-mailako protokolo bakarra erabiltzen, batzuk baizik. Protokolo bakarra erabiltzen da mezuak igortzeko, SMTP izenekoa, eta beste batzuk mezuak hartzeko (POP3, IMAP, HTTP). SMTP bezeroaren eta zerbitzariaren arteko komunikazioetan erabiltzen da, baita bi zerbitzariren artekoetan ere. Besteak bezeroaren eta zerbitzariaren artean bakarrik erabiltzen dira.
- RFC 822 agiriak definitzen du posta elektronikoko mezuak nolakoak izan behar diren. Posta-helbideen egitura ere definitzen da agiri horretan.

Irudian hiru motatako posta-erabiltzaile ageri dira:

- A izeneko erabiltzaileak posta-zerbitzaria duen konputagailu berean du bere posta-bezeroa. Irudian, posta-irakurle esaten diogu bezero horri, posta-zerbitzariarekiko harremana ez delako sarearen bidez egiten. Gainera, A erabiltzaile honek zuzenean atzitzen du irakurlea eta zerbitzaria egikaritzen dituen konputagailua, eta ez sarearen bidez. Mota honetako erabiltzailea gutxitan aurkituko dugu gaur egungo sistemetan.
- B erabiltzailea A bezalakoa da, baina ez du irakurlea eta zerbitzaria egikaritzen dituen konputagailuarekin atzipen zuzenik. Sarearen bidez urruneko lan-saio bat ireki behar du konputagailu horretan, telnet aplikazioa erabiliz, eta orduan posta-irakurlea egikaritu, A-k egiten duen bezala. Modu horretan lan eginez gero, B-k posta bidez jasotako fitxategi bat bere konputagailu lokalera eraman nahi badu, beste aldetik FTP saio bat ireki behar du, fitxategi hori transferitzeko. Gainera, atzipena telneten bidez egiten denez, ezin dira interfaze grafikoak erabili eta, ondorioz, B-k gaur egun posta elektronikoz hain maiz bidaltzen diren multimedia fitxategiak ezingo ditu ikusi. Horretarako bere konputagailu lokalera ekarri beharko ditu multimedia

fitxategiak eta hor bistaratu, dagokion bisorea erabiliz. A motako erabiltzaileak bezala, urriak dira B motakoak.

- C erabiltzailea da arruntena gaur egun. Honek bere konputagailuan egikaritzen du posta-bezeroa, eta bezero hori eta posta-zerbitzaria sarearen bidez komunikatzen dira. Oro har, posta-zerbitzariaren konputagailua erabiltzailearen erakundearena edo enpresarena izaten da edo, etxeko erabiltzaileen kasuan, Internet hornitzailearena.

Posta-helbideak eta MX erregistroak

Interneteko posta-helbideek honako egitura hau dute:

erabiltzailea@posta-zerbitzaria

Lehenengo zatiak erabiltzaile baten postontzia identifikatzen du. Bigarrena DNS izen bat da, erabiltzaile horren postontzia duen posta-zerbitzariarena. Izen horri dagokion MX erregistroan gordetzen da konputagailu horren «benetako» izena. Mezu baten igorlearen posta-zerbitzariak DNSri galdetu beharko dio ea zein den helburuko helbidearen bigarren zatiari dagokion MX erregistroa, hartzailearen posta-zerbitzaria nor den jakiteko, eta berarekin harremanetan jartzeko mezua emateko.

Gerta daiteke helbidearen aurreko zatiak erabiltzaile talde bat identifikatzea, eta ez erabiltzaile bakar bat. Kasu horretan helbidea posta-zerrenda batena da. Zerrenden mesedea da mezu berbera bidalketa bakarra eginez pertsona talde bati bidal dakiokeela. Hartzailearen posta-zerbitzariak ugalduko du mezua, zerrendakide bakoitzari bere kopia igorriz. Adibidez, Linux talde batek *posta.linux.eh*-n *linuxzaleok* izeneko posta-zerrenda bat badauka instalatuta, *linuxzaleok@posta.linux.eh*-ra bidalitako mezu oro taldearen zerbitzarira bideratuko da, eta zerbitzariak posta-zerrendako kide guztientzako mezu indibidualak sortuko ditu, kideak munduko edozein bazterretan daudela ere. Zerrendakide guztiek beren postontzia zerrenda kudeatzen duen posta-zerbitzari berean baldin badute, sarean ez da bestelako mezurik sartu behar.

4.4.2. Aplikazioaren protokoloak

SMTP

Simple Mail Transfer Protocol (SMTP) Interneteko posta elektronikorako lehen aplikazio-protokoloa da. 4.14 irudian ikus daitekeenez, bi zerbitzarien arteko komunikazioetarako erabiltzen da protokolo hau. Hala eta guztiz ere, bezero/zerbitzari moduko protokolo bat da, non igorlearen posta-zerbitzariak bezeroarena egiten du (eskaerak bidaliz), eta hartzailearen posta-zerbitzariak benetako zerbitzariarena egitendu (erantzunak emanez). Datu-transferentzia fidagarria izan dadin, TCP zerbitzua erabiltzen da (25. portua).

Aurreko atalean adierazi denez, mezuen bidaia bi eratakoa izan daiteke:

- Igorlearen eta hartzailearen zerbitzarien arteko komunikazioa zuzena da, ez dago inongo bitartekaririk bi zerbitzarien artean. Nahiz eta bi zerbitzariak munduko bi muturretan egon, igorleak hartzailearen IP helbidea eskatuko dio DNSri (dagozkion MX eta A erregistroak), eta berarekin zuzenean ezarriko du TCP konexio bat mezuak bidaltzeko. Hartzailearen zerbitzaria ez badago mezuak jasotzeko prest, igorleak ez dizkio mezuak beste inongo tarteko zerbitzariri ematen; geroago saiatuko da berriro bidaltzen. Saiakera batzuen ostean bidaltzea lortu ez badu, erabiltzailea ohartaraziko du, errore-mezu baten bidez.
- Bi zerbitzarien artean beste posta-zerbitzari batzuk daude. Orduan mezua ez da zuzenean hartzailearen posta-zerbitzarira joaten, tarteko zerbitzari batera baizik (*relay mail system*). Igorleak DNStik lortutako informazioak bitartekariarengana eramango du mezua, hori hartzailearen zerbitzaria balitz bezala. Izan ere, igorlearen aurrean, bitartekaria da mezuaren helburuko makina. Bitartekariak jakingo du zein den benetan hartzailearen zerbitzaria, edo, behintzat, zein den helburura heltzeko hurrengo bitartekaria. Zeharkako mekanismo honen erabilgarritasunaren adibide bat hau da: Interneteko etengabeko konexioa ez duten posta-zerbitzariak (telefono bidezko konexioa besterik ez duten tokietan kokatzen diren zerbitzariek, adibidez) aldiro konektatzen dira bitartekari batekin posta bidaltzeko eta hartzeko.

SMTP konexioaren bidezko protokolo bat da. Mezuak bidaltzen hasi baino lehenago, lan-saioa (edo aplikazio-mailako konexioa) ezarri behar da bi SMTP zerbitzarien artean. Gero mezuak bidaltzen dira, banan-banan eta, mezu gehiagorik ez dagoenean, lan-saioa amaitzen dute zerbitzariek. Mezu bakoitza bidaltzeko ere, hiru urrats egiten dira: igorlea identifikatzea, hartzailea identifikatzea, eta mezua bera bidaltzea.

Urrats horiek guztiak 4.15. irudian agertzen dira, bakoitzean erabilitako protokoloaren komandoekin batera. Kontuan izan mezuak beti joaten direla TCP konexioa ezarri duen posta-zerbitzaritik (bezeroarena egiten duena) hartzailearen posta-zerbitzarira (zerbitzariarena egiten duena). Horregatik esaten da SMTP *sartu* erako protokoloa dela (*push protocol)*.

Sintaxiari dagokionez, SMTP protokoloa karaktere moduko protokoloa da, 7 biteko ASCII kodean oinarritua. Horrek irakurgarriak egiten dizkigu gizakioi SMTP «elkarrizketak», esaterako, 4.15. irudian agertzen dena.

Komandoa	Parametroak	Esanahia
HELO	Igorlearen posta-zerbitzariaren izena	Lan-saioa irekitzeko eskaera
MAIL FROM	Igorri nahi den mezu baten igorlearen identifikazioa	Mezu bat bidaltzeko lehen urratsa
RCPT TO	Igorri nahi den mezu baten hartzailearen identifikazioa	Mezu bat bidaltzeko bigarren urratsa
DATA	Parametrorik ez	Mezu bat bidaltzeko hirugarren urratsa
QUIT	Parametrorik ez	Saioa amaitzeko agindua

4.2. taula. SMTP komandorik erabilienak.

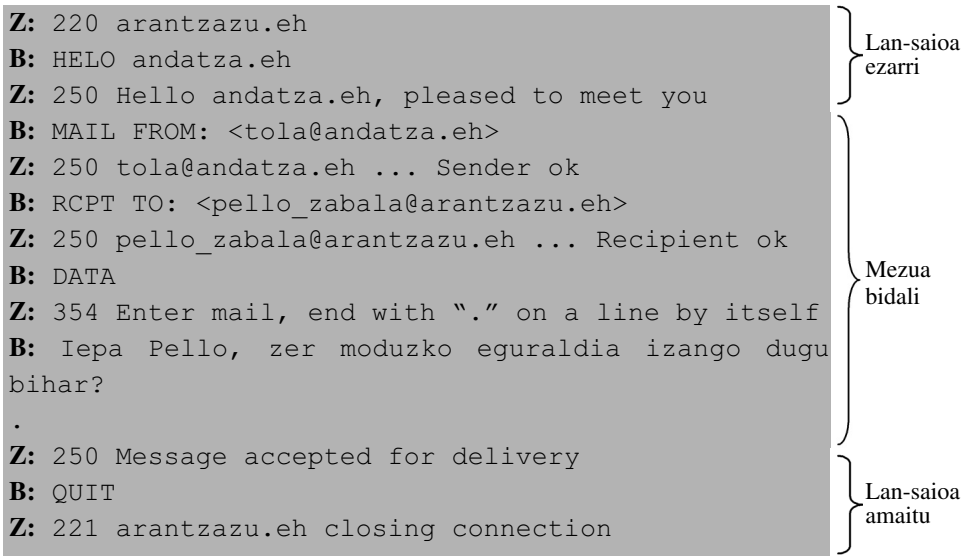
Komando bakoitzean bi eremu besterik ez daude: komandoaren identifikazioa eta komandoaren argumentuak, baldin badaude. Identifikazioa hitz batek edo bik egiten dute, 4 ASCII karaktereko hitzek gehienez ere; argumentuak ere ASCII kodean agertzen dira. Komandorik erabilienak 4.2. taulan agertzen direnak dira.

Erantzunaren kodea	Esaldia	Esanahia
220	<posta-zerbitzariaren izena> Service ready	Lan-saioa ezartzeko lehenengo urratsa
421	<posta-zerbitzariaren izena> Service not available	Lan-saioa ezartzeari uko
250	OK	Ados bidalitakoarekin
251	User not local; will forward to <posta-zerbitzaria>	Hartzaileak ez du postontzirik zerbitzari honetan, baina hau bitartekaria da helburura heltzeko
354	Enter mail, end with “.” on a line by itself	Mezua bidaltzeko baimena
221	<posta-zerbitzariaren izena> closing connection	Lan-saioa amaitzeko onarpena. Hartzailearen posta-zerbitzariak TCP konexioa amaituko du

4.3. taula. SMTP erantzun batzuk. Esaldia aplikazioaren inplementatzailearen araberakoa da.

Zerbitzariak komando bakoitzari erantzun bat ematen dio. Erantzunen sintaxia honako hau da: hiru digituko ASCII zenbakiak dira, gehi hautazkoa den ASCII testu bat, erantzuna adierazten duena. 4.3. taulan daude erantzun batzuk, beren mezu posibleekin. Taulan agertzen diren lehenengo erantzunak (220 eta 421 kodekoak) bereziak dira: hartzailearen zerbitzariak bidaltzen du bata edo bestea 25 portuan TCP konexioa ezarri bezain laster, igorleak inongo komandorik bidali baino lehen.

4.15. irudian SMTP lan-saio bat dugu. Hasieran I letra duten lerroak igorlearen posta-zerbitzariak (`anboto.eh` izeneko makinak) bidalitako komandoak dira. H letra dutenak hartzailearen erantzunak dira (arantzazu.eh izeneko makinarenak). Irudiko saioan mezu bakarra bidaltzen da ("`Iepa Pello, zer moduzko eguraldia izango dugu bihar?`"). Mezuaren bukaera adierazteko, puntu hutsa duen lerroa bidali behar da.



4.15. irudia. SMTP lan-saio bat. «Z:»-z hasten diren lerroak zerbitzariari dagozkio, eta «B:»-z hasten direnak bezeroari.

POP3

Gogoan izan SMTP sartu moduko protokoloa dela. Beraz, ez du inongo komandorik posta-zerbitzari bati erabiltzaile baten postontzian dauden mezuak eskatzeko. Eta hori da, hain zuzen ere, bere posta-zerbitzariarena ez den beste konputagailu batean egikaritzen diren posta-bezeroen beharra (4.14. irudian, C erabiltzailearena). SMTP sortu zen garaian ez zegoen beste inongo beharrik, posta-erabiltzaile guztiak 4.14. irudiko A eta B erabiltzaileak bezalakoak zirelako. Testuinguru horretan, posta elektronikoari zegozkion sare-komunikazio guztiak posta-zerbitzarien artean egiten ziren, eta beti mezuak bidaltzeko (*push*) eta ez jasotzeko. Baina C erabiltzailearen posta-bezeroak bere mezuak posta-zerbitzariari *eskatu* behar dizkio, sarearen bidez. Gutxienez erabiltzailea identifikatu beharko du, bereak diren mezuak bakarrik jasotzeko, eta ez beste erabiltzaileen postontzietakoak. Eta SMTP protokoloak ez du hori egiteko komandorik. Horregatik sortu ziren beste protokoloak, atera modukoak (*pop* protokoloak, ingelesez), posta-bezeroek posta-zerbitzaritik mezuak irakurtzeko erabiltzen dituztenak. Bidaltzeko, aldiz, betiko SMTP erabiltzen dute.

Gaur egun POP3 da posta irakurtzeko gehien erabiltzen den protokoloa (RFC 1939an deskribatuta). Oso protokolo xumea da, SMTPren antzekoa. Neurri handi batean, SMTPren zabalpena besterik ez da. Bere aita bezala, POP3 ASCII kodean oinarritutako karaktere moduko protokoloa da, eta konexio bidezkoa. Komandoak SMTPrenak bezalakoak dira, eta erantzunak are sinpleagoak: bi erantzun posible besterik ez daude. Bata `+OK` da, komandoari dagokionez dena ondo dagoela adierazteko. Erantzunarekin batera ohiko esaldi adierazgarria joan daiteke, edota datuak (mezuak). Beste erantzun posiblea `-ERR` da, komandoari dagokionez zer edo zer gaizki dagoela erantzuten duena. Ez dago errore-koderik; esaldi adierazgarria badago, horrek argitu dezake arazoa.

Konexio bidezko beste aplikazio-protokoloetan bezala, POP3n ondoko hiru urrats hauek egiten dira bezeroaren eta zerbitzariaren arteko lan-saio batean:

- Konexio-ezarpena: POP3n **kautotze-fasea** da hau (*authorization phase*). Bezeroak 110 portuan TCP konexio bat ezarri eta gero, erabiltzaileak, lanean hasi baino lehen, bere izena eta pasahitza eman behar dizkio zerbitzariari, zerbitzari horren erabiltzailea dela bermatzeko.
- Transakzioak: bezeroak ekartzen ditu erabiltzailearen postontzian dauden posta-mezuak.
- Konexio-amaiera: SMTPkoa bezalakoa da: bezeroak lan-saioaren amaiera adierazten dio zerbitzariari eta, erantzuna eman eta gero, honek amaitzen du erabilitako TCP konexioa.

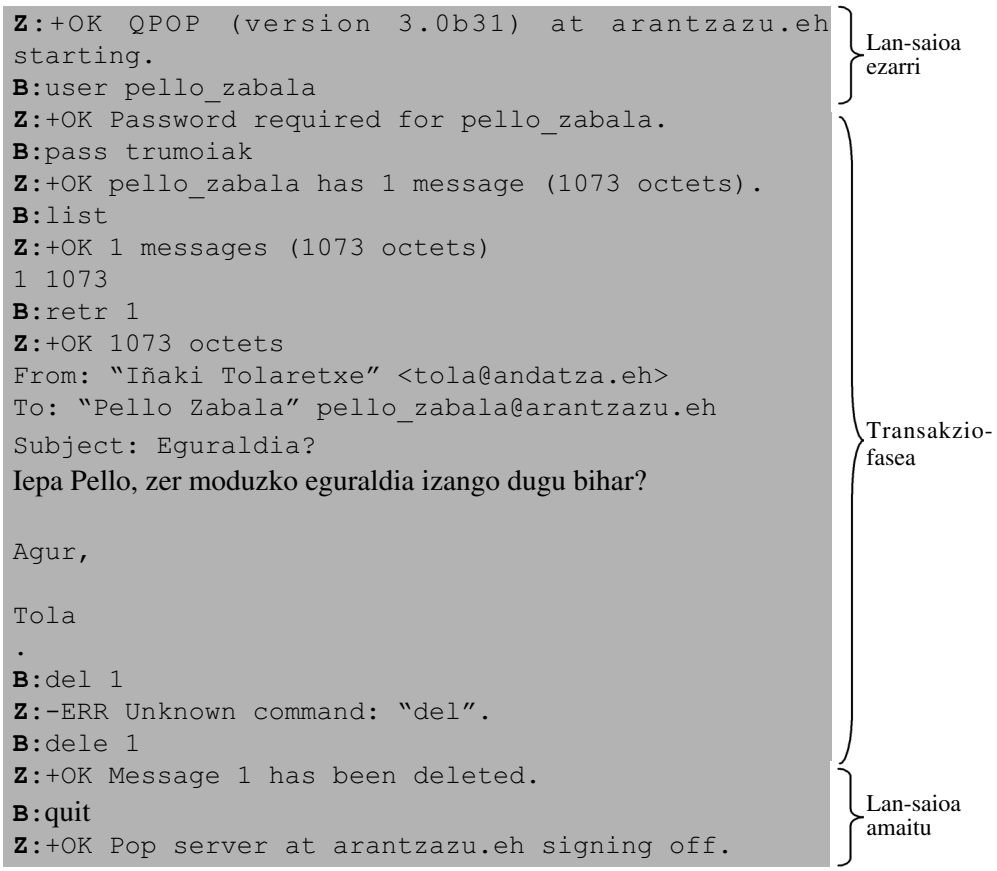
Ondoko taula honetan POP3 komandorik esanguratsuenak agertzen dira.

Komandoa	Parametroak	Esanahia
USER	Erabiltzailearen izena zerbitzarian	Lan-saioa irekitzeko eskaera
PASS	Erabiltzailearen pasahitza	Kautotze-fasearen bigarren urratsa
LIST	Parametrorik ez	Postontzian dauden mezuen zerrenda bidali
RETR	Mezu baten zenbakia zerrendan	Mezu bat bidaltzeko eskaera
DELE	Mezu baten zenbakia zerrendan	Mezu bat postontzitik kentzeko eskaera
QUIT	Parametrorik ez	Saioa amaitzeko agindua

4.4. taula. POP3 komandoak.

Aipatzekoa da mezu bat jasotzea eta mezu hori postontzitik kentzea ekintza desberdinak direla. Hau da, zerbitzariak ez ditu posta-mezuak postontzitik kentzen bezeroari bidaltzen dizkionean, baizik eta hark espresuki ezabatzeko eskatzen duenean, DELE komandoaren bidez.

4.16. irudian POP3 lan-saio batean elkarri bidalitako komando eta erantzunak ikus ditzakegu. Irudia irakurgarriago egiteko, ez dira jasotako posta-mezuaren goiburuko gehienak agertzen. Hurrengo atal batean aztertuko ditugu nolakoak diren mezu baten goiburukoak.



4.16. irudia. POP3 lan-saio bat. Errore bat sartu dugu mezu bat ezabatzean. «Z:»-z hasten diren lerroak zerbitzariari dagozkio, eta «B:»-z hasten direnak bezeroari.

IMAP

Erabiltzaileak bere posta konputagailu desberdinetatik irakurtzen duenean, POP3 ez da egokia (adibidez, ziberkafe batetik batzuetan eta etxetik beste batzuetan). POP3 aurreko sekzioan ikusi dugun moduan erabiliz, erabiltzailearen mezuak bere konputagailuetan zehar sakabanatzen dira; etxeko PCra jaitsitako mezuak hurrengo egunean lantokiko PCtik ez dira atzigarriak izango.

Konponbidea DELE komandoa ez erabiltzea izan daiteke. Izan ere, posta-bezeroek badute era horretan lan egitea, hau da, badute postontzitik ezabatu gabe mezuak zerbitzaritik jaistea. Hala eta guztiz ere, irtenbide hau noizean behin

egitekoa besterik ez da. Erabiltzailea normalki konputagailu desberdinetan badabil, oso deserosoa da (eta batzuetan oso garestia) posta elektronikoa irakurtzen duen bakoitzean aurreko saioetan jadanik jaitsi diren mezuak berriro ekartzea. Gainera, erabiltzaileek katalogoetan antolatzen dituzte jasotako mezuak. Zerbitzaria atzitzen duen bakoitzean bere mezu guztiak berriro sailkatu behar ditu konputagailu lokaleko katalogoetan? Ez da bideragarria; egin behar dena katalogoaren antolaketa zerbitzarian mantentzea da, baina POP3 protokoloak ez du horretarako komandorik.

Arazo hori eta beste batzuk konpontzeko asmatu zuten IMAP protokoloa (Internet Mail Access Protocol*)*, RFC 3501 agirian definitua. POP3 bezala, IMAP posta irakurtzeko protokoloa da (*atera* moduko protokoloa). POP3k baino ezaugarri gehiago ditu eta, beraz, konplexuagoa da. IMAPek baditu komandoak, erabiltzaileak zerbitzarian duen postontzia lokala balitz bezala manipula dezan. Besteak beste, IMAPen bidez erabiltzaileak badu katalogoak sortzea, katalogo horien artean mezuak mugitzea, edota zerbitzarian mezuak ezabatzea. Oro har, IMAPek zerbitzarian duen postontzia kudeatzeko tresnak eskaintzen dizkio erabiltzaileari. Horregatik IMAP inplementatzea POP3 inplementatzea baino askoz konplexuagoa da, eta IMAP posta-zerbitzariek konputagailuaren baliabide gehiago kontsumitzen dituzte.

IMAPek ondoko ezaugarri garrantzitsu hau ere badu: bezeroak heldu berrien mezuen goiburukoak edo beste zatiak bakarrik jaitsi ditzake. Hau oso erabilgarria da erabiltzaileak abiadura txikiko konexioa duenean; audio- edo bideo-klipak, adibidez, ekidin daitezke.

SMTP eta POP3 bezala, IMAP ASCII kodean oinarritutako konexioaren bidezko aplikazio-protokoloa da. TCP erabiltzen du, eta 143 portua du esleituta. POP3n bezala, bezeroaren eta zerbitzariaren arteko harremana hiru urratsetan egituratzen da: kautotze-fasea, elkarrekintzak, eta konexio-amaiera. IMAPen elkarrekintzak POP3renak baino askoz aberatsagoak dira.

Web posta sistemak

Gero eta posta-erabiltzaile gehiagok erabiltzen dituzte webean oinarritutako posta-zerbitzuak. Web zerbitzuak erabiltzen direnean, web bezeroak posta-bezeroarena egiten du, eta mezuak igortzeko eta irakurtzeko HTTP protokoloa erabiltzen da bezeroaren eta zerbitzariaren arteko komunikazioetan. Posta-zerbitzariak, hala ere, SMTP erabiliko du beste posta-zerbitzariei erabiltzaileak bidalitako mezuak igortzeko.

Posta-zerbitzua atzitzeko irtenbide hau oso komenigarria da posta edozein toki eta konputagailutatik atzitzeko: etxetik, lanetik edo ziberkafe batetik. Behar den gauza bakarra web bezeroa eta Internet konexioa da. IMAPen bezala, erabiltzaileak bere mezuak zerbitzarian antola ditzake, ez konputagailu lokal batean.

IMAPekin alderatuta, weba erabiliz ez dugu konputagailu lokalean inongo bezero berezirik izan behar gure datuekin konfiguratuta. Web bezeroak Internet konexioa duten konputagailu guztietan daude instalatuta; IMAP bezeroak, aldiz, ez. Gainera, web bidezko posta-zerbitzari askok dohainik eskaintzen dute zerbitzu hau.

Posta-zerbitzuetarako weba erabiltzeak dituen desabantailak ondoko hauek dira:

- Zerbitzaria motela izan daiteke, IMAP edo POP3 zerbitzariekin alderatuta. Kontuan izan behar dugu web zerbitzariek beste zerbitzu asko ere ematen dituztela.
- Askotan zerbitzua oso ezegonkorra da, erabiltzaile-kopuru ikaragarriagatik.
- Sarearen atzipena etxeko telefonoaren bidez egiten badugu, eta Internet erabiltzeko tarifa finkorik ez badugu, IMAP edo POP3 baino askoz garestiagoa da web posta erabiltzea, telefono-deia mantendu egin behar baita mezuak idazten edota irakurtzen ditugun bitartean. POP3 eta IMAP sistemetan, aldiz, mezuak bidaltzen edota hartzen diren bitartean soilik irauten dute telefono-deiek, baina ez editatzen ditugunean edo katalogoak arakatzen ditugun bitartean.

4.4.3. Mezuen formatua

Eskutitz fisikoak gutun-azaletan sartzen ditugu, eta gutun-azaletan mezuari dagokion informazioa ipintzen dugu: nori zuzenduta dagoen eta non dagoen hartzaile hori, gutxienez. Posta-bulegoetan gutun-azala zigilatuko dute, igortzeko data eta jasotzeko data grabatuz. Mezuak, beraz, bi osagai izango ditu: gutun-azalaren barruan dagoen mezuaren edukia, eta gutun-azalean dagoen informazioa.

Antzera, mezu elektronikoek bi zati izango dituzte: edukia eta goiburukoa, biak ASCII kodean idatzirik. Edukia da erabiltzaile batek besteari helarazi nahi diona, eta goiburukoan kontrol-informazioa agertzen da, posta arrunteko gutun-azaletan bezalaxe.

RFC 822k definitzen ditu goiburukoaren sintaxia eta semantika. Lerro zuri batek banatzen ditu mezuaren goiburukoa eta edukia. Goiburukoko eremu bakoitzak testu-lerro bat osatzen du (horregatik, askotan, posta-mezuen goiburukoko eremuei *lerroak* besterik ez zaie deitzen), bi zatitan banatua bi punturen bidez: hasieran hitz bat edo bi, eremu horren identifikazioarekin, eta, bi puntuen ondotik, eremu horren balioa. Eremu batzuk nahitaezkoak dira, eta beste batzuk, ordea, hautazkoak. Goiburuko guztietan agertuko dira `From` eta `To` izeneko lerroak. Normalean `Subject` izenekoa ere agertuko da, nahiz eta hautazkoa izan. Kontuz ibili SMTP komandoak eta goiburukoaren eremuak ez nahasteko; 4.2. taulako MAIL FROM eta RCPT TO komandoek erlazio zuzena dute mezuaren goiburukoaren `From` eta `To` izeneko lerroekin, baina batzuk programaren arteko elkarrizketaren zati bat dira, eta besteak bidalitako mezuaren zati bat.

4.16. irudian agertzen den mezuaren goiburukoa ondoko hau da:

```
From: "Iñaki Tolaretxe" tola@andatza.eh
To: "Pello Zabala" pello_zabala@arantzazu.eh
Subject: Eguraldia?
```

`From`, `To` eta `Subject` izeneko lerroak igorlearen posta-zerbitzariak eransten dizkio mezuari. Baina goiburukoaren beste lerro batzuk hartzailearen posta-zerbitzariak eraikitzen ditu, posta-bulegoan gutun-azalean zigilua jartzen duten moduan, hain zuzen ere. Hartzailearen zerbitzariak, mezua jasotzean, `Received` lerroa gehitzen dio goiburukoari. Lerro horretan honako hau adierazten da: mezua bidali duen SMTP zerbitzariaren izena («from»), mezua hartu duen SMTP zerbitzariaren izena («by»), helmugako postontzia («for»), eta jasotzeko data eta ordua. Horrela, 4.16. irudiko mezuaren helmugako erabiltzaileak hartuko duen goiburukoak honako itxura hau izango du:

```
Received: from postaria.hegointernet.eh by mail.isp.com
for <pello_zabala@arantzazu.eh>; Wed, 23 Oct 2002
09:16:27 +0200
From: "Iñaki Tolaretxe" tola@andatza.eh
To: "Pello Zabala" pello_zabala@arantzazu.eh
Subject: Eguraldia?
```

Horrelako goiburuko batean ikus dezakegu zein diren igorlearen eta hartzailearen posta-helbideei dagozkien DNSko MX erregistroak.

Ez da batere arraroa `Received` lerro bat baino gehiago dituen posta-mezua jasotzea. Hori gertatzen da mezuaren bidean bitarteko posta-zerbitzariren bat egon denean. Lerro horiek aztertuz jakin dezakegu zein SMTP posta-zerbitzaritatik pasatu den guk jasotako mezua. `Received` lerro guztien informazioa biltzen duen `Return-Path` izeneko lerroa ere ager daiteke mezuaren goiburukoan. `Return-Path` lerroa hartzailearen posta-zerbitzariak betetzen du; askotan igorlearen helbidea ipintzen du soilik, eta ez ibilbide guztia.

MIME luzapena: datu motak

Lehen aipatu dugun legez, RFC 822 agirian definitzen den mezu-formatuak ASCII moduko mezuak bakarrik bidaltzeko balio du; ez da nahikoa multimediako mezuak (irudiak, audio- eta bideo-mezuak) edo ASCII ez diren testuak (ingelesa ez den beste hizkuntzetan erabiltzen diren karaktereak dituztenak) garraiatzeko. ASCII ez diren edukiak bidaltzeko definitu da RFC 822aren luzapena den MIME (Multipurpose Internet Mail Extensions) estandarra (RFC 2045 eta RFC 2046).

MIMEk mezuen goiburukorako eremu berriak definitzen ditu. Multimedia garraiatzeko bi eremu garrantzitsuenak `Content-Type` eta `Content-Transfer-Encoding` izenekoak dira. `Content-Type` goiburukoko lerroak mezua nolakoa den adierazten dio hartzailearen bezeroari, adibidez, mezuaren

edukia JPEG irudi bat dela adierazteko. `Content-Transfer-Encoding` goiburukoko lerroaren beharra ulertzeko, gogoratu ASCII ez diren edukiak ASCII eran kodetu behar direla SMTP komandoekin ez nahasteko. `Content-Transfer-Encoding` eremuak adierazten dio hartzaileari jatorrizko edukiarekin nolako ASCII kodeketa egin den. Hau da, bezero batek bi lerro horiek dituen mezua jasotzen badu, lehenengo `Content-Transfer-Encoding` lerroaren balioa aztertuko du mezuaren edukia ASCII ez den jatorrizko formatura bihurtzeko, eta gero `Content-Type` eremua erabiliko du mezuaren edukia nola interpretatu behar duen jakiteko.

Hori guztia argitzeko, azter dezagun MIME mezu baten egitura:

```
From: pello_zabala@arantzazu.eh
To: tola@andatza.eh
Subject: isobara-mapa
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

Mezuaren edukia [...
.........................
...] base64 eran kodetuta
.
```

Mezuaren goiburukoan ikus dezakegu ezen igorleak JPEG irudi bat bidaltzen duela (isobara-mapa bat, `subject` eremuak dioenari kasu egiten badiogu), eta base64 kodeketa erabili duela JPEG formatuan zegoen jatorrizko fitxategia ASCII bihurtzeko. Edozein bit multzo 7 biteko ASCII formatura bihurtzeko MIMEk onartzen duen kodetzeko teknika estandarretako bat da Base64. Hartzaileak base64 deskodeketa ezarriko dio mezuaren edukiari, JPEG irudi bat lortzeko. MIME bertsio desberdinak daudenez, hartzaileak jakin behar du igorleak zein erabili duen bere mezua osatzean. Horretarako dago `MIME-Version` eremua goiburukoan. Goiburuko MIME lerro berezi horiek ez ezik, mezuko beste guztiak ere RFC 822 formatua betetzen du. Zehazki, goiburukoaren eta edukiaren artean lerro zuria dago, eta mezuaren amaiera adierazteko puntu bakarra duen lerroa dago.

MIME luzapena: mezuaren egitura

Multimedia mezuek mota desberdinetako zatiak izaten dituzte. Oso arraroa da irudi edo soinua besterik ez duen mezu bat bidaltzea, azalpenen bat ematen duen inongo testurik gabe. Ohikoagoa da testu bat bidaltzea, eta horrekin batera irudi bat eta, beharbada, irudiarekin bat datorren melodia bat. Orain arte ikusitako goiburukoaren lerroek mezu homogeneoak soilik eraikitzeko balio dute, baina ez media ezberdinak garraiatzen dituen posta-mezua deskribatzeko. Falta ditugu

mezuari egitura emateko lerroak, mezuaren zatiak bereizteko goiburukoko eremuak, alegia. MIMEk hori ere bideratzen du, `Content-type` eremuan *multipart* balioa erabiliz.

Zehazki, multimedia mezu batek objektu bat baino gehiago daramatzanean `Content-type: multipart/mixed` lerroa izango du goiburukoan, mezuak objektu asko dituela adierazteko. Hartzailearen bezeroak honako datu hauek jakin beharko ditu:

(i) non hasten den eta bukatzen den objektu bakoitza,

(ii) ASCII ez den objektu bakoitza nola dagoen kodetuta, eta

(iii) objektu bakoitza nolakoa den.

Azken finean, mezuari egitura eman behar zaio, hau da, zatiak bereizi eta zati bakoitzaren deskribapena egin. Zatiak bereizteko `Content-type: multipart/mixed` lerroan definituko da mezuaren zatien arteko muga egiten duen karaktere-sekuentzia. Adibidez, `=======67843%` izan daiteke sekuentzia hori. Definitutako sekuentzia zati bakoitzaren hasieran agertuko da, eta beti izango ditu bi gidoi aurretik eta lerro-aldaketa atzetik. Gero, mezuaren zati bakoitzak bere goiburuko propioa izango du, `Content-type` eta `Content-Transfer-Encoding` eremuek osatuta, zatiaren edukia deskribatzeko. Hori guztia honako mezu honetan ikus dezakegu:

```
From: pello_zabala@arantzazu.eh
To: tola@andatza.eh
Subject: isobara-mapa
MIME-Version: 1.0
Content-Type: multipart/mixed; Boundary="=======67843%"
--=======67843%
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: 8bit
Aspaldiko Tola:
Hona doakizu gaurko isobara-mapa. Asma ezazu zu zeuk
biharko eguraldia!

Pello
--=======67843%
Content-Type: image/jpeg
Content-Transfer-Encoding: base64
Mezuaren edukia [...
. . . . . .
...] base64 eran kodetuta
.
```

Aurreko mezuan bi zati daude: lehenago testu soileko zatia dago, eta, ondoren, horrekin batera bidalitako JPEG irudi bat.

4.5. IP TELEFONIA (VoIP)

Oro har, VoIP sistemek ematen duten oinarrizko zerbitzua ahots-zerbitzu arrunta da, hots, telefonoz hitz egitea. Hala ere, oinarri beretik abiatuta, oso bestelakoak izan daitezke sistema desberdinek emandako zerbitzuak. Adibidez:

- Ahotsa ez ezik, irudiak ere transmititu daitezke. Kasu horretan bideokonferentzia-zerbitzua dugu.
- Erabilera-esparruari dagokionez, sistema lokalak edo unibertsalak izan daitezke. Hau da, gure sarean dauden erabiltzaileen artean hitz egiteko soilik, edo Interneten bidez edozeinekin hitz egiteko. Sistema lokal bat da, esaterako, bere bertako sarea barruko telefono-komunikazioetarako erabiltzen duen erakunde bat. Sistema unibertsalak, adibidez, Interneten bidezko telefonia eskaintzen duten konpainiak dira (*Skype* edo *Yahoo! Voice* besteak beste).
- Sistema batek konputagailu bidezko komunikazioetarako soilik eman dezake zerbitzua, eta beste batzuek, aldiz, telefono konbentzionalekin hitz egiteko ere balio dute.

Sistema telefoniko konbentzionaletan bezala, dei telefoniko batean honako bi fase hauek betetzen dira: lehenago, deia ezarri (dei-bideraketa), eta gero elkarrizketa gauzatu. Deia ezartzeko lan nagusia da deitutakoa non dagoen aurkitzea. Arazoak DNS ebazpenaren antza du, baina kasu honetan konplexuagoa da, dei hartzailea toki desberdinetan egon daitekeelako une bakoitzean (IP mugikorrean eta telefonia mugikorrean gertatzen den bezala), eta oso teknologia desberdineko sare asko (IP sareak eta telefonia konbentzionaleko sareak) inplikatuta egon daitezkeelako. Bilaketa egiteko zenbait eragilek hartu beharko dute parte. Elkarrizketa gauzatzean, dei-bideraketan ez bezala, eragile mota bakarra dago: solaskidea. Bere lan nagusia ahotsa eta irudiak denbora errealean digitalizatzea eta erreproduzitzea da. Hemen dei telefonikoaren lehenengo faseari bakarrik ekingo diogu, bilaketari, alegia.

4.5.1. Aplikazioaren osagaiak

Gaur egun dauden VoIP sistema gehienak bezero/zerbitzari ereduari jarraitzen diote deia ezartzeko. Kasu horretan, honako hauek dira aplikazioaren osagaiak:

- Bezeroak.

  Erabiltzailearen konputagailuan egikaritzen den softwarea da. Honako zeregin hauek ditu:

  – Erabiltzailearekiko interfazea.

  - Ahotsaren kodeketa eta deskodeketa.
  - Zerbitzariekiko komunikazioa IP sarearen bidez.
- Zerbitzariak.

  Bi motatako zerbitzariak daude:
  - Deia bideratzeaz arduratzen diren zerbitzariak. VoIP sistema gehienetan beste bi azpimotatako zerbitzariak agertzen dira bideratze-lan hau betetzeko: proxiak eta erregistratzaileak. Erabiltzaile bakoitza lotuta dago proxi batekin eta erregistratzaile batekin (bi lanak konputagailu berak egitea badago). Hurrenez hurren, beraien zereginek DNS bertako zerbitzariarenarekiko eta DNS jatorrizko zerbitzariarenarekiko antza handia dute.
  - IP sarearen eta sare telefoniko arrunten arteko pasabideak. Hauen lana datagrametan datozen laginak sare telefonikoan txertatzea da, baita kontrakoa ere. Gainera, VoIP protokoloaren eta sare telefonikoen kontrol-seinalizazioaren arteko lotura (pasabidea, alegia) ere egiten dute.
- Protokoloak.

  Kapitulu honetan ikusi ditugun beste aplikazioetan ez bezala, ez dago, oraindik, estandar bakarra, gehiengoak erabiltzen duena. Konpainia batzuek bere protokolo propioak garatu dituzte (Skype kasu). Beste alde batetik, badaude bi estandar ofizial, biak nahiko erabiliak eta guztiz desberdinak. Honako hauek dira:
  - H.323 ITUk ateratako estandar multzoa da, hau da, konpainia telefonikoen inguruan sortutako estandarra. Oso konplexua eta asmo handikoa da.
  - SIP (Session Initiation Protocol). Internet inguruko munduak egindako proposamena. Bere filosofia Internetena bera da, KISS siglak biltzen dutena: *Keep It Simple Stupid*. Hau da, oso protokolo erraza da. Bere erabilera ez da telefonia-aplikazioetara mugatzen, eta, bere izenak dioenez, bi aplikazioko entitateen artean edozein denbora errealeko lan-saioak ezartzeko balio du. Lan-saio horiek, telefoniaz gain, bideokonferentzia, bideo *streaming*, berehalako mezularitza edo sare-jokoak izan daitezke.

Osagai hauen arteko elkarrekintza ondoko atalean adierazten dugu.

4.5.2. Funtzionamendua

Adibide bat erabiliz azalduko dugu horrelako sistema tipiko baten funtzionamendua, dei-bideraketari dagokionez. Adibiderako SIP protokoloa erabiltzen dela hartuko dugu. H.323 erabiltzen duten sistemen oinarrizko funtzionamendua antzekoa da.

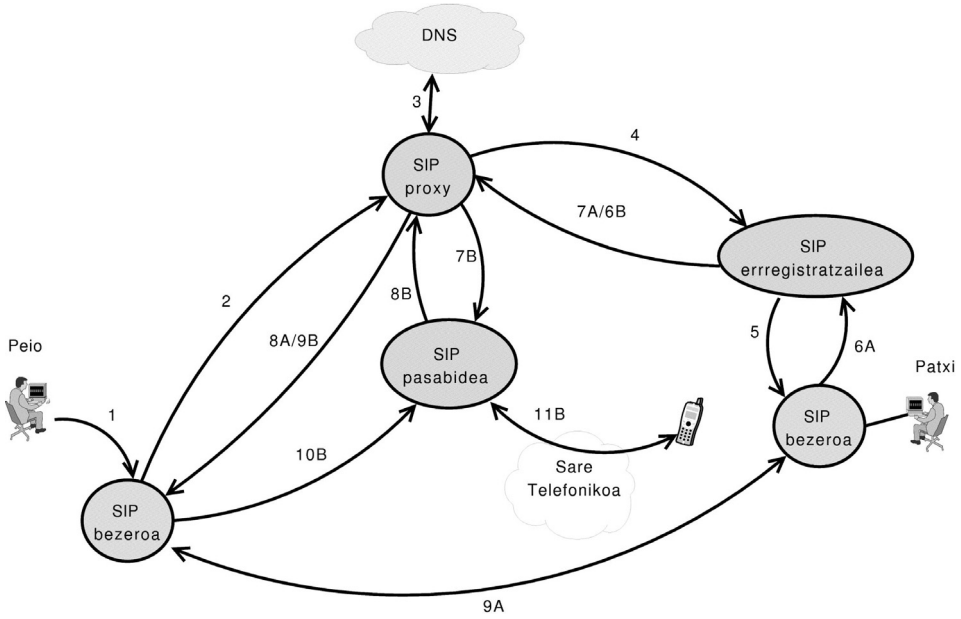
Demagun `patxi@sip_register.org` SIP identifikazioa duen erabiltzaile bat. Deiak jaso ahal izateko, erabiltzaile horrek bere kokapenaren berri eman behar dio bere SIP erregistratzaileari. Adibidez, demagun `patxi@sip_register.org` atzigarri egongo dela 155.222.30.87 helbidea duen konputagailuan, 5060 UDP portuan, edo, hor ez badago, 677 711433 telefono-zenbakian. Horren berri emango dio bere erregistratzaileari, eta horrek informazio hori gordeko du, Patxirentzako deia jasotzen duenean erabiltzeko.

Demagun beste erabiltzaile batek, izan bedi Peio, Patxirekin hitz egin nahi duela, eta horretarako bere konputagailuan duen SIP bezeroa erabiliko duela. Honako hauek dira bi lagunak hitz egiten hasi arte emandako urratsak (ikusi 4.17. irudia):

(1) Peiok bere SIP bezeroari Patxiren helbidea (hau da, `patxi@sip_register.org`) emango dio, deia ezartzeko.

(2) SIP bezeroak, SIP protokoloa erabiliz, bere SIP proxiari birbidaliko dio eskaera. SIP proxi hori zein den, SIP bezeroaren konfigurazioan ezartzen da. Datua VoIP hornitzaileak eman behar dio Peiori.

(3) Proxiak DNSri eskatuko dio emandako helbidearen eskuinaldean dagoen izenari dagokion SRV erregistroa. Hau da, posta-zerbitzariek bezala egiten du, baina MX erregistroaren ordez, SRV motakoa eskatzen da. Erregistro horrek gordeko du Patxiren SIP erregistratzailearen izen kanonikoa. Hortik abiatuta, erregistratzailearen IP helbidea ere lortuko du proxiak.



4.17. irudia. SIP dei bat egiteko urratsak (sinplifikaturik).

(4) Proxiak deiaren berri emango dio erregistratzaileari, berriro SIP protokoloa erabiliz.

(5) Erregistratzaileak begiratuko du Patxiri buruz duen informazioa, eta saiatuko da 155.222.30.87 konputagailuan egon beharko lukeen SIP bezeroarekin kontaktatzen, deiaren berri emateko.

Hemendik aurrera, bi aukera aztertuko ditugu adibidean. Hasierakoa 6A urratsetik 9A urratsera deskribatzen dugu, eta bigarrena 6Btik 11Bra.

(6A) Demagun konputagailu hori piztuta eta sarean konektatuta dagoela, SIP bezeroa abiatuta dagoela, eta Patxi konputagailu aurrean ari dela lanean. Orduan, bezeroak Patxiri jakinaraziko dio dei bat duela, eta aurkeztuko dizkio pantailan dei horren ezaugarriak (nork deitzen duen, gutxienez). Demagun Patxik onartzen duela deia (hau telefonoa hartzearen baliokidea da): orduan bere bezeroak deiaren onarpenaren berri emango dio, SIP erabiliz, erregistratzaileari. Horrekin batera esango dio zein IP helbidetan eta portutan jasoko duen deia.

(7A) Erregistratzaileak datu horiek helaraziko dizkio Peioren proxiari, SIP erabiliz.

(8A) Proxiak datu berberak emango dizkio eskaera bidali zion SIP bezeroari, Peiorenari, alegia.

(9A) Peioren bezeroak deia egingo dio jasotako IP helbide/portuari, eta bi lagunak hasiko dira hizketan. Bezeroek ahots-laginak mikrofonotik jaso, kodetu, eta bidaliko dizkiote elkarri, eta saretik jaso, deskodetu, eta bozgorailura bidali egingo dituzte.

(6B) Demagun Patxiren konputagailua ez dagoela sarean, edo Patxik ahaztu duela bere SIP bezeroa abiatzea, edo, nahiz eta konputagailua sarean egon eta SIP bezeroa abiatu izan, Patxi joan dela une horretan kafetegira, eta, berez, erregistratzaileak ez duela erantzunik jaso 155.222.30.87 konputagailutik. Kasu horretan erregistratzaileak Peioren proxiari bidalitako SIP erantzunean 677 711433 zenbakian saiatzeko esango dio.

(7B) Peioren proxiak 677 711433 telefono arruntarekin kontaktatzeko zein den pasabide egokia ebatzi behar du. Batzuetan lan zaila izango da aukeraketa hori egitea. Irizpideak teknikoak (helburuko sarearekin konexio zuzena duen pasabide bat bilatzea, edo une horretan elkarrizketa gutxien duen pasabidea aurkitzea, adibidez) eta ekonomikoak izaten dira (helburuko sare arruntarekin komunikazioetarako tarifarik merkeena ematen digun pasabidea aukeratu, adibidez). Aukeratutako pasabideari SIP mezu bat bidaliko dio, deiaren berri emanez.

(8B) Pasabideak proxiari SIP mezu batean esango dio bere zein IP helbidetan/portutan egin behar den konexioa deia gauzatzeko.

(9B) Proxiak jasotako informazio hori Peioren bezeroari helaraziko dio beste SIP mezu batean.

(10B) SIP bezeroak saioa hasiko du emandako pasabidearen IP/portuan.

(11B) Pasabideak emandako telefonora egingo du deia. Patxik erantzuten badu, sare konbentzionaletik jasotako hitz-jarioaren formatua aldatu, eta SIP saioaren bidez Peioren bezeroari helarazi. Horrek, aurreko kasuan bezala, deskodetu eta bozgorailuan jarriko ditu hitzak, baita Peiok esandakoak mikrofonotik jaso, kodetu, eta pasabideari bidali ezarritako SIP saioaren bidez. Horrekin elkarrizketa abiatu da.

SIP protokoloa

Aurreko adibidean erabilitako SIP protokoloaren ezaugarri nagusiak honako hauek dira:

- Komunikazio-eredua: konexio bidezko protokoloa da, bere izenak iradokitzen duen moduan («saioak» ezartzeko protokoloa da).
- Garraio-mailako edozein zerbitzu, TCP ala UDP, erabil dezake.
- Sintaxiaren aldetik, karaktereka moduko protokoloa da. HTTPn dago oinarrituta, eta, beraz, mezuak goiburukoetan eta edukietan egituratzen dira.

Xehetasunak ezagutzeko, jo estandarrera (RFC 3261-3265).

4.6. P2P APLIKAZIOAK

Gero eta garrantzi handiagoa dute P2P ereduari jarraitzen dioten sare-aplikazioek. Izan ere, zenbait txostenaren arabera, gaur egun Interneten ibiltzen den trafiko gehiena horrelako aplikazioei dagokie. Hasiera batean, fitxategiak konpartitzeko sistemetarako garatu eta erabili izan dira P2P aplikazioak, baina haien erabilera beste era bateko aplikazioetara hedatzen ari da (IP telefonia- eta telebista-sistemetara, hain zuzen ere).

Edozein sare-aplikaziotan, honako bi urrats hauek ematen dira erabiltzaileak eskatutako zerbitzua ematearren:

1. Aurkitu, sarean zehar, zerbitzua emateko behar diren beste aplikazioko entitateak.
2. Beste entitate horiekin elkarlanean aritu zerbitzua gauzatzeko.

Bezero/zerbitzari ereduari jarraitzen dioten sare-aplikazioetan, lehenengo urrats hori erabiltzaileak emandako informaziotik abiatuta betetzen da. Aplikazio horietan, erabiltzaileak, zerbitzua eskatzearekin batera, zerbitzu hori eman behar duen zerbitzariaren identifikazioa ere hornitu behar du. Web orri bat eskatzean, adibidez, orriaren identifikazioarekin batera, zerbitzariarena dugu adierazita URLan. Postako mezu bat bidaltzean (edo SIP dei bat egiteko), hartzailearen helbideko @

karakterearen atzetik dugun izenak zehazten du zein SMTP zerbitzariri eman behar zaion mezua (edo zein SIP erregistratzaileri helarazi behar zaion deia). DNSren kasua zertxobait desberdina da, erabiltzaileak emandako identifikadoreak (izen bat, berriro ere) ez baitu zuzenean adierazten zein den bilatzen dugun DNS erregistroa gordetzen duen DNS zerbitzaria. Baina, hala ere, zerbitzari horren bilaketa gauzatzeko behar den informazioa izen horretan dugu: izenaren azken osagaiak erro-zerbitzariari balioko dio dagokion TLD zerbitzaria identifikatzeko, eta hortik, haritik tiraka, behar den jatorrizko zerbitzariraino ailegatuko gara.

P2P aplikazioetan, partaideak elkarlanean aritzen dira erabiltzaileei zerbitzua emateko, bezero/zerbitzari aplikazioetan bezeroak eta zerbitzariak aritzen diren modu berean. Baina P2P aplikazioetan, aldiz, erabiltzaileak ez daki, hasiera batean, zein kidek emango dion eskatutako zerbitzua. Hau da, erabiltzaileak *zer* nahi duen adieraziko dio bere bezeroari[23], baina ez *nork* emango dion. Horregatik, aplikazioko lehenengo lana partaideen artean zerbitzua emateko moduan daudenak toptzea izango da. Bilaketa hori egiteko eraren arabera sailkatzen dira P2P aplikazioak.

P2P izaera

Aplikazioaren bi aldeek (zerbitzua emango duenaren bilaketa eta zerbitzua bera gauzatzea) P2P izaera badute, ez dago zalantzarik aplikazioa P2P erakoa dela esateko. Baina bigarren aldea besterik ez bada P2P erakoa, batzuetan aplikazioa P2P erakotzat hartzen da eta beste batzuetan ez. Portaera kontraesankor horren adierazlea Napster eta SIP telefonia dira. Bietan bilaketa bezero/zerbitzari ereduari jarraituz egiten da, eta, gero, zerbitzua kideen arteko lanetan betetzen da. Baina Napster P2P aplikaziotzat hartu zen (izan ere, P2P ereduari bultzada eman ziona da), eta SIP telefonia, aldiz, ez. Ondoko taulan duzu beste aplikazio batzuen bi aldeko izaera.

23. Nahasgarria bada ere, *bezero* izena ematen zaie P2P aplikazioko entitateei. Bezero bat baldin badago, zerbitzaria ere egongo dela dirudi. Laster ikusiko dugunez, P2P aplikazio gehienak ez dira hain P2P erakoak, nolabaiteko zerbitzari-lana egiten duten aplikazioko entitateak beharrezkoak baitituzte.

Aplikazioa	Bilaketa-sistema	Bezero/zerbitzaria	Aplikazioaren eredua
DNS	Bezero/zerbitzaria	P2P	Bezero/zerbitzaria
Berehalako mezularitza	Bezero/zerbitzaria	P2P	Bezero/zerbitzaria
SIP telefonia	Bezero/zerbitzaria	P2P	Bezero/zerbitzaria
Napster fitxategi-konpartizioa	Bezero/zerbitzaria	P2P	P2P
FastTrack fitxategi-konpartizioa	P2P	P2P	P2P
Skype telefonia	P2P	P2P	P2P

4.5. taula. Aplikazioen sailkapena, ereduaren arabera eta aplikazioaren aldeak bereizita.

4.6.1. Kideak bilatzeko sistemak P2Pn

Ondoko hiru taldetan sailka ditzakegu P2P aplikazioetako solaskideak bilatzeko sistemak:

- Aurkibide zentralizatua.
- Zunda-uholdea.
- Talde hierarkizatua.

Ikus ditzagun banan-banan.

Aurkibide zentralizatua

Zerbitzari batek gordetzen du taldeko kideen zerrenda. Kide berri batek taldean sartu nahi duenean, zerbitzari horri jakinaraziko dio, eta aurkibidean sartuko du zerbitzariak kide berria. Kide baten bila gabiltzanean, zerbitzariari bidali beharko diogu eskaera.

Ikusten denez, hau ez da P2P sistema bat, zerbitzari batean baitatza. Hala ere, P2P terminoa zabaldu zuen lehenengo aplikazioak (fitxategiak konpartitzeko erabiltzen zen Napster izenekoa) horrela egiten zuen kideen bilaketa, eta horregatik agertzen da sistema hau P2Pri buruzko testuetan. Napsterren P2P izaera aplikazioko beste aldean zegoen, fitxategiak kideen artean bidaltzen zizkioten elkarri, eta ez zerbitzari batetik jaitsi, web edo ftp sistemetan egiten den bezala.

Eskema honen indarra eta ahulezia bere izaera zentralizatua da aldi berean. Indarra, oso sinplea delako. Baina soiltasun horren truke ondoko bi arazo larri ditu: bata, zerbitzariak kale egiten badu, sistema osoa bertan behera gelditzen dela, eta, bestea, sistema ez dela eskalagarria. Hau da, kide kopurua hazten den heinean, gero eta zailagoa izango da zerbitzariak bere lana egitea. Napsterren era zentralizatu

honek oso zaurgarria egiten zuen sistema, eta jabego intelektualaren eta kopia-eskubidearen arteko borroka judiziala hasi zenean, Napster izan zen lehenengo biktima.

Aurkibide zentralizatuaren oso kasu berezitzat har daiteke BitTorrent sistema. Bere aurreko fitxategiak konpartitzeko sistemekin alderatuta, honako bi aldeak ditu BitTorrent-ek:

- Ez du definitzen fitxategiak aurkitzeko inongo sistemarik. Fitxategi baten konpartizioan parte hartzeko (deskargatzen edo bidaltzen), fitxategiari dagokion `.torrent` formatuko fitxategia aurkitu behar du erabiltzaileak. Fitxategi horretan aurkituko duen zerbitzariak (*tracker*) emango du behar den informazioa (taldeko kideen identifikazioa, bereziki) fitxategia konpartitzeko.
- Beste sistemetan fitxategi bat deskargatu nahi denean, nortzuek duten fitxategia aurkitu eta gero, horietako bat aukeratu eta hortik deskargatzen da fitxategi osoa. BitTorrent-en, aldiz, fitxategia kide desberdin askotatik deskargatzen da aldi berean, zatika. Fitxategi bat konpartitzen duten kideek BitTorrent P2P sare bat (*torrent* bat) osatzen dute.

Beraz, BitTorrent ez da sistema zentralizatu bat, baizik eta sistema zentralizatu txiki pilo bat. Horregatik ez ditu izan «benetako» sistema zentralizatuek izan dituzten arazoak.

BitTorrent protokoloak ez du zehazten nola aurkitu behar dugun `.torrent` fitxategia, baina badaude mota horretako fitxategiak gordetzen dituzten beste zerbitzariak (*BitTorrent index*). BitTorrent sare batean parte hartu nahi denean, lehenengo urratsa horrelako zerbitzari batengana jotzea izango da, dagokigun `.torrent` fitxategia hor bilatzeko.

Zunda-uholdea

Eskema honetan taldekide bakoitzak bere bizilagunak diren beste hainbat taldekiderekin sortutako loturei eusten die. Kideen arteko lotura horiek bilaketetarako sare bat[24] osatzen dute. Kide batek beste kideren batekin lan egin nahi duenean (fitxategi bat kopiatzeko, edo elkarrizketa telefoniko bat izateko, adibidez), bere bizilagunei zunda bat bidaltzen die, nolako kidearen bila dabilen adieraziz. Zunda hartzen duten kideek, eskatutakoa betetzen badute, erantzungo dute, eta, edozein kasutan, zunda birbidaliko diete beren lagunei. Horrela zunda zabaltzen da sarean, uholde moduan. Mugarik ez bazaio jartzen zundari, uholdea sare osora zabaltzen da, eta eskalagarritasun-arazo bat sortu (kide asko daudenean uholdeek sortutako trafikoak sarea kolapsatuko du). Horregatik, IP datagramen TTL antzeko

24. Ingelesezko jatorrizko testuetan *overlay network* terminoa erabiltzen da. Horren itzulpen zuzena egitea ('geruza-sarea') baino egokiagoa iruditu zait 'bilaketetarako sarea' terminoa erabiltzea.

kontagailu bat ezartzen zaio hasierako zundari, zenbat aldiz birbidal daitekeen mugatzeko. Zunda birbidaltzen duen kide bakoitzak kontagailu horri bat kentzen dio, eta, zeroraino heltzen denean, ez da gehiago zabalduko.

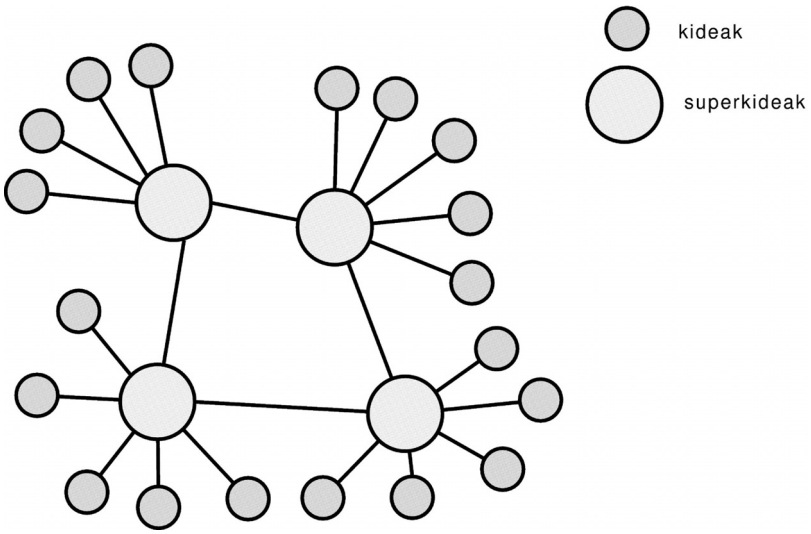
Sistema honen arazo nagusia hasieratzea da (*bootstrap problem*): nola gehitu sarean kide berriak? Ezinbestekoa da dagoeneko sarean dagoen kideren baten IP helbidea ezagutzea. Arazo horretarako ez dago irtenbide garbirik. Aukera bat kide bakoitzak bere partaidetza publiko egitea da bere ezagunen artean (bere webgunean, edo posta elektronikoko mezuen sinaduran), baina oso mugatua da. Beste bat da taldekide batzuen identifikazioa gordetzen duten zenbait aurkibide argitaratzea. Ohartu bigarren aukera ez dela aurkibide zentralizatuko sistemak bezalakoa, honako arrazoi hauengatik: hasieratze-aurkibide hauetan ez da gordetzen taldekideen zerrenda osoa (konektatuta egoten diren gutxi batzuekin nahikoa da), zerrenda desberdinak egon daitezke (eta ez osoa den bakarra), eta kideek ez dituzte erabiltzen aurkibide hauek zerbitzua gauzatuko duten beste kideak bilatzeko (adibidez, fitxategi bat bidaliko digun beste nodo bat bilatzeko), baizik eta bilaketetarako sarean beren hasierako bizilagunak izango direnak topatzeko.

Zunda-uholdea erabiltzen zuen sistema baten adibidea jatorrizko Gnutella zen. Hasieratzeko arazoa gainditzeko, banatzen ziren Gnutella bezeroek (askotan, P2P aplikazioko entitateei «bezero» deitzen zaie) aurrez grabatuta zeukaten normalki konektatuta egoten diren zenbait kideren helbideen zerrenda.

Talde hierarkizatua

Bilaketa-sistema hau aurreko bien artekoa da. Bilaketetarako sarea ez dute kide guztiek osatzen, zunda-uholdeko sistemetan bezala, bereziak diren batzuek baizik. Horiei superkide deituko diegu. Kide arrunt bakoitzak superkide batekin izango du lotura. Bilaketa bat egin behar denean, kideak bere superkideari helaraziko dio eskaera, eta horrek, beste superkideekin lankidetzan, erantzuna topatu beharko du. Erantzuna lortzeko bide posible bat superkideen artean zunda-uholdea hedatzea da, baina arinagoak diren DHT (Distributed Hash Table) sistemak gehiago erabiltzen dira. Horrelako bi mailako P2P sarea 4.18. irudian dugu.

Lehenengo talde hierarkizatuko P2P sistema FastTrack izan zen. Geroko Gnutellak ere erabiltzen du, baita FastTrack asmatu zutenek geroago sortutako Skype telefonia-sistemak ere. BitTorrent fitxategiak banatzeko sistemari egindako hedapenetan DHT erabiltzen da dagokigun `.torrent` fitxategia aurkitzeko, baina ez dago superkiderik haren bilaketetarako sarean. Kide guztiek hartzen dute parte bilaketan, baina ez, ordea, uholde moduan.



4.18. irudia. P2P sare hierarkizatuak.

LABURPENA

Sare-aplikazioen osagaiak eta egitura aurkeztu ditugu kapitulu honen hasieran, protokoloen espezifikazioak egiteko proposamen batekin batera. Gero, Interneten erabiltzen diren aplikaziorik garrantzitsuenak aukeratu ditugu benetako aplikazioak nolakoak diren ikasteko, eta hainbestetan erabiltzen ditugun aplikazio horien barrukoak ulertzeko. Izendegi-zerbitzua, posta elektronikoa, informazio-banaketa eta IP telefonia dira aplikazio horiek.

DNSk ematen du izendegi-zerbitzua Interneten. Berari esker ez ditugu IP helbideak erabili behar beste aplikazio guztietan zerbitzariak identifikatzeko. Beraz, oso aplikazio estrategikoa da DNS, ia beste aplikazio guztiek erabiltzen baitute. Sare batean DNS zerbitzua ez badabil, beste zerbitzu gehienak bertan behera gelditukо dira. DNS datu-base banatutzat har dezakegu: bezeroak, ebazlea izena duenak, galderak egiten dizkio izen-zerbitzariek gordetzen duten datu-base horri. Datu-basea izen-zerbitzariek gordetzen dituzten DNS erregistroek osatzen dute. DNS galdera-erantzunak bidaltzeko, UDP erabiltzen zen hasiera batean, baina gaur egun TCP ere erabiltzen da. DNS protokoloak definitzen ditu DNS izenak, beste aplikazio gehienetan ere erabiltzen direnak. DNS izenak hierarkikoki egituratzen dira. DNS zerbitzariak ere era hierarkikoan daude antolaturik; hierarkia horren goren puntuan erro-zerbitzariak daude, eta domeinu bakoitzean bertako zerbitzariak eta jatorrizko zerbitzariak daude.

Weba da, zalantzarik gabe, gaur egungo Internetaren izarra. Informazioa argitaratzeko eta banatzeko erabiltzen da, hipermediaren bidez. Horretarako

oinarrizkoa izan da HTML formatuaren definizioa. Informazioaren kokapena adierazteko URLak erabiltzen dira. Arakatzaileen (hori da web bezeroen izen berezia) eta web zerbitzarien arteko harremana HTTP protokoloari jarraituz gauzatzen da. Horrek ere TCP erabiltzen du, baina posta elektronikorako protokoloak ez bezala, konexiorik gabeko aplikazio-mailako protokolo bat da HTTP. Hala eta guztiz ere, HTTPk posta elektronikoko mezuen sintaxia definitzen duen RFC 822aren elementuak jasotzen ditu: ASCII kodean oinarritutako protokoloa da, eta bere komando eta erantzunen formatuan RFC 822aren antzeko goiburukoak aurkituko ditugu. Web proxi/cacheak gero eta garrantzitsuagoak dira zerbitzu azkarra jasotzeko. Webaren gainean garatzen diren aplikazio banatuak web aplikazioak dira. Horrelako aplikazioak askoz errazagoak eta azkarragoak dira garatzeko, prozedurak burutzeko softwarea besterik ez baita egin behar. Beste guztia (erabiltzailearekiko interfazea, sare-komunikazioak, eta biltegiratze-sistema) webak eta datubaseetarako softwareak egiten baitute.

Posta elektronikoa, webarekin batera, Interneteko aplikaziorik ezagunena eta erabiliena da. Oinarria posta-zerbitzariek osatutako nazioarteko posta-sarea da. Zerbitzarien artean hitz egiteko SMTP protokoloa erabiltzen da. SMTP ere ASCII kodean oinarritutako konexioen bidezko protokolo bat da. Bezeroen eta zerbitzarien arteko harremanetan SMTP eta beste protokolo bat erabiltzen dira: SMTP mezuak bidaltzeko, eta bestea zerbitzariak gordetzen duen erabiltzailearen postontzitik mezuak jaisteko. Bigarren protokolo hori POP3, IMAP edo HTTP izan daitezke. Protokolo horiek guztiek TCP konexioak erabiltzen dituzte. Posta-mezuen formatua RFC822 arauak definitzen du. Formatu horrek ere eragina izango du beste aplikazioetan, webaren eskaera eta erantzunen formatuan, hain zuzen ere. RFC822 araua oso zaharra da; definitu zenean, ASCII mezuak besterik ez ziren bidaltzen posta elektronikoaren bidez. Multimediari eustearren, MIME arauak RFC 822aren ahalmena zabaldu du.

Gero eta gehiago erabiltzen dugu Internet telefonoz hitz egiteko, IP telefoniasistemei esker. Sistema horietan, betiko telefonian bezala, bi fase daude dei batean: lehenago, deia ezarri, eta, gero, solasaldia gauzatu. Deia ezartzeko honako bi protokolo hauek dira nagusiak: SIP eta H.323. Bietan egiten denak DNS ebazpenaren antza du: lehenago aurkitu behar da zein den solaskidearen uneko kokaguneari buruzko informazioa duen zerbitzaria (erregistratzailea), eta gero zerbitzari horri galdetu.

P2P eredua fitxategiak konpartitzeko aplikazioetan hasi zen erabiltzen, eta gero eta gehiago zabaltzen ari da denbora errealeko multimedia aplikazioetan, IP telefonia eta IP telebista kasu. P2P sistemak sailkatzeko gehien erabiltzen den iritzia kideak bilatzeko sistema da. Horren arabera, sistema zentralizatuak, zundauholdekoak, eta bilaketa hierarkikoko P2P sistemak ditugu.

5. Segurtasuna sarean

Kapitulua ikasi eta gero, ikasleak jakin beharko du:

- Zer diren suhesiak eta zertarako erabiltzen diren.
- Zer diren mugasareak eta zeintzuk diren haien topologiak.
- Zein diren komunikazio seguruaren ezaugarriak.
- Zein diren riptografia simetriko eta asimetrikoaren oinarriak, eta berauen erabilera komunikazio segurua lortzeko.
- Zer den sinadura digitala.
- Zer diren ziurtagiriak eta nola erabiltzen diren komunikazio seguruak lortzeko.
- Zer diren VPNak, IPsec, eta SSL/TSL, eta zertarako erabiltzen diren.

5.1. SARRERA

5.1.1. Sareak eta segurtasuna

Segurtasunaz eta sareez hitz egiten denean, beharrezkoa da honako bi arlo hauek bereiztea:

- Sare seguruak.

  Arlo hau sareak emandako zerbitzuen eta sarean dauden ekipoen segurtasunaz arduratzen da. «Sarea» diogunean, bertako sare pribatu batez ari gara, nahiz eta publikoari zerbitzu batzuk eman edota gure sareko konputagailu batzuetarako atzipena onartu kanpoko konputagailu batzuei. Sarea segurua izango da sareko aplikazioko zerbitzarien, erabiltzaileen ekipoen, eta interkonexiorako ekipoen kontrako mehatxuei (kanpokokoak zein barrukoak) aurre egiteko prest baldin badago. Sareko segurtasunak bi atal ditu:

  – Sarrera-kontrola. Beronen oinarria mugasareak dira. Laster aztertuko ditugu.

– Segurtasunaren kudeaketa. Alde batetik segurtasun-arauak (politikak) definitu behar dira, eta beste alde batetik, arau horiek gauzatzeko prozedurak definitu behar dira.

- Komunikazio seguruak.

  Arlo honen helburua sare-aplikazio seguruak egitea da. Beronen oinarria kriptografiaren erabilera da.

5.1.2. Sare-segurtasunerako arriskuak

Segurtasun informatikoaz hitz egiten denean, sistema informatikoari edo haren zatiren bati eragin diezaieketen arrisku guztiak hartzen dira kontuan, nahita sortutakoak eta nahigabekoak. Nahita egindakoak gizakiek egindakoak dira, hainbat asmoren bila (etekinak lortu, kaltea egin, ondo pasatu, informazioa lortu…). Nahigabekoen iturria gizakia bera izan daiteke (akatsak, prestakuntzarik eza, utzikeria…) edo naturako fenomenoak. Testu honetan kontuan hartuko ditugun arriskuak ondoko bi irizpidek mugatzen dituztenak dira:

- Alde batetik, sare informatiko bat edukitzearen ondorioz sortzen diren arriskuak soilik hartuko ditugu kontuan. Beraz, ez ditugu kontuan hartuko arrisku fisikoak (gela bat urez betetzea, edo zerbitzari bat matxuratzea, adibidez), edo lan-kudeaketari dagozkionak (disko baten segurtasun-kopia ez egitea, adibidez).
- Beste alde batetik, nahigabeko arriskuak ez ditugu kontuan hartuko.

Murrizte horiek hartuta, arriskuez hitz egitea erasoez hitz egitea bihurtzen da, eta, zehatzago, sarearekin lotutako erasoez. Helburuaren arabera, honako hiru sare-eraso bereiziko ditugu:

1. Sare-zerbitzuen erabilgarritasunaren kontrako erasoak.

   Eragin handiena duten erasoak dira hauek. Bere formarik ohikoenak ondoko hauek dira:

   - Zerbitzarien kontrako zerbitzu-ukapena (DoS – Denial of Service). DoS terminoa erabiltzen denean, zerbitzaria itotzen duen eraso motari egiten zaio erreferentzia. Eraso hauek erabiltzaileentzako aplikazioen kontrakoak izan daitezke (web zerbitzaria, posta-zerbitzaria…), edo sareko zerbitzuak edota ekipoen kontrakoak (DNS zerbitzariak, DHCP zerbitzariak, bideratzaileak…). Eraso honen aldaera bat sare osoa itotzea da, hau da, sare osoa (eta ez bere zerbitzari edo zerbitzu bat) kolapsatzeko adina trafiko sartzea lineetan.
   - Zerbitzu baten konfigurazioaren kontrako erasoak. Zerbitzarien konfigurazio-fitxategiak aldatzean dautza. Eraso hauek sailkatzeko irizpide bat beraien eragin-eremua izan daiteke:

– Eraso orokorrak: erabiltzaile eta egoera posible guztietarako zerbitzariak ondo ibiltzeari uzten dio erasoaren eraginez. Begi bistakoak dira eraso hauek, hau da, erasoa gertatzen denean, ez da inongo teknikarik behar erasoa atzemateko.

– Zuzendutako erasoak: konfigurazio-aldaketak egoera edo zerbitzu konkretu batzuei besterik ez die eragiten. Askotan oharkabean gelditu nahian egiten dira; eraso isilak dira kasu horretan. Zaila izan daiteke eraso hauek atzematea. Horretarako sarea zelatatzeko tresnak behar dira.

Beste iritzi baten arabera, zerbitzuen konfigurazioaren kontrako erasoak honako bi talde hauetan sailka daitezke:

– Sare-zerbitzuen kontrako erasoak: kaltegarrienak dira, sareak ematen dituen erabiltzaile-zerbitzu guztiei eragiten baitiete. Adibidez, DNS zerbitzarien konfigurazioen kontrako erasoak edota bideratzaileen konfigurazioaren kontrakoak (bideratze-taulen aldaketak).

– Aplikazio konkretu baten zerbitzarien konfigurazioaren kontrako erasoak. Beronen adibideak web zerbitzarien edo posta-zerbitzarien kontrako erasoak dira.

2. Informazioaren kontrako erasoak.

Eraso hauek zerbitzarien edota erabiltzaileen ekipoen kontra gerta daitezke. Ondoko bi eratakoak izan daitezke:

- Informazio-lapurreta (espioitza). Ondoko bi uneotan egin daitezke:
  – Informazioa gordetzen duen konputagailutik bertatik datuak lapurtu, sarearen bidez.
  – Informazioa sarean zehar bidaltzen denean lapurtu (*sniffing*).

  Bi kasuetarako defentsarik onena sarrera-kontrol egokia edukitzea (bai sarerako, baita ekipoetarako sarrera-kontrola ere) gehi kriptografiaren erabilera da. Informazio-lapurreta gehienak beste eraso motaren baten aurretiko pausoak dira (urrutiko kontrola hartzea, informazioa aldatzea, zerbitzuaren kontrako erasoak…). Horrelakoak dira *ping* masiboa, portu-eskaneatzea, edo izenak eta pasahitzen bila egindako sare-miaketa.
- Informazioa aldatzea. Aldatutako informazioa publikoa izan daiteke (web zerbitzari batek argitaratzen duena, adibidez) edo ez (enpresa baten datu-basea, adibidez). Berriz, erasoaren helburua izan daiteke informazioaren aldaketa begi-bistakoa izatea (web orriaren kasua), eta orduan errazak dira atzemateko, edo guztiz kontrakoa (kontu korronte baten saldoa edo ikasle baten nota aldatzea, adibidez).

  Aurreko kasuan bezala, informazioaren aldaketak ere honako bi talde hauetan sailka ditzakegu: sare-aplikazioei (web orriarena, adibidez), eta

sare-zerbitzuei (DNS erregistroen edukia aldatzen dutenak, adibidez) eragiten dietenak.

Askotan, informazioaren aldaketarik okerrena informazioa hori ezabatzen duena da.

3. Ekipoen urrutiko kontrola lortzea.

   Eraso hauen helburua erasotako makina beste baten kontrako erasorako abiapuntu gisa erabiltzea izaten da (zonbiak), erasotzailearen identifikazioa zailagoa egiteko asmoz. Kasu honetan, erasoek isilak izatea bilatzen dute, hau da, ahal den gehiena atzeratu nahi dute erasoa atzematea. Informazioaren kontrako erasoetan bezala, zerbitzarien edo erabiltzaileen konputagailuen kontrakoak izan daitezke eraso hauek, baina askoz ugariagoak dira erabiltzaileen ekipoen kontrakoak, askoz errazagoa baita babesik gabeko erabiltzaile-ekipo bat sarean aurkitzea, egoera horretan dagoen zerbitzari bat topatzea baino.

5.1.3. Segurtasun-gabeziaren iturriak

Behin sare-segurtasunerako arriskuak badaudela onartuta, azter dezagun zein diren arrisku horiek benetako segurtasun-arazo bihurtzea errazten duten baldintzak. Honako hiru bide hauek handitzen dute gure sareko segurtasun-gabezia:

- Ahulezia teknologikoak.

  Termino honek aplikazioen eta sistema eragileen softwarean etengabe agertzen diren *segurtasun-zuloak* erreferentziatzen ditu. Software-akats hauek saihestezinak dira: ez dago software bat % 100ean segurua dela esaterik, edozein unetan horren kontra software horrek babesik ez duen eraso bat sor daitekeelako, besteak beste. Horregatik, softwarearen ahulezien kontrako defentsarik onena kode irekiko softwarea erabiltzea da, bere kodea erarik sakonenean aztertua izaten delako. Gainera, ezinbestekoa da softwarearen eguneraketak egitea, argitaratzen diren konponketak (adabakiak) instalatuz.

- Segurtasun gabeko sare-diseinuak.

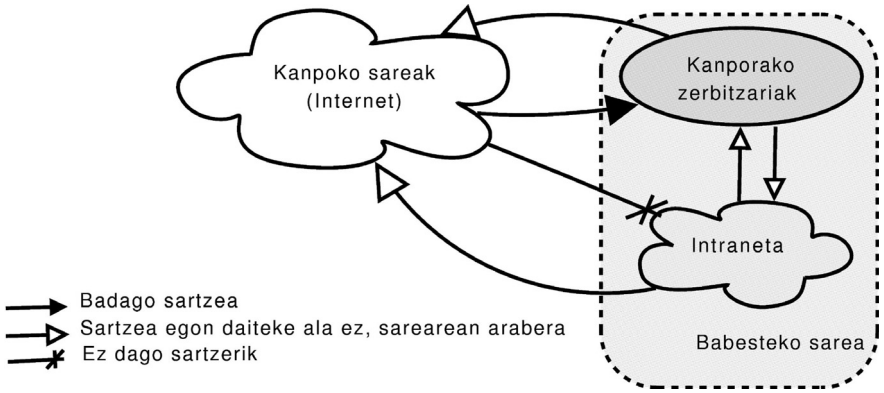
  Diseinua sare-segurtasuneko aldeak kontuan hartu gabe egiten denean gertatzen da. Zehazki, sare bat diseinatzean, kontuan hartu behar dira sarrera-kontrolerako aferak, sare-monitorizaziokoak, baita komunikazioen konfidentzialtasun-beharrak ere.

- Segurtasunaren kudeaketa txarra.

  Sare askotan, segurtasunaren kudeaketa arazoak albait hoberen konpontzea besterik ez da (*a posteriori* lana). Hau da, ez dago segurtasunaren inongo planifikaziorik, erasotzaileen lana dezente erraztuz.

5.2. SARRERA-KONTROLA

Sare batean dauden konputagailuak bi taldetan bana ditzakegu: kanpotik atzigarri egon behar dutenak, eta egon behar ez dutenak. Lehenengo taldean egoten dira, adibidez, posta elektronikoa jaso behar dutenak, DNS jatorrizko zerbitzariak, edo kanporako web zerbitzariak. Bigarren taldekoak barruko zerbitzariak eta erabiltzaileen konputagailuak izaten dira. Bigarren talde horri *intranet* izena ematen zaio[25]. Hiru mundu horien arteko atzigarritasuna 5.1. irudian duzu. Muga horiek zaintzeko teknikek osatzen dute sarrera-kontrola. Hau da, sarrera-kontrolak zaindu behar du Interneten eta gure konputagailu talde hauen arteko trafikoa baimendua dela, baita gure bi konputagailu taldeen artekoa ere.



5.1. irudia. Gure sarerako atzigarritasuna. Babesteko sarea, gehienetan, gure sare pribatua da, eta, kasu horretan, kanpoko sarea Internet izango da. Hala ere, daitekeena babestu behar den sarea gure sare pribatuko zati bat besterik ez izatea.

Sarrera-kontrola irudiko mugetan kokatzen diren konputagailuek egin behar dituzten ondoko bi zereginetan datza:

- Alde batetik, trafikoa miatu behar da, tartean trafiko maltzurra ez dela ezkutatzen bermatzeko. Hau da, trafikoa iragazi egin behar da. Iragazketalan hau **suhesi**ek[26] (*firewall*) egiten dute.
- Beste alde batetik, urrutiko konputagailuak intranetean sartu ahal izateko, egin beharko den urrutiko konexioa kautotu egin beharko da. Lan hori RAS zerbitzariek egiten dute (Remote Access Server). Zerbitzari horiek telefono bidezko konexioetarako asmatu ziren, baina gaur egun Internetetik datozen konexioetarako ere erabiltzen dira. Ohar zaitez urrutiko konexio horiek 5.1. irudiko eskemarako salbuespentzat har daitezkeela.

25. Intranet bat, oro har, TCP/IP teknologia erabiltzen duen sare pribatu bat da. Hala ere, batzuek, barruko erabilerarako soilik den web zerbitzariari deitzen diote intranet.

26. Euskaraz erabiltzen den *firewall* hitzaren beste itzulpena *suebaki* da.

Suhesiak eta RAS zerbitzariak lan horietarako espresuki jarritako makinak izan daitezke, baina, gaur egun, lan horiek sareen artean kokatzen diren bideratzaileek egiten dituzte gehienetan.

5.2.1. Suhesiak

Ingelesezko *firewall* terminoaren erabilera ez dago estandarizatua, hau da, errealitate desberdinak (eta, batzuetan, oso desberdinak) izendatzeko erabiltzen da. Hemen erabiltzen dugun definizioa honako hau da: bi sareen arteko trafikoak halabeharrez zeharkatu behar duen makina bat da suhesia, eta makina horretan trafikoa aztertzen duen softwarea egikaritzen da.

Suhesiak sailkatzeko irizpide asko daude. Guk bi talde bereiziko ditugu:

- Pakete-iragazkiak.

  IP eta garraio-mailako iragazketa-lanak konbinatzen dituzte. IP mailako iragazketarako datagramaren iturburuko eta helburuko helbideak erabiltzen dira nagusiki. Horrela kontrola daiteke babestutako sareko zein konputagailutarako sarrera gaituko den, baita zein konputagailutarako trafikoa atera daitekeen gure saretik ere. Protokoloaren identifikadorea ere miatzen bada, badago konputagailu baterako trafikoa era zehatzagoan iragaztea. Adibidez, badago ICMP trafikoa blokeatzea, baina TCP eta UDP trafikoa onartzea.

  Aurreko kontrolekin batera, badago garraio-mailako kontrolak ezartzea, iragazketa are meheagoa egiteko. Garraio-mailako iragazketa hori iturburuko eta helburuko portuen arabera egiten da, eta estatikoa edo dinamikoa izan daiteke. Estatikoa aplikazioei egindako portu-esleipenean datza, hau da, portu erreserbatuen balioan. Balio du gure sareko konputagailu baten aplikazio batzuk besterik ez izateko atzigarriak, edo jakiteko zein zerbitzu eska daitezkeen gure sareko konputagailuetatik.

  Garraio-mailako iragazketa dinamikoari egoera-iragazketa ere deitzen zaio. Gure sareko bezeroek sortzen duten trafikoa kontrolatzeko balio du iragazketa honek, eta bezero bakoitzak ezartzen dituen TCP konexioetan erabilitako portuen jarraipena egiten du. Gogoratu bezeroek edozein portu, libreen artean, erabil dezaketela konexio bat ezartzean. Inongo jarraipenik egiten ez bada, gure bezeroen edozein portutara igorritako trafikoa utzi beharko litzateke barrura igarotzen, eta edozein portutatik bidalitakoa ateratzen utzi. Egoera-iragazkiak erabiltzen badira, ordea, TCP konexioetan erabiltzen ari diren portuetarako trafikoa soilik onartzen da barrura igarotzeko, eta portu horietatik bidalitakoa besterik ez da ateratzen utziko.

- Aplikazioko proxiak eta pasabideak.

  Trafikoa aplikazio-mailan aztertzen dute hauek. Azterketa hori aplikazio-mailako goiburukoaren eremu batzuen balioaren araberako iragazketa hutsa baino gehiago izaten da. Komunikazioko bi muturretakoren bat ordezkatzen dute suhesi hauek: proxi izenekoak bezeroaren eta zerbitzarien artean kokatzen dira, eta pasabideak bi zerbitzarien artean.

  Proxiek kanpoan kokatutako zerbitzarietara bezeroek sortutako trafikoa bidean atzematen dute, eta ordezkatzen dute. Hau da, proxiak bezeroarena egingo du zerbitzariaren aurrean, eta zerbitzariarena bezeroaren aurrean (hala ere, proxiaren lana ezkutuan gelditzen da erabiltzailearentzat). Ez dute lan egiten aplikazio-mailan bakarrik, TCP/IP goiko hiru mailetan baizik. Honako bi eratako proxiak aurkituko ditugu:

  – Aplikazio batenak. Kasu honetan proxiak aplikazio baten trafikoa besterik ez du atzematen. Oso erabiliak dira dagoneko ezagutzen ditugun web proxiak. Aurreko kapituluan proxi horien cache lana ikusi dugu, baina, horrez gain, proxiaren bidez badago murriztea gure sareko zein konputagailutatik atzitu daitekeen kanpoko zerbitzari edo web orri bat.

  – Orokorrak. Hauek aplikazio askotako trafikoa (agian, aplikazio guztietakoa) atzematen dute. Atzemandako trafikoari ezarritako tratamenduak edozein aplikaziotarako baliagarria izan behar duenez, bere ahalmena mugatua da, aplikazio bakoitzak bere segurtasunerako berezko beharrak baititu. Izan ere, proxi hauek ez dute benetako iragazketa egiten aplikazio mailan (bai, ordea, iragaz dezakete IP eta garraio-mailan), eta bere lana RAS (kautotzea) eta VPN zerbitzariarena (gero ikusiko dugunez, zifratzea) egitea da. Oso ezaguna eta erabilia den proxi orokorra SOCKS izenekoa da. Internet estandar izateko proposatuta dagoena (RFC 1928). Esanguratsua da SOCKSen bertsio baten izena SOCKSVPN izatea.

  Pasabideak proxiak bezalakoak dira, baina bi zerbitzariren arteko komunikazioetan, horietako bat ordezkatzen dute[27]. Beraien erabileraren adibidea postarako pasabideak dira.

Bideratzaile/suhesiak sareko mugan kokatzen direnez, IP/garraio/aplikazio iragazketaz gain honako lan hauek ere egin ditzakete (eta, askotan, egiten dituzte) gaur egun:

- DHCP/NAT zerbitzariarena. Kontuan izan NAT zerbitzari baten lana, azken finean, IP mailako proxi batena dela.
- RAS zerbitzaria. Hau da, urrutiko erabiltzaileen intraneterako sarrera-kontrola egiten dute askotan suhesiek.

27. Berriz ere, termino honen erabilera estandarizatua ez dagoenez, «pasabide» hitza aurki dezakegu hemen definitu dugunaz bestelako kontzeptu edo teknikak izendatzeko.

- VPN zerbitzaria. Gero ikusiko dugunez, honek gure intranetaren mugak gure sare fisikotik harago hedatzen ditu.
- IDS zerbitzaria (Intrusion Detection System). Sistema hauek sarean dabilen trafikoa arakatzen dute, gure sarean baimenik gabeko sarrerak atzemateko nahian. Trafiko-miaketa hori iragazketarekin oso erraz integratzen denez, ez da harritzekoa makina berak bi lanak egitea.
- SNMP kudeaketarako agentea. SNMP protokoloan (Simple Network Management Protocol) oinarritutako sare-kudeaketarako aplikazioak erabiltzea ahalbidetzen du honek.

5.2.2. Mugasareak

Gure sarerako sarrera babesteko modurik xumeena gure sarearen eta kanpoaldearen artean suhesi bat kokatzea da. Baina kanpotik atzigarriak diren zerbitzariak baditugu sarean, hori ez da nahikoa: horietako zerbitzari baten kontrako erasoren batek arrakasta lortzen badu, intraneta osoa dago arriskuan. Hortik dator 5.1. irudiko intraneta eta kanpoko zerbitzariak bilduko dituen sareak bereizteko ideia.

Mugasare bat babestu nahi dugun sarearen eta kanpoaldearen artean kokatzen den sare lokal bat da. Hau da, intranetaren eta kanpoko sarearen (normalki, Internet) artean kokatuko dugun sare lokala. Sare lokal horretan, suhesiek babestuta, kanpotik atzigarri egon behar duten konputagailuak soilik kokatu behar dira, 5.2. irudian agertzen den moduan. Bereizketa honi esker, intranetean sartzen den trafikoa kontrolatzea errazagoa izango da, baita mugasareko zerbitzari horiek babestea ere. Orain, kanpoko zerbitzarien kontrako eraso batek ez du zuzenean arriskuan jarriko intraneta. Beste alde batetik, gure intranetean sortutako erasoen kontrako babesa ere eskaintzen du mugasareak.

5.2. irudia. Mugasare arrunta.

Mugasarean kokatutako konputagailuetarako sarrera-kontrola zerbitzari horietan bertan egingo da, suhesiek sareko sarrera-irteerekin egiten duten modu berean. Izan ere, mugasareko zerbitzarietan kontrol hori egiten duen softwarea eta suhesietan egiten duena, baliokideak izaten dira. Horregatik, zerbitzarietan instalatutako trafikoa kontrolatzeko softwareari suhesi lokala deitzen zaio, eta horrelako softwarea erabiltzen duen zerbitzariari, gotorlekua (*bastion* ingelesez). Aipatzekoa da mugasareak deitzeko DMZ sigla ere erabiltzen dela maiz (DeMilitarized Zone).

Azpimarratu behar da mugasareak gure sarera sartzen den trafikoa kontrolatzeaz gain, gure saretik ateratzen dena behatzeko ere balio duela. Zonbi-konputagailuez egindako erasoen garrantzia handitzen den heinean, gure sarean sortutako segurtasun-arazoei arreta eskaintzeko beharra ere handitu egin da.

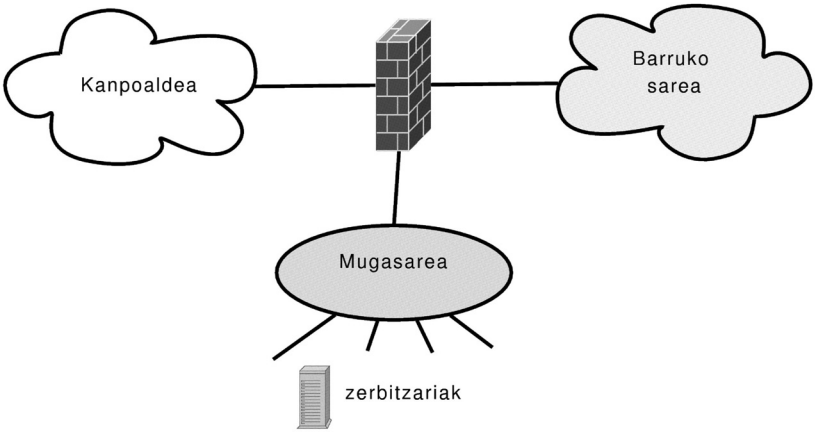
Beste alde batetik, ohartu mugasareen erabilera ez dela gure sarea Internetetik isolatzea bakarrik. Gure sare barruan ere, azpisareen arteko mugasareak jar daitezke.

Mugasareko topologiak

Zenbait aukera topologiko ditugu mugasare bat osatzeko. Ohikoenak ondokoak dira:

- Suhesi bakarreko mugasarea.

  Hau aukerarik sinpleena eta merkeena da. 5.3. irudian duguna da. Suhesia hiru saretarako lotunea da. Ondoko bi desabantailak ditu: suhesia gune kritikoa da (kale egiten badu, hiru sareak guztiz deskonektatuta gelditukodira), eta bere konfigurazioa beste aukerena baino zailagoa da.



5.3. irudia. Suhesi bakarreko mugasarea.

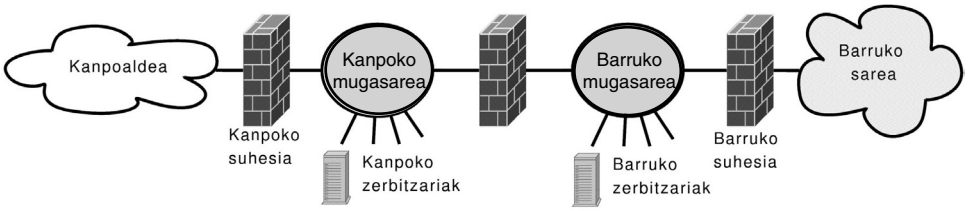
- Mugasare arrunta.

  5.2. irudian duguna da. Aurrekoarekin alderatuta, sendoagoa da suhesian arazoak sortzen direnean. Kanpoko suhesian arazorik badago, barruko erabiltzaileek ez dute atzemango zerbitzariekin lanean ari direnean. Barruko suhesiak baldin baditu arazoak, zerbitzariekin lan egiten duten kanpoko erabiltzaileek ez dute atzemango.

- Mugasare bikoitza.

  Hau egokia da barruan sor daitezkeen arriskuak kontuan hartzekoak badira. Topologia honetan honako bi talde hauetan bereizten dira zerbitzariak:

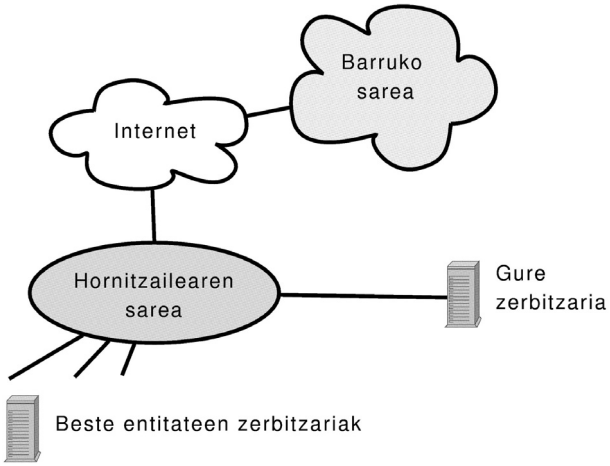
kanpotik atzigarriak izango direnak, eta barrutik atzigarriak direnak. Horietako talde bakoitzeko mugasare bat sortuko dugu. Hori da 5.4. irudian agertzen zaiguna.



5.4. irudia. Mugasare bikoitza.

Topologia horietan guztietan, suhesien betebehar nagusia trafikoaren iragazketa egitea da, baina, gainera, askotan DHCP/NAT zerbitzariarena eta VPN/RAS zerbitzariarena ere egiten dute.

Kanpotik atzigarriak izan behar duten zerbitzariak isolatzeko beste aukera bat, mugasarearenaz gain, zerbitzari horiek kanporatzea da (*outsourcing*). Web zerbitzarien kasuan oso erabilia da aukera hau, 5.5. irudian duguna. Aukera honen abantailarik handiena da zerbitzarien segurtasunaz arduratzeari uzten diogula. Horrek aurrezte handia eragin dezake, ekipoetan eta langileetan kendutako inbertsioengatik. Aurka, gure zerbitzarien kontrola eta gordetzen duten informazioa beste entitate baten esku uztea du.



5.5. irudia. Outsourcing.

5.3. SEGURTASUNAREN KUDEAKETA

Hurrengo bi agiri multzo hauek dira segurtasunaren oinarriak:

- Sareko segurtasunerako arauak (*network security policies*).

  Agiri hauetan honakoak definitzen dira:

  – Zer babestu behar den, eta zehazterik badago, zertaz babestu behar den.

  – Norberaren ardurak (erabiltzaileak, sare-teknikariak, eta erakundeko kudeatzaileak).

  – Norberaren portaera, oro har.

  Arau multzo honi *sareko segurtasunerako politikak* izena ere ematen zaio. Testu honetan *politika* baino, *araua* edo, beharbada, *agiria* hitza erabiliko dugu, ingelesezko *policy* hitzaren itzulpen literala ez baita egokia testuinguru honetarako.

- Sareko segurtasunerako prozedurak.

  Aurreko arauak betetzeko eta, horrela, sareko segurtasuna bermatzeko zer, nola, eta noiz egin behar den zehazten duen beste agiri multzo bat da hau.

5.3.1. Sareko segurtasunerako arauak

Arau multzo hau osatzen duen agiri zerrenda eta agiri bakoitzaren izaera (nahitaezkoa edo gomendioa) oso aldakorra da, sare bakoitzaren ezaugarrien eta beharren araberakoa baita. Hala ere, duten garrantziagatik, honako hauek agertzen dira normalki:

- Sareko deskribapena. Hau beste agirietarako oinarria izaten da. Sarea eta sarea erabiltzen eta kudeatzen duen erakundea deskribatzen ditu. Sarearen topologiaz gain, sareak ematen dituen zerbitzuak eta atzematen diren segurtasun-beharrak adierazten dira.
- Sareko kudeaketarako, oro har, eta zehazki sareko segurtasunerako arduradunen definizioa.
- Informazioaren babeserako arauak. Sarean atzigarri dagoen informazioa sailkatzen du agiri honek, bere babeserako beharraren arabera.
- Sarea erabiltzeko arau orokorrak. Gomendatzekoa da erabiltzaileei agiri hau sinaraztea, beronen edukia ezagutzen dutela bermatzearren.
- Kanpotik sarean sartzeko arauak. Agiri honek nor sar daitekeen kanpotik gure intranetean, eta zein baldintzatan, definitzen du.
- Birusen kontrako arauak. Ondokoak, adibidez, definitzen dira agiri honetan:
  – Birusek gu ez kutsatzeko, eta infekzioak ez zabaltzeko portaera orokorrak.

  - Erabili behar den birusen kontrako softwarearen definizioa, baita horren eguneraketarako eta birusak atzemateko txantiloien eguneraketarako maiztasunarena ere.
  - Erabili behar diren sistema eragileen eta aplikazioen definizioa.
- Segurtasun-kopietarako arauak. Honakoak definitu behar dira, bereziki:
  - Zein edukiren kopia egin behar den. Hau zehazteko, sare eta datuen erabileraz eta beharraz gain, alde legalak hartu behar dira kontuan (adibidez, Hego Euskal Herriko ISPen kasuan, *Ley de Servicios de la Sociedad de Información y Comercio Electrónico* izeneko legea).
  - Kopien maiztasuna eta izaera (osoa edo aldaketena soilik).
  - Nor den kopiak egitearen arduraduna.

Askotan agertzen diren beste agiriak honako hauek dira:

- Pasahitzei buruzko arauak.
- Haririk gabeko sareetan aritzeko arauak.
- Informazioa zifratzeko arauak.
- Aplikazio bakoitzeko segurtasunerako arauak.
- VPNei buruzko arauak.
- Erregistro-fitxategiei (*log files*) dagozkien arauak.

Honako eragile hauei dagozkie segurtasun-arauak:

- Sareko erabiltzaileak. Oso bestelakoak izan daitezke hauen profilak. Batzuek prestakuntza sakona izango dute informatikan eta segurtasunean, eta beste batzuek ez dute bat ere izango.
- Sareko ustiaketaren eta kudeaketaren arduradunak. Hauek beti izan beharko lukete teknikari informatikoak.
- Sareko jabea den entitatearen kudeatzaileak. Normalki sareko erabiltzaileak ere badira aldi berean, baina jabeak direnez, haien papera desberdina da: sareko segurtasunaren alde askoren definizioa beraiek egin beharko dute, betiere informatikariek, sareko diseinatzaileek eta lege-aholkulariek aholkatuta.
- Lege-aholkulariak. Hauen lana garrantzitsua izaten da arauak definitzean eta ezartzean.

Hala ere, argi dago eragile hauek guztiak ez direla agertzen kasu guztietan. Erakunde txikien kasuan, agian aktore berberak jokatu beharko ditu paper guztiak.

Segurtasun-arauek sortzen dituzten jarrerak desberdinak izan daitezke eragile bakoitzarengan, baina oso ohikoa da, bereziki sareko erabiltzaileen artean,

mesfidantza, goganbeharrak, eszeptizismoa, eta, oro har, jarrera negatiboak. Askotan egiten diren kritikak honako hauek dira.

- Sarean egindako lanerako oztopotzat hartzen dira. Horregatik garrantzitsua da ahal dela arau minimoak ezartzea, eta, batez ere, ondo orekatu behar dira sareko segurtasun-beharrak eta sareko erabilgarritasuna. Histeria eta paranoia baztertu behar dira, baina utzikeria ere bai.
- Mesfidantza eta tentsioa sortzen duten gehiegizko kontrolerako mekanismoak izatea leporatzen zaie. Hau saihesteko edo murrizteko, aurreko bi gomendioak azpimarratu behar dira: arau minimoak ezarri, eta orekatu segurtasuna eta erabilgarritasuna.
- Baliagarritasuna kolokan jartzen dute. Hau ekiditeko, benetako beharrei egokituak izan behar dute arauek, eragile guztiak kontuan hartuta. Eta, oroz gain, egingarriak izan behar dute.

Laburbilduz, kritikak ekiditeko, segurtasun-arauek izan behar dute:

- Egingarriak, bai gauzatzeko bai gauzatzen direla egiaztatu ahal izateko.
- Ulertzeko errazak.
- Segurtasunaren eta eraginkortasunaren arteko orekari eutsi behar diote.

Gainera, arauen eragin nekagarria atzemango dutenek ezagutu behar dituzte haien zergatiak, beharrezkotasuna, eta dakarten onura.

Estandarrak

Badaude segurtasun-arauak definitzeko laguntza edo gida moduan erabiltzeko estandar eta agiri batzuk. Ondokoak dira aipatzekoak:

- RFC 2196. Internetekin lotutako sareetako segurtasunerako arauak eta prozedurak definitzeko zertxobait zaharkituta (1997) gelditu den gida bat da.
- ISO/IEC 27000 estandar-saila. Informazio-sistemetarako segurtasunari buruzko oso agiri multzo zabala da hau. Garrantzitsuenak honako hauek dira:
  - ISO/IEC 27001: ISO segurtasun-ziurtagiria lortzeko behar diren betebeharrak zehazten ditu, ISOk jaulkitako beste ziurtagirietarako bezala (adibidez, hain ezaguna den ISO 9001 kalitate-ziurtagiria).
  - ISO/IEC 27002: 2007ko uztailean ISO/IEC 17799 araua berrizendatzeko hartutako kodea da. 27001 ziurtagiria lortzeko exijitzen diren gomendioak biltzen dituen estandarra da.

Segurtasun-arauak erredaktatzeko zenbait adibide eta laguntza aurki daitezke http://www.sans.org/resources/policies gunean.

Sareko segurtasunerako arauak sortzeko prozedura

Proposamen bat honako urrats hauei jarraitzea da:

1. Segurtasun-arauak egin beharko dituen taldea osatzea.

   Talde horretan egon beharko dute, gutxienez, ondokoek:

   – Sareko jabea den entitatearen kudeatzaileetako bat.
   – Sarea kudeatzen duten teknikarietako bat.
   – Erabiltzaileen ordezkari bat.
   – Legelari bat.
   – Agiri bat erredaktatzen dakien norbait.

2. Definitu behar diren arauak zehaztu.

   Oro har, hobe dugu agiri asko eta laburrak sortzea, gutxi eta lodiak baino. Kontuan hartu behar da saihestezina izango dela hainbat gai agiri batean baino gehiagotan agertzea.

3. Zirriborroak sortu. Agiri bakoitzak eragiten dienen artean aukeratutako talde bati helarazi behar zaio agiri horren kopia, azter eta ebalua dezan.
4. Arauen berrikuspena egin, zirriborroetatik eta eragindakoek egindako iradokizunak eta eskaerak kontuan hartuz.
5. Arauak jakinarazi arauek eragindako guztiei.
6. Laugarren eta bosgarren urratsak periodikoki errepikatu.

5.3.2. Segurtasun-prozedurak

Segurtasun-arauek ezarritakoa nola gauzatu behar den zehazten dute prozedura hauek. Askotan, arau bakoitzeko prozedura bat izango dugu (adibidez, segurtasun-kopietarako araua, eta segurtasun-kopiak egiteko prozedura), baina ez beti: lotutako prozedurarik ez duten arauak aurkituko ditugu (sareko deskribapenaren kasu), baita prozedura bat baino gehiago lotuta dituzten arauak ere.

Prozedurarik garrantzitsuenak honako hauek dira:

- Arrisku-analisia egiteko prozedura.
- Ekipoak instalatzeko eta konfiguratzeko prozedura.
- Monitorizaziorako prozedura.
- Segurtasun-arazoak sortzen direnerako jarraibideak.
- Analisi forenserako prozedura.
- Formaziorako prozedurak. Alde batetik sarean erabili behar diren segurtasun-tekniketarako trebakuntza eman behar zaie teknikariei eta erabiltzaileei, eta beste alde batetik segurtasun-arauen berri zabaldu behar da.

- Segurtasun-kopiak egiteko prozedurak.

Azter ditzagun horietako batzuk.

Arrisku-analisia egiteko prozedura

Helburua gure sareak zertarako egon behar duen prest identifikatzea da. Oro har, honako bi motatako arriskuak identifikatu behar dira: edozein sareri dagozkionak (arrisku orokorrak), eta gure sareari, espezifikoki, dagozkionak (arrisku partikularrak). Arrisku-analisia aldiro egin behar den lana da.

Arrisku-identifikazio hori egiteko erreferentzia ona CERT/CSIRT izenekoak dira (Computer Security Incident Response Team). CSIRT bat sare talde batean atzemandako segurtasun-arazoak konpontzeko urratsak koordinatu, lagundu eta abiatu egiten dituen talde bat da. Nazioartean, FIRST erakundeak (Forum of Incident Response and Security Teams, www.first.org) biltzen ditu CSIRT taldeak. Haren kideen artean munduko CSIRT talde nagusiak daude (hainbat gobernutako agentziak, industriako taldeak, irakaskuntza eta ikerkuntzarako sareak, eta abar). CSIRT batek egin behar duena RFC 2350 agiriak biltzen du. Gaur egun CSIRTek ez dute beren lana gertakari baten osteko ekintzetara mugatzen, eta lan handia egiten dute segurtasunerako trebakuntzan.

Arrisku-analisirako oso erabiliak diren bi teknika ahulezia-testa eta sarkin-testa dira. Ahulezia-testak software konkretu baten (aplikazio bat, sistema eragile bat) azterketa egiten du. Software horren konfigurazioa berrikusten dute ahulezien bila. Sarkin-testa gure sarearen kontrako erasoak guk geuk (edo horretarako kontratatutako norbaitek) abiatzea da, horretarako ezagunak diren eta normalki erabiltzen diren teknikak eta softwarea erabiliz.

Monitorizaziorako prozedurak

Prozedura hauen ardatza IDS (Intrusion Detection System) izeneko softwarearen erabilera da. Haren lana sarean gertatzen dena zelatatzea da, eraso ezagunen zantzuak bilatuz. Horretarako, ondoko bi atazak gauzatzen dituzte nagusiki: sarean dabilen trafikoa arakatu (*sniffing*), eta sareko konputagailuetako *log* fitxategiak aztertu.

Segurtasun-arazoak sortzen direnerako jarraibideak

Agiri hau aldakorra izango da, sare bakoitzaren arabera, baina, gutxienez, honako urrats hauek agertu behar dira:

1. Segurtasun-arazoaren berri eman segurtasunaren arduradunei.
2. Gertakariak eragindako konputagailuak isolatu.

3. Aztertu ea antzeko konputagailuak ere kaltetuta dauden (sare berekoak, edo zerbitzu bera ematen dutenak, adibidez).
4. Gertakariari buruzko informazioa bildu (sintomak, kaltetutako makinak…) eta gure CSIRT taldea jakinaren gainean jarri.
5. Eraso mota identifikatu. Hau errazteko CSIRTekiko elkarlana eta sareko segurtasunaren arduradunen prestakuntza eta eskarmentua garrantzitsuak izango dira.
6. Analisi forensea. Helburu bikoitza du: erasoak eragindako kaltea balioetsi, eta erasoaren jatorriari eta egileei buruzko informazioa bildu. Analisi forensea egiteko erreferentzia bat RFC 3227 agiria da.
7. Arazoa berriro gerta ez dadin neurriak hartu. Erasoaren izaeraren eta gure sareko konfigurazioaren arabera, neurri horiek oso bestelakoak izan daitezke. Hoberena litzateke konfigurazio txar batzuk zuzendu behar izatea, besterik ez (pasahitz ahul batzuk aldatu, adibidez), baina askotan software-adabakiren bat instalatu beharko da. Okerragoa izaten da software-aldaketa egitera behartuta bagaude, erabiltzen dugunak ez duelako inolaz ere hurrengo eraso baten aurreko babesa bermatzen (kasurik okerrena, sistema eragilea aldatu behar denean). Batzuetan haren segurtasuna hobetzeko sareko birdiseinua egitea ere mahaigaineratu beharko da, topologia edo ematen diren sare-zerbitzuak aldatuz. Edozein kasutan, segurtasun-arauak eta prozedurak berrikusi beharko dira.
8. Arazoak bertan behera utzitako konputagailuak edo zerbitzuak berrabiatu, eta egindako kalteak konpondu, ahal bada.
9. Sarkin-testa egin, jasotako erasoa kontuan hartuta.

5.4. KOMUNIKAZIO SEGURUAK

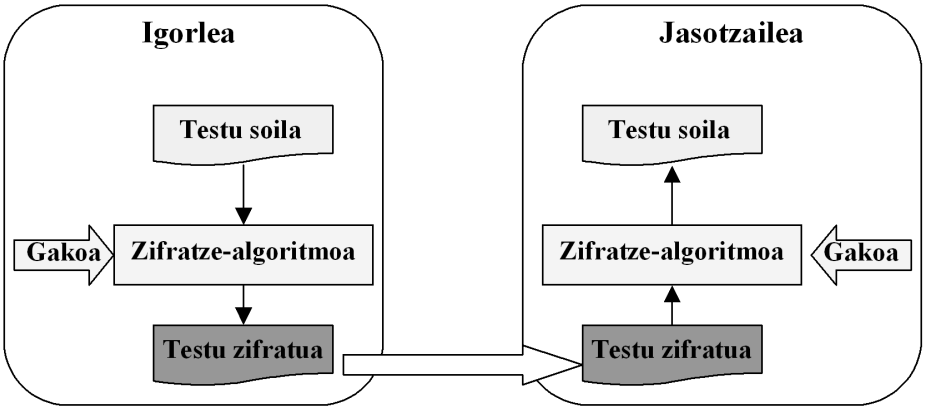
Sare-aplikazio bat segurua izateko, haren osagaien arteko komunikazioak segurua izan behar du. Horretarako, honako ezaugarri hauek izan behar ditu:

- **Konfidentzialtasuna:** igorleak eta hartzaileak izan ezik beste inork ez du ulertu behar mezuaren edukia. Horrela izanik, nahiz eta sudurluze batek mezua atzeman bidean zehar, ezingo du inongo informaziorik jaso.
- **Kautotasuna:** igorleak eta hartzaileak ez dute zalantzarik izan behar bestearen nortasunari dagokionez. Hau da, nortasun-ordezkapenak ezinezkoa izan behar du.
- **Osotasuna eta ukorik ez:** hartzaileak ziur egon behar du jasotako mezuak ez duela aldaketarik izan. Gainera, bidaltzaileak ezin du ukatu mezua berak sortu duela.

Hauek guztiak lortzeko oinarri teknikoa kriptografia da. Hurrengo ataletan banan-banan aztertuko ditugu.

5.4.1.Konfidentzialtasuna

Kriptografiako teknikek igorleak informazioa desitxuratzea ahalbidetzen dute, baten bat tartean sartu eta bidean mezua atzematen badu, inolako informazio ulergarririk lor ez dezan. Hartzaileak, aldiz, jatorrizko informazioa lortzeko gai izan behar du, noski. 5.6. irudiak kriptografia erabiltzean agertzen diren osagaiak azaltzen ditu.



5.6. irudia. Prozesu kriptografikoetan agertzen diren osagaiak.

Jatorrizko mezua **testu soila** da, ukitu gabekoa. Bidali baino lehen, igorleak mezu hori **zifratze-algoritmo** baten bidez zifratuko du. Zifratzeko prozesu horretan **zifratze-gako** bat erabiliko du. Algoritmoaren emaitza **testu zifratua** da, bidaliko duguna. Testu zifratuari **kriptograma** ere esaten zaio. Hartzaileak alderantzizko prozesua abiatu beharko du, testu zifratutik jatorrizko mezua berreskuratzeko. Horretarako algoritmo bera erabiliko du, eta dagokion **deszifratze-gakoa**. Zifratze-gakoa eta deszifratze-gakoa berdinak direnean (edota bata bestetik erraz erator daitekeenean) **kriptografia simetrikoa** erabiltzen ari gara. Bestela, **kriptografia asimetrikoa** izango da.

Kriptografia simetrikoa

Kriptografia simetrikoa antzinako kontua da. XX. mendera arte erabilitako teknikek kriptografia klasikoa osatzen dute. Honako hauek dira:

- **Zifratze monoalfabetikoa:** letra edo letra multzo bakoitza beste letra edo letra multzo batekin ordeztean datza. Ezagutzen den zifratzerik zaharrenetako bat Zesarren zifratzea da. Teknika horretan, testu zifratuaren alfabetoa k letraz mugitzen da, non k zifratze-gakoa den. Adibidez, $k = 3$ hartzen badugu, *sarea* hitza *vduhd* bihurtzen da. Euskarazko alfabetoan gakoak 26 balio posible bakarrik izan ditzakeenez, oso teknika ahula da Zesarrena.

Zesarrena baino ordezte monoalfabetiko hobea, honako hau dugu: edozein letra beste edozein letrarekin ordeztu, betiere letra bakoitzak ordezkari bakarra izanda eta alderantziz. Gakoa alfabeto osoari dagokion 27 letrako katea izango da, ondoko hau bezalakoa:

Alfabetoa	a	b	c	d	e	f	g	h	i	j	k	l	m	
Gakoa	q	w	e	r	t	y	u	i	o	p	a	s	d	
Alfabetoa	n	ñ	o	p	q	r	s	t	u	v	w	x	y	z
Gakoa	f	g	h	j	k	l	ñ	z	x	v	b	n	m	c

5.1. taula: Zifratze monoalfabetiko baten gakoa.

Aurreko gakoarekin, *sarea* mezua *ñqltq* testu zifratua bilakatuko litzateke. Argi dago sistema hau Zesarrena baino seguruagoa dela; izan ere, nahiz eta kriptoanalistak sistema orokorra ezagutu (letrak letra ordeztea), ez daki 27! gako posibleen artean zein erabiltzen ari den. Zesarren zifratuaren kasuan ez bezala, denak probatzea ez da bideragarria. Irtenbide posible bakoitzeko 1 μseg erabilita ere, konputagailu batek 10^{14} urte baino gehiago beharko lituzke gako guztiak probatzeko! Hala ere, testu soilaren hizkuntzaren analisi estatistikoak kontuan izanda, kodea haustea erraz samarra da.

- **Zifratze polialfabetikoa:** duela 500 urte asmatua, Blaise de Vigènere-ri oker egotzi izan zaio urteetan zehar, eta horregatik Vigènere-ren zifratze izenaz ere ezagutzen da. Ideia da zifratze monoalfabetikorako gako bat baino gehiago erabiltzea, eta testuan letrak agertzen diren tokiaren arabera, letra bakoitza zifratzeko zein gako erabili aukeratu. Horrela, mezu berean letra bera era ezberdinean kodetuta ager daiteke, mezuko zein lekutan dagoen, gako ezberdina erabili delako. Ondoren Vigènere zifratzearen adibide bat dugu, bi gako erabiliz, G_1 eta G_2:

Alfabetoa	a	b	c	d	e	f	g	h	i	j	k	l	m	n
G_1	q	w	e	r	t	y	u	i	o	p	a	s	d	f
G_2	f	g	h	j	k	l	ñ	z	x	v	b	n	m	c
Alfabetoa	ñ	o	p	q	r	s	t	u	v	w	x	y	z	
G_1	g	h	j	k	l	ñ	z	x	v	b	n	m	c	
G_2	q	w	e	r	t	y	u	i	o	p	a	s	d	

5.2. taula: Zifratze polialfabetiko baten gakoa.

Gakoez gain, zifratze polialfabetikoetan gakoak erabiltzeko patroia ere definitu egin behar da. Demagun aurreko adibidearen patroia $G_1G_2G_2G_1$ dela. Orduan, *sarea* mezua, *ñfttq* testu zifratua bilakatuko litzateke. Aipagarria da nola mezuaren lehenengo «a» G_2 erabiliz kodetzen den, eta bigarren «a», berriz, G_1 erabiliz.

- **Ordezte homofonikoa:** irteerako alfabetoa ez da testu soila sortzeko erabili den bera, eta gainera, sinbolo gehiago ditu. Ordezteko teknika honek ezinezkoa egiten du testu soilaren jatorrizko hizkuntzaren analisi estatistikoetan oinarritutako kriptoanalisia, sortutako kriptogramen sinbolo guztiek agertzeko probabilitate bera baitute. Dena dela, ordezte homofonikoak arazo ugari ditu praktikan gauzatzeko.
- **Transposizio-zifratzea:** Aurreko teknika guztiak ordezkapen-zifratzeak dira; testu normalaren ordena gordetzen dute, baina mozorrotu egiten dute. Transposizio-zifratzeak, ordea, letrak berrordenatu egiten ditu, baina mozorrotu gabe. Beharbada, ezagutzen den transposizio-zifratzerik zaharrena *escitalo* izeneko makilarena da. Horren baliokidea, papera eta arkatza besterik behar ez duena, teknika hau da: testua N zutabe duen taula batean idatzi eta, testu zifratua sortzeko, letrak zutabeka hartu (ez errenkadaka), eta gainera ez hartu zutabeak ezkerretik eskuinera, baizik eta gako batek adierazitako hurrenkeran. 5.3. taulan dugu horren adibide bat. Jatorrizko esaldia «Gaur zure etxean elkartuko gara.» da. Gakoa 83425176 bada, testu zifratua ondoko hau izango da: «rnu.uel rtkga eo xaaGe kuatazerr».

1	2	3	4	5	6	7	8
G	a	u	r		z	u	r
e		e	t	x	e	a	n
	e	l	k	a	r	t	u
k	o		g	a	r	a	.

5.3. taula. Transposizio-zifratze baten adibidea.

Kriptografia modernoa konputagailuen erabilerarekin batera abiatzen da. Kriptografia klasikoaren oinarrizko ideia berberak dauzka, transposizioa eta ordezkapena alegia, baina konputagailuen erabilerak askoz algoritmo konplexuagoak erabiltzea ahalbidetu du.

Kriptografia simetrikoari dagokionez, teknika modernoetan bi multzo bereiziko ditugu: zatikako zifratzea eta fluxuzko zifratzea. Zatika zifratzea jatorrizko testua zatitu eta zati bakoitza bere aldetik zifratzea da. Ondoko hauek dira zatikako algoritmoak; gero aurkeztuko ditugu fluxuzkoak.

- DES algoritmoa (Data Encription Standard).

  AEBko NSAk garatua (National Security Agency*)*, 70eko hamarkadaren hasieran IBMk sortutako LUCIFER algoritmoan oinarrituta. 1977ko urtarrilean, Estatu Batuetako gobernuak informazio ez-sekreturako estandar ofizialtzat hartu zuen eta, laster, industriak berea egin zuen segurtasun-produktuetan erabiltzeko; hortik gutxira gehien erabiltzen zen algoritmo simetrikoa bilakatu zen. 1998. urtera arte, nahiko sendo agertu da, baina

data horretan indar hutsaren bidez eta diru gehiegi xahutu gabe apurtzea posible dela frogatu zen. Dena dela, ahulezia ez dago algoritmoan bertan, jatorrizko forman erabiltzen duen gakoaren luzeran baizik, motzegia baita. Beraz, jadanik ez da segurua jatorrizko forman, baina bere aldaera sendoagoa erabil daiteke, ondoan aurkeztuko duguna.

Ez dugu algoritmoaren deskribapen zehatza eta osoa egingo, baina algoritmo klasikoekin alderatuta teknika modernoen konplexutasunaren ideia egitearren, haren oinarriak azalduko ditugu. Jatorrizko mezua 64 biteko puskatan zatitzen du, eta puska horietako bakoitza 56 biteko gako batekin zifratzen da (hau da DES algoritmoaren jatorrizko bertsioan motzegia den gakoa). Gakoaren bidezko zifratze horri ekin baino lehen, aurreko zati zifratuarekin XOR funtzioa egiten zaio zifratu behar den zatiari. Gero, benetako zifratzea dator. Hasteko, zatiaren 64 bit permutatzen dira (transposizioa); gero, 16 aldiz zifratzen da horren emaitza, 56 biteko gakoan oinarriturik sortutako 48 biteko 16 azpigako erabiliz (ordezkapenak), eta horien 16 zifratze bakoitza baino lehenago zatiaren bi erdien arteko beste transposizio erraza egiten da; bukatzeko, hasierakoa bezalako beste permutazio bat ezartzen zaio sortutako 64 biteko blokeari.

- DES hirukoitza (3DES).

  56 biteko DES ahultzat jotzen bada, algoritmoa hainbat aldiz egikaritu daiteke gako desberdinak erabiliz. Zati bakoitzeko 64 biteko irteera hurrengo egikaritzapenaren sarrera izango da. DES hirukoitzean hiru aldiz egikaritzen da algoritmoa, bi gako erabiliz, ondoko eskema honi jarraituz:

  $$\text{Zati zifratua} = Z_{g1}\,(Z_{g2}{}^{-1}\,(Z_{g1}\,(\text{jatorrizko zatia})))$$

  Hau da, g1 gakoarekin zifratzen da jatorrizko zati bakoitza, emaitza g2 gakoa erabiliz deszifratzen da, eta horren emaitza g1 gakoarekin zifratzen da berriro. Hau guztia 112 biteko gakoa (56 + 56) erabiltzearen baliokidea da. 3DES PPP protokoloan erabiltzeko proposaturiko zifratze estandarra da (RFC 2420).

  3DESek duen arazoa azkartasuna da: algoritmoa 3 aldiz egikaritzeak denbora asko kontsumitzen du. Horregatik beharrezkoa izango da beste estandar bat.

- IDEA algoritmoa (International Data Description Algorithm).

  1991. urtean deskribatu zuten, DES algoritmoaren alternatiba gisa. Segurutzat jotzen dute, ez baitzaio eraso bideragarririk aurkitu. Bere oinarriak DESen antzekoak dira, transposizioak eta ordezkapenak egiten baititu jatorrizko mezua zatitu eta gero. Zatiak, berriro ere, 64 bitekoak dira, baina gakoak 128 bit ditu. Patentatuta dago 2010-11 urte arte, baina dirua irabazteko asmorik gabe erabiltzea badago. Gaur egun azkarragoak diren algoritmo seguruak ditugu.

- AES (Advanced Encrytion Standard)

  DES ordezkatzeko garatutako estandar hau zifratze simetrikoetan gehien erabiltzen dena da. Bere indarra ez datza bere ezaugarri matematikoetan bakarrik, baizik eta sorrerarako jarraitutako prozesuan ere, konfiantza handia sortu baitu algoritmo honetan. Honen garrantzia ulertzeko aurreko estandar kriptografikoen historia ezagutu behar da. Historia horren erakusle ona DES algoritmoa da. Lehen aipatu dugu AEBko NSAk garatu zuela DES estandarra, IBMren LUCIFER algoritmoan oinarriturik. Bada, algoritmoaren bi bertsioen artean alderik handiena gakoaren luzeran dago: IBMren jatorrizko bertsioaren gakoa 128 biteko luzerakoa zen, eta DESen gakoa, aldiz, 56koa bakarrik. Zergatik? Batzuen arabera, garai hartan, luzera horrekin posible zen NSArentzat, ez beste inorentzat, algoritmoa apurtzea, hau da, DESen bidez zifratutakoa deszifratzea. Kontuan izan NSA dela munduan zehar baliabide kriptologiko (gizakiak eta dirua) gehien duen erakundea. Izan ere, 1998ra arte DES «nahiko» segurutzat hartu izan da, nahiz eta 1977. urtean bertan, bere sorreraren urtean, Stanford-eko Unibertsitatean erakutsi zuten, teorikoki, algoritmoa apur zitekeela. Praktikan garestiegia zen… NSArentzat izan ezik? 1998. urteaz geroztik, argi gelditu da DES apurtzea jadanik ez dela hain garestia.

  AES sortzeko bidea oso bestelakoa izan da, jakinda AEBk ateratako edozein estandar susmagarria izango zela, eta horregatik, inork ez erabiltzeko arriskua zegoela. 1997ko urtarrilean deialdi publiko bat egin zuen AEBko NIST gobernu-erakundeak (National Institute of Standards and Technology) estandar berri baten proposamenak aurkezteko. Proposamenek bete behar zituzten baldintzak honako hauek ziren:

  – Algoritmo simetrikoa eta zatikakoa izatea.
  – Diseinua publikoa izatea.
  – 128, 192, eta 256 biteko gakoak erabiltzea.
  – Hardware eta softwarearen bidez inplementatzea bideragarria izatea.
  – Algoritmoaren erabilera publikoa izatea, lizentziarik gabea.

  Hiru urte igarota, 2000ko urrian, Rijndael algoritmoak irabazi zuen lehia (‘reindal’ ahoskatzen omen da, gutxi gorabehera). Algoritmo horretan oinarritu zen urtebete geroago, 2001eko azaroan, argitaratu zen AES estandarra.

  AES estandarrean 128 biteko zatiak erabiltzen dira. Nahiz eta gakoak 128, 192, edo 256 bitekoak izan, ez da espero 192ko aukera oso erabilia izatea.

Ikus dezagun orain kriptografia simetrikoan dugun bigarren teknika multzoa, fluxuzko zifratzea, alegia. Fluxuzko zifratzea erabiltzen duten tekniketan mezua bezain luzea den sasi-zorizko bit kate bat sortzen da, eta kate horren eta mezuaren artean XOR funtzioa ezartzen da. Horren emaitza izango da testu zifratua. Deszifratzeko, bit katea sortu behar da berriro, eta testu zifratuarekin XOR egin.

Bit katea sortzeko gako bat (hazia) eta katea sortuko duen funtzio bat behar dira. Teknika honi jarraitzen dioten algoritmoek sasi-zorizko funtzioa definitzen dute. Azkarrenak hardwarean inplementatutakoak dira, baina interesgarrienak softwarean inplementa daitezkeenak dira. Gehien erabiltzen den fluxuzko zifratzean oinarritutako algoritmoa RC4/5 da. Software kriptografikoa garatzen duen konpainia batek sortu zuen RC4 1987an; beraz, algoritmo hau erabiltzeagatik ordaindu egin behar da. Sekretupean garatua izanda, ez zen fidatzekoa. Izan ere, 1994. urtean publikoki deskribatu zen, eta laster aurkitu zitzaizkion ahuleziak. Konpainia berak ordezkoa den RC5 plazaratu du. Oraingoz, RC5 segurutzat hartu da, baina hau ere ordaintzekoa da. Hala eta guztiz ere, RC4 oso erabilia da oraindik. RC4ren arrakastaren gakoak izan dira bere inplementatzeko erraztasuna eta azkarra izatea, bai softwarean bai hardwarean.

RC4ren esportaziorako bertsio zaharra, hau da, AEBtik at erabiltzekoa, bereziki ahula da. 1999ko abendu arte, herrialde horren legeak zifratzeko teknologia armatzat jotzen zuen, eta ez zien baimentzen enpresei algoritmoak bere horretan esportatzea: beste herrietara esportatutako zifratze-softwarea «zikiratu» behar zuten. RC4ren kasuan, horrek suposatu zuen jatorrizko algoritmoak erabiltzen dituen gakoaren 128 bitetatik 88 finkatu eta publikoak izatea esportaziorako bertsioan. Hau da, benetan, RC4 algoritmoa erabiltzen duten aplikazio ugaritan 40 biteko gakoak erabiltzen dira, 2000. urtea baino lehenago ekoitzi zirenetan hain zuzen ere.

Kriptografia asimetrikoa

2.000 urtez baino gehiagoz (Zesarren zifratzearen garaitik 1970eko hamarkadara), zifratutako komunikazioetan bi alderdiek sekretu bat konpartitu behar zuten: gako simetrikoa. Horren arazoa bi alderdiek gakoa zein den nolabait adostu behar izatea da eta, horretarako, komunikazio-kanal seguru bat behar dute. Adibidez, zifratutako komunikazioa burutu baino lehen, bi alderdiak pertsonalki elkartu eta gakoa adostu dezakete. Konputagailu-sareen munduan, hala ere, bi alderdiek ezin dute elkarrekin inoiz hitz egin edo elkartu, sarea erabilita ez bada. Arazo horri aurre egiteko sortu zen kriptografia asimetrikoa.

1976an, Diffie eta Hellman-ek algoritmo bat aurkeztu zuten (*Diffie-Hellman gakoaren elkar trukaketa* izenarekin ezagutua gaur egun) aurretik adostutako gakorik gabe komunikazio segurua gauzatzeko. Algoritmo horrekin batera kriptografia asimetrikoa edo gako publikoko kriptografia sortu zen. Hauek dira kriptografia asimetrikoaren bi oinarriak:

- Bi gako desberdin erabiltzen dira, bata zifratzeko eta bestea deszifratzeko. Horregatik deitzen zaio «asimetriko» kriptografia honi.
- Kriptografia simetrikoan bezala, horietako gako bat ezkutua da (pribatua esaten zaio askotan), eta bestea, aldiz, ez. Horregatik deitzen zaio kriptografia honi «gako publikokoa» ere. Eta horregatik kriptografia simetrikoari «gako pribatuko kriptografia» ere esan izan zaio 1970eko hamarkadaz geroztik.

Gako publikoen funtzionamendua hau da. Erabiltzaile bakoitzak bere bi gakoak ditu, publikoa eta ezkutua. Gako publikoa berarekin komunikatu nahi duen guztiak ezagutu behar du. Horretaz, oraingoz, ez gara kezkatuko; suposatuko dugu hori lortzea erraza dela, adibidez, bakoitzak bere web orri pertsonalean bere gako publikoa argitaratuz. Demagun Ainhoak Beñatekin[28] komunikatu nahi duela. Horretarako Beñaten gako publikoa eskuratuko du eta, zifratze-algoritmo estandar bat erabiliz, bere mezua zifratu eta Beñati bidaliko dio. Beñatek Ainhoaren mezua jasotzen duenean bere gako ezkutua eta zifratze-algoritmo estandarizatua erabiliz mezua deszifratuko du.

Kriptografia asimetrikoan, sudurluze batek nahi dituen {testu soila, testu zifratua} bikote guztiak eskura ditzake, zifratzeko behar dituen gakoa eta algoritmoa ezagunak baitira. Bikote horien azterketatik abiatzen dira kriptoanalisi-teknikarik eraginkorrenak. Beraz, algoritmoek bereziki sendoak izan behar dute mezua-kriptograma bikotetik gako ezkutua erator ezina dela bermatzeko. Sendotasun hori lortzeko erabili behar diren gakoak oso handiak dira. Gehien erabiltzen diren algoritmoetan gomendioa da gutxienez 2.048 biteko gakoak erabiltzea. Alderatu kriptografia simetrikoak behar dituen 128 biteko gakoekin. Horren guztiaren ondorioa hau da: algoritmo asimetrikoak simetrikoak baino askoz garestiagoak dira konputazionalki. Hau da, zifratzeko eta deszifratzeko askoz denbora gehiago behar da.

Hori dela eta, algoritmo asimetrikoak ez dira normalki erabiltzen komunikazio orokorrak zifratzeko, baizik eta komunikazio-bide segurua ezartzeko: kriptografia asimetrikoa mezuak zifratzeko erabiliko den kriptografia simetrikorako gako ezkutuak trukatzeko erabiliko da. Horretarako mezu labur gutxi batzuk besterik ez dira asimetrikoki zifratu behar eta, beraz, kriptografia asimetrikoa erabiltzea bideragarria da. Beraz, konfidentzialtasunari dagokionez, kriptografia asimetrikoa simetrikoaren osagarria da.

28. Mundu anglosaxoian Alice eta Bob izenak erabiltzen dira sare-segurtasunari buruzko liburuetan horrelako adierazpenetan, A eta B letren ordez, testua ulerterrazagoa egiteko.

Hona hemen kriptografia asimetrikorako algoritmo batzuk:

- Diffie-Hellman.

  Lehen aipatu dugun algoritmoa da, 1976an kriptografia asimetrikoari hasiera eman ziona. Aurretik elkar ezagutzen ez duten bi kidek kanal arriskutsu baten bidez gako simetriko bat ebazteko metodo bat da. Bere izen osoa *Gakoak elkarrekin trukatzeko Diffie-Hellman algoritmoa* da. Beste teknika kriptografikoetako oinarri bezala erabiltzen da. Adibidez, erabil daiteke bi kideren arteko komunikazioaren konfidentzialtasuna bermatuko duen zifratze simetriko baten gakoa ebazteko.

- RSA algoritmoa (Rivest Shamir Adleman[29]).

  Algoritmo asimetrikorik seguruenetako bat da. 2000. urtera arte bere erabilera ordaintzekoa izan da. Kriptografia asimetrikorako *de facto* erabilitako estandarra da. Bere segurtasuna zenbaki handien faktorizazioaren zailtasunean datza. Kriptograma bat eta dagokion gako publikotik abiatuta jatorrizko mezua lortu nahi duenak arazo horri egin behar dio aurre. Dena dela, nahiz eta analitikoki segurua izan, algoritmoa gaizki erabiliz gero ondoko ahulezia hauek sor daitezke:

  – Gako multzo ahul bat dago. Gako horiek zifratzeko erabilita, mezua ez da aldatzen. Inplementatzaileek kontuan hartu behar dute hori, gako horien erabilera ez onartzeko.

  – Gakoen luzerarekin kontuz ibili behar da. 512 biteko gakoak erabiltzea nahikoa zela uste zen, baina, 1999. urtean, faktorizaziorako algoritmoen optimizazioak 1.024 biteko gakoen luzera minimoa gomendatzea ekarri zuen. Gero, 2003. urtean, gakoen luzera horren fidagarritasuna kolokan jarri zuten, eta harrezkero 2.048 biteko luzera erabiltzea gomendatzen da. Oraingoz, ez da aurreikusten luzera horretarako iraungitze-datarik.

- Elgamal zifratze-algoritmoa.

  Diffie-Hellman algoritmoan oinarritutakoa da, 1984. urtean argitaratua. Posta elektronikoa zifratzeko hainbat sistemak erabiltzen dute (GPG eta PGP, adibidez). Zifratze-algoritmoarekin batera, beronen asmatzaileak sinadura elektronikorako beste sistema bat ere aurkeztu zuen. Horregatik Elgamal izen bereko bi teknikak —zifratzekoa eta sinadura digitalekoa— maiz nahasten dira.

5.4.2. Kautotasuna

Kautotzea norbaiten identitatea egiaztatzea da. Atal honetan, sarearen bidez komunikatzen ari diren bi alderdiek nola kautotu dezaketen elkar ikusiko dugu. Horretarako komunikazioko bi alderdiek eman behar dituzten urratsek kautotze-

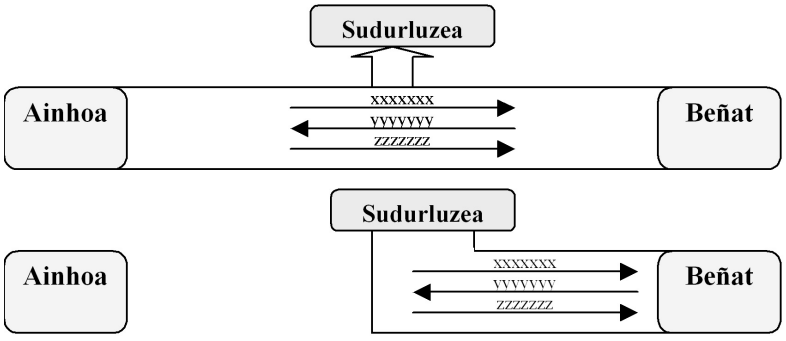
29. Algoritmoa sortu zutenen abizenak dira.

protokoloa osatzen dute. Eskuarki, kautotzeko protokoloa bi alderdiek beste edozein protokolo egikaritu aurretik abiatuko da (esaterako, HTTP, bideratze-taulak eguneratzeko protokoloa, edo SMTP protokoloa). Kautotzeko protokoloak lehenik alderdien identitateak finkatzen ditu; kautotzearen ondoren soilik has daitezke bi aldeak informazioa bidaltzen. Normalki, kautotzeaz gain, prozesu horretan konfidentzialtasuna bermatuko duen zifratzerako beharko diren gakoak (simetrikoak, gehienetan) ere ezarriko dituzte.

Zifratzean oinarritutako kautotzea

Kriptografiak, konfidentzialtasunaz gain, mezua zifratu duenaren nortasuna ere berma dezake. Kriptografia simetrikoa erabilita, gako simetrikoa bi solaskideek beste inork ez duela ezagutzen bermatuta badago, jasotako mezua nork bidali duen ere bermatzen da. Era berean, kriptografia asimetrikoan, jasotakoa ongi deszifratzen badugu gure solaskidearen gako publikoarekin, ez dago zalantzarik: mezua berak zifratu du, berak soilik ezagutzen duen bere gako ezkutua erabiliz. Horrelako zifratzean oinarritutako kautotzeak ez du inongo protokolorik behar bi aldeak identifikatzeko, eta zuzenean has daitezke informazioa elkarri bidaltzen.

Zoritxarrez, horrelako kautotzea ez da bideragarria hainbat arazorengatik. Hasteko, kriptografia asimetrikoaren kasuan, motela litzateke dena zifratu behar izatea. Lehen ikusi dugunez, kriptografia asimetrikoa ez da erabiltzen, bere kostuagatik, komunikazio orokorrak zifratzeko. Kriptografia asimetrikoa baztertuta, simetrikoa erabiltzea gelditzen zaigu, baina ikus dezagun 5.7. irudiak deskribatzen duen erasoa. Hor, Ainhoak eta Beñatek saio bat izan dute (irudiko goiko aldea), dena beraien arteko gako simetrikoak zifratuta, horrela konfidentzialtasuna eta kautotasuna bermatzearren. Baina sudurluze batek trafiko guztia atzeman eta kopiatu badu (beheko irudia), geroan, sudurluzeak badu Ainhoaren nortasuna ordezkatzea eta kopiatutako saioa Beñatekin errepikatzea. Horri errepikapen-erasoa deitzen zaio.



5.7. irudia. Errepikapen-erasoa. Lehenengo, sudurluzeak mezu zifratuen trukea grabatzen du, eta gero errepikatzen du, Ainhoarena eginez.

Errepikapen-erasoa itsua da, erasotzaileak ez baitaki benetan zertan ari den. Eraso mota hau, behar den informazioarekin konbinatuta, kaltegarria izan daiteke. Adibidez, demagun Ainhoak dirua zor diola norbaiti, eta Beñat Ainhoaren bankua dela. Telebankako aplikazio baten bidez, Ainhoak Beñati bere hartzekodunari ordainketa egiteko agintzen dio. Hartzekodunak sudurluzearena egiten badu, eta errepikapena gauzatzen badu, nahi duen adina transferentzia egin ditzake Ainhoaren kontutik bere kontu korrontera. Arazo horri aurre egiteko bidea Ainhoaren eta Beñaten arteko lan-saio bakoitzeko gako simetriko ezberdina erabiltzea da, eta horrek kautotze-protokolo baten beharra dakar, gako hori ezarriko duena. Horrelako protokolo baten adibidea honako hau da:

(1) Ainhoak zifratu gabeko mezu bat bidaltzen dio Beñati, non bere burua identifikatu besterik ez duen egiten. Mezu horren bidez Ainhoak Beñati lan-saio bat hasteko eskatzen dio.

(2) Beñatek bien arteko gako simetrikoarekin zifratutako mezu batekin erantzuten dio aurrekoari. Erantzunean lan-saio horretarako espresuki sortutako gako simetrikoa bidaltzen dio. Gako hori sortzeko prozesuak bermatu behar du bien arteko komunikazioetan gako hori ez dela inoiz berriz erabiliko, errepikapen-erasoak ekiditeko.

(3) Hortik aurrera adostutako gako simetrikoa erabiliko dute bi aldeek lan-saioa amaitu arte.

Aurreko protokoloak ez du inongo ahulezia teorikorik, baina arazo praktiko bat badu. Bere ahulgunea bi aldeen artean aurretik adostutako gako simetriko bat egon beharra da. Bi aldeek sarean zehar komunikatu baino lehen elkarrekiko harremana baldin badute, ez dago problemarik, nolabait gako simetriko hori aurreadostu baitezakete. Baina aurreneko harremana sarearen bidez egiten bada, jai dute; ez dago lan-saiorako gako simetrikoa ezartzerik.

Beraz, kautotzea zifratzean oinarritutako protokoloen bidez egiteko, lehenago gakoen ezarpenaren arazoa konpondu beharko dugu. Laster ikusiko dugu nola.

Pasahitzean oinarritutako kautotzea

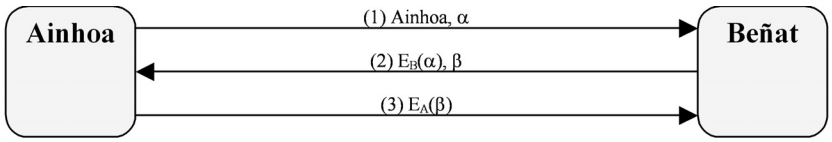
Kautotzeko protokolorik sinpleena eta, beharbada, erabiliena, erabiltzailearen izenaren eta pasahitzaren bidalketan datzana da. Aplikazio-mailako protokolo askoren lehen urratsa hori da; izan ere, kasu askotan kautotzeko prozesu horrekin batera aplikazio-mailako konexioak ezartzen dira. Baina besterik ez bada egiten, {*erabiltzailea, pasahitza*} bikotea bidaltzen duten protokoloak ez dira batere seguruak. Informazio hori daramaten datagramek zeharkatuko duten sareren edota bideratzaileren batean sniffer bat baldin badago, jai dugu: snifferrak bildutako informazioa jasotzen duen arrotzak gure nortasuna ordezka dezake, nahi duenean. Garbi dago segurtasuna bermatzeko bidalketa horiek zifratu egin behar direla. Hau da, erabiltzailearen izena eta pasahitza bidali baino lehenago, kanal konfidentzial

bat ezarri behar da bi muturren artean, hau da, zifratzerako gako bat ezarri behar da, zifratzean oinarritutako kautotzearen kasuan ikusi dugun bezala. Hala eta guztiz ere, kanal konfidentzial gehi pasahitzeko mekanismo honek baditu honako gabezia hauek:

- Sistema hau bideragarria da soilik solaskide bat aurretik baldin badago erregistratuta bestearen aurrean, hau da, besteak onartuko duen erabiltzaile eta pasahitza bikote bat baldin badu.
- Edozelan ere, protokolo honek ez du balio komunikazioko bi aldeak kautotzeko, alde bakar bat baizik. Sare-aplikazio batzuetan nahikoa izango da bezeroa kautotzea, baina beste askotan ez. Ez litzateke bideragarria bezero bakoitzak zerbitzarien nortasuna eta pasahitzak erregistraturik izatea.
- Eta, gainera, protokolo hauen beste arazo bat aurreko atalaren bukaeran aurkitutako bera da: nola lortu hasierako kanal konfidentziala eratzeko behar den gakoa (simetrikoa ala publikoa)? Gakoen ezarpenaren arazoa hor dugu oraindik konpondu gabe.

Erronka-protokoloak

Erronka-protokoloetan solaskide bakoitzak besteari eskatzen dio bere nortasuna frogatzea zifratze baten bidez. Kriptografia simetrikoa erabiltzen bada, Ainhoak eskatuko dio Beñati zenbaki bat zifratzea (hori da erronka). Ondo egiten badu, behar den gakoa daukala erakusten du, eta, beraz, bere nortasuna frogatuta dago (kontuan izan gako simetrikoa bi solaskideek beste inork ez duela ezagutzen suposatzen dugula). Kriptografia asimetrikoan, Ainhoak Beñati bere gako ezkutuarekin zenbaki bat zifratzeko erronka botako dio. Ainhoak, Beñaten gako publikoa erabiliz, ondo deszifratzen badu Beñatek zifratutakoa, ez dago zalantzarik: Beñaten gako ezkutua erabili du zifratzeko, eta, beraz, Beñat da. Hori da 5.8. irudiak adierazten duena:



5.8. irudia. Kriptografia asimetrikoan oinarritutako erronka-protokoloa.

(1) Ainhoaren hasierako mezuan zifratu gabeko bi gauza bidaltzen dizkio Beñati. Bata, Ainhoaren identifikazioa eta saioa ezartzeko eskaera besterik ez da. Bestea, espresuki sortutako α zenbaki bat da. Zenbaki hori sortzeko prozesuak bermatu behar du bien arteko hurrengo kautotze-saio batean α ez dela berriro erabilia izango, errepikapen-erasoak ekiditeko. Berme hori betetzeko, α-ren sorrera uneko datan eta orduan oinarritzen da normalki.

(2) Beñaten erantzunak ere bi zati ditu. Bata α zenbaki bera izango da, baina Beñaten gako ezkutua erabiliz zifraturik (E_B). Bestea Beñatek sortutako β zenbakia izango da, errepikapenak saihesteko α-k dituen ezaugarri berberak dituena.

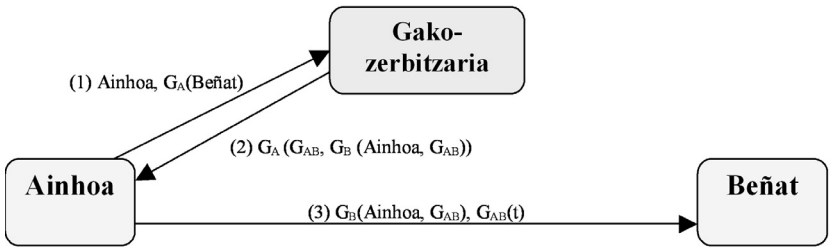
(3) Ainhoak Beñatek zifratutakoa deszifratuko du, Beñaten gako publikoa erabiliz, eta α zenbakia lortuko du. Horrek beste aldean benetan Beñat dagoela bermatzen du, berak bakarrik zifratu dezakeelako horrela α zenbakia. Orain, Ainhoak Beñatek bidalitako β zenbakia zifratuko du bere gako ezkutuarekin (E_A), eta hori bidaliko dio Beñati. Beñatek deszifratzen duenean, Ainhoaren gako publikoa erabiliz, beste aldean Ainhoa dagoela jakingo du.

Orain komunikazioaren bi muturrak kautotuta daude. Normalki, komunikazioaren hurrengo urratsa konfidentzialtasuna bermatzea litzateke eta, horretarako, bi muturrek saio honetarako gako simetriko bat ezarriko dute, zifratzean oinarritutako kautotze-protokoloetan ikusi dugun era berean. Adibidez, 3. urratsean, zifratutako β zenbakiarekin batera, Ainhoak lan-saiorako sortutako gako simetriko bat bidal dezake. Gako hori ezkutatzeko, Beñaten gako publikoa erabiliko zuen.

Zoritxarrez, zifratzearen bidezko kautotzean gertatzen zen bezala, nahiz eta protokolo honek arazo teorikorik ez izan, badu arazo praktiko bat: Ainhoak Beñaten gako publikoa ezagutu behar du, eta alderantziz, Beñatek Ainhoaren gako publikoa ezagutu behar du aurretik. Gakoen arazoa konpontzeko lehenengo proposamena gako-zerbitzariak erabiltzea da.

Gako simetrikoetarako zerbitzariak

Gakoen banaketaren arazoa gako-zerbitzari batek konpon dezake, baldin komunikazioko parte-hartzaile potentzial guztiak zerbitzari horretan erregistratuta badaude. 5.9. irudiak erakusten du gako-zerbitzari baten bidezko kautotze-protokoloen jarduera sinplifikatua.



5.9. irudia. Gako-zerbitzariaren bidezko kautotze-protokoloa, kriptografia simetrikoa erabiliz. Protokolo honetan, bi aldeak kautotzeaz gain, kanal konfidentziala ezartzen da.

Hau da irudiko urratsetan egiten dena:

(1) Ainhoak bere burua aurkezten du gako-zerbitzariaren aurrean. Bere burua kautotzeko, haren eta gako-zerbitzariaren artean aurretik adostutako gakoan (G_A) oinarritutako kautotze-protokoloren bat erabiliko du, erabiltzailea + pasahitza edo erronka modukoa. Hala ere, irudia sinplifikatzeko, G_A Ainhoak norekin komunikatu nahi duen (Beñat) zifratzeko besterik ez da erabiltzen. Kautotzeaz gain, G_A erabiliko dute bi aldeek saioko konfidentzialtasuna bermatuko duen beste gako bat adosteko. Berriz ere, hori ez da agertzen irudian, eta, eskema argitzearren, suposatuko dugu G_A gako bera erabiltzen dutela mezuak zifratzeko.

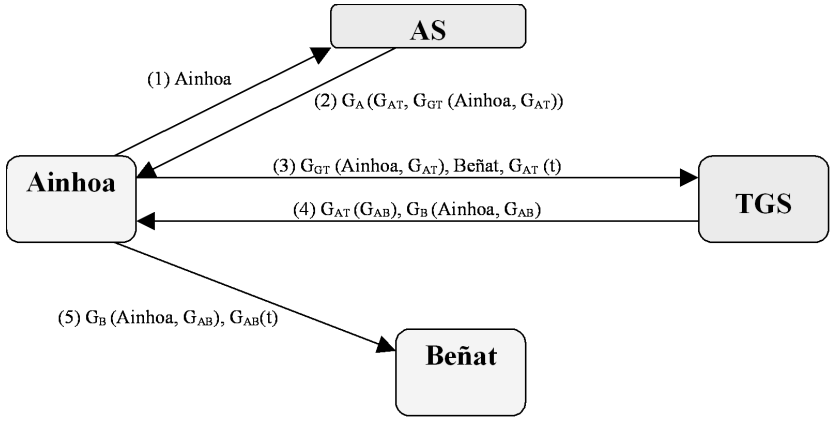
(2) Gako-zerbitzariaren erantzunean bi gauza ematen zaizkio Ainhoari. Alde batetik, Ainhoaren eta Beñaten arteko lan-saiorako gako simetriko berria, horretarako espresuki sortua (G_{AB}). Beste alde batetik, Beñaten eta gako zerbitzariaren arteko gako simetrikoarekin zifratuta (G_B), Ainhoaren identifikazioa eta Ainhoaren eta Beñaten arteko lan-saiorako berriki sortutako gako simetrikoa. Gako-zerbitzariaren erantzunaren bigarren zati hau ulertezina da Ainhoarentzat. Nolabait, gutun-azal itxi bat da berarentzat. Hau da Ainhoak Beñati aurkeztu behar dion egiaztagiria, gako-zerbitzariak emanda.

(3) Orain Ainhoak Beñatengana jo dezake, bere burua identifikatzen duen egiaztagiria baitu. Gako-zerbitzariak emandako egiaztagiria bidaltzen dio Beñati. Gainera, errepikapen-erasoak ekiditeko, denbora-marka bat bidaltzen du, bien arteko gako simetrikoarekin zifratuta. Beñatek egiaztagiria deszifratuko du, eta hor aurkituko du Ainhoaren identifikazioa eta berarekin komunikatzeko erabili behar duen gako simetrikoa. Orain bi aldeak, Ainhoa eta Beñat, kautotuta daude eta badute lan-saio honetan, eta honetan bakarrik, erabiliko duten gako simetrikoa. Norbaitek bidalketa hau eta Ainhoaren eta Beñaten artean egingo diren hurrengoak kopiatzen baditu, eta geroago birbidaltzen baditu, Beñatek errepikapena dela atzemango du, denbora-marka desfasatuta egongo delako.

MITen (Massachusets Institute of Technology) garatutako Kerberos sistema da gako-zerbitzariaren bidezko kautotasuna erabiltzen duen softwarerik ezagunena. Internet estandar izateko proposatuta dago (RFC 4120). Kerberos sareko zerbitzariak atzitzen zituzten erabiltzaileak kautotzeko diseinatu zen, eta hasiera batean domeinu administratibo bakar batean erabiltzeko diseinatuta dago, campus batean edo enpresa batean, esaterako. Guk ikusitako kautotze-protokolo generikoa baino konplexuagoa da Kerberos, zerbitzu gehiago ematen dituelako, baina bere oinarria ez da aldatzen. Kriptografia simetrikoa erabiltzen du Kerberosek, azkarragoak baitira algoritmo kriptografiko simetrikoak asimetrikoak baino. Hiru zerbitzari agertzen dira Kerberosen bidezko komunikazioetan:

- Aplikazio-zerbitzaria. Hau da, norekin hitz egin nahi duen Ainhoak (gure adibideetan, Beñatekin).
- Gako-zerbitzaria. Kerberos sisteman, honi AS izena ematen zaio (Authentication Server). Erabiltzaileek zerbitzari honekin aurreadostutako gako simetrikoa beharko dute. Gizakiok gako simetriko horiek gogoratzea oso zaila denez, erabiltzaileak sortutako pasahitz batetik abiatuta kalkulatzen dira.
- Tiket-zerbitzaria. Kerberosen jargoian, TGS izena du (Ticket-Granting Server). Zerbitzari hau ez da guk deskribatutako protokolo generikoan agertzen. Aplikazio-zerbitzariek zerbitzari honekiko aurreadostutako gako simetrikoa izango dute.

Ikus ditzagun bezero baten (Ainhoa) eta aplikazio-zerbitzari baten (Beñat) arteko komunikazioa kautotzeko Kerberosen 4. bertsioan egiten diren urratsak. Ondoko irudi honetakoak dira.



5.10. irudia. Kerberosen bidezko kautotzea.

(1) Ainhoak gako-zerbitzariaren aurrean bere burua identifikatzen du. Hasierako mezu hau zifratu gabe doa.

(2) Gako-zerbitzariak Ainhoaren gako simetrikoa erabiliz zifratuko du bere erantzuna. Mezu hori jasotzen denean, aplikazioko bezeroak Ainhoari bere pasahitza eskatuko dio. Pasahitz horretatik, bezeroak lortuko du Ainhoaren gako simetrikoa, eta AS zerbitzariaren erantzuna deszifratuko du. Hor Ainhoak tiket-zerbitzariarekin lan-saio bat ezartzeko behar duena aurkituko du: gako simetriko bat (G_{AT}), eta egiaztagiri bat, gako-zerbitzariaren eta tiket-zerbitzariaren arteko gako simetrikoa erabiliz sortua (G_{GT}).

(3) Orain Ainhoaren bezeroak tiket-zerbitzariarengana joko du. Gako-zerbitzariak emandako egiaztagiria, aplikazio-zerbitzariaren identifikazioa (Beñat), eta zifratutako denbora-marka bidaliko dizkio.

(4) Tiket-zerbitzariak Ainhoaren eta Beñaten arteko lan-saiorako gako simetrikoa sortuko du (G_{AB}), eta bi aldiz zifratuko du: batean Ainhoarekin duen gako simetrikoarekin, eta bestean tiket-zerbitzariarekin Beñatek duen gako simetrikoa erabiliz. Bigarren hori Ainhoak Beñati aurkeztu beharko dion egiaztagiria da.

(5) Ainhoaren bezeroak Beñati berriki lortutako egiaztagiria bidaliko dio, eta, berriro errepikapen-erasoak ekiditeko, denbora-marka bat bien arteko gako simetrikoarekin zifratuta. Dagoeneko, Ainhoak eta Beñatek badute bien arteko gako simetriko bat.

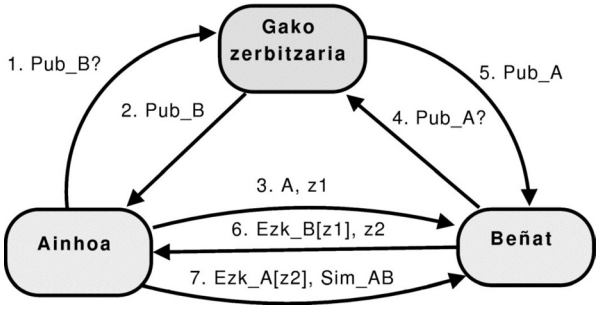
Baina zerbitzarien bidezko kautotzeko protokolo hauek ere ez dira perfektuak, berauen segurtasun osoa gako-zerbitzariaren segurtasunean baitago errotuta. Gako-zerbitzariaren bidezko kautotze-protokoloak erabiltzeko ondoko bi baldintza hauek bete behar dira:

- Parte-hartzaile guztiek gako-zerbitzariarenganako konfiantza osoa izan behar dute, bere gakoak beste inork ez dituela atzemango segurutzat jotzeko. Zerbitzaria gure sarekoa bada, hau ez da arazo bat. Baina gure saretik kanpokoekin aritzeko arazo bat dugu: gure zerbitzaria ez da, agian, fidatzekoa besteentzat (bestea nor den), besteen zerbitzariak guretzat, agian, fidatzekoak izango ez diren modu berean.
- Gako-zerbitzariaren erabiltzaile guztiek aurretik adostutako gakoak izan behar dituzte zerbitzariarekin komunikatzeko.

Kerberos eta antzeko sistemen erabilera nagusia inguru lokaletan kokatzen da, non kautotze-sistemaren erabiltzaile guztiek erlazio administratibo bat duten. Testuinguru horretan, eskuarki, gako-zerbitzaria kudeatzen duena eta sare lokala kudeatzen duena berbera da, eta gako-zerbitzariaren erabiltzaileak sare lokalaren erabiltzaileak dira. Egoera horretan, aurreko bi baldintzak erraz betetzen dira.

Kriptografia asimetrikorako gako-zerbitzariak

Kriptografia asimetrikoan oinarritutako erronka-protokoloetarako gako-zerbitzarien kasua desberdina da. 5.11. irudian dugu horrelako zerbitzari bat erabiltzen duen protokolo posible bat. Ainhoa eta Beñaten arteko saio konfidentziala eta kautotua ezartzeko emandako urratsak honako hauek dira:



5.11. irudia. Kautotze-protokoloa gako publikoetarako zerbitzari bat erabiliz.

1. eta 2. bidalketa: Ainhoak zerbitzariari Beñaten gako publikoa (Pub_B) eskatzen dio. Ikusi mezu hauek ez direla zifratu behar.

3. bidalketa: Ainhoak Beñati eskatzen dio saioa ezartzeko. Horretarako bere burua identifikatzen du (A), eta erronka-zenbaki bat bidaltzen dio Beñati (z1).

4. eta 5. bidalketa: Beñatek zerbitzariari Ainhoaren gako publikoa (Pub_A) eskatuko dio.

6. bidalketa: Beñatek bere nortasuna bermatzen du, bere gako ezkutuarekin erronka zifratuz (Ezk_B[z1]). Aldi berean, beste erronka bat bidaltzen dio Ainhoari (z2), bere burua identifikatu dezan. Ainhoak, mezu hori jasotzean, Beñaten nortasuna egiaztatuko du, zifratutakoa Beñaten gako publikoarekin deszifratuz.

7. bidalketa: Ainhoak, erronka zifratuta itzuliko du (Ezk_A[z2]). Aldi berean, saiorako kalkulatutako gako simetrikoa (Sim_AB) bidaltzen du. Mezua jasotzean, Beñatek badu Ainhoa kautotzea, Pub_A horretarako erabiliz.

Zerbitzariak banatzen dituen gakoak publikoak direnez, kriptografia simetrikorako ikusi ditugun zerbitzariek zituzten bi arazoak desagertzen dira:

- Zerbitzariarenganako konfiantza-arazorik ez: edozeinek kontsulta ditzake zerbitzari horrek gordetzen dituen gakoak. Are gehiago: horretarako dago zerbitzaria.
- Zerbitzariarekin komunikatu ahal izateko ez da behar zerbitzari horretan erroldatuta egotea aurretik.

Horrela izanik, badirudi azkenean lortu dugula testuinguru global batean erabiltzeko kautotze-protokolo bat, gako simetrikoak banatzeko zerbitzariek ahalbidetzen duten testuinguru lokala gaindituz. Nahikoa litzateke bakoitzak nahi duen webgunean bere gako publikoak argitaratzea, eta, beste inorekin saio bat hasterakoan, webgune horren berri eman kautotze-protokoloaren lehenengo urratsean, beste aldeak gure gako publikoa hor lor dezan. Baina, oraindik ere, arazoak daude. 5.11. irudiko protokoloak honako bi arazo hauek ditu:

- Gako publikoetarako zerbitzariaren kudeatzaileak bermatu beharko du, nolabait, bere webgunean gako publiko bat argitaratu nahi duena benetan dela esaten duena. Hau da, Ainhoak webgune horri bere gako publikoa bidaltzen badio, nola jakingo du Beñatek, webgune horretatik gako hori jaisten duenean, benetan Ainhoa dela gako hori hor utzi duena? Ez al da izango Ainhoaren nortasuna ordeztu nahi duen arrotz batek utzitako gako tranpa? Finean, konfiantza-arazo bat dago, gako publikoak banatzen dituzten zerbitzariek erabiltzaile guztien konfiantzazkoak izan behar baitute eta, berriz, eskema hau inguru lokaletik at erabiltzea zail bilakatzen da.
- Gainera, bi ezezagunen arteko harremana hasteko, beharrezkoa izango da besteari gako publikoa gordeta non dugun jakinaraztea, eta gero, gakoen bila hara joatea. Horrek denbora gehiegi eska dezake, Ainhoaren eta Beñaten sarearen egoeraren arabera eta aukeratutako gako-zerbitzariaren arabera. Imajina nolako web nabigazioa izango genukeen, web zerbitzari bakoitzarekin HTTPz hitz egiten hasi baino lehenago beste zerbitzari baten bila joan beharko bagenu (auskalo nola dagoen trafikoa haraino heltzeko, eta nola dabilen gako-zerbitzari bera), behar dugun gako publikoa hor eskuratzeko. Prozesua motelegia bihurtuko litzateke kasu askotan. Askoz arinagoa litzateke bakoitzak, hasierako mezuetan, bere gako publikoa beste aldeari helaraztea. Baina horrek arazoaren hasierara garamatza berriro: nola jakin bestea dela esaten duena?

Horren guztiaren ondorioa hau da: kriptografia asimetrikoan oinarritutako kautotzeak arazo praktiko larria du inguru irekietan, kriptografia simetrikoari gertatzen zitzaion bezala. Baina kriptografia simetrikoan ez bezala, arazo horrek badu konponbidea: badago bide bat nork bere gako publikoa beste aldeari helarazteko zuzenean, hasierako mezu batean, eta gako hori fidatzekoa izan. Auziaren muina egiaztagutun iraunkorrak lortzean eta erabiltzean egongo da. Kontuan izan gako simetrikoko zerbitzariek jaulkitzen dituzten egiaztagutunak «erabili eta bota» erakoak direla, errepikapen-erasoak ekiditeko, eta, beraz, lan-saio bakoitzeko egiaztapen berri bat lortu behar dela. Kriptografia asimetrikoa erabiliz egiaztagutun iraunkorrak lortzea badago: **ziurtagiri elektronikoak** edo **ziurtagiri digitalak** dute izena. Baina ziurtagiri elektroniko bat zer den eta nola erabiltzen den jakiteko, lehenago sinadura elektronikoa zer den jakin behar dugu. Hori aztertuko dugu hurrengo atalean, eta gero ekingo diogu berriro ziurtagirien aferari.

5.4.3. Osotasuna eta ukorik ez: sinadura digitala

Saretik at sortutako harremanetan, mundu fisikoan alegia, agiri bat sinatzen dugunean adierazten dugu agiriaren edukia ezagutzen dugula, horrekin ados gaudela, edota agiria bera guk sortu dugula. Sinaduraren bidez, honako hauek bermatzen ditugu:

- Agiriaren osotasuna: ezabadurak, urraketak edo zuzenketak dituen sinatutako agiriak bere balioa galtzen du.
- Norbaitek sinatutako agiria benetan pertsona horrek sinatu duela froga daiteke; betiere, sinadura egiaztatzea badago.
- Sinatu duenak ezin du ukatu bera dela sinatzailea: ez dago beste inork sinadura hori egin dezakeenik, ezin da sinadura faltsutu.

Mundu digitalean ere, sinaduraren beharra izango dugu, gure komunikazioetan trukatutako (sarearen bidez edo sarerik gabe) agirien egilearen nortasuna eta osotasuna bermatzeko. Horretarako asmatu da sinadura digitala (edo sinadura elektronikoa), kriptografia asimetrikoan eta *hash* funtzioetan oinarrituta. Gainera, faltsutzaile onek sinadura fisiko bat faltsutu dezakete; sinadura digital bat, aldiz, inolaz ere ez.

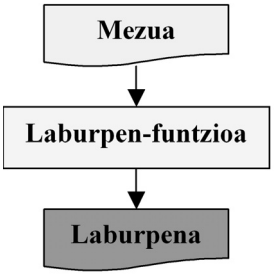
Agiri baten egilearen nortasuna bermatzea eta komunikazio bateko parte-hartzaile baten nortasuna egiaztatzea gauza bi dira, antzekoak izanda ere. Komunikazioaren kautotasuna besterik ez badugu bermatzen, ziurta dezakegu nork *bidali* digun agiri bat, baina ez nork *sortu* duen agiri hori. Gerta daiteke gure solaskide elektronikoak, kautotze-protokoloak haren nortasuna bermatu digularik, berak egin ez duen agiri bat guri bidaltzea. Bi arazoen arteko aldea eta erlazioa argiago ikusiko dugu ziurtagirien afera aztertzen dugunean. Orduan ikusiko dugu nola, askotan, kautotze-protokoloen oinarria ziurtagiri baten sinadura digitala izango den: gure solaskideak gure konfiantzazko beste norbaitek sinatutako ziurtagiri bat aurkeztu beharko digu bere burua kautotzeko.

Hash kriptografikoak

Hash funtzio kriptografikoak sinadura digitala sortzeko erabiltzen diren laburpen-funtzioen aldaera bat dira. Oro har, laburpen-funtzioek bit kopuru aldakorreko iturri batetik laburragoa den emaitza sortzen dute. Hash kriptografikoaren kasuan, ondoko ezaugarriak betetzen dira laburpena sortzean:

- Prozesu atzeraezina da: ezin da laburpenetik jatorrizko mezua lortu.
- Ez da posible laburpen bera sortuko duten bi mezu aurkitzea, bi mezu horiek existitzen badira ere.
- Prozesu azkarra eta erraza da.
- Laburpenak luzera finkoa du, jatorrizko mezuaren luzera edozein izanda ere.

Badaude laburpena gako baten arabera kalkulatzen duten laburpen-funtzioak, baina ez dira gehien erabiltzen direnak. Laburpen-funtzioen kontzeptua 5.12. irudian adierazten da. Sinadura digitalak kalkulatzeko ondoko bi hash kriptografikorako algoritmo hauek erabiltzen dira nagusiki:



5.12. irudia. Laburpen-funtzioak.

- MD5 (Message Digest 5) algoritmoa.

  128 biteko laburpena sortzen du. Posta elektronikoaren segurtasunerako asmatutako PGP softwarearen hasierako bertsioetan erabiltzen zen, eta horrek ospe handia eman dio. Ahulezia teoriko batzuk aurkitu dizkiote, baina oraindik asko erabiltzen da. Hala ere, bere erabilera jaisten ari da, SHA algoritmoen mesedetan.

- SHA (Secure Hash Algorithm) algoritmo sorta.

  NSAk sortutako algoritmoak dira, AEBko estandarrak izateko. Une honetan bost algoritmok osatzen dute sorta, SHA-1, SHA-224, SHA-256, SHA-384, eta SHA-512 izenekoek. Azkeneko laurak SHA-2 izenarekin biltzen dira. SHA-1 algoritmoak 160 biteko laburpena sortzen du; besteek, beren izenetan daramaten zenbakiak adierazten duen bit kopurukoa.

  Gehien erabiltzen dena SHA-1 da, 2005. urtean ahulezi posible bat aurkitu zioten arren. SHA-2 aldaeren kontrako erasorik ez da oraindik agertu, baina algoritmikoki antzekoak direnez, sendoagoa izango den beste alternatiba bilatzeari ekin diote. AES algoritmoarekin egin zen antzeko lehiaketa antolatu dute SHA-3 izena hartuko duen hash funtzio berria lortzeko. 2012. urterako espero da prozesua bukatuta egotea.

Hash kriptografikoarekin askotan nahasten diren beste laburpen-funtzioak ondoko taulan agertzen dira.

Izena ingelesez	*Izena euskaraz*	*Emaitza*	*Erabilera*
hash function	hash funtzioa	hash, laburpena	tauletan egindako bilaketak azkartzeko
checksum, hash sum	cheksum, erroreak atzemateko funtzioa	cheksum, erroreak atzemateko batura	transmisio- edo biltegiratze-erroreak atzemateko
fingerprint	hatz-marka, hatz-aztarna funtzioa	hatz-marka, hatz-aztarna	fitxategietan aldaketak atzemateko (backup sistemetan, cacheak eguneratzeko, log fitxategien zaintzan…)
criptographic hash	laburpen kriptografikoa	digest, mezuaren laburpena	sinadura digitalak sortzeko

5.4. taula. Laburpen-funtzioak.

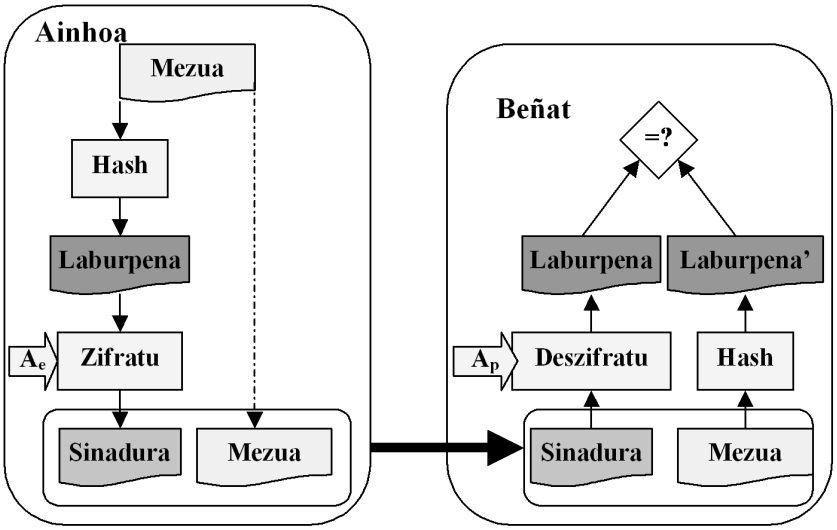
Sinadura digitalak

Laburpen-funtzioen eta kriptografia asimetrikoaren[30] erabilera konbinatuak sinadura digitalak gauzatzeko aukera ematen du. Horretarako egin behar diren urratsak 5.13. irudian agertzen dira. Hasteko, mezua sinatu behar duenak mezuaren laburpena kalkulatzen du. Gero laburpen hori bere *gako ezkutua*rekin zifratzen du, ez hartzailearen gako publikoarekin, konfidentzialtasuna lortzeko ikusi dugun moduan. Lortutakoa da mezuaren sinadura digitala. Igorleak biak batera bidaliko ditu, mezua eta bere sinadura. Hartzaileak bikote hori hartzean, sinadura hiru urratsetan egiazta dezake: bat, mezuaren laburpena kalkulatu; bi, igorlearen gako publikoa erabiliz sinadura deszifratu, eta hiru, aurreko bi urratsen emaitzak alderatu. Lortutako bi laburpenak berdinak ez badira, sinadurak ez du balio, faltsua da: edo norbaitek aldatu du mezua, edo mezuari dagokion sinadura aldatu dute. Ohartu horrela sinatutako mezuek honako ezaugarriak betetzen dituztela:

1. Sinadurak egilea identifikatzen du, beste inork ezin baitu mezuaren laburpena bere gako ezkutuarekin zifratu. Hartzaileak berak ezin izan du mezua asmatu, ez baitu egilearen gako ezkutua.
2 Egileak ezin du ukatu sinatutako mezua berak sortu duela.

Baina aurreko ataletan bezala, gakoen banaketaren arazoa dugu aurrean: sinadura digital bat egiaztatzeko, sinatzen duenaren gako publikoa behar dugu, eta hori era seguruan lortzea, nahiz eta publikoa izan gako hori, ez da hain erraza.

30. Badago kriptografia simetrikoa erabiliz sinadura digitalak sortzea, baina horrelako sinadurak egiaztatzeko beti behar da zerbitzari baten parte-hartze interaktiboa. Horregatik ez da erabiltzen sinadura mota hori.



5.13. irudia. Sinadura digitala eta bere egiaztapena. A_e eta A_p Ainhoaren gako ezkutua eta gako publikoa dira, hurrenez hurren.

5.4.4. Gako publikoen banaketa: ziurtagiriak

Ziurtagiriak eta konfiantza-ereduak

Ziurtagiri digital bat gako publiko bat eta nortasun bat uztartzen dituen agiri elektronikoa da. Ziurtagirian agertu behar dira, beti, ondoko lau atal hauek:

- Ziurtagiriaren jabearen nortasuna.
- Jabearen gako publikoa.
- Ziurtagiria jaulki duenaren nortasuna.
- Ziurtagiriaren sinadura, jaulkitzaileak egina.

Ziurtagiri-jaulkitzailearen eginkizuna funtsezkoa da. Lan hori nork egiten duen, hauetako bi konfiantza-ereduetako bat izango dugu:

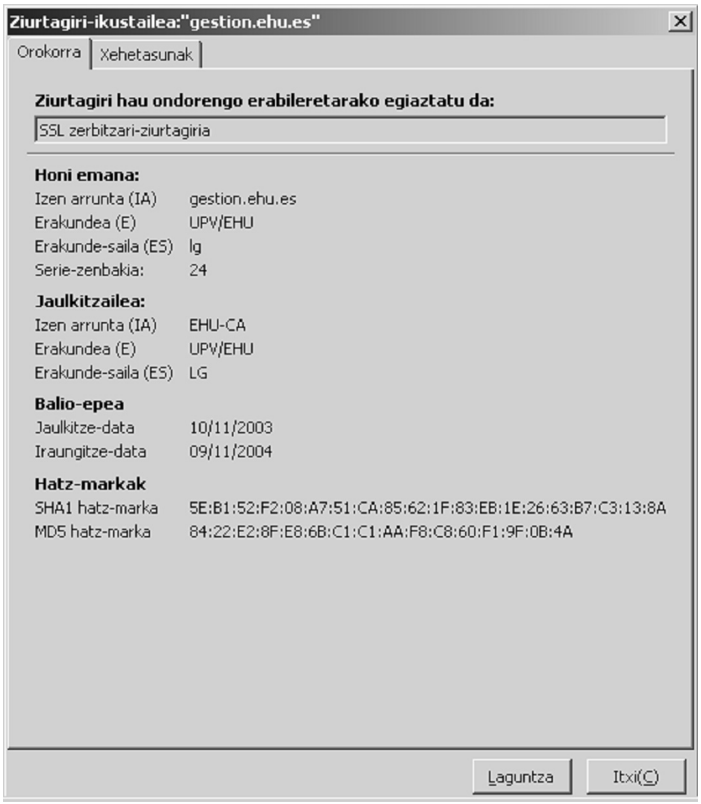
- Eredu horizontala: ziurtagiriaren sinatzailea (jaulkitzailea, alegia) gertuko norbait denean. Eredua mugatua da, hasiera batean behintzat, bakoitzak bere gertukoek (bere konfiantzakoek, alegia) sinatutako ziurtagiriak besterik ez baitu onartu ahal izango. Norberaren konfiantzako horiek osatzen dute bere konfiantza-taldea. Taldekide bakoitzaren gako publikoa ezagutuko du, inongo zalantzarik gabe (horregatik dira konfiantzakoak) taldea onartzen duenak. Taldea taldekideek babestutakoekin zabaltzen da. Adibidez, Ainhoak ezagutzen du Beñat, bere gako publikoa gordeta du bere giltza-zorroan, eta, beraz, onartzen ditu Beñatek sinatutako ziurtagiriak. Une

batean, Ainhoak ezagutzen ez duen Enara izeneko kide batekin saio seguru bat ezartzen du, eta horretarako Enarak Beñatek sinatutako ziurtagiri bat aurkeztuko dio Ainhoari. Ainhoak onartuko du, Beñaten sinaduraz fidatzen baita, eta, are gehiago, Enaraz fidatu nahi badu, bere gako publikoa gehitu dezake bere giltza-zorroan. Ohartu ondoko honetaz: desberdina da Enara benetan Enara dela jakitea (Beñaten sinadurak bermatzen diguna), eta Enarak sinatutakoa egia izatea.

Eredu horizontal hau posta zifratzeko sortutako PGP sisteman bultzatu dute. Hala ere, haren erabilgarritasuna mugatua da. Merkataritza elektronikorako ez da nahikoa, salerosketa batean, gehienetan, ez baitago inongo harremanik saltzailearen eta eroslearen artean, eta, beraz, beraien konfiantza-taldeak multzo disjuntuak dira. Horregatik, gehienetan, ezingo dira ziurtagiriak onartu, alde baten ziurtagiriaren sinatzailea bestearen konfiantza-taldean ez delako egongo. Arazoa konpontzeko dago eredu hierarkikoa.

- Eredu hierarkikoan, ziurtagiria sinatzen duena espresuki horretarako sortutako entitate bat izaten da. Haren papera mundu errealean notarioek jokatzen dutena da. Jaulkitzaile hauei ziurtatze-agintari deitzen zaie, edo, maiz, ingelesezko siglak erabiltzen dira (CA – Certification Authority). Laster aztertuko dugu hobeto haren lana eta eredu hierarkikoaren funtzionamendua.

ITUk (International Telecomunication Union) bere X.509 estandarra sortu du, non kautotze-zerbitzu bat eta ziurtagirien sintaxia zehazten diren. 5.14. irudian webgune baten X.509 ziurtagiri baten alderik garrantzitsuenak agertzen dira, arakatzaile batek pantailan erakusten dituen moduan. Saleroste elektronikoak sortzen dituen transakzio seguruen beharra dela-eta, ziurtagiri jaulkitzaileen inguruko interesa handitu egin da azkenaldian.



5.14. irudia. Webgune baten X.509 ziurtagiriaren eduki batzuk. Irudian «hatz-marka» deitzen duena ziurtagiriaren sinadura da. Arakatzaileak ez du erakusten leiho honetan datu garrantzitsuenetako bat: ziurtagiriaren jabearen gako publikoa. «Xehetasunak» izeneko leihoan ikus daiteke gako hori.

Ziurtatze-agintariak

Ondoko hauek dira ziurtatze-agintari baten lanak:

- Kautotu.

  Ez dago kautotze hau nola egin behar den agintzen duen prozedurarik, baina agintariak zorrotza izan behar du norbaiten nortasuna kautotzean. Gehienetan kautotzeko prozedura horiek saretik at egiten dira, agiri fisikoen bidez eta, askotan, lagunen bat fisikoki agertzeko ere eskatzen da.

- Ziurtagiriak sortu eta eman.

  Ziurtagiria sortzea 5.13. irudiko ezkerreko prozedura betetzea da (Ainhoarena), non mezua ziurtagiriaren zati bat den (jabea, bere gako publikoa, eta jaulkitzailearen identifikazioa). Behin mezua sinatuz gero, ziurtagiria lortzen dugu. Bere jabeari helarazteko sarea erabil daiteke, edo zuzenean diskete batean eman. Kontuan izan ziurtagiria agiri publiko bat dela eta,

beraz, ez dago inongo arazorik sarean zehar bidaltzean. Bai, ordea, ziurtagiriak daraman gako publikoarekin lotuta dagoen gako ezkutua bidaltzean. Ziurtagiriak jaulkitzeko zailtasuna aurreko urratsean datza: kautotzean.

- Ziurtagiriak kudeatu.

  Jaulkitzaileak kontrolatuko du nori eman dizkion ziurtagiriak, eta, beharrezkoa denean, ziurtagiriak baliogabetuko ditu. Adibidez, jabeak ziurtagirian agertzen den gako publikoari dagokion gako ezkutua galtzen badu, ziurtagiri hori baliogabetu egin behar da.

Konfiantza-hierarkiak

Nork jaulkiko ditu ziurtagiriak? Izan dezakegu erakunde bakarra, munduan zehar zerbitzari eta bulego asko barreiatuta dituena. Baina eredu zentralizatu horrek arazo ugari ditu, bereziki antolatzeko arazoak. Herri batzuetan erakunde hori gobernuarena izatea nahi izango dute, eta beste batzuetan, aldiz, gobernuarena ez izatea ezinbestekoa izango da. Koska dago konfiantza ea norengan dugun.

Arazo horiek direla eta, onartu egin da ziurtagirien jaulkitzailea edozein izatea, eta erabiltzaileek aukeratzea norengan duten konfiantza. Horrela izanik, ziurtagiriak erabili nahi dituen edonork bere konfiantzakoen **zerrenda** osatu beharko du (eredu horizontalean ikusi dugun konfiantza-taldea kontzeptu berdina, baina, testuinguru hierarkiko honetan, beste izen batekin). Hau da, onartuko dituen ziurtagirien jaulkitzaileen izenak eta gako publikoak gorde beharko ditu, eta komunikazio bat hastean beste aldeak bere ziurtagiria bidaltzen dionean, ondoko bi urrats hauek beteko ditu:

(1) Begiratu ea ziurtagiriaren sinatzailea bere konfiantzakoen zerrendan agertzen den. Ez badago, hiru gauza egin ditzake: komunikazio saioa bertan behera utzi, saio honetarako bakarrik onartu ziurtagiria, edo sinatzailea bere konfiantzakoen zerrendan sartzea onartu.

(2) Jasotako ziurtagiriaren sinatzailea jadanik bere konfiantzakoen zerrendan baldin badago, egiaztatu ziurtagiria. Hau da, 5.13. irudiko eskuineko aldean agertzen dena egin, Beñatek egiten duena, hain zuzen.

Hala eta guztiz ere, prozedura hau oso murriztailea da. Webean zehar ibiltzen garenean, askotan topatuko ditugu ziurtagiri-sinatzaile arrotzak, guztiz ezezagunak, errefusatu beharko ditugunak. Zorrotzak bagara, eta gure konfiantzakoen zerrendan benetan gure konfiantza duten erakundeak soilik onartzen baditugu, oso ziurtagiri gutxi onartuko ditugu, eta ziurtagirien erabilgarritasuna pikutara joango zaigu. Eta zorrotzak ez bagara, ziurtagirien balioa galduta dago.

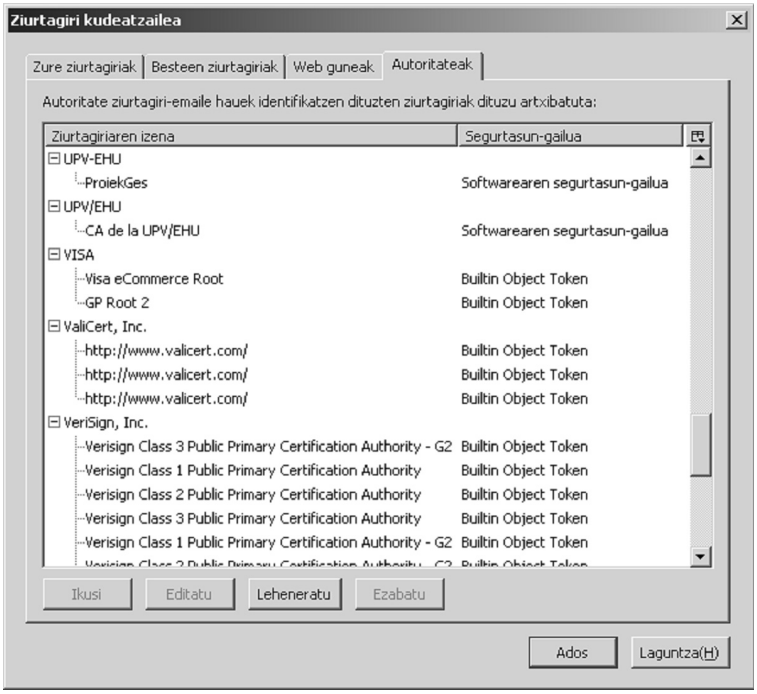
Korapilo honen irtenbidea ziurtagiri-jaulkitzaileen hierarkia sortzean dago. Ideia hau da: ziurtagiri-jaulkitzaile baten fidagarritasuna hierarkikoki bere gainean dagoen beste jaulkitzaileak, edo jaulkitzaileek, bermatzen du(te). Ikus dezagun adibide batekin. Demagun Ainhoak Beñaten ziurtagiria jasotzen duela, Usurbilgo

Gazte Asanbladak sinatuta. Ainhoak ez du ezagutzen erakunde sinatzaile hori eta, beraz, besterik ezean, ez du ziurtagiria onartuko. Beñatek badaki hori eta, horregatik, bere ziurtagiriarekin batera, Usurbilgo Gazte Asanbladaren ziurtagiria ere bidaltzen dio Ainhoari, bigarren hau Udalbiltzaren Ziurtagiri Zerbitzuak sinatuta. Ainhoak Udalbiltzaren Ziurtagiri Zerbitzu hori ezagutzen badu, eta bere gako publikoa badu, bi ziurtagiriak egiazta ditzake, Beñatena eta Usurbilgo Gazte Asanbladarena. Baina demagun Ainhoa Zeelanda Berrikoa dela, eta Udalbiltzaren Ziurtagiri Zerbitzua ezezaguna duela. Orduan, hirugarren ziurtagiri bat beharko du Udalbiltzaren Ziurtagiri Zerbitzuaren gako publikoa egiaztatzeko. Beñatek ez badio bidali, eskatu egin beharko dio. Demagun Udalbiltzaren ziurtagiria NBEk (Nazio Batuen Erakundea) sinatzen duela, eta demagun NBE hierarkiaren maila gorenean dagoela. Gune honetan ez dago harago joaterik: Ainhoak konfiantza izan behar du NBEk sinatutako ziurtagirietan edo, beste era batean esanda, Ainhoak NBEren gako publikoa izan behar du, eta ziur egon behar du gako hori benetan NBErena dela. Hau bezalako ziurtagiri-kate bati, erabiltzaile baten ziurtagiritik hierarkiako maila goreneraino doanari, **konfiantza-katea** edo **egiaztatze-bidea** deitzen zaio (*chain of trust* edo *certification path* ingelesez). Hierarkia osatzen duen konfiantza-egiturari **konfiantza-zuhaitza** edo **konfiantza-hierarkia** esango diogu.

Hala eta guztiz ere, arazoaren koska mantentzen da: nork kudeatzen du hierarkiaren maila gorena? Nor dago konfiantza-zuhaitzaren erroan? Aurreko adibidean NBE agertzen da lan horretan, baina adibideko ziurtatze-agintariak asmakizunak dira. Errealitatean ez dago konfiantza-zuhaitz bakarra munduan, asko baizik, eta bakoitza bere erroarekin. Izan ere, norberak bere konfiantza-zuhaitza sor dezake. Erroek **konfiantza-aingura** (*trust anchors*) edo **konfiantza-erroa** izena dute. Erabiltzaileak erro horien izenak eta gako publikoak bere konfiantza-zerrendan gorde beharko ditu, 5.14. irudikoa bezalako ziurtagiri autosinatuak bilduz. Tira, zer edo zer hobetu dugu egoera; behintzat, konfiantza-zerrenda horiek tamaina onargarria dute… oraingoz. Eta, hala ere, nola jakingo du erabiltzaile arrunt batek zeintzuk diren erro horiek eta beren gako publikoak? Egiten dena hau da: ziurtagiriak masiboki erabiltzen dituzten aplikazioek aurretik dute osatuta konfiantza-zerrenda. Une honetan, hori da web arakatzaileen kasua. Gaur egungo arakatzaileek 100 bat erro daramate aurrekargaturik. 5.15. irudian dugu arakatzaile baten zerrendaren zati bat, pantailan agertzen den moduan. 5.14. irudiko ziurtagiria zerrenda horietako bat da: sinatzailea eta jabea bera da (RSA konpainia).

Ikusten denez, azkenean, konfiantzaren afera softwarearen ekoizlearen eskuetan gelditzen da askotan, berak erabakitzen baitu zein erro sartu bere softwarean. Dena dela, erabiltzaileak aurrekargatutako konfiantza-zerrenda kudea dezake, eta hortik ziurtagiriak kendu edota gehitu. Aurrekargatutako zerrendak, normalki, diruaren truke osatzen dira, hau da, erro-erakundeak software-ekoizleari ordaintzen dio bere zerrendan agertzeagatik. Une honetan aipatu behar da konfiantza-erro askok kobratu egiten dutela ziurtagiriak jaulkitzeagatik. Izan ere, gaur egun dauden

konfiantza-erro gehienak horretarako bereziki sortutako konpainiak dira. Hau da, konfiantza ere negozio bihurtu da Interneten. Konfiantza handikoa bilakatzen den ziurtatze-agintariak dirutza kobra diezaioke bere ziurtagiria eskuratu nahi duen zerbitzariari. Ziurtagiri komertzialen arteko borroka ahalik eta erabiltzaile kopururik handienak bere ziurtagiria onartzean datza, horrela sortzen baita beraien ziurtagirien eskaria. Horregatik agertzen dira beren produktuen konfiantza-zerrendetan agertzeagatik software-ekoizleei ordaintzeko prest daudenak.



5.15. irudia. Arakatzaile baten konfiantza-zerrendaren zati bat.

Konfiantza-zuhaitzen afera eta beronen inguruan sortzen diren arazoak biltzeko (erabiltzaileak, ziurtagiriak, jaulkitzaileak, ziurtagiri-katalogoak, baliogabetuen zerrendak, formatuak, protokoloak…) PKI siglak (Public Key Infraestructure) erabiltzen dira gaiari buruzko agiri eta testu askotan.

5.5. KOMUNIKAZIO SEGURURAKO TEKNOLOGIAK

Aurreko atalean komunikazio segurua zer den eta komunikazio seguru hori lortzeko oinarri teorikoa, kriptografia, aztertu dugu. Oraingo honetan oinarri teoriko horiek aplikatzen dituzten hainbat teknologia ezagutuko ditugu.

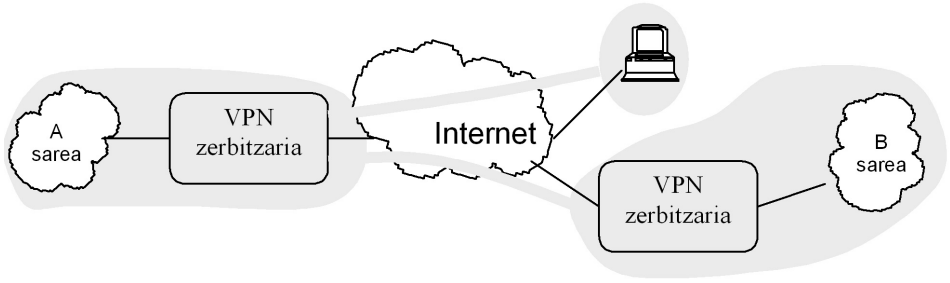
5.5.1. Sare Pribatu Birtualak (VPN)

VPN kontzeptua zabala eta zehaztugabea da, oso teknologia desberdinak sartzea baitago izen horren pean. Testu honetan, sare publiko baten bidez (gehienetan Internet) gure sareko bi zatiren arteko komunikazio segurua bermatzen duen teknologiara mugatuko gara VPNez aritzen garenean.

VPNetan tunelak eta kriptografia konbinatzen dira. Funtsean, VPN bat sareko zatietako mugetan dauden konputagailuen artean tunelak ezartzea da, gero, tunel horietatik bidaltzen den guztia zifratzeko. Bi muturren arteko tunela edota zifratzeko eskema ezartzeko kautotze-prozesu bat egin behar denez, VPNek RAS zerbitzariarena ere egiten dute. Gainera, askotan, VPNetatik bidalitako mezu guztiak sinatzen dira, komunikazioaren osotasuna bermatuz. Hau da, VPN teknologiak aurreko atalean ikusitako komunikazio segururako baldintzak betetzea ahalbidetzen du.

VPNak ondoko bi taldeetan sailka daitezke topologiaren arabera:

- Sareen arteko VPNak: gure intraneteko hainbat zati konektatzen dira VPNen bidez. Sareko zati bakoitzean VPN zerbitzari batek egon behar du, eta zerbitzarien artean ezarriko dira VPNak gauzatuko dituzten tunelak. Sareko zati batean kokatuta dagoen konputagailu batetik beste zati batera doan trafikoa VPN zerbitzarira bideratuko da, hortik beste aldeko VPN zerbitzarira iristeko, eta hortik bere helmuga den konputagailurako bideari jarraitzeko.
- Urrutiko sarrerarako VPNa: konputagailu soil bat gure intranetean sartzeko modua da VPNa. Hau aurrekoaren kasu partikulartzat har daiteke. VPN zerbitzari bakarra izango dugu, gure intraneteko mugan kokatuta, eta, intranetean sartu nahi duen konputagailuan VPN bezero bat egikaritu beharko da. Bezeroaren eta zerbitzariaren artean tunela ezarriko dute, eta tunelean zehar igorritako guztia zifratuko (baita, agian, sinatuko ere) dute. Tunela ezartzeko eta zifratzerako gakoak adosterakoan, bezeroa egikaritu duen erabiltzaileari bere nortasuna frogatzea eskatuko dio zerbitzariak, RAS zereginak betez.



5.16. irudia. VPN motak, topologiaren arabera.

TCP/IP metaren zein mailatan zifratzen den, bada, horren arabera ere sailka daitezke VPNak. Horrela, beste bi VPN talde hauek bereiziko ditugu:

- IP mailako VPNak: zifratzea IP mailan egiten da. Horretarako gehien erabiltzen diren protokoloak IPsec eta MPPE dira. Aurrekoa TCP/IP protokoloen taldekoa da, hau da, Interneteko estandarra (laster ikusiko duguna). MPPE (Microsoft Point to Point Encryption) protokoloak RC4 algoritmoa erabiltzen du PPP paketeak zifratzeko.
- Aplikazio-mailako VPNak: aplikazioko mezuen edukia zifratzen dute, IP eta garraio-mailako informazioa (zein konputagailu ari diren hizketan, eta zein portu —zein aplikazio, alegia— erabiltzen ari diren) agerian utziz. VPN mota honetarako gehien erabiltzen dena SSL/TSL da. Berez, SSL ez zen garatu VPNak gauzatzeko teknologia gisa, baizik eta garraio-mailako komunikazioa segurua izateko. Haren jatorrizko asmoa edozein aplikaziotako entitateen arteko komunikazioa seguru bilakatzea da. Baina bi entitate horiek VPN zerbitzariak eta bezeroak izan daitezke; hori da SOCKSVPN moduko SSLn oinarritutako aplikazioko VPNen funtsa.

L2TP eta PPTP

Hauek dira tunelak ezartzeko gehien erabiltzen diren VPNetarako protokoloak. Lehenengoa (Layer 2 Tunneling Protocol) Interneterako proposatutako estandarra da (RFC 3931). Bigarrena, Microsoftek sortutako PPTP (Point-to-Point Tunneling Protocol), CISCOk egindako L2F protokoloan dago oinarrituta. Guztien funtzionamendua tuneleko bi muturren artean PPP konexio bat ezartzean datza, gero datagramak konexio horretan zehar bidaltzeko. Bi protokoloen arteko aldeak honako hauek dira:

- PPTP prest dago IP sareekin lan egiteko (ohikoena dena) soilik. L2TPk edozein sare motaren gainean lan egin dezake.
- PPTPk tunel bakarra ezar dezake bi muturren artean. L2TPk tunel anitz ezar ditzake. Hau interesgarria izan daiteke trafiko motak bereizteko.
- L2TPk goiburukoak trinkotzea badu, eta PPTPk, berez, ez. Hala ere, PPTP MPPCrekin batera (Microsoft Point to Point Compression) erabiltzen denean, badago konprimitze hori egitea.
- Azkenik, eta garrantzitsuena, L2TPk IPsec erabiltzen du datagramak zifratzeko, eta PPTPk, aldiz, Microsoften jabegoko protokoloak erabiltzen ditu (MPPE, EAP-TLS…).

L2TPk UDP erabiltzen du (1701 portua du erreserbaturik). Horrela izanik, aproposagoa dirudi L5TP izenak L2TP baino.

5.5.2. IPsec (IP segurua)

TCP/IP metako IP mailari segurtasuna gehitzeko estandar multzo bat da IPsec (RFC dozena bat baino gehiago argitaratu dira). Bidalitako IP datagramen edukia zifratzea eta kautotzea ahalbidetzen du. IPv4rako definitu zen, eta, IPv6ren kasuan, protokoloaren zati bat denez, nahitaezkoa da IPsec inplementatzea. IPsec da VPNak gauzatzeko gehien erabiltzen den aukera. Funtsean, ondoko biak definitzen ditu:

- IP mailako konexio seguruak ezartzeko protokoloa.

  IPsec protokoloak konexio bidezko komunikazio-ereduari jarraitzen dio: bi konputagailuren arteko datagramen harremana segurua abiatu baino lehen, konexio bat ezarri bien artean. IPsec konexio hauek *akordioak* edo *segurtasun-loturak* izendatu ditugu, edo, askotan, bere ingelesezko siglen bidez (*SA,* Security Association) dira ezagunak. Sinplex erako konexioak direnez, bi noranzkoko komunikazioa edukitzeko horietako bi SA ezarri behar dira, bat komunikazioko noranzko bakoitzeko.

  IPsec akordioak ezartzeko (zifratzeko gakoen ezarpena barne) eta amaitzeko IKEv2 protokoloa (Internet Key Exchange) definitu egin da (RFC 4306). 2005eko bertsio honek lehenik zeuden ISAMKP eta IKE izenekoak, oso kritikatuak izan zirenak, ordezkatzen ditu.

- IP datagramari gehitu behar zaizkion goiburukoak.

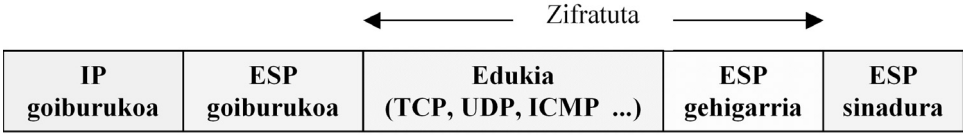
  Bi dira definitu direnak, lortu nahi den segurtasun motaren araberakoak. AH formatuan (Authentication Header*,* RFC 4302) datagramak sinatzen dira, osotasuna eta kautotzea bermatuz, baina ez dira zifratzen. AH goiburukoa, sinadura daramana, IP datagramaren edukiari gehitzen zaio (ikusi 5.17. irudia). AH goiburukoa hor dagoela adierazteko, 51 balioa eman behar zaio IP datagramaren goiburukoko *protokolo* eremuari.

IP goiburukoa	AH goiburukoa	Edukia (TCP, UDP, ICMP ...)

5.17. irudia. AH kapsulatze IP datagrama baten barnean.

ESP (Encapsulation Security Payload*,* RFC 4303) formatua harago doa, eta osotasunaz eta kautotzeaz gain, konfidentzialtasuna ere ematen du, datagramak daramana zifratuz. 5.18. irudian agertzen den bezala, ESP datagrama jatorrizko IP datagramako edukiari ESP goiburukoa eta bi gehigarri erantsiz sortzen da. Gero, hori guztia IP datagrama arrunt batean sartzen da. IP datagramaren goiburukoko protokolo-eremuari 50 balioa emango zaio, helburuko IP mailari barruan dagoena ESP datagrama bat dela adierazteko. 5.18. irudiak adierazten duen bezala, jatorrizko IP datagramaren edukia eta ESP lehenengo gehigarria zifratzen dira, baina ez ESP goiburukoa ezta bigarren gehigarria ere. Goiburukoa ez da zifratu behar, hain zuzen ere, hor dagoelako SPI zenbakia, SA identifikatzen duena, eta deszifratzeko jakin

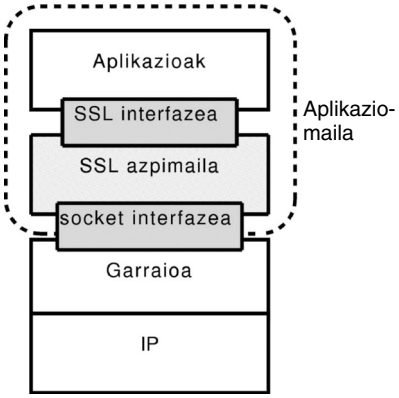
behar dugu ea zer SAri dagokion datagrama, gako zuzenak erabiltzeko deszifratzean. Bigarren gehigarrian sinadura dago. Sinadura ez zifratzeak ez du konfidentzialtasuna kolokan jartzen, eta, beste alde batetik, eraginkorragoa da ez zifratzea, igorleak paraleloan egin dezakeelako sinadura eta datagrama zifratzea.



5.18. irudia. ESP goiburukoa eta gehigarriak IP datagrama batean.

5.5.3. Garraio-maila segurua: SSL/TLS

SSL (Secure Sockets Layer) protokoloa Netscape konpainiak garatu zuen 1995. urtean, bere web zerbitzari eta arakatzaileen arteko komunikazioak seguruak izateko. Gaur egun arakatzaile eta zerbitzari guztiek erabiltzen dute SSL edo bere ordezkoa den TLS (Transport Layer Security, RFC 5246) eta, are gehiago, bere erabilgarritasuna ez dago mugatuta webean, edozein aplikaziorekin erabil baitaiteke. SSLk API (Application Programming Interface) bat eskaintzen die aplikazioei, socket interfazearen gainetik. 5.19. irudiak adierazten duenez, SSL aplikazio-mailan kokatzen bada ere, programatzaileari garraio-mailan balego bezala agertzen zaio.



5.19. irudia. SSL interfazea TCP/IP arkitekturan.

SSLk aukera asko ematen ditu. Hauta daiteke, besteak beste, konpresioa erabili ala ez, zer algoritmo kriptografiko erabili nahi dugun, eta nolako gakoak erabiliko diren.

SSL komunikazioetan bi fase daude: konexio segurua ezartzea eta konexio hori erabiltzea. Konexio segurua ezartzeko urratsak 5.20. irudian agertzen dira. Honako hauek dira:

(1) Ainhoak Beñati SSL konexio-eskaera bidaltzen dio. Horretan erabilitako SSL bertsioaren berri ematen du, baita erabil ditzakeen algoritmo eta gakoena ere. Gainera, α zenbaki bat ere bidaltzen du, kautotzeko protokoloetan erabiltzen diren zenbakien moduan sortua.

5.20. irudia. SSL konexioa ezartzeko azpiprotokoloa. Sinplifikatuta dago.

(2) Beñatek Ainhoak adierazitako kriptografia eta konpresio-aukeren artean aukeratzen du, eta hautatutakoak bidaltzen ditu. Horrekin batera bere kautotze-zenbakia, β, eta SSL bertsioa ere bidaltzen ditu.

(3) Beñatek bere ziurtagiria bidaltzen du. Beharrezkoa bada, ziurtagiri-kate osoa bidaliko du, eta ez ziurtagiri bakar bat.

(4) Orain, Beñatek nahi badu, beste mezu bat bidaliko dio Ainhoari, bere ziurtagiria eskatzeko. Errealitatean, ordea, normalki zerbitzaria bakarrik kautotzen da. Horrela egiten da adibide honetan eta, beraz, orain, Beñatek beste mezu bat bidaltzen dio Ainhoari, bere aldetik nahikoa dela esanez.

(5) Ainhoak zenbaki berri bat sortzen du, γ, eta Beñaten gako publikoa erabiliz zifratuko du. Horren emaitza Beñati bidaliko dio. γ, α eta β zenbakietatik abiatuta, lan-saiorako gako simetrikoa kalkulatuko dute Ainhoak eta Beñatek. Mezua jaso eta gero, bi aldeek daukate behar duten gako simetrikoa.

(6) Ainhoak zifratzen hasteko prest dagoela adierazten du.

(7) Ainhoak konexioa ezartzeko fasearen bukaera jakinarazten du.

(8) Beñatek Ainhoaren 6. urratseko mezuaren onespena bidaltzen du.

(9) Beñatek Ainhoaren 7. urratseko mezuaren onespena bidaltzen du.

Konexioa ezarrita, aplikazioa mezuak bidaltzen has daiteke. SSL mailak zatituko ditu mezuak, gehienez ere 16 KB-eko pusketatan eta, garraio-mailari pasatu baino lehen, zati bakoitza konprimitu, sinatu, zifratu eta bere SSL goiburukoa gehituko dio.

LABURPENA

Sarea erabiltzen duten konputagailu guztiek kontuan hartu behar dituzte erabilera horrek sortzen dituen arriskuak. Hackerrak, birusak eta abar, gero eta sofistikatuagoak eta ugariagoak dira. Horien eraginetik babestuta egoteko, alde batetik sareak segurua izan behar du, eta, bestetik, sarea erabiltzen duten aplikazioek ere seguruak izan behar dute. Sarea segurua izateak sarrera-kontrola egiten dela eta sarearen segurtasuna ondo kudeatuta dagoela esan nahi du. Aplikazioak seguruak izateak burututako komunikazioak seguruak direla esan nahi du.

Sarrera-kontrolaren oinarriak suhesiak eta mugasareak dira. Sareko segurtasunaren kudeaketarenak, sare-segurtasunerako arauak eta prozedurak. Prozedurarik garrantzitsuenak arrisku-analisia, gertaeretarako jarraibideak, ekipoen konfiguraziorako jarraibideak, eta monitorizaziorako prozedura dira beharbada.

Komunikazio seguruaren hiru baldintzak konfidentzialtasuna, kautotasuna eta osotasuna dira. Hirurak bermatzeko oinarrian dago kriptografia. Bi motatako kriptografia dugu: simetrikoa eta asimetrikoa. Kriptografia simetrikoan gako bakarra erabiltzen da mezuak zifratzeko eta deszifratzeko. Kriptografia asimetrikoan, aldiz, bi gako daude; batek zifratzen duena besteak deszifratzen du. Horietako gako bat publikoa izango da, eta bestea ezkutua. Kriptografia simetrikoa azkarragoa da asimetrikoa baino, eta horregatik erabiltzen da normalki konfidentzialtasuna bermatzeko. Horretarako gehien erabiltzen diren estandarrak 3DES eta AES berria dira. Kautotze-protokoloek gakoen banaketaren arazoa dute. Kriptografia simetrikoa erabiltzen bada, gako-zerbitzariek konpon dezakete arazoa, baina inguru lokaletan besterik ezin dira erabili. Testuinguru irekietan, Internet bidezko komunikazioetan, alegia, kriptografia asimetrikoa erabili behar da. Kautotze-protokolo horien fidagarritasuna ziurtagirien erabileran datza. Ziurtagiri elektroniko bat agiri bat da (elektronikoa, noski), non gako publiko bat eta bere jabea uztartzen diren. Ziurtagiriaren fidagarritasuna, berriz, bere sinaduran datza. Sinadura elektronikoak sinadura fisikoak lortzen dituen berme berberak ahalbidetzen ditu mundu elektronikoan erabilitako agirietarako. Hash funtzioen eta kriptografia asimetrikoaren bidez sortzen dira sinadura elektronikoak. Horretarako gehien erabiltzen diren estandarrak MD5 eta SHA-1 (hash) eta RSA (algoritmo asimetrikoa) dira.

VPNek sare publiko baten bidez (gehienetan Internet) gure sareko bi zatiren arteko komunikazio segurua bermatzeko balio dute. Ideia da konektatu behar diren bi muturren artean tunel bat egitea, gero tunel horretatik bidaliko den informazio guztia zifratzeko. Tunelak egiteko gehien erabiltzen den protokoloa L2TP da, eta zifratzeko gehien erabiltzen dena IPsec da. IPsec IP protokoloaren bertsio segurua da.

SSL/TLS da, beharbada, gaur egun gehien erabiltzen den sistema kriptografikoa. Zifratzea garraio-mailan egiten du. Oso erabilia da arakatzaile eta web zerbitzarien arteko komunikazioetan, baina edozein aplikaziorekin erabiltzeko diseinatuta dago. SSLren fidagarritasuna ziurtagiri elektronikoen erabileran oinarrituta dago.

6. Eranskina: socket interfazea

6.1. BERKELEY SOCKETAK

Gogoan izan konputagailu berean dauden bi maila ezberdinetako entitateen arteko komunikazioa beheko mailaren interfazearen bidez egiten dela. Sare-arkitekturaren inplementazio bakoitzak bere interfaze propioak ditu, interfazeak ez baitira arkitekturaren definizioan sartzen. Horregatik, sistema eragilearen barnean dagoen IP entitateak sarbide-maila atzitzeko, sistema eragileak sare-txartelari dagokion driverra eskatuko digu. Garraio-mailaren eta IP mailaren arteko komunikazioa sistema eragilearen araberakoa da, biak sistema eragilearen barnean baitaude. Garraio-mailaren eta aplikazioen arteko komunikazioa ere sistema eragilearen araberakoa izango da, sarea baita aplikazioek eskuragarri duten sistemaren beste baliabide bat.

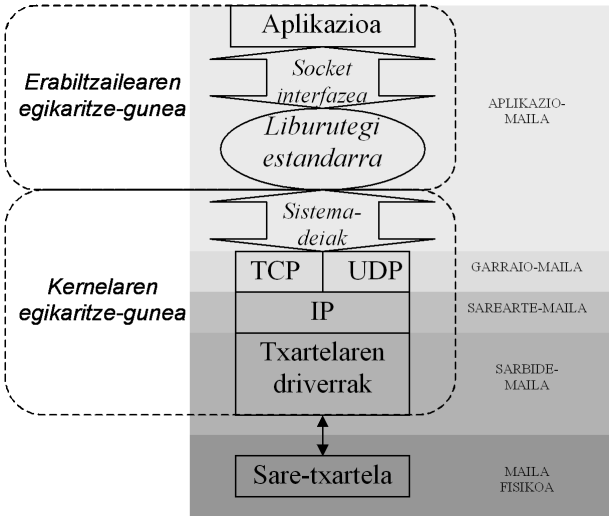
TCP/IP sare-arkitektura, bere sorreratik, oso lotuta egon da Unix sistemekin. Arkitekturaren lehenengo inplementazio publikoa 1983. urtean plazaratu zen, 4.2 BSD Unix sistemaren barnean. Unix inplementazio horretan **socket interfazea** ere agertu zen, sarearen bidezko komunikazioa konputagailu bereko beste edozein bi prozesuren arteko komunikazioa bezala egiteko asmoarekin. Hasierako API hau (Application Programming Interface) C programazio-lengoaiarako liburutegi bat zen, *Berkeley sockets* izenarekin ere ezagututa. Hortik aurrera, TCP/IPren arrakasta eta hedapenarekin batera, socket interfazea zabaldu da, eta gaur egun sare-aplikazioak sortzeko erabiltzen den ia bakarra da. TCP/IP arkitekturarekin gertatu den bezala, socketak ere Unix ez diren beste sistema eragile guztietan inplementatu dira. C ez diren beste programazio-lengoaia askotarako, C-rako Berkeley socketsen antza handiko APIak ere agertu dira.

Socketen inplementazioa Unix sistemetan

Sistema eragile guztiek badute interfaze bat aplikazioei kernel zerbitzuak eskaintzeko. Unix sistemetan sistema-deiak dira. Kernel zerbitzuen artean sare-zerbitzuak daudenez, sistema-deien azpimultzo bat dago sare-komunikazioak egiteko; azpimultzo hori socket interfazea da. Dei horiek prozesuen arteko komunikazioetarako erabiltzen diren sistema-deien multzoaren zati bat dira.

Dena dela, aplikazioek ez dituzte normalki sistema-deiak zuzenean erabiltzen. Egiten dena da aplikazioaren programazio-lengoaiaren liburutegi estandarrean funtzio bat sartzea, sistema-deiaren izen berekoa. Aplikazioak liburutegiko

funtzioari deitzen dio, eta horrek sistema-deia erabiliko du. Bitartekari hori sartzeak agindutakoa betetzeko denbora gehiago eskatzen du, baina bere alde onak ere baditu. Adibidez, sistema eragilearen bertsioa aldatzen denean, sistema-dei berriak erabiltzeko nahikoa da liburutegi estandarra eguneratzea, baina aplikazioa ez da ukitu behar. e.1. irudian osagai hauen guztien arteko erlazioa agertzen da.



e.1. irudia. TCP/IP arkitekturaren inplementazioa Unix sistemetan.

Ondoko taula honetan socket interfazearen funtziorik garrantzitsuenak agertzen dira. Eranskin honen beste atal batean aztertuko ditugu funtzio horiek.

Funtzioaren izena	**Zeregina**
socket()	Komunikazio-puntu bat sortu.
bind()	Garraio-mailako helbide bat socket batekin lotu. Normalki zerbitzariek bakarrik erabiltzen dute.
listen()	Konexio-eskaerak jasotzeko ilara gaitu.
accept()	Socket batean konexio-eskaera baten zain gelditu (konexio-ezarpen pasiboa).
connect()	Konexio-eskaera bidali (konexio-ezarpen aktiboa).
send()	Datuak bidali socket baten bidez.
receive()	Datuak jaso socket baten bidez.
close()	Socketa deuseztatu.

e.1. taula. C lengoaiarako socket interfazearen funtzio batzuk. C liburutegi estandarrean barruan daude. Javarenak oso antzekoak dira.

Zer da socket bat?

Sare-zerbitzuak erabiltzeko ate bat da, komunikazio-puntu bat, alegia. Unix taldeko sistemetan, aplikazio batek TCP/IP sarea erabili nahi badu, egin behar duen lehenengo gauza socket bat sortzea da, *socket()* deia erabiliz. Horrekin batera, sistema eragileak socketari dagokion datu-egitura sortuko du, eta datu-egitura hori identifikatzen duen deskriptorea aplikazio-prozesuaren deskriptore-taulan erantsiko du. Hurrengo batean, aplikazio-prozesuak socketa erabili behar duenean, deskriptore horren bidez erreferentziatuko du. Hori da, zehatz-mehatz, Unix sistemetan sistemaren beste edozein baliabide (fitxategiak, diskoak…) erabiltzeko mekanismoa. Hori da socket interfazearen helburuetako bat: sarea erabiltzea beste edozein baliabide erabiltzea bezala izatea.

Socketaren datu-egituran bi buffer egoten dira (ez beti, socket motaren araberakoa baita), bata bidaltzeko eta bestea jasotzeko. Prozesu batek socket baten bidez informazioa bidaltzeko dei bat egiten duenean, sistema-deiak informazio hori socketari dagokion irteerako bufferrean kopiatuko du, eta kernelen prozesu batek (gehienetan UDP edo TCP prozesua, baina badaude aukera gehiago) socket horretan datuak bidaltzeke daudela ohartaraziko du. Era berean, socket horri bidalitako segmentu bat heltzen denean, garraio-mailako prozesuak segmentuaren edukia socketari dagokion sarrera-bufferrean kopiatuko du, eta aplikazio-prozesuari horren berri emango dio.

Socket baten domeinua

Egia esan, socketek ez dute sare-komunikazioetarako bakarrik balio, edozein bi prozesuren arteko komunikazioetarako baizik. Horregatik, socket bat sortzen dugunean, sistemari socket hori nolako komunikazioetarako erabiliko dugun adierazi behar diogu. Socket horren bidez komunikatuko diren prozesuak konputagailu berean egikaritzen badira, socketa domeinu lokalekoa dela esango dugu. PF_LOCAL sistemaren konstanteak identifikatzen du domeinu lokala (Protocol Family LOCAL). Prozesu horiek TCP/IP sare baten bidez konektatuta dauden bi konputagailu desberdinetan kokatzen badira, socketak Internet domeinukoa (PF_INET konstanteak adierazten duena) izan beharko du. Domeinu horren socketak dira hemen landuko ditugunak. Domeinu gehiago badaude, baina liburu honetan ez zaizkigu interesatzen.

Socketaren domeinuak arlo hauek ezartzen ditu:

- Socket beraren identifikazioaren formatua. PF_INET domeinuko socketak IP helbide batek eta portu batek osatzen duten bikote baten bidez identifikatzen dira. IP + portu bikote horri socketaren helbide deitzen zaio (ez nahastu socketaren deskriptorearekin). Helbide-formatu hori AF_INET (Address Family InterNET) konstanteak identifikatzen du.

- Socket horren bidezko komunikazioetan erabil daitezkeen protokoloak. PF_INET domeinuko socketetan TCP eta UDP dira protokolo horiek. Egia esan, protokolo gehiago daude eskuragarri PF_INET socketetan, baina une honetan ez zaizkigu interesatzen.

Socket baten helbidearen formatua

Aplikazioak socket baten helbidea erabili behar duenerako, datu-egitura estandar bat definitu da. PF_INET domeinuko socketen kasuan *sockaddr_in* egitura da hori. Bere definizioa goiburuko fitxategi orokorretan dago, *netinet/in.h* fitxategian Linux sistemaren kasuan. Ondoko lau eremu hauek ditu:

- sin_family: bere balioak sistemaren konstante bat izan behar du, zein helbide mota erabiliko den adieraziz. PF_INET domeinuko helbideak, IP + portu bikoteak alegia, AF_INET konstanteak identifikatzen ditu.
- sin_port: socketari lotutako portua. Socket honetatik eskaerak bidali behar baditugu, beste aldearen portua adierazten da hemen (gehienetan zerbitzariarena dena). Socketatik eskaerak jaso behar baditugu (ezarpen pasiboak), aldiz, gure portua adierazten da. Bigarren kasu honetan, aplikazioak zein portu dagokion baldin badaki (adibidez, erreserbatua denean), aplikazioak berak beteko du eremua. Bestela, sistema eragileak esleitutako portu dinamiko bat bada, aplikazioak eskatu behar dio sistemari eremu horretan esleitutako portu dinamikoaren balioa grabatzeko (*getsockname* funtzioa erabiltzen da horretarako).
- sin_addr: socketari dagokion IP helbidea. Lehen bezala, beronen balioa alda daiteke socketa eskaerak bidaltzeko edo jasotzeko den aintzat hartuta. Socketa aktiboki erabiltzen bada (eskaerak bidaltzeko), aplikazioak eremu honetan beste aldeko socketaren IP helbidea grabatuko du. Bestela, bere IP helbidea jarri behar du.
- sin_zero: hau eremu betegarri bat da. Mota desberdinetako socket helbideek oso luzera desberdina dutenez, sistemak eremu hau erabiltzen du helbide mota guztiekin datu-egitura bera erabiltzeko. Aplikazioek ez dute eremu hau ezertarako erabiltzen.

Bezeroek datu-egitura hau erabili ohi dute datagrama bat bidali behar dutenean edo konexio bat ireki nahi dutenean sistemari zerbitzariaren helbidea adierazteko.

Zerbitzariek, aldiz, sistemari bere socketaren helbidea adierazteko erabiltzen dute datu-egitura hau, konexio-ezarpen pasiboa egiten dutenean, edo UDP eskaerak jasotzeko socketa sortzen dutenean.

Sare-sintaxia eta sintaxi lokala

Konputagailuek era desberdinetan gordetzen dituzte datuak beren barneko memorietan. Batzuek *big endian* izeneko era erabiltzen dute, eta besteek, aldiz, *little endian*. Lehenengo taldeko konputagailuetan, zenbaki bat osatzen duten byteak handitik txikira ordenatzen dira, hau da, balio handiena duen bytea memoriako helbide txikienean kokatzen da. *Little endian* erako konputagailuetan kontrako eran interpretatzen dira zenbakiak: memoriako helbiderik baxuenean balio txikiena duen bytea kokatzen da.

Horrek guztiak arazo bat sortzen du bi konputagailuen arteko komunikazioetan. Igorleak datagrama batean portu bat grabatzen duenean, bere barneko sintaxi lokalari jarraituz egingo du eta, horren arabera, lehenengo bytea balio handienekoa edo txikienekoa izango da. Hartzaileak, beste aldetik, bere sintaxi lokalari jarraituz interpretatuko du jasotakoa. Bi konputagailuek ez badituzte datuak era berean interpretatzen, jai dute elkar ulertzeko.

Arazoa konpontzeko, komunikazio-protokoloak komunikaziorako sintaxi komuna ezarri behar du, edo **sare-sintaxia**. TCP/IP arkitekturan zenbakientzat erabiltzen den sintaxia *big endian* da. Beraz, aurreko datu-egituraren *sin_port* eta *sin_addr* eremuen balioa *big endian* eran idatzi behar da, konputagailuaren sintaxi lokala edozein izanda ere. Konputagailu era bakoitzeko aplikazioen bertsio desberdina sortzea eta erabiltzaileari zein den bere makinaren barruko sintaxia jakin beharra ekiditearren, C liburutegi estandarrean badaude funtzio batzuk behar diren sintaxi-bihurketak egiteko. Programatzaileak funtzio horiek erabili behar ditu bere socketaren helbidea grabatzean (sintaxi lokala sare-sintaxi bihurtzeko), baita beste muturreko socketaren helbidea irakurtzean ere (sare-sintaxia bere sintaxi lokalera bihurtzeko). Lehenengo bihurketan *htonl*() eta *htons*() funtzioak erabiltzen dira (Host TO Network Short, Host TO Network Long); sare-sintaxitik sintaxi lokalera pasatzeko, *ntohs()* eta *ntohl()* (Network TO Host Short, Network TO Host Long). Funtzio horien erabilerari buruzko xehetasunak ezagutzeko, sistemaren laguntza erabili (man komandoa Unix sistemetan).

6.2. SOCKETEKIN LAN EGITEKO OINARRIZKO FUNTZIOAK

e.1. taulan agertzen diren funtzioen eta beste batzuen erabilera aurkeztuko dugu atal honetan. Horri buruzko xehetasun gehiago ezagutzeko, sistemaren laguntza erabili behar da.

Socketa sortu: socket()

Dei honetan socketaren ezaugarriak adierazi behar dira. Hauek dira:

- Socketaren domeinua. Ikusi dugunez, TCP/IP inguruan lan egiteko PF_INET adierazi behar da.
- Socket mota. Honekin aplikazioak behar duen zerbitzua adierazi behar diogu sistemari. PF_INET domeinuan aukera asko daude, eta batzuk ez daude sistema eragile guztietan erabilgarri. Baina oinarrizkoenak diren garraio-mailako zerbitzuak hor beti daude: konexioaren bidezko komunikazioa eta datagrama erako komunikazioa. Lehenengoa SOCK_STREAM izeneko sistemaren konstantearekin identifikatzen da, eta bestea SOCK_DGRAM konstantearekin.
- Protokoloa. Batzuetan, zerbitzu bat emateko protokolo bat baino gehiago erabil daitezke. Kasu horietarako, aplikazioak badu sistemari esatea zein protokolo erabili behar duen. SOCK_STREAM eta SOCK_DGRAM zerbitzuetarako protokolo horiek finkatuta daudenez (TCP eta UDP, hurrenez hurren), programatzaileak 0 balioa adierazten du protokolo gisa.

Socketa eta helbidea lotu: bind()

Socket bat sortzen dugunean (*socket()* deia erabiliz) ez diogu inongo IP helbiderik ezta porturik ere esleitzen. Dei honek esleipen hori egiteko balio du.

Socketa era aktiboan erabiltzen badugu (bezeroek egiten dutena), jaso baino lehen socketaren bidez bidaliko dugu; lehenengo bidalketa hori egitean, sistemak socketari dagokion helbidea esleituko dio eta, beraz, dei hau ez da erabili behar. Horregatik bezeroek ez dute normalki dei hau erabiltzen (badaude salbuespenak).

Erabilera pasiboan, aldiz, sistemari esan behar zaio IP eta portu konkretu bat dakarren segmentu baten informazioa socketaren zein sarrera-bufferretan sartu. Beraz, erabilera pasiborako *socket()* deia egin eta gero, zerbitzariek beren helbide propioa *in_addr* datu-egitura batean gorde behar dute, eta gero sortutako socketa helbide horrekin lotu, *bind()* erabiliz. Aurretik inongo porturik esleituta ez duten aplikazioen zerbitzariek 0 zenbakia adierazi behar dute helbidearen *sin_port* eremuan, sistemak berak aukeratutako portu dinamiko bat socketari esleitu diezaion.

Socket baten helbidea lortu: getsockname()

Gure socketari helbidea sistemak esleitu badio, eta programak zein den helbide hori jakin behar badu, dei hau erabiltzen da.

Socketa konexio-ezarpen pasiborako prestatu: listen()

Dei honen bidez aplikazioak (normalki zerbitzariak) sistema ohartarazten du socket batean konexio-eskaerak jasoko direla. Sistemak socket horretarako eskaera-ilara bat prestatuko du. listen() dei baten bidez markatutako socket bati **adi-socketa** esaten zaio (*listening socket*). Gero ikusiko dugunez, socket hauen

bidez ez dago inoiz informazio-trafikorik; adi-socketetan hiru urratseko akordioari dagokion trafikoa bakarrik ibiltzen da.

SOCK_STREAM motako socket bat *listen()* baten bidez markatzearekin batera, prozesu-zerbitzaria konexio-eskaeren zain gelditzen da (ezarpen pasiboa). Hortik aurrera, adi-socketerako konexio-eskaera heltzen denean, hiru urratseko akordioa betetzen bada, socketari lotutako ilaran sarrera bat gehituko du sistemak.

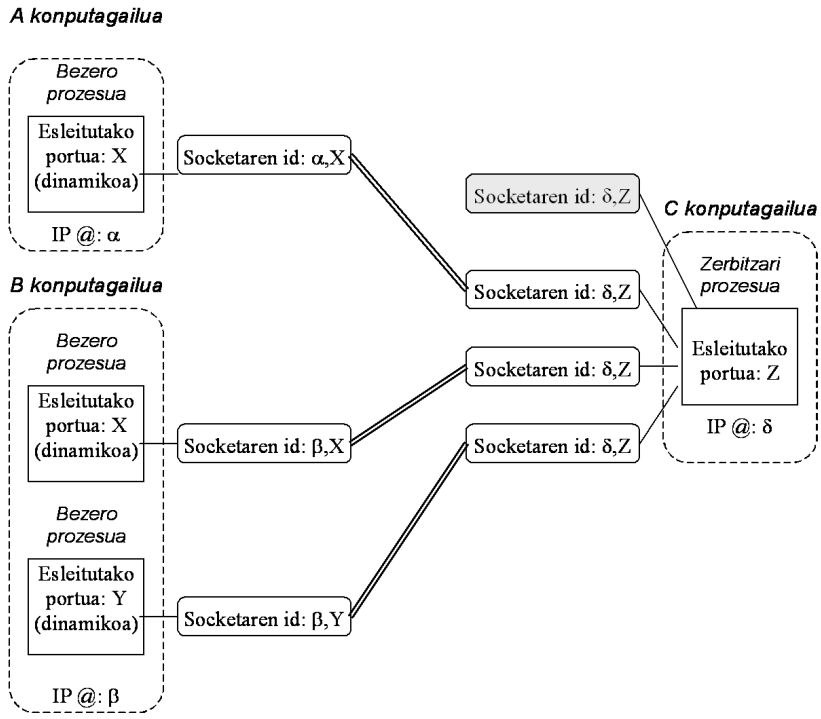
Konexio-ezarpen pasiboa: accept()

TCP zerbitzari batek *accept()* deia erabiliko du adi-socket bati lotutako ilaran dagoen lehenengo sarrera ateratzeko. Ilara hori hutsik badago, zerbitzaria hor gelditzuko da blokeatuta, sarrera bat sortu arte (hori da besterik ezean egiten dena; zerbitzaria ez blokeatzeko aukera ere badago). Ilaran zerbait badago, sistemak socket berri bat sortuko du, **socket konektatua** deituko duguna (*connected socket*), eta horren deskriptorea itzuliko dio *accept()* egin duen prozesuari. Hortik aurrera, aplikazioak badu datuak igortzea eta jasotzea socket konektatu horren bidez.

Adi-socketak eta socket konektatu berriak helbide bera dutela ohartu. Are gehiago: zerbitzari batek onartzen duen konexio bakoitzeko, socket konektatu desberdin bat sortuko da, eta socket horiei guztiei IP helbide bera eta portu bera dagozkie. TCP entitateak (sistema eragileak) bereizi beharko du zein socket konektaturi dagozkion helbide eta portu horretara heltzen diren segmentuak. Horretarako TCP goiburukoan datorren sorburuko helbidea erabiliko du. Aplikazioak ere badu jakitea nor dagoen socket konektatu bakoitzaren beste muturrean, zeren *accept()* deiaren parametroetako batean grabatuko baitu sistemak konexioa eskatu duen bezeroaren identifikazioa.

e.2. irudian ikus daiteke zein socket multzo behar den zerbitzari batek une berean 3 bezerori kasu egiten dienean. Irudiko hiru bezero bi makina desberdinetan egikaritzen dira. A eta B konputagailuetan dauden bi bezerok portu dinamiko bera erabiltzea (X portua) kasualitate handia da, baina guztiz zilegi.

Sistemak socket konektatuak horrela ugalduko ez balitu, zerbitzariak une batean ezarrita dituen bere konexio guztien bidalketak adi-socket bakarretik jasoko lituzke, eta berak (aplikazio-mailako zerbitzariaren softwareak) bereizi beharko luke mezu bakoitza zein bezerori dagokion. Baina aplikazio-mailak ez du horretarako behar den informazioa (sorburuko IP helbidea eta TCP portua). Horregatik erabili behar dira adi-socketak eta socket konektatuak zerbitzariaren aldean. Bezeroak, bere aldetik, ez du behar bi motatako socketen mekanismo hau, berak sortutako socket bakoitzari sistema eragileak helbide desberdina (portu dinamiko desberdina) esleituko diolako.



e.2. irudia. Adi-socketa eta socket konektatuak zerbitzari batean. Adi-socketa itzalean dago.

Konexio-ezarpen aktiboa: connect()

Normalki TCP zerbitzua erabiltzen duten bezeroek erabiltzen dute dei hau (badaude beste erabilera batzuk). Hiru urratseko akordioa abiatzen du, argumentuetan adierazitako helbidearekin konexio bat ezartzearren.

Socketa deuseztatu: close()

Prozesuaren deskriptore-taulatik kenduko dio socketa. Socket konektatua bada, hori egin baino lehenago konexio-amaiera bati dagozkion segmentuak elkarri bidaliko dizkiote bi garraio-entitateek, informazio-galerarik ez dela egongo bermatuz.

Dei honek ez du bezeroa blokeatzen: sistemak berehala itzultzen dio kontrola, eta bere kabuz egingo du konexioaren amaiera (egin behar bada).

SOCK_STREAM motako socket baten bidez datuak igorri: write()

Adierazitako socket konektatuaren irteerako bufferrean kopiatzen ditu datuak. Sistemak jasoko ditu datuak eta segmentu batean edo gehiagotan igorriko ditu. Ez

dago erlazio zuzenik *write* deien eta segmentu-bidalketen artean; *write* egiteak segmentu bat, segmentu batzuk edo inongo segmenturik ez bidaltzea ekar dezake. Era berean, segmentu batean doazen datuak irteerako bufferrean kopia daitezke, *write* bakar batean edo gehiagotan.

Send() eta *sendto()* deiek ere gauza bera egiteko balio dute, eta TCPren aukerak hobeto kontrola ditzake aplikazioak horiek erabiliz. Baina gehiago erabiltzen da *write()*, bere helburua orokorragoa delako eta, beraz, programatzaileen artean ezagunagoa delako.

SOCK_STREAM motako socket baten bidez datuak jaso: read()

Aplikazioak socketari dagokion sarrerako bufferrari kasu egiten dio. Ezer ez badago, datuen zain blokeatzen da (ez blokeatzea ere badago, baina ez da erabilera arrunta). Ez dago overflow arriskurik sarrerako bufferrean, TCP lokalak beste aldeari emandako kreditua bere tamainakoa delako, hain zuzen ere.

Recv() edo *recvfrom()* ere erabil daitezke datuak jasotzeko baina, *write()* kasuan gertatzen den bezala, ohikoena *read()* erabiltzea da.

SOCK_DGRAM motako socket baten bidez datuak igorri: sendto()

Lehen aipatu dugun bezala, dei hau SOCK_STREAM socketekin ere erabil daiteke. Deiaren parametroetan esplizituki adierazi behar da helburuko socketaren helbidea. Ohartu socket konektatuetan *write()* deia erabiltzen dugunean hori ez dela beharrezkoa.

Sendmsg() deia ere erabil daiteke. Honekin UDP segmentu bat baino gehiago bidal daiteke helbide batera. *Sendto()* deia batzuk jarraian egitea baino eraginkorragoa da, baina haren erabilera konplexuagoa ere bada.

Kasu honetan badago erlazio zuzena *sendto()* deien eta bidalitako UDP segmentu kopuruaren artean. Horregatik, programatzaileak kontuz ibili behar du dei bakar batean UDP segmentu batean sartzen direnak baino byte gehiago ez emateko. Horrela egiten badu, errore bat egongo da eta ez da inongo segmenturik bidaliko. Aplikazioak ezar dezake UDP segmentuaren gehienezko tamaina hori (*setsockopt()* deia erabiliz), baina sistemak onartzen dituen mugen artekoa izan behar du. Sistema bakoitzak baditu bere mugak, 9.000 byte inguru izaten direnak. Zoritxarrez, ez dago bide estandarizatu eta argia muga horiek zeintzuk diren jakiteko. Normalki sistemaren konstante bat da, Unix erako sistemetan limits.h fitxategian gordetzen dena. UDP segmentuak bidaltzean, sarbide-sareak dituen mugak errespetatzea programazioko ohitura ona da. Horrela ekidingo dugu IP datagramak beren sorburuko sarean bertan zatitzea. Gure konputagailua Ethernet sare batean baldin badago, adibidez, aplikazioak gehienez ere 1.472 byteko UDP segmentuak bidali beharko lituzke.

SOCK_DGRAM motako socket baten bidez datuak jaso: recvfrom()

Hau ere SOCK_STREAM motako socketen bidez datuak jasotzeko erabil daiteke. Bere portaera *read()* deiaren antzekoa da, baina kontuan izan UDP erabiltzen duten komunikazioetan (eta SOCK_DGRAM motako socketek erabiltzen dute) ez dagoela inongo fluxu-kontrolik eta, beraz, igorlea hartzailea baino azkarragoa denean, sarrera-bufferrean overflow gerta daiteke. Hori gertatzen bada, informazioa galdu egingo da, eta garraio-mailak ez dio aplikazioari inongo berririk emango.

Beste aukera bat *recvmsg()* erabiltzea da. Dei honek segmentu bat baino gehiago jasotzeko balio du.

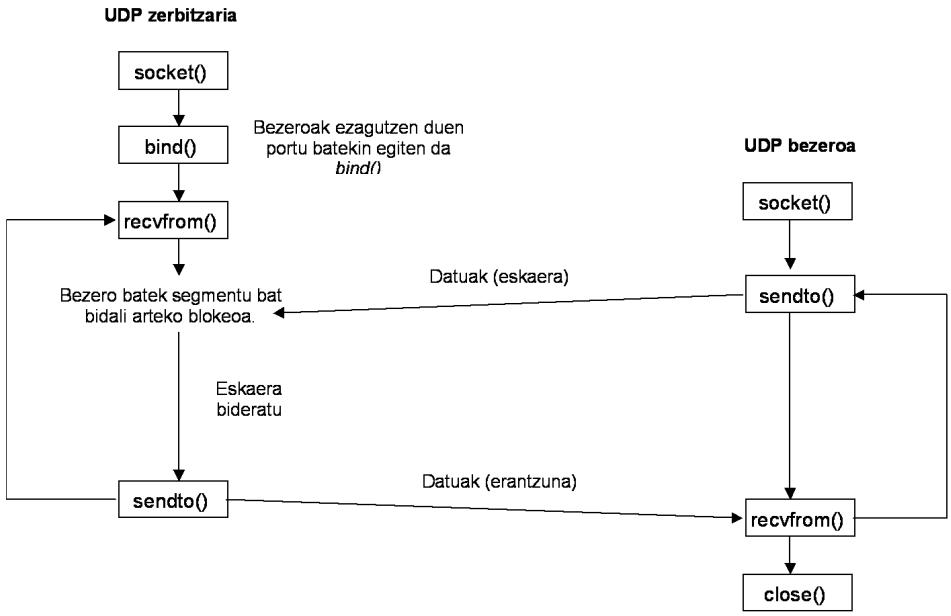
Socketekin lotutako beste dei interesgarriak

- IP helbideen erabilerarako: *inet_aton(), inet_ntoa()*. Karaktereen eta zenbaki osoen barruko sintaxiaren arteko IP helbideen itzulpena egiten dute.
- Konputagailuen identifikazioa: *gethostbyname(), gethostbyaddr()*. Konputagailu bati buruzko informazioa lortzen da, tartean bere IP helbideak. Batak konputagailuaren izena erabiltzen du, eta besteak bere IP helbide bat.
- Socket baten ezaugarriak lantzeko: *setsockopt(), getsockopt()*, fcntl(), *ioctl()*.

Dei hauen erabilerari buruzko informazio zehatza lortzeko, erabili sistemaren programaziorako laguntza (*man* komandoa, Unix erako sistemetan).

Socketen erabilera datagramen bidezko aplikazioetan

e.3. irudian bi muturrek lehen aztertutako deien erabilera egiten dute. Irudian agertzen dena ez da erabilera bakarra: aplikazio bakoitzak berea egingo du, baina irudikoa da gehien ikusiko duguna.



e.3. irudia. Datagramen bidezko ohiko komunikazioa.

Konexiorik gabeko zerbitzari baten eta bezero baten adibidea

Sekzio honetan UDP erabiltzen duen aplikazio baten programazioa ikus dezakezu (zerbitzaria eta bezeroa), C lengoaian. Ondoko irudietako zerbitzariak eta bezeroak e.3. irudiko urratsak betetzen dituzte. Agertuko diren programazioko beste adibideak bezala, C lengoaian daude eginda.

Bezeroa eta zerbitzaria konputagailu berean egikaritu ditzakegu, erosoago lan egiteko. Erabiltzaileak bezeroari eman behar dizkion parametroak bi dira: zerbitzariaren konputagailuaren izena eta zerbitzariaren portua. Zerbitzariak ez du parametrorik behar.

```
# include <sys/types.h>
# include <sys/socket.h>
# include <netinet/in.h>
# include <stdio.h>

#define DATA "erantzuna hau da.\n"

main()
{
       int sock, luze, luze2;
       struct sockaddr_in helb, bez_helb;
       char buf[1024];
```

```
// Socketa sortu
       sock=socket(PF_INET, SOCK_DGRAM, 0);
       if (sock<0)
     {
               perror("datagramako socket-a sortzen");
               exit(1);
               }

// Zerbitzariaren socketaren helbidea eraiki
       helb.sin_family=AF_INET;
       helb.sin_addr.s_addr=htonl(INADDR_ANY);
       helb.sin_port=htons(0);

// Eraikitako helbidea esleitu socketari
       if (bind(sock, (struct sockaddr *)&helb, sizeof helb)< 0)
               {
          perror("socket-i izena ematen");
               exit(1);
               }

// Lortu socketari esleitutako portua, eta pantailan atera
       luze = sizeof helb;
       if (getsockname(sock, (struct sockaddr *)&helb,&luze) < 0)
               {
      perror("socketaren identifikazioa lortzen");
               exit(1);
               }
       printf("Erabilitako portua: ->%d\n",ntohs(helb.sin_port));

// Itxaron bezeroak eskaera bat bidali arte. Orduan jaso
// bidalitako mezua eta pantailan atera.
       if (recvfrom(sock, buf, 1024, 0,(struct sockaddr *) &bez_helb,
&luze2) < 0)
               perror("datagrama bat jasotzen");
       printf("%s\n", buf);

// Itzuli erantzuna bezeroari
    if (sendto(sock,DATA,strlen(DATA) + 1 ,0,(struct sockaddr
*)&bez_helb,sizeof bez_helb)<0) perror("datagrama erantzuten");
// Deuseztatu socketa eta amaitu programa
       close(sock);
       exit(0);
}
```

e.4. irudia. Konexiorik erabiltzen ez duen zerbitzari baten kodea.

```
# include <sys/types.h>
# include <sys/socket.h>
# include <netinet/in.h>
# include <stdio.h>

# define DATA "Hau da bezeroak bidalitako mezua..."

main(int argc, char **argv) {
       int sock;
       struct sockaddr_in zerb_helb;
       struct hostent *hp, *gethostbyname();
       int luze;
       char buf[1024];

// Socketa sortu
       sock=socket(PF_INET, SOCK_DGRAM, 0);
       if (sock<0) {
              perror("datagrama-socketa sortzen");
              exit(1);
              }

// Zerbitzariaren helbidea eraiki
       hp = gethostbyname(argv[1]);
       if (hp == 0) {
              fprintf(stderr, "%s: host ezezaguna", argv[1]);
              exit(2);
              }
       memcpy( (char *)&zerb_helb.sin_addr, (char *)hp->h_addr,
              hp->h_length);
       zerb_helb.sin_family=AF_INET;
       zerb_helb.sin_port=htons(atoi(argv[2]));

// Eskaera bidali
       if (sendto(sock, DATA, strlen(DATA)+1, 0,(struct sockaddr
*)&zerb_helb, sizeof zerb_helb) < 0)
              perror("datagrama bidaltzean");

// Erantzuna jaso
       if(recvfrom(sock,buf,1024,0,(struct sockaddr *)&zerb_helb,
&luze)<0)
          perror("zerbitzariaren erantzuna jasotzean");

// Zerbitzariaren erantzuna atera pantailan
    printf("%s\n",buf);

// Deuseztatu socketa eta amaitu
       close(sock);
       exit(0);
}
```

e.5. irudia. Konexiorik erabiltzen ez duen bezero baten kodea.

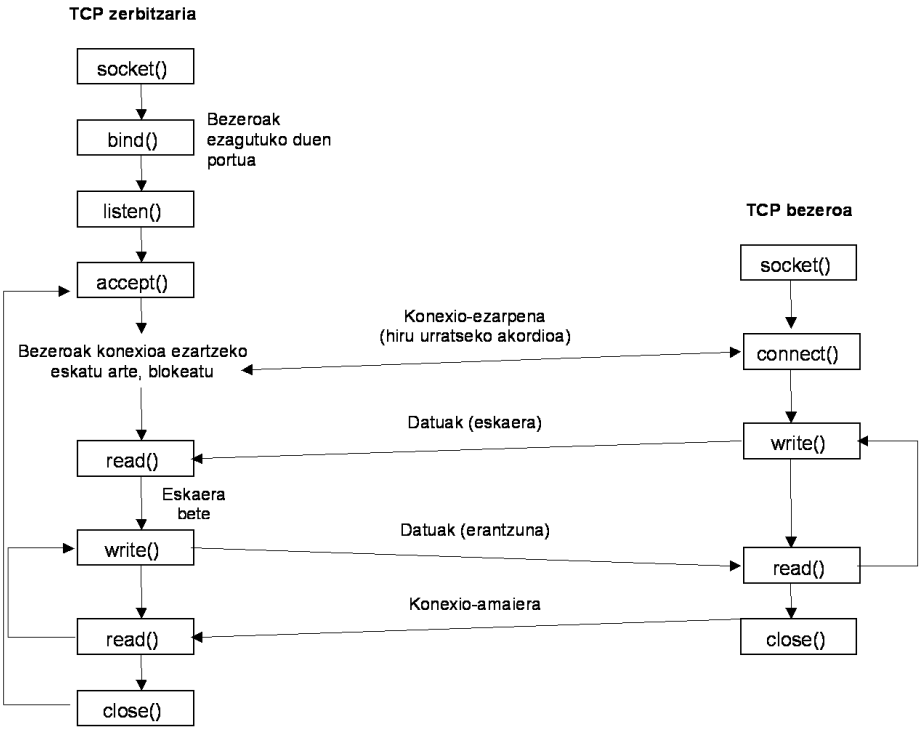
e.4. irudiko zerbitzaria xume-xumea da. Bezero bakar batentzako zerbitzua emango du, eta gero desagertu egingo da. Hori ez da zerbitzari normal batek egin behar duena: zerbitzariak etengabeko begizta batean sartzen dira, datozen eskaerei aurre egiteko. Irudiko zerbitzaria bezero baten eskaeraren zain dago, eta hori heltzen denean egiten duen gauza bakarra da bezeroak bidalitako mezua pantailan ateratzea. Hori egin baino lehen, *bind()* bidez bere socketari helbidea esleitu eta gero, pantailatik aterako du zein den sistemak esleitu dion portu-zenbakia. Kontuan izan gure zerbitzari esperimental honek ez duela inongo porturik erreserbatuta. Kasu horietan programatzaileak sistemari portu dinamiko bat esleitzeko eska diezaioke, edo programatzaileak zuzenean har dezake horietako portu bat, betiere libre egongo dela baldin badaki. Aplikazio honetan lehenengo aukera hartu dugu, eta horrek bigarren arazo bat sortzen du: nola jakingo du bezeroak zein den zerbitzariaren portua? Horretarako ateratzen du zerbitzariak pantailan datu hori.

Bezeroak, bere aldetik, argumentuetatik jasoko du zerbitzariaren portu-zenbakia (ikusi e.5. irudia). Datu hori ezinbestekoa da eskaera bidaltzeko. Bezeroak egikaritzen duen erabiltzaileak zerbitzariaren exekuzioa aztertu behar izango du lehenago, datu hori pantailatik hartzeko.

Zerbitzariak bere socketaren helbidea `helb` aldagaian eraikitzean, ematen dion IP helbidea INADDR_ANY konstantea da. Konstante horrek sistemari adierazten dio berak aukeratzeko behar den IP helbidea. Kontuan izan konputagailu batek IP helbide bat baino gehiago izan ditzakeela.

Socketen erabilera konexioaren bidezko aplikazioetan

e.6. irudian TCP erabiltzen duen aplikazio baten bezeroaren eta zerbitzariaren eskema dugu. Berriro ere, kontuan izan irudian agertzen ez diren beste liburutegiko funtzioak ere erabil daitezkeela. Programatzaile bakoitzak aukeratuko ditu zeintzuk erabili, bere ohiturei jarraituz eta aplikazioaren beharrak kontuan harturik.



e.6. irudia. Konexioaren bidezko garraio-zerbitzua erabiltzen duen aplikazio baten eskema.

Konexioaren bidezko garraio-zerbitzua erabiltzen duen aplikazio baten adibidea

Orain TCP erabiltzen duen aplikazio baten programazioa ikusiko dugu (zerbitzaria eta bezeroa), C lengoaian berriro ere. Ondoko bi irudietan horrelako aplikazio baten bezeroaren eta zerbitzariaren C kodea duzu. Konexiorik gabeko zerbitzua ematen zuen aplikazioaren antzekoa da, baina orain bi partaideek elkarren arteko komunikazioetan TCP erabiliko dute.

e.7. irudian, UDP zerbitzariarekin alderatuz, aldaketa batzuk ikus daitezke:

- Bezeroen edozein eskaera hartu baino lehen, konexioa ezartzeko, *listen* eta *accept* egin behar da. Hau da, aplikazio-mailako eskaera hartu baino lehen, garraio-mailako konexio-eskaera onartu behar da.
- Zerbitzari honek ematen du, bai, etengabeko zerbitzua, ez da desagertzen bezero bakar bati kasu egin eta gero.
- Zerbitzari honek portu dinamiko bat hartzen du, ez dio sistemari eskatzen. Portu hori Z_PORTUA konstantean definitzen da, sistemako IPPORT_RESERVED konstantean oinarrituz. IPPORT_RESERVED konstanteak adierazten du zein den lehenengo portu dinamikoa.

Programatzailearen ardura da bezeroetan eta zerbitzarian Z_PORTUA libre egongo dela egiaztatzea.

- Socket konektatu batetik irakurtzea UDP socket batetik egitea baino konplexuagoa da. Socket konektatuetatik fitxategietatik bezala irakurtzen da. Socket itxita dagoenean, *read()* funtzioak balio negatibo bat itzuliko du. Socketaren bufferra hutsik dagoenean, zerbitzaria blokeatu egiten da, datuen zain.
- Zerbitzari honek ez dio ezer itzultzen bezeroari. Hau arraroa da aplikazio banatuetan, baina gerta daiteke.

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include <stdio.h>

#define STDOUT 1
#define Z_PORTUA (IPPORT_RESERVED+1)

main() {
    int byte_kop,luz;
    int sock, sock_k;
    struct sockaddr_in zerb;
    char buf[1024];

    luz = sizeof(struct sockaddr);

// Adi-socketa sortu
    sock = socket(PF_INET, SOCK_STREAM, 0);
    if (sock < 0) {
        perror("Ezin izan da adi-socketa lortu");
        exit(1);
           }

// Eraiki eta esleitu adi-socketaren helbidea */
    zerb.sin_family = AF_INET;
    zerb.sin_addr.s_addr = htonl(INADDR_ANY);
    zerb.sin_port = htons(Z_PORTUA);
    if (bind(sock, (struct sockaddr *)&zerb, sizeof zerb ) < 0)
       {
        perror("Ezin izan dut helbidearen esleipena egin ");
        exit(1);
       }
```

```
// Konexio-eskaerak jasotzeko gaitu socketa */
    listen(sock, 5);

// Eskaerak jaso eta kasu egin, etengabe
    do {

    // Konexio-eskaeraren zain egon. Heltzendenean, onartu
    // eta sortu socket konektatua (sock_k)
        sock_k = accept(sock, (struct sockaddr *)&bez, (int *) &luz);
           if (sock_k == -1){
            perror("Konexioa ez da onartu !!!");
               exit(-1);
               }

    // Konexioaren bidez bidalitako byteak irakurri eta
    // pantailan atera
         do {

                // Bufferra garbitu
            memset(buf, 0, sizeof buf);

                // Jaso datuak eta pantailan atera
            byte_kop = read(sock_k, buf, 1024);
            if (byte_kop <0) perror("Datuak jasotzen");
            else write(STDOUT, buf, byte_kop);
         } while (byte_kop > 0);

     // Socket konektatua deuseztatu
            close(sock_k);
    } while (1);
  }
```

e.7. irudia. TCP erabiltzen duen zerbitzari baten kodea.

e.8. irudiko bezeroari dagokionez, hau aipatu behar da:

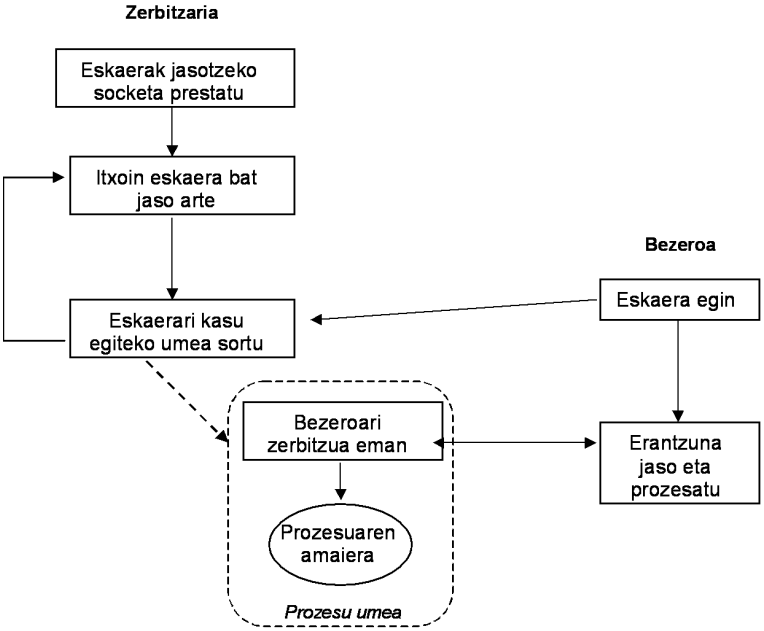
- Aplikazio-mailak ezer bidali baino lehenago, garraio-mailako konexioa irekitzen du, *connect()* deiaren bidez.
- Kasu honetan, erabiltzaileak ez dio aplikazioari argumentuetan zerbitzariaren portu-zenbakia eman behar, portu hori finkatuta dagoelako (IPPORT_RESERVED+1).

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>

#define Z_PORT (IPPORT_RESERVED+1)
#define ESKAERA "Bezeroak bidaltzen duena"

main(int argc, char *argv[]) {
    int sock;
    struct sockaddr_in zerb;
    struct hostent *hp, *gethostbyname();

    if (argc < 2) {
        printf("Erabilera: %s zerb_izena\n", argv[0]);
        exit (1);
           }

// Sortu socketa
    sock = socket(PF_INET, SOCK_STREAM, 0);
    if (sock < 0) {
        perror("Ezin izan da socketa sortu");
        exit(1);
           }

// Eraiki zerbitzariaren helbidea
    zerb.sin_family = AF_INET;
    hp = gethostbyname(argv[1]);
    if (hp == 0) {
        fprintf(stderr, "%s:konputagailu ezezaguna\n",argv[1]);
        exit(2);
           }
    memcpy((char*)&zerb.sin_addr, (char*)hp->h_addr,
        hp->h_length);
    zerb.sin_port = htons(Z_PORT);

// Konexio-eskaera bidali
    if (connect(sock, (struct sockaddr *)&zerb,sizeof zerb)<0){
        perror("Ez du konexioa onartu !!!");
        exit(1);
           }

// Aplikazio-mailako zerbitzu-eskaera bidali
    if (write(sock, ESKAERA, strlen(ESKAERA)+1) < 0)
         perror("Ezin izan dut mezua idatzi");

// Socketa deuseztatu
close(sock);
 }
```

e.8. irudia. Konexioak erabiltzen duen bezero baten kodea.

6.3. ZERBITZARI KONKURRENTEAK

Adibideetan ikusitako zerbitzariak ez dira konkurrenteak, une berean bezero bakar bati zerbitzua emateko gauza baitira, eta ez bezero bati baino gehiagori. Hori bideragarria izan daiteke sarean aplikazioaren bezero bakarra baldin badago, edo nolabait bermatuta baldin badago bi bezerok zerbitzariari ez dizkiotela inolaz ere beren eskaerak une berean egingo. Baina aplikazio banatu gehienetan, bezeroei emandako zerbitzuan konkurrentzia behar da, bezero bati kasu egiteko aurrekoa bukatu arte itxaronarazi gabe.



e.9. irudia. Zerbitzari konkurrenteen jarduera.

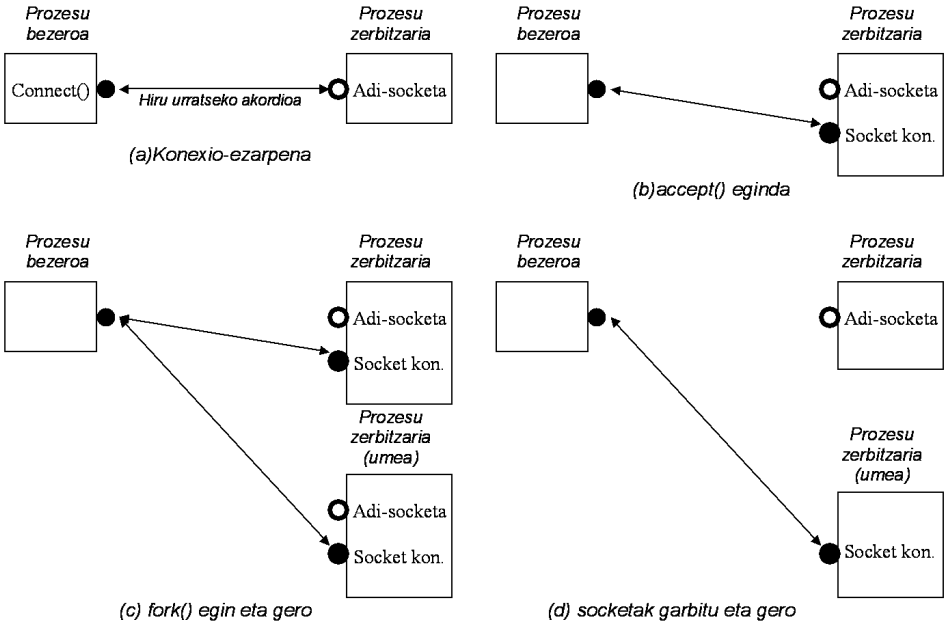
Zerbitzari konkurrenteen programazioan, gehienetan egiten dena bezero batetik jasotzen den eskaera bakoitzeko zerbitzariaren prozesu ume bat sortzea da. Umeak jasotako bezeroen eskaerez arduratzen diren bitartean, prozesu gurasoak heltzen diren eskaera berriei kasu egiten die, eta eskaera bakoitzeko ume berri bat sortzen du. Beren bezeroarentzako zerbitzua bete eta gero, umeak deuseztatzen dira. Hori guztia e.9. irudian adierazten da era grafikoan.

e.10. irudiko kodea konexioak erabiltzen dituen zerbitzari konkurrente baten eskema da. Irudi horretan konkurrentzia lortzeko sistema-dei esanguratsuenak bakarrik adierazten dira. Socket eta prozesuen arteko dinamika hobeto ulertzeko, ikusi e.11. irudia.

```
main() {
[...]
int adi, konektatua;
[...]
adi = socket(PF_INET, SOCK_STREAM, 0));
[...]
bind(adi, ...);
    listen(adi, ...);

    while(1) {
    konektatua = accept(adi, ...);

    switch(fork())
{
        case 0 :              // umea da
  close(adi);          // adi-socketa deuseztatu
  zerbitzua (konektatua);     // Eskaerari kasu egin
  close(konektatua);          // Itxi socket konektatua
  exit(0);                    // Umeak bere burua deuseztatu
default:
    close(konektatua);//Socket konektatua deuseztatu
       }
}
```

e.10. irudia. TCP zerbitzari konkurrente baten eskema.

e.10. kodea aztertu eta gero, ezinbestekoa da honako galdera hau egitea: prozesu gurasoak socket konektatua deuseztatzeko *close()* egiten duenean, ez al du horrek suposatzen FIN segmentu bat bidaltzea, eta bezeroak konexioa amaitutzat jotzea? Ondoren, nola komunikatuko dira bezeroaren eta zerbitzariaren prozesu umeak, azken honi bere gurasoak konexioa amaitu egin badio? Erantzuna sistema eragileak dauka. Unix sistemetan, erabiltzen ari den baliabide bakoitzeko (fitxategiak, socketak, diskoak…) kontagailu bat dago, non zenbatzen den unean baliabide horrekin lan egiten ari diren prozesuen kopurua. Socket konektatuaren kontagailuak biko balioa izango du *fork()* egin eta gero, gurasoak eta umeak erabiltzen dutelako socket hori. Gero, gurasoak socket horretan *close()* egiten duenean, sistemak ez du benetan amaiera-prozedura martxan jartzen; ez du FIN segmenturik bidaltzen, kontagailuaren balioari bat kentzen baizik, besterik ez. Kontagailuak 1 balio duenean eta socket horretan *close()* egiten dugunean bakarrik amaituko du sistemak konexioa benetan.



e.11. irudia. Socketen eta prozesuen arteko dinamika TCP erabiltzen denean.

6.4. inetd() SUPERZERBITZARIA

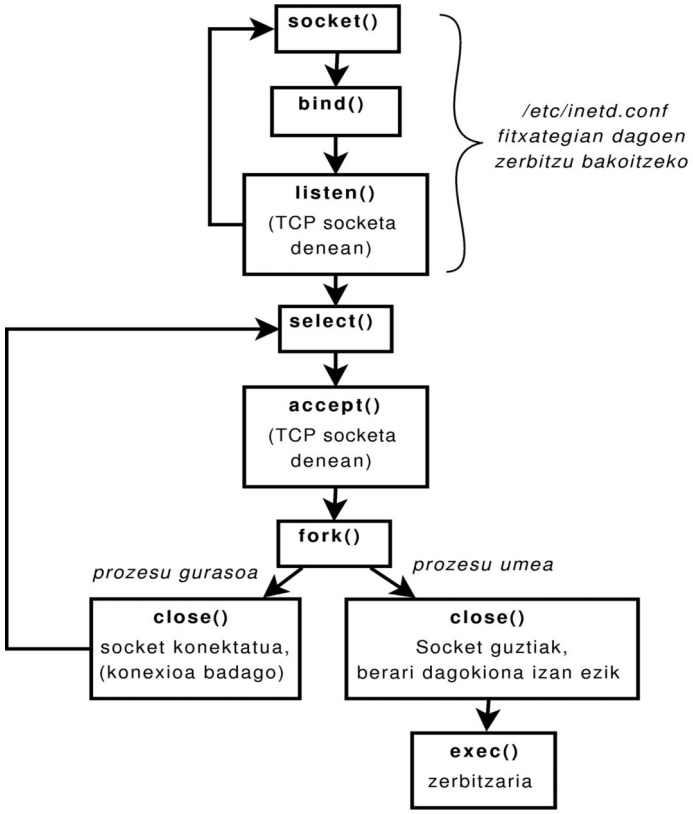
Eskuarki, konputagailu batean zerbitzari asko daude martxan. Zerbitzari bakoitza abiatzean prozesu bat sortzen da, eta prozesu horrek zerbitzariaren adi-socketa zaintzen du, eskaerak jasotzeko. Sarritan, zerbitzari guztiak *accept()* batean blokeatuta egongo dira, bezeroen eskaeren zain. Ez dute ezer egiten egoera horretan, baina sistemari lana ematen diote eta CPU denbora kontsumitzen dute.

Hori ekiditeko, Unix sistemetan badago zerbitzariak kudeatzeko beste era eraginkorragoa. Ideia hau da: zerbitzari bat ez da abiatuko berarentzako eskaera bat heltzen ez den bitartean. Bere adi-socketa beste prozesu batek zainduko du bitartean, superzerbitzariarena egiten duen prozesu batek. Hori da *inetd()* prozesua. Konfigurazio-fitxategi baten bidez adierazten zaio *inetd()* prozesuari zeintzuk diren zaindu behar dituen zerbitzuak, zein UDP eta TCP porturi kasu egin behar dien, alegia. Horietako portu batean eskaera bat hartzen denean, *inetd()* prozesuak ume bat sortzen du, eta ume horri dagokion zerbitzariaren kodea ematen dio egikaritzeko.

e.12. irudian *inetd*ek betetzen dituen urrats esanguratsuenak agertzen dira. Ondoko hauek ditugu:

- Hasteko, konfigurazio-fitxategian agertzen den zerbitzu bakoitzeko dagokion erako socket bat sortzen du, eta bere helbidea esleitzen dio.
- Sortutako TCP adi-socketetan, gainera, *listen()* egin behar du.

- Gero, superzerbitzaria sortutako socketetan edozein eskaeraren zain gelditzen da blokeatuta. Hori *select()* deiaren bidez egiten da. UDP socketetan datagrama bat heltzea nahikoa da desblokeatzeko. TCP adi-socketetan hiru urratseko akordioa bete behar da.
- Eskaera TCP socket batean hartu bada, *accept()* egiten da.
- Eskaerari kasu egiteko, superzerbitzariak ume bat sortzen du. Ume horrek bezeroarekin komunikatzeko behar ez dituen socket guztiak deuseztatuko ditu, eta bakar batekin gelditzuko da (socket konektatuta TCPren kasuan, eta jatorrizko UDP socketa bestean). Gero, *exec()* deiaren bidez, zerbitzariari dagokion kodea egikarituko du.



e.12. irudia. *inetd()* prozesuak egindako urratsak.

Superzerbitzariaren eskema hau Unix ez diren beste sistema askotan ere erabiltzen dute. Jatorrizko *inetd()* prozesuak adierazi dituen segurtasun-arazoengatik, gaur egun haren ordezkoak direnek, askotan *xinetd()* izenekoek, ordezkatu dute.

7. Bibliografia

Liburu honetan ez dago TCP/IP arkitekturari buruzko guztia, are gutxiago konputagailu-sareei buruzko guztia. Interesa duen irakurleak honako liburu hauetan aurki dezake informazio gehiago, sakonagoa eta zehatzagoa.

7.1. LIBURU OROKORRAK

Kurose, J. F. eta Ross, K. W. (2008): *Computer Networking*, Addison Wesley [4. argitaraldia].
Hau da gaur egun dagoen libururik onena TCP/IP arkitektura ikasteko, nire iritziz. Ez du jorratzen maila fisikoa, baina bai beste guztiak. Liburu bikaina, oso argia eta zehatza da.

Tanenbaum, S. (2003): *Computer Networks*, Prentice Hall [4. argitaraldia].
Ziur aski konputagailu-sareei buruzko inoiz idatzitako libururik osatuena. Arazoaren maila guztiak lantzen ditu, fisikoa barne. Liburua nahiko entziklopedikoa da. Azken argitaraldi hau hasi da zahar gelditzen arlo batzuetan. Hirugarren bertsioa eskuragarri dago euskaraz, UPV/EHUko argitalpen-zerbitzuaren bidez. Hirugarren eta laugarren bertsioen arteko aldea ez da handia.

Stallings, W. (2004): *Data and Computer Communications*, Prentice Hall [7. argitaraldia].
Urteetan, Tanenbaumen liburuaren lehiakide zuzena. Hau ere liburu entziklopedikoa da. Gai gehienetarako bestea bezain erabilgarria da, baina aplikazioetan, segurtasunean, eta multimedia arloan 6. edizioaren gabezia harrigarri bereak ditu.

7.2. GAI KONKRETUETARAKO LIBURUAK

Medhi, D. eta Karthikeyan, R. (2007): *Network routing: algorithms, protocols, and architectures*, Elsevier/Morgan Kaufmann.
Interneten bideratzea sakonki ezagutzeko liburua, oso eguneratua. Interneten egitura, oro har, ezagutzeko oso iturri ona da.

Stevens, W. R. (1994): *TCP/IP Illustrated*, 1. bol., Addison Wesley.
TCP/IP inguruko xehetasunak ezagutzeko iturririk onena. Aurreko liburuetan detaileak falta direla iruditzen bazaizu, hartu hau. Irakurle adituentzat bakarrik.

Lucena, M. (2008): *Criptografía y seguridad en computadores*, 4-0.7.51 bertsioa [2008ko ekaina].
Kriptografia eta bere aplikazioa komunikazioetarako segurtasunean ikasteko balio du liburu honek. Liburu hau ez dago eskuragarri paperean: bertsio elektronikoa soilik egin da. Ondoko URL honetan lor daiteke, dohainik: <http://wwwdi.ujaen.es/~mlucena/>.

Brown, C. (1994): *UNIX: distributed programming,* Prentice Hall.
Aplikazio banatuen programazioa. Antzekoak diren hainbat liburu aurki daitezke. Gaztelaniaz dauden horietako batzuk honako hauek dituzu:
Miguel, J. (1998): *TCP/IP en UNIX*, Ra-Ma.
López, A. (1999): *Novo. Protocolos de Internet,* Ra-Ma.
Márquez, F. M. (1996): *UNIX: Programación avanzada*, Ra-Ma.

Stevens, W. R. (1998): *UNIX Network Programming*, 1. bol., Prentice Hall [2. argitaraldia].
Zerbitzari konkurrenteei eta *inetd* superzerbitzariari buruzko informazio gehiago nahi izanez gero, ikusi liburu hau.

7.3. RFCak

Askotan agertzen dira RFCetarako erreferentziak liburuan zehar, agiri horiek baitira, askotan, informazio-iturririk onena TCP/IP eta Interneti dagokienez. Hainbat tokitan daude eskura RFCak Interneten. Hoberena zuzenean Internet Society-ko RFC Editor entitateak kudeatutako gunea erabiltzea da (www.rfc-editor.org). Webgune horretan RFCak bila daitezke izen, zenbaki edo egilearen arabera, eta RFC bakoitzaren egoera (uneko estandarra, proposatutakoa, historikoa, eta abar) azaltzen dute, baita zein beste RFCk eguneratu edo ordeztu duten bilatutakoa ere.

Aurkibide alfabetikoa

Sailean argitaratu diren beste liburu batzuk

Algoritmika
Rosa Arruabarrena
1997an argitaratua
ISBN: 84-86967-82-1

Ordenadore bidezko irudigintza
Joseba Makazaga, Asier Lasa
1998an argitaratua
ISBN: 84-86967-90-2

Oinarrizko programazioa. Ariketa-bilduma
Arantza Diaz de Illarraza, Kepa Sarasola
1999an argitaratua
ISBN: 84-8438-002-5

Zirkuitu elektriko eta elektronikoen oinarrizko analisia
Olatz Arbelaitz, Txelo Ruiz
2001ean argitaratua
ISBN: 84-8438-018-1

LINUX Sistemaren eta sarearen administrazioa
Iñaki Alegria
2003an argitaratua
ISBN: 84-8438-040-8

Sistema Digitalen Diseinu-hastapenak. Oinarrizko kontzeptuak eta adibideak
Olatz Arbelaitz eta beste
2005ean argitaratua
ISBN: 84-8438-069-6

Softwarearen ingeniaritza [I. atala: Softwarearen garapenaren zenbait arlo]
Jose Ramon Zubizarreta
2006an argitaratua
ISBN: 84-8438-085-8

Softwarearen ingeniaritza [II. ATALA: Garapen monolitikotik hiru mailako arkitekturara bezero/zerbitzariak bisitatuz]
Jose Ramon Zubizarreta
2009an argitaratua
ISBN: 978-84-8438-165-5

Linux: Sistemaren eta sarearen administrazioa 2. argitaraldia (Debian eta Ubuntu)
Iñaki Alegria eta Roberto Cortiñas
2008an argitaratua
ISBN: 978-84-8438-178-5

TAPE Testu-analisirako Perl erremintak
Aitzol Astigarraga eta beste
2009an argitaratua
ISBN: 978-84-8438-233-1