7. Internet eli verkkojen verkko

Johdanto

Kaikkien tuntema Internet[291] on kansainvälinen tietoverkko, joka koostuu suuresta joukosta sovitulla tavalla yhteen liitettyjä verkkoja. Tunteminen jää kuitenkin useimmilla hyvin pintapuoliseksi, sillä Internetin käyttö sinänsä ei vaadi juuri minkäänlaista käsitystä siitä, miten Internet on saanut alkunsa tai miten se teknisesti toimii. Sen sijaan Informaatioteknologian alalla toimivan asiantuntijan on tunnettava IP-teknologian (IP = Internet Protocol) keskeisimmät periaatteet ja ominaisuudet. Samalla tämä osuus myös syventää muutamien aikaisempien osuuksien tarkasteluja. Erityisesti on huomattava, että Internetin verkkolaitteet ovat tietokoneita, joiden ohjelmistot ovat erittäin monimutkaisia.

Tämän osan keskeisimmät aiheet ovat:

1. Miten internetistä on tullut sellainen kuin se on
2. IP-verkon rakenneosat ja protokollatasot
3. IP-paketin rakenne, sekä IPv4 että IPv6
4. TCP:n toimintaperiaate ja sen vaikutus Internetin palveluun
5. Turvallisuushaasteet Internetissä

Teoreettinen osuus käsittelee jonojärjestelmien mallinnusta.

Internetin kehittyminen

Historiaa

Internetin alku ajoittuu 1950- ja 60-lukujen vaihteeseen. Idän ja lännen välinen kilpavarustelu oli saanut uutta vauhtia Sputnikin laukaisusta 1957, minkä seurauksena Yhdysvaltojen puolustusministeriö perusti ARPA:n (Advanced Research Projects Agency[292]) edistämään tutkimusta ja kuromaan kiinni oletettua Neuvostoliiton etumatkaa huipputekniikan alueella. Samaan aikaan RAND Corporationissa[293] työskennellyt Paul Baran esitti ajatuksen täysin hajautetusta verkon toimintaperiaatteesta, jonka avulla voitaisiin varautua myös ydinsodan aiheuttamiin äärimmäisiin tilanteisiin. Hänen ajatuksenaan oli, että jokainen

[291] Melko vakiintuneen käytännön mukaan Internet isolla I:llä on erisnimi, joka viittaa yhteen kaikkien tuntemaan maailmanlaajuiseen tietoverkkoon. Pienellä kirjoitettuna internet tarkoittaa joukkoa reitittimillä yhdistettyjä verkkoja, joissa kaikissa käytetään IP-yhteyskäytäntöä eli protokollaa. Käytäntö on sama myös englannin kielessä.

[292] Nimiä ARPA ja DARPA (Defense ARPA) on käytetty vuoron perään saman organisaation nimenä, W. Isaacson, *The Innovators*, p. 228 tai http://en.wikipedia.org/wiki/DARPA

[293] RAND on vuonna 1948 perustettu tutkimuslaitos Yhdysvalloissa, joka alun perin teki tutkimusta Yhdysvaltain ilmavoimille, mutta on myöhemmin laajentanut toimintaansa muille aloille.

verkon solmu yhdistettäisiin muutamaan muuhun solmuun ja jokainen solmu pystyisi tekemään reitityspäätöksiä täysin itsenäisesti verkon kulloisenkin tilanteen mukaan. Baranin arvion mukaan kolme tai neljä vaihtoehtoista yhteyttä muihin solmuihin olisi riittävä takaamaan lähes yhtä hyvän verkon luotettavuuden kuin mihin teoriassa voitaisiin päästä.[294]

Ajatus itsenäisestä reitityksestä johti myös siihen ajatukseen, että välitettävä informaatio tuli pilkkoa määrämittaisiin paketteihin, jotka sisältävät reitityksessä tarvittavan tiedon. Tämä periaate poikkesi olennaisesti silloisissa verkoissa, erityisesti puhelinverkossa, käytetyistä periaatteista. Puhelinverkossa yhteys muodostettiin ensi erillisen merkinannon avulla, jolloin itse puhelun aikana ei tarvinnut enää välittää mitään puhelun reititykseen liittyvää tietoa. Pakettipohjainen verkko, jossa reititys tapahtuu periaatteessa itsenäisesti jokaiselle paketille erikseen, oli varsin radikaali.

Kehitys kohti Internetiä, sellaisena kuin sen nyt tunnemme, ei kuitenkaan ollut suoraviivaista. Baran yritti vakuuttaa AT&T:n insinöörit pakettikytkentäisyyden eduista, laihoin tuloksin. Vaikka Baran kirjoitti laajan artikkelin kehittämistään pakettiverkon toimintaperiaatteista[295], hänen ajatuksensa eivät ilmeisesti vaikuttaneet merkittävästi ARPA:ssa tehtyyn pakettiverkkojen kehitystyöhön ainakaan ennen vuotta 1967.

Toinen olennainen kehitykseen vaikuttanut tekijä oli tietokoneiden ja niiden välisen tiedonsiirron kehittyminen. Vuonna 1965 yhdistettiin Yhdysvaltojen länsi- ja itärannikoilla sijaitsevat tietokoneet datayhteydellä valintaisen puhelinverkon kautta. Tätä voidaan pitää ensimmäisenä laajan alueen tietokoneverkkona. Ensimmäiset käytännön kokemukset osoittivat, että kaukanakin olevat tietokoneet voidaan yhdistää toisiinsa, mutta että piirikytkentäinen puhelinverkko oli tähän tarkoitukseen väärä teknologia. Pakettikytkentäinen verkko oli olennaisesti parempi ratkaisu. Ensimmäinen suunnitelma verkolle, jota kutsuttiin ARPANET:iksi, julkaistiin vuonna 1967. Samaan aikaan oli käynnissä, toisistaan tietämättä, kaksi muutakin samaa aihealuetta tutkinutta projektia: jo edellä mainittu RAND sekä NPL Isossa-Britanniassa.[296]

Varsin yleisesti on esitetty väittämää, että Internet olisi suunniteltu kestämään ydinsota. ARPANETin keskeiset kehittäjät, Bob Taylor ja Larry Roberts, ovat kuitenkin vakuuttaneet, ettei ydinsotaan varautumisella ollut mitään vaikutusta ARPANETin toimintaperiaatteiden valinnassa. Internetin arkkitehtuuri ei siten suoranaisesti perustu tarpeeseen suunnitella verkko ydinsotaa varten. Asialla on myös kolmas näkökulma (Baran siis ensimmäinen), nimittäin ARPANETin kehitystyön rahoittaminen. ARPAn johtajien tasolla ydinsotaan

[294] Tämä vastaa hyvin niitä tuloksia jotain esitettiin luvussa 4 kohdassa saatavuusanalyysi.
[295] P. Baran: On Distributed Communications: IX: Security, Secrecy, and Tamper-free Considerations. Defense Technical Information Center, 1964.
[296] NPL, National Physical Laboratory, Iso-Britannia.

varautumisella oli huomattava vaikutus. Jos johtajat olisivat pitäneet ARPANETin tavoitteena vain muutamien yliopistojen ja tutkimuslaitosten tietokoneiden yhdistämistä, se tuskin olisi saanut niin paljon rahoitusta kuin mitä se 1960-luvulla sai. Voidaan jopa spekuloida, että jos joku muu taho olisi tarjonnut ydinsotaan varautumisen kannalta paremman ja uskottavamman ratkaisun, rahoitus olisi suunnattu sinne. Eli vastaus onko Internet alun perin suunniteltu ydinsodan varalle, on kyllä tai ei riippuen näkökulmasta.

Joka tapauksessa jo Internetin alkuvaiheessa korostettiin vaatimusta, että verkon tulee olla mahdollisimman toimintakykyinen silloinkin, kun merkittävä osa verkosta tulee toimintakyvyttömäksi. Syitä saattoi olla monia, eivätkä ne välttämättä liittyneet mitenkään sotaan, vaan pitkien yhteyksien epäluotettavuuteen, hajautetusta toiminnasta mahdollisesti aiheutuviin yhteensopivuusongelmiin ja akateemisen ympäristön epähierarkkisuuteen.

Käytännössä tasavertaisiin solmuihin perustuva verkko ilman keskitettyä hallintoa täytti parhaiten hajautetun dataliikenteen vaatimukset. Tällaisessa verkossa jokainen solmu pystyy itsenäisesti välittämään ja vastaanottamaan viestejä. Tieto liikkuu verkossa pieninä paketteina,[297] jotka etsivät tiensä vastaanottajalle ennalta määräämätöntä reittiä pitkin. Tämä vastasi siis hyvin Paul Baranin esittämiä ajatuksia. Toisaalta on varsin vaikea jälkikäteen sanoa, kuka ensimmäisenä esitti jonkun myöhemmin tärkeäksi havaitun idean. Pakettikytkennän osalta tätä kunniaa on sovitettu myös Leonard Kleinrockille, eikä vähiten hänen itsensä toimesta. Vaikka Kleinrock on ollut ansiokas sekä pakettikytkentäisten verkkojen teoreettisessa analyysissä että ARPANETin teknisessä kehittämisessä, ei häntä ilmeisesti voida pitää pakettikytkennän periaatteen varsinaisena keksijänä.[298]

Ajatus hajautetusta pakettipohjaisesta tietoverkosta levisi nopeasti. Vuoden 1969 loppuun mennessä verkkoon liitettiin ensimmäiset neljä solmukonetta. Koneet olivat sen aikaisia supertietokoneita ja niiden ylläpitäjät kehittivät yhdessä tarvittavat protokollat ja ohjelmistot. Tässä on yksi olennainen ero perinteisiin televerkkoihin, joita (ehkä aivan alkuvaihetta lukuun ottamatta) ei ole kehitetty niiden pääasiallisten käyttäjien toimesta vaan suurten valtiollisten tai kaupallisten toimijoiden tutkimus- ja tuotekehitysosastoilla.

Alkuvaiheessa ARPANETissa käytettiin NCP-protokollaa (Network Control Protocol), jolla oli kuitenkin monia rajoituksia, mm. verkossa käytettyjen osoitteiden suhteen. NCP ei kyennyt selviytymään pakettien hukkumisesta, sillä ARPANET oletettiin niin luotettavaksi, ettei pakettien hukkumisia tarvinnut ottaa huomioon. Koska tavoitteena oli liittää yhteen

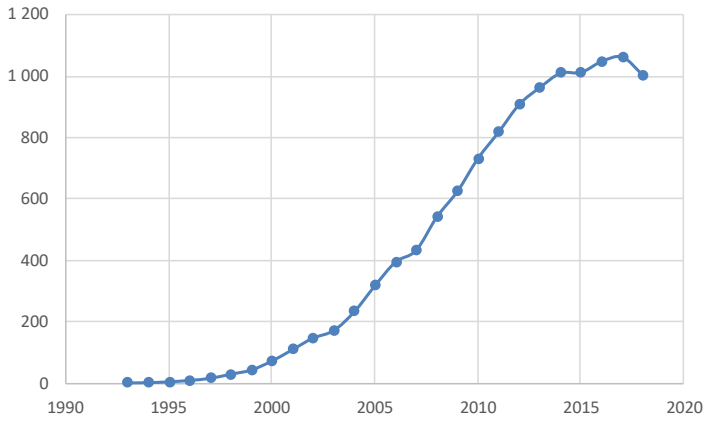
[297] Termi ”packet” eli paketti tuli NPL:n puolelta.

[298] Isaacsonin *The Innovators* –kirjan lukuun 7, The Internet. Pakettikytkennällä, kuten monella muullakaan teknologialla, ei ole yhtä keksijää, myös A. Schaffer, Tech’s Enduring Great-Man Myth, MIT Technology Review, August 4, 2015, http://www.technologyreview.com/review/539861/techs-enduring-great-man-myth/

monenlaisia, myös vähemmän luotettavia verkkoja, tarvittiin uusi protokolla. Suunnittelun pohjaksi otettiin seuraavat periaatteet:

- Verkkoja piti voida yhdistää toisiinsa ilman muutoksia itse verkkotekniikkaan.
- Liikennöinti tapahtuisi ”best effort” periaatteen mukaan, eikä verkko itse pyrkisi varmistamaan, että paketti saada kuljetettua verkon läpi.
- Verkon tulisi toimia myös tilanteissa, joissa paketteja hukkuu.
- Verkkojen yhdistämiseen käytettäisiin ”mustia laatikoita” (black boxes), jotka eivät pyrkineet tietämään mitään läpikulkevista informaatiosta. Näin ”laatikoista” saatettiin tehdä mahdollisimman yksinkertaisia hyödyntäen kaupallisesti saatavilla olevia tietokoneita. Mustia laatikoita alettiin kutsua myöhemmin reitittimiksi (router) ja yhdyskäytäviksi (gateway).
- Mitään maailmanlaajuista verkon valvontaa ei toteutettaisi.
- Osoitteiden tulisi kattaa koko maailmanlaajuinen verkko.
- Päätelaitteet hoitaisivat yhteyksien hallinnan (flow control).

Vuonna 1977 Internetiin oli liitettynä noin 100 tietokonetta, joista vain muutama oli Yhdysvaltain mantereen ulkopuolella satelliittilinkin takana. Armeijan verkko erkaantui ARPANETista vuonna 1983 omaksi MILNET-verkokseen. Tämä helpotti verkon laajentumista Yhdysvaltain ulkopuolelle. 1980- ja 90-lukujen vaihteessa ARPANET muuttui Internetiksi, minkä jälkeen verkon kasvu nopeutui entisestään (kuva 7.1). Internetin luonne on myös muuttunut olennaisesti siinä mielessä, että alkuvaiheen tarve hyödyntää keskitettyjä, kalliita tietokoneresursseja on vaihtunut tarpeeksi toteuttaa täysin yleiskäyttöinen ja hajautettu tiedonsiirtojärjestelmä.



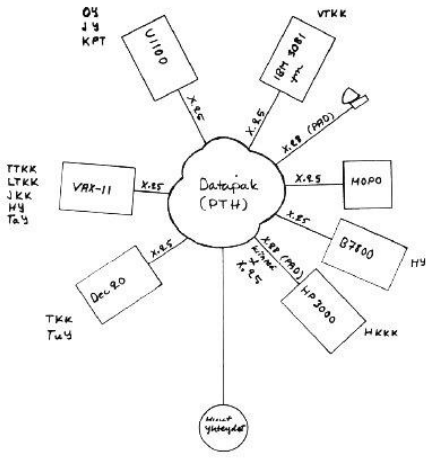
Kuva 7.1. Internetiin liitettyjen (isäntä)koneiden (host) lukumäärän kehitys (miljoonaa).[299]

[299] Data: https://www.statista.com/statistics/264473/number-of-internet-hosts-in-the-domain-name-system/

Suomi Internetiin

Päätös Suomen liittymisestä Internetiin tehtiin 2.12.1986 Funetin johtoryhmän kokouksessa.[300] Käytännössä liittyminen tapahtui kuitenkin vasta vuoden 1988 puolella. Funet tulee sanoista Finnish University and Research Network eli se on Suomen korkeakoulujen ja tutkimuksen tietoverkko. Kuvassa 7.2 on esitetty Funetin ensimmäisen vaiheen verkon suunnitelma, jossa käytettiin silloisen Posti- ja telehallituksen Datapak-palvelua. Datapak-palvelu perustui X.25-protokollaan, joka mahdollisti yhteyksien muodostamisen tietokoneiden välille sanomien lähettämistä varten. Liikennemaksut olivat niin kalliita (sisältäen liittymismaksun, kiinteän kuukausimaksun ja siirrettyyn tietomäärään perustuvan maksun), että tutkimuslaitosten ja yliopistojen oli pakko hakea edullisempia tiedonsiirtotapoja.

USA:n hallinnoimaan Internetiin liittyminen ei 1980-luvulla ollut suoraviivainen asia, sillä USA:ssa oli epäilyksiä Suomen suhteesta Neuvostoliittoon. Toisaalta Suomi muiden Pohjoismaiden mukana suhtautui myönteisemmin Internetiin kuin useat Länsi-Euroopan maat, jotka ehkä halusivat suojella kansallisia tietoliikennealan monopolejaan. Liittymislupa joka tapauksessa saatiin syksyllä 1988. Ensimmäisessä vaiheessa yhteys pohjoismaista Yhdysvaltoihin toimi 56 kbit/s satelliittilinkillä. Muun muassa Funet on päivittänyt ulkomaanyhteytensä 100 Gbit/s nopeuteen.

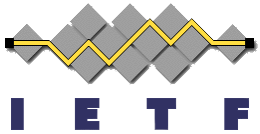


Kuva 7.2. Kuva Funetin projektisuunnitelmasta vuodelta 1984.[301]

[300] Tämä tieto sekä pääosa Suomen osuudesta perustuu kirjaan: P. Ahonen, Suomen tie internetiin, CSC – Tieteen tietotekniikan keskus Oy, Art Print Oy, Helsinki, 2008.

[301] Kuva ja historiatiedot: http://www.skrolli.fi/internetit-ennen-interneti%C3%A4-modeemipurkkien-nousu-ja-tuho. Sama kuva on myös P. Ahosen kirjassa Funet Suomen tie internetiin (s. 37). Arpanetin runkoverkon nopeus oli vuonna 1986 56 kbit/s, päivitys nopeuteen 1,544 Mbit/s tehtiin vuonna 1988, http://www.zakon.org/robert/internet/timeline/

Standardisointi



Internetiä kehittää suuri joukko erilaisia kansainvälisiä yhteisöjä. Niistä merkittävin on tietotekniikan ammattilaisista koostuva IETF (Internet Engineering Task Force) joka pyrkii tunnistamaan Internetin ongelmia ja tekemään ehdotuksia niiden korjaamiseksi.[302] Sen paremmin IETF kuin muutkaan vastaavat yhteisöt eivät ole virallisia standardointiorganisaatioita. Useimmat valmistajat pyrkivät silti noudattamaan niiden suosituksia. IETF on normeja tekeväksi organisaatioksi varsin poikkeuksellinen. Sen keskeisen toimintaperiaatteen on muotoillut David Clark seuraavasti ”We believe in rough consensus and running code” eli ”uskomme karkeaan yhteisymmärrykseen ja toimivaan koodiin” sekä ”We reject kings, presidents and voting” eli ”hylkäämme kuninkaat, presidentit ja äänestykset.” Päätöksiä on tietysti joskus tehtävä, mutta varsinaisia muodollisia äänestyksiä ei järjestetä. Sen sijaan yksimielisyyttä voidaan testata läsnä olevien ihmisten hyrinän (humming) voimakkuudella. Tällainen periaate on täysin poikkeuksellinen standardointiorganisaatiolle.

Internetin toimintaperiaatteet

Oliko kaikki se kehitys, joka on johtanut nykyiseen Internetiin, väistämätöntä? Tuskin. Datapalvelut olisivat saattaneet pysyä paljon tiukemmin suurten teleyritysten hallinnassa. Jos Paul Baran olisi pystynyt vakuuttamaan AT&T:n insinöörit pakettikytkentäisyyden eduista ja ARPA olisi ryhtynyt rahoittamaan AT&T:n kehitystyötä, niin verkon arkkitehtuurista ja palvelumallista olisi varmasti tullut keskitetympi ja tiukemmin kontrolloitu kuin nykyisessä Internetissä. Entä voidaanko Internetin nykyisestä dominoivasta asemasta päätellä, että Internet on erityisen erinomainen tekninen ratkaisu? Ei. Jos Internet voitaisiin nyt suunnitella alusta alkaen ilman, että otettaisiin huomioon jo toiminnassa olevia verkkoja ja laitteita, niin siihen tehtäisiin merkittäviä muutoksia esimerkiksi verkko-osoitteiden osalta.[303] Toisaalta on vaikea sanoa mihin tarkkaan ottaen päädyttäisiin, sillä toiveet ja näkemykset ovat ristiriitaisia. Mutta nyt meillä on Internet sellaisena kuin se on ja jokaisen informaatioteknologian alalla toimivan on tunnettava sen keskeiset toimintaperiaatteet.

Miten liikenne tai pikemminkin data ylipäätään voi kulkea onnistuneesti niin monimutkaisen verkon kuin Internet läpi? Päätelaitteita on miljardeja, palvelimia (server) kymmeniä miljoonia, verkon solmupisteitä miljoonia ja verkkoa operoivia tahoja tuhansia. Periaat-

[302] IETF:n toimintaperiaatteista, katso: P. Hoffman (ed.) The Tao of IETF: A Novice's Guide to the Internet Engineering Task Force, http://www.ietf.org/tao.html.
[303] Esimerkiksi: Day, J. (2007*). Patterns in network architecture: a return to fundamentals*. Pearson Education.

teessa ongelma on samankaltainen kuin lähetettäessä (fyysistä) postia silloin, kun vain henkilön nimi on tiedossa. Ensin henkilölle täytyy löytää osoite, jonka avulla paketti voidaan kuljettaa haluttuun kohteeseen. Osoitetieto (esimerkiksi Konemiehentie 2, 02150 Espoo, Suomi) on sarja kirjaimia ja numeroita, joiden perusteella paketti voidaan kuljettaa perille, mutta se ei varsinaisesti ole paikkatieto. Sen sijaan koordinaattitieto, esimerkiksi 60°11'13"N 24°49'16"E, määrittelee fyysisen paikan tarkasti mutta ei sisällä reititystietoa paketin kuljettamista varten.

Tietoverkoissa osoite viittaa yleensä laitteeseen, samaan tapaan kuin osoite Konemiehentie 2 viittaa rakennukseen. Toinen mahdollisuus on viitata liitäntään, eli rakennuksen tapauksessa tiettyyn ulko-oveen. Vaikka ero tuntuu pieneltä, sillä on merkitystä reitityksen toiminnan kannalta varsinkin, jos osoitteesta ei voi päätellä mitkä ovet vievät samaan fyysiseen paikkaan, eli mitkä liitännät johtavat samaan aliverkkoon (subnetwork tai subnet). Aliverkon voi ajatella vastaavan rakennuksen sisäisiä osoitteita, esimerkiksi ”Konemiehentie 2, B251” johtaa yhteen Aalto-yliopiston työhuoneeseen. IP-verkossa aliverkko tarkoittaa verkon osaa, joka on loogisesti erotettu OSI-mallin kolmannella eli verkkokerroksella (OSI-malli selostetaan tarkemmin hieman myöhemmin tässä luvussa). Koneet, jotka kuuluvat samaan aliverkkoon, käyttävät osoitteita siten, että niissä on täsmälleen samat merkitsevimmät bitit määrättyyn rajaan saakka. Kyseinen raja jakaa IP-osoitteen aliverkon osoitteeseen ja aliverkon sisällä tietokoneen yksilöivään osaan.

Toinen ilmiö, joka joskus voi aiheuttaa lisävaivaa on se, että osoitetieto ei yleensä sisällä luotettavaa tietoa kahden osoitteen välisestä etäisyydestä. Internetissä käytettävä paketin osoite, historiallisesta taustasta johtuen, ei kerro sellaisenaan mitään tietokoneen fyysisestä sijainnista eikä edes kovin paljon loogisesta sijainnista verkossa. Esimerkiksi IP-osoite 130.233.199.243 on käytössä Aalto-yliopistossa Espoossa, mutta osoite 131.233.199.243 on käytössä Philadelphian kaupungin lähistöllä Yhdysvalloissa.[304]

Tässä vaiheessa voimme olettaa, että oikean talon tai oikean tietokoneen löytäminen on riittävää, jotta paketti löytää perille. Käytännössä tarvitaan siis hakemisto, joka yhdistää nimen ja osoitteen. Puhelinten tapauksessa tällainen oli vuosittain päivitetty puhelinluettelo, josta löytyi tieto henkilön puhelinnumerosta ja yleensä myös osoitteesta. Internetin tapauksessa ei tietenkään ole mielekästä käyttää puhelinluettelon tapaisia opuksia, vaan osoitteen haunkin täytyy tapahtua Internetin välityksellä mahdollisimman automaattisesti.[305] Tällaista kyselyä varten tarvitaan joku osoite, josta sitten muita osoitteita voidaan hakea. Tätä varten IETF on kehittänyt protokollan nimeltä DHCP (Dynamic Host

[304] IP-osoitteita koskevat tiedot: http://www.ip-adress.com/ip_tracer/

[305] Miljardin IP-osoitteen tiedot vaatisivat noin 4 miljoonaa sivua tyypillisessä puhelinluettelon formaatissa. Luetteloista voisi muodostaa noin sata metriä korkean pinon. Lisäksi IP-osoitteet muuttuvat jatkuvasti.

Configuration Protocol), joka mahdollistaa tietokoneen siirtymiseen uuteen verkkoon ilman, että kenenkään tarvitsee tehdä manuaalisia muutoksia mihinkään tietokantaan.[306]

Verkon ylläpitäjällä on käytössä tietty IP-osoiteavaruus, josta verkko voi jakaa IP-osoitteita verkkoon liittyville laitteille. Päätelaite pyytää käynnistyksen yhteydessä DHCP-palvelimelta oman IP-osoitteen, joka on yleensä voimassa ennalta määrätyn ajan. Päätelaite pyytää tarvittaessa osoitteen voimassaoloajan jatkamista. Yleensä jatkaminen tapahtuu automaattisesti ilman mitään häiriöitä verkkoyhteydessä. Jos palvelin ei jostain syystä jatka voimassaoloaikaa, niin päätelaite ei voi enää käyttää aikaisemmin annettua IP-osoitetta.

Mutta miten laite osaa uudessa verkossa kysyä mitään DHCP-palvelimelta, jos se ei tiedä sen IP-osoitetta? Päätelaitteen lähettämä paketti voi löytää perille, vain jos sillä on tiedossa vastaanottajan IP-osoite eikä IP-osoitteeksi kelpaa ”DHCP.” Tähän tarpeeseen tarvitaan ennalta tarkasti määritelty menettely. Koska päätelaite ei tässä vaiheessa tiedä vastaanottajan IP-osoitetta, sen on käytettävä yleislähetystä (broadcast). Kun DHCP-palvelin vastaanottaa yleislähetyksen, jossa on DHCP:n osoitetta koskeva tiedustelu, se esittää tarjouksen, joka sisältää IP-osoitteen ja sen voimassaoloajan ja joitakin muita tietoja. IP-paketti sisältää automaattisesti lähettäjän, eli tässä tapauksessa DHCP-palvelimen, IP-osoitteen. Laite saattaa saada eri palvelimilta useita eri tarjouksia, joista se sitten valitsee yhden ja ilmoittaa tästä kyseiselle palvelimelle.

Periaatteessa DHCP voi jakaa mitä tahansa asetustietoja, mutta keskeisimmät näistä ovat oletusyhdyskäytävän (default gateway) ja nimipalvelimen (Domain Name System, DNS) IP-osoitteet. Tämän jälkeen päätelaitteella on tieto siitä, miten aliverkon ulkopuolelle päästään ja mistä voidaan kysyä muita IP-osoitteita.

Nimipalvelu on keskeinen osa Internetin toimintaa. Ilman toimivia nimipalvelimia verkko ajautuisi kaaokseen. IP-verkko kyllä toimisi, koska se perustuu IP-osoitteisiin, mutta sen sijaan se Internetin palvelu, jota pääosin käytämme eli WWW (World Wide Web) olisi erittäin hankalakäyttöinen.[307] Tämä esimerkki osoittaa sen, miten tärkeää on erottaa käsitteellisesti toisistaan:[308]

- Internet, joka on maailmanlaajuinen avoin tietoverkko, joka ytimeltään perustuu TCP/IP-yhteyskäytäntöjen käyttöön, ja
- WWW, joka on palvelujärjestelmä, jonka avulla julkaistaan verkkosivuja ja hyödynnetään niitä.

[306] RFC 1541, https://tools.ietf.org/html/rfc1541
[307] Sanastokeskuksen mukaan voidaan käyttää joko muotoa www tai WWW, mutta ei mielellään termiä web. http://www.tsk.fi/tepa/fi/haku/www
[308] Nämä ovat sanastokeskuksen mukaiset määritelmät.

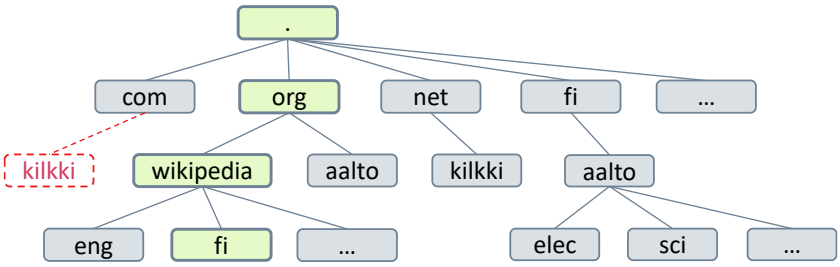
Internet on siis verkko, joka toimii tiettyjen protokollien mukaisesti, kun taas WWW on Internetin päälle rakennettu palvelu, joka toimii OSI-mallin ylimmällä eli sovelluskerroksella. Ilman DNS-palvelimia verkkoon liitetyt laitteet on edelleen mahdollista löytää, mutta selaimesi löytää vain ne verkkosivut, joita vastaavat IP-osoitteet sillä on tiedossa.

Nimipalvelun toiminta perustuu verkkotunnuksiin (domain name) ja URL (Uniform Resource Locator) -osoitteisiin. Esimerkkinä URL:sta on https://www.aalto.fi/fi/opiskelu-aallossa, jonka osat ovat:

- https (hypertext transfer protocol secure) määrittelee yhteyskäytännön,
- www.aalto.fi määrittelee palvelimen, josta resurssi on löydettävissä ja
- loppuosuus /fi/opiskelu-aallossa/ määrittelee verkkosivun, jonka perusteella palvelin osaa palauttaa halutun sisällön käyttäjälle.

Erisnimien (suurten yritysten nimet näyttävät olevan erityisessä suojeluksessa) käytöllä verkko-osoitteena on jonkin verran rajoituksia, silti samankaltaisilla mutta hieman eroavilla verkkotunnuksilla voi päätyä täysin eri sivustoille.[309] Sitten on tietysti lukematon määrä muodollisesti oikeita sivuston nimiä, jotka eivät johda mihinkään.

Vaikka *.fi* on ylimmällä hierarkiatasolla Suomen maatunnus, se ei tarkoita sitä, että osoitteen takana oleva laite tai palvelin olisi Suomessa, vaan ainoastaan että kyseinen tunnus on myönnetty Suomessa.[310] (*fi*-juuren alaisia nimiä jakaa keskitetysti Viestintävirasto). Muilla tasoilla *fi* ei välttämättä liity mitenkään Suomeen (tosin Wikipedian tapauksessa *fi* viittaa suomenkieliseen Wikipediaan). Vaikka joku sivusto olisikin tarkoitettu WWW-käyttöön, sivuston URL:ssa ei tarvitse olla missään kohtaa ”www”. Toki www-alku helpottaa asiakkaita havaitsemaan mikä on yrityksen verkko-osoite.[311] Verkkotunnukset muodostavat siten hierarkkisen rakenteen, josta kuvassa 7.3 on esitetty yksi esimerkki.

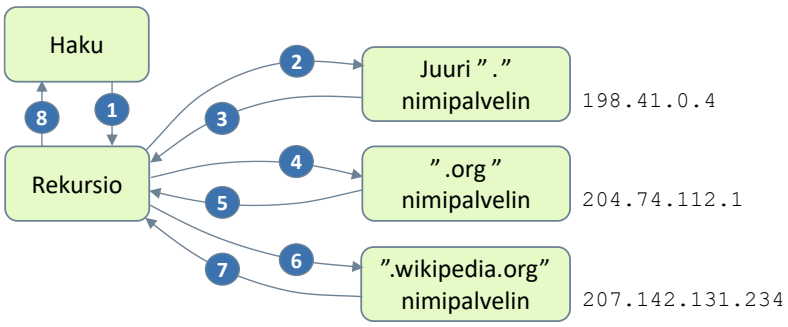


Kuva 7.3. Verkkotunnusten hierarkkinen rakenne (kilkki.com –verkkotunnus on periaatteessa mahdollinen, mutta sitä ei verkosta löydy).

[309] Kokeile esimerkiksi osoitteita aalto.fi, aalto.org ja aalto.net.
[310] fi-juuren alaisia nimiä jakaa keskitetysti Viestintävirasto https://www.viestintavirasto.fi/index.html.
[311] Kotisivuni osoite on kilkki.net ilman www-alkua. Selaimet esittävät sen silti usein muodossa www.kilkki.net.

Kuvassa 7.4 on esitetty osoitehaku verkkotunnukselle fi.wikipedia.org. Haku sisältää tässä tapauksessa kahdeksan vaihetta:

1. Palvelin tekee hakupyynnön rekursiiviselle osalle kohteelle *fi.wikipedia.org*.
2. Rekursiivinen[312] osa kysyy juuripalvelimelta *fi.wikipedia.org*:n osoitetta. Juuripalvelimen osoitteet vaihtuvat yleensä hyvin harvoin, joten voidaan olettaa, että nimipalvelinohjelmisto tietää ainakin yhden juuripalvelimen osoitteen, tässä tapauksessa osoitteen `198.41.0.4`.
3. Tehtyyn hakuun juuripalvelin vastaa lähettämällä takaisin tiedon, että *.org* verkkotunnusten nimipalvelin löytyy IP-osoitteesta `204.74.112.1`.
4. Seuraavaksi kysytään *.org* nimipalvelimelta mistä IP-osoitteesta löytyvät osoitteet *fi.wikipedia.org* –verkkotunnuksille.
5. Vastaus: palvelin osoitteessa `207.142.131.234` tietää Wikipedia.org osoitteet.
6. Nyt rekursiivinen osa voi kysyä viimeiseksi annetusta IP-osoitteesta, mikä on IP-osoite verkkotunnukselle *fi.wikipedia.org*.
7. Wikipedian verkkopalvelin lähettää kysytyn IP-osoitteen rekursiiviselle osalle.
8. Rekursiivinen algoritmi havaitsee, että sillä on nyt vastaus alun perin tehtyyn kyselyyn ja lähettää vastauksen eteenpäin.



Kuva 7.4. Nimipalvelinjärjestelmän osoitehaun rekursiivinen periaate.

Kysely jatkuu, kunnes haettu IP-osoite löytyy tai nimipalvelin toteaa, ettei haettua kohdetta vastaa mikään IP-osoite (kuten käy esimerkiksi kilkki.com-haulle). Tämän jälkeen, kun haluttu verkko-osoite on löydetty onnistuneesti, käyttäjä ja hänen laitteensa ja selaimensa voivat aloittaa liikennöinnin halutulle verkkosivustolle, eli hakea jotain tietoa vaikkapa Wikipedian suomenkielisiltä sivuilta.

[312] Rekursiivinen tarkoittaa ominaisuutta, jossa sama rakenne voidaan toistaa periaatteessa rajattoman monta kertaa.

IP-paketin rakenne

Ennen kuin paketteja voidaan siirtää, on sovittava vielä lukuisista muista asioista ja käytetyistä menetelmistä. Ensinnäkin on määriteltävä tarkasti, mikä on IP-paketin muoto, koska mitään tulkinnanvaraisuutta ei saa jäädä. Tässä on merkittävä ero fyysiseen postipalveluun, jossa kokeneet postinjakelijat osasivat (ainakin aikaisemmin) toimittaa perille kirjeitä ja paketteja varsin puutteellisin tiedon; pelkkä nimi ja paikkakunta saattoivat hyvin riittää. Jokaisella koneella Internet-verkossa on siis oma osoite. Samalla koneella voi olla myös useampia osoitteita. Kahdella eri koneella ei voi kuitenkaan olla samaa osoitetta.[313]

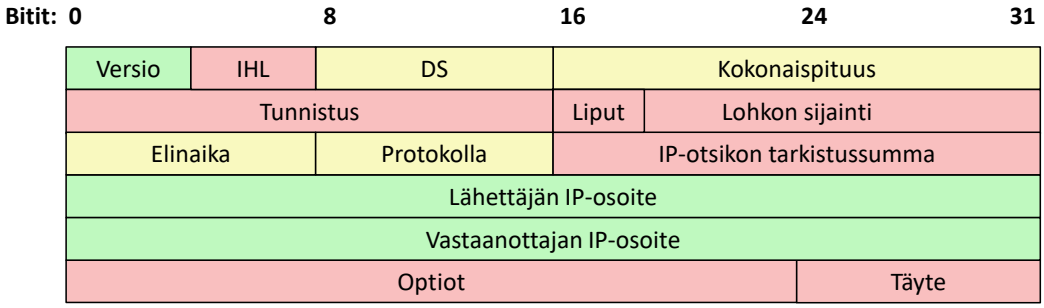
Vastaanottajan osoitteen ja lähettävän data lisäksi IP-paketissa on myös muuta informaatiota. Periaatteessa IP-paketin koko voi olla enimmillään 2^{16} tavua eli $65536 \cdot 8$ bittiä. Useimmat verkot eivät kuitenkaan pysty käsittelemään näin suuria yksittäisiä paketteja vaan tyypillisesti suurin mahdollinen koko on 1500 tavua. Tämän vuoksi data yleensä pilkotaan jo lähtiessä sopivan kokoisiin osiin.

Paketin dataosuutta edeltävää tiedonsiirtoa ohjaavaa osaa kutsutaan otsikoksi (header). Otsikon rakenne riippuu siitä, onko kyseessä IPv4- vai IPv6-paketti. Edelleen laajimmin käytössä oleva IPv4 on määritelty jo vuonna 1979 ja sen ollut käytössä kohta 40 vuotta. Voisi olettaa, että seuraava versio olisi IPv5, mutta näin ei käytännössä ole, sillä IPv5 oli kokeellinen protokolla. 1990-luvulla huomattiin, että IPv4:n osoiteavaruus (2^{32} = 4 294 967 296) ei riitä loputtomiin, kun WWW:n myötä käyttäjien ja verkkoon liitettyjen tietokoneiden määrät lähtivät räjähdysmäiseen kasvuun. Ongelmaa pahensi se, että osoiteavaruus oli jaettu sangen epätasaisesti.[314] Niinpä IETF suunnitteli uuden version, IPv6:n, jolla osoiteongelma ja samalla myös joitakin muita ongelmia ratkaistiin. Vaikka IPv6 on ollut käytössä jo pitkään, siirtymävaihe on kestänyt paljon pidempään kuin alun perin arvioitiin. Varovaisen arvion mukaan IPv6-liikenteen osuus ylittää IPv4-liikenteen ensi vuosikymmenen alkupuolella. Niinpä IP-paketin esittely on edelleen syytä aloittaa v4-muodosta.

IPv4:n otsikon vähimmäispituus on $5 \cdot 32$ bittiä (kuva 7.5). Useimmissa tietokoneissa otsikoiden käsittely nopeutuu, mikäli otsakkeen pituus on 32 bitin kerrannainen. Sen vuoksi otsikon loppuun lisätään tarvittaessa täytettä. IHL (Internet Header Length) kertoo IP-otsikon pituuden, jonka avulla täytebittien ja varsinaisen datan raja tunnistetaan. IPv4-osoite käsitellään 32-bittisenä lukuna. Osoitteen alkuosa on verkko-osoite. Loppuosan osoitteista vastaava organisaatio voi jakaa paikalliseen verkko-osoitteeseen ja laiteosoitteeseen. Jotta osoitteita olisi helpompi käsitellä, ne ilmaistaan tavuja vastaavina kymmenjärjestelmän lukuina (0-255) pisteillä erotettuina, esimerkiksi `207.142.131.234`.

[313] Joskus käytetään tarkoituksellisesti samaa osoitetta eri paikoissa verkkoa, esimerkiksi nimipalvelimien tapauksessa. Käyttäjän kannalta on tällöin sama mikä nimipalvelin vastaa.

[314] Yksittäinen amerikkalainen yliopisto saattaa hallinnoida suurempaa osoitemäärää kuin Kiina.



Kuva 7.5. IPv4-paketin otsikon rakenne. Punaisella merkityt alueet poistuivat IPv6:ssa, keltaisella merkityt ovat IPv4:ssä hieman eri muodossa tai eri paikassa.

Alun perin IP-osoitteet jaettiin kolmeen eri luokkaan (A, B, C) ajatuksena, että on muutamia todella suuria verkkoja, jonkin verran keskikokoisia verkkoja ja paljon pieniä verkkoja. Jako on sikäli tehoton, että on paljon organisaatioita, joissa on yli 250 konetta, mutta selvästi vähemmän kuin 65000 konetta. Osoitteita jäi siis paljon käyttämättä. Ratkaisuksi kehitettiin luokaton reititys, jossa verkkokohtaisesti voidaan valita verkko-osan pituus. Vaikka uusia IPv4-osoitteita ei ole enää jaettavissa, jo jaettuja osoitteita voi ostaa vapailta markkinoilta.

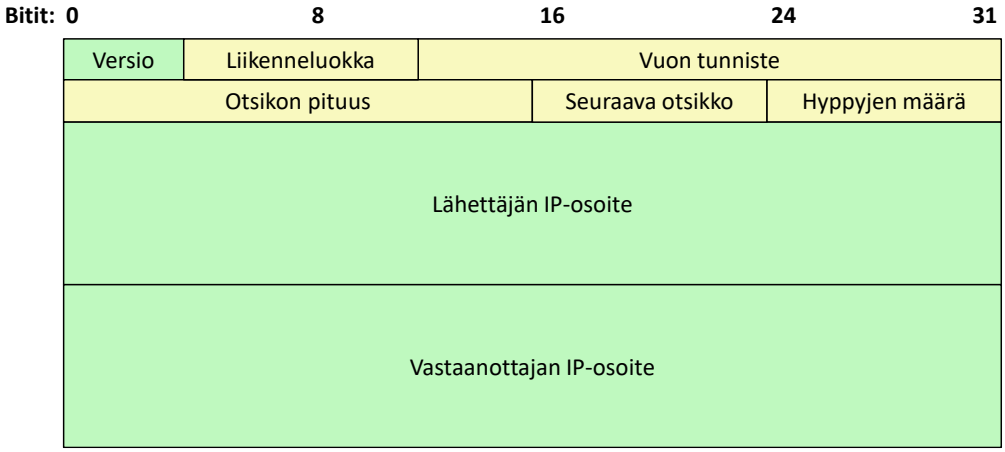
DS-tavua (Differentiated Services, eriytetty palvelu) käytetään paketin luokitteluun ja priorisointiin. Tämän tavun merkitys ja tulkinta riippuvat käytettävästä verkosta. DS-kentän avulla lähettäjä voi esimerkiksi toivoa, että paketti siirretään erityisen pienellä viiveellä. DS-tavua on käytetty lähinnä puheyhteyksien viiveen minimoiseen ja erottelemaan joitain muita kriittisiä sovelluksia tavallisesta dataliikenteestä.[315]

Koska erilaisissa verkoissa pakettien sallittu koko vaihtelee, voidaan paketteja joutua pilkkomaan pienemmiksi palasiksi. Pilkkominen on monella tavoin ongelmallista: jos yksikin osa paketista hukkuu, käytännössä koko IP-paketti täytyy lähettää uudelleen. Pilkkomista (fragmentation) pyritään käytännössä välttämään (IPv6:ssa ei edes ole tätä mahdollisuutta). Tunnistus-, ohjaus- ja lohkon sijaintitietoja käytetään apuna, kun palasista kootaan alkuperäistä viestiä. Elinaika ilmaisee ajan, jonka paketti voi olla verkossa; kyseessä ei kuitenkaan ole aika, vaan jokainen reititin vähentää lukua yhdellä. Kun luku tulee nollaksi, paketti hävitetään, joten paketti ei voi jäädä verkkoon ikuisesti. Osoitteiden lisäksi otsikossa on erilaisia optioita ja täytettä. Käytännössä optioiden käyttö on ollut hyvin harvinaista.

IPv4:n osoitteiden riittävyydestä ja muista IPv4:n rajoituksista on väitelty 20 vuotta. Tärkein IPv6:n etu on osoiteavaruuden huomattava kasvu: IPv4:n 32 bitin sijasta IPv6:ssa on käytettävissä 128 bittiä, mikä periaatteessa tarkoittaa $2^{128} = 3{,}4 \cdot 10^{38}$ osoitetta. Erityisesti IoT-laitteet (eli kaikenlaiset, usein pienet, Internetiin liitettävät laitteet) voivat jatkossa tarvita niin paljon osoitteita, ettei IPv4 siihen veny. Otsikon rakenteessa on myös

[315] K. Kilkki, *Differentiated Services for the Internet*, MacMillan, 1999, saatavilla osoitteessa http://kilkki.net/book

muita muutoksia, joilla on pyritty tehostamaan reitittimien toimintaa ja ottamaan huomioon muuttuneita vaatimuksia. Eräs IPv6:n merkittävistä eduista on, että suuri osoiteavaruus helpottaa automaattisen konfiguraation (auto-configuration) toteuttamista.



Kuva 7.6. IPv6-paketin otsikko. Keltaisella merkityt kentät ovat lähes samoja kuin IPv4:ssä.

OSI-malli

Protokolla (myös yhteyskäytäntö, protocol) on säännöstö, jota kahden tai useamman laitteen on noudatettava, jotta niiden välinen yhteys olisi mahdollinen. Monet menetelmät, vaikkapa reititysalgoritmit, toteutetaan erilaisten protokollien avulla. Tietoliikennetekniikassa käytetään tavallisesti yhtä aikaa useita eri protokollia, jotka huolehtivat kukin omasta tarkoin rajatusta tehtävästään tiedon siirrossa. Yhdessä nämä protokollat muodostavat protokollapinon. Kansainvälinen standardointijärjestö ISO[316] on standardoinut tietoliikennetekniikassa käytettävän protokollapinon eri kerrosten tehtävät. ISOn protokollapinoa kutsutaan OSI-malliksi.[317] OSI-malli pyrkii kuvaamaan tietoliikennejärjestelmän rakennetta ja protokollia sähköiseltä tasolta käyttäjän tasolle asti.

OSI-malli jakaantuu seitsemään kerrokseen. Kerrosten tehtävät ja niiden väliset rajapinnat on tarkkaan määritelty. Kerros voi antaa ylä- ja alapuolellaan olevalle kerrokselle vain tietynlaisia pyyntöjä ja ilmoituksia, joihin se saa vasteita ja vahvistuksia.

Kuva 7.7. OSI-mallin kerrokset

[316] ISO = The International Organization for Standardization, http://www.iso.org
[317] OSI = ISO Reference Model for Open Systems Interconnection

Eri kerrosten tärkeimmät tehtävät ovat:

1. Rakenteellisella eli fyysisellä kerroksella (physical layer) määritellään konkreettisia, mitattavia asioita. Muut kerrokset sisältävät ohjelmistomäärittelyitä. Fyysisen kerroksen alueeseen kuuluvat esimerkiksi liittimet, johdot ja sähköiset tasot.
2. Siirtoyhteyskerros tai siirtokerros (data link layer) määrittelee, kuinka verkossa rakennetaan yhteyksiä solmusta toiseen. Siirtokerroksen protokollat huolehtivat virhesuojauksesta ja palautumisesta normaalitoimintaan virhetilanteiden jälkeen.
3. Verkkokerros (network layer) reitittää kehykset tai paketit määränpäähänsä usein monimutkaisen verkon yli.
4. Kuljetuskerros (transport layer) tarjoaa ylemmille kerroksille suoran liikenneyhteyden ja häivyttää erityyppiset siirtojärjestelmät näkyvistä. Kuljetuskerroksen protokollat tarjoavat usein myös virheenkorjauksen.
5. Istuntokerros (session layer) idea on muodostaa ja purkaa yhteydet liikennöivien sovellusten väliltä ja jaksottaa liikenteen loogisiin osiin.
6. Esitystapakerros (presentation layer) sisältää (ainakin periaatteessa) muunnokset, joita tarvitaan esimerkiksi tietojen suojauksessa ja erilaisten aakkosten käytössä.
7. Sovelluskerros (application layer) palvelee suoraan loppukäyttäjää.

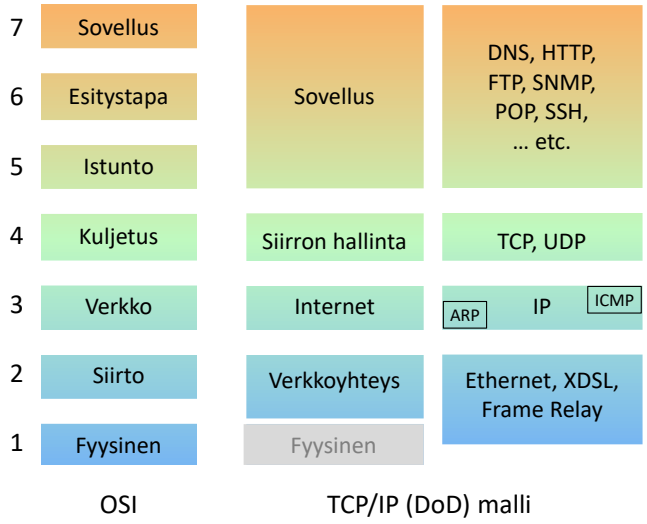
Huolimatta tunnettuudestaan OSI-mallia ei ole sellaisenaan käytetty juuri missään järjestelmässä. Informaatioteknologian asiantuntijan kuuluu kuitenkin tuntea OSI-malli ja sen tasot. OSI-malliin viitataan myös Internetin puolella, vaikka IP-verkot eivät noudatakaan täsmälleen OSI-mallia ja sen kerroksia.

TCP/IP-protokollaperhe

TCP (Transmission Control Protocol) -protokolla sijoittuu OSI-mallissa IP-protokollan yläpuolelle. Se on yhteydellinen protokolla, joka tarjoaa ylemmille kerroksille luotettavan kuljetuspalvelun IP-kerroksen toimintaa valvomalla. Ensimmäinen versio TCP/IP-protokollasta esiteltiin vuonna 1973. Aluksi tavoitteena oli, että TCP:tä voitaisiin käyttää kaikkeen tiedonsiirtoon. TCP suunniteltiin protokollaksi, joka selviää hukatuista ja väärään järjestykseen joutuneista paketeista havaitsemalla ja uudelleen lähettämällä hukatut paketit. Toisaalta on tilanteita, missä pakettien uudelleen lähettäminen ei ole mielekästä, esimerkkinä puhelut. Tarvittiin siis toinen protokolla, joka hoitaa kyseiset sovellukset. Tätä varten kehitettiin UDP (User Datagram Protocol).

Internet perustuu siis ARPANETin pohjalta syntyneeseen TCP/IP-arkkitehtuurin, josta käytetään myös nimitystä DoD-arkkitehtuuri.[318] TCP/IP-malli on yksinkertaisempi kuin OSI-malli, eikä se pyri samanlaiseen yleispätevyyteen kuin OSI. TCP/IP-arkkitehtuurin mukaiset protokollat eivät noudata täsmällisesti OSI-mallin kerrosjakoa, mutta yksinkertaisuuden vuoksi niitä jatkossa käsitellään kuin ne olisivat OSI-yhteensopivia.

OSI-mallissa IP-protokolla sijoittuu kolmostason yläosaan. Koska IP-protokolla ei odota paljon alla olevalta verkolta, se tarjoaa yhteydettömän palvelun riippumatta siitä, minkälaisia verkkoja sen alapuolella on. Jokaiseen pakettiin liitetään vastaanottajan täydellinen osoite ja paketit lähetetään matkaan toisistaan riippumatta. Pakettien kulkureitti ja järjestys saavat vaihdella matkalla (tosin yhden yhteyden pakettien kulkeminen eri reittejä on harvinaista ja yleensä liittyy johonkin vikatilanteeseen).



Kuva 7.8. Vasemmalla OSI-malli, keskellä TCP/IP-malli ja oikealla Internetissä käytettyjä yleisimpiä protokollia.

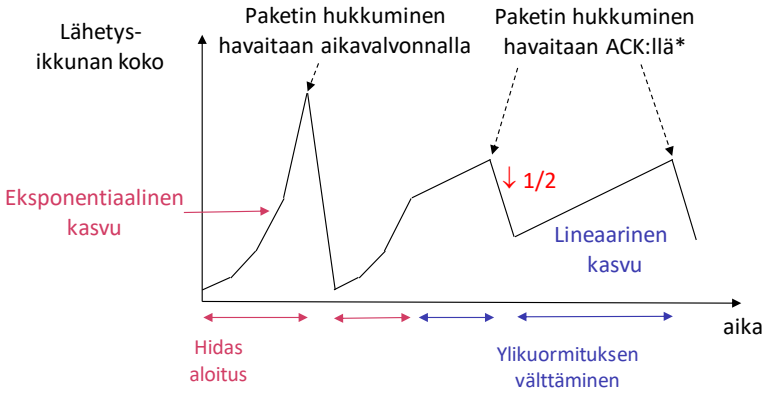
TCP/IP-mallissa (kuvassa 7.8 oikealla) verkkoyhteyskerros kattaa kaikki Internet-kerroksen alapuolella olevat verkot. Se, miten kukin verkko kuljettaa IP-paketteja eteenpäin, on määritelty IETF:n julkaisemissa RFC (Request for Comment) -julkaisuissa. Verkkoyhteyskerroksen yläpuolella ovat Internet-kerros (Internet layer) ja siirronhallintakerros (transmission control layer), jotka ovat saaneet nimensä suoraan IP- ja TCP-protokollilta. Ylimpänä TCP/IP-mallissa on prosessi- ja sovelluskerros (process / application layer), johon sijoittuu erilaisia protokollia ja apuohjelmia, kuten tiedostojen siirtoon tarkoitettu FTP (File Transfer Protocol), postin kuljettamiseen tarkoitettu SMTP (Simple Mail Transfer Protocol) ja SNMP (Simple Network Management Protocol).[319]

[318] DoD tulee nimestä Department of Defense, eli Yhdysvaltain puolustusministeriö.
[319] IETF tuntuu suosivan ”yksinkertaisia” (”simple”) protokollia, mutta käytännön yksinkertaisuus on toinen asia.

Kun IP-paketti ja kriittisimmät protokollat on nyt määritelty, paketti voidaan lähettää verkkoon olettaen, että se löytää tiensä perille. Merkittävänä ongelmana kuitenkin on se, että ilman mitään etukäteisvalvontaa käyttäjät saisivat lähettää mielin määrin liikennettä verkkoon ottamatta huomioon verkon kuormitustilannetta. Lopputuloksena on merkittävä riski verkon ylikuormittumisesta, ellei ennakoiviin tai reagoiviin toimenpiteisiin ryhdytä.

Kun Internetin liikenne alkoi 1980-luvulla kasvaa, törmättiin karuun todellisuuteen. Verkko ylikuormittui vuonna 1986 useita kertoja niin pahasti, että verkon välityskyky romahti täysin. Ongelman ydin oli siinä, että TCP:n säätömekanismi oli suunniteltu ottamaan huomioon vastapään rajallisen kapasiteetin (jotta vastaanottopään puskuri ei vuotaisi yli), mutta ei verkon kuormitustilannetta. Kun verkon kuormitus nousi tietyn rajan yli, uudelleenlähetyspyynnöt lisäsivät verkon kuormitusta ilman, että mikään mekanismi olisi palauttanut verkon toimivaan tilaan.

Ensimmäisen toimivan ratkaisun tähän ongelmaan esitti Van Jacobson.[320] Ratkaisu perustuu siihen, että jos päätelaite havaitsee, että paketteja hukkuu (= päätelaite ei saa määräajassa kuittausta lähetetyistä paketeista), se ”pudottaa lähetysnopeuden puoleen.” Lainausmerkit johtuvat siitä, että TCP ei suoranaisesti säätele lähetysnopeutta, vaan sitä määrää dataa (ns. ikkunan kokoa), jonka laite voi lähettää ennen vastaanottajan kuittausta. Käytännössä ikkunan koko yhdessä päästä-päähän viiveen kanssa määrää keskimääräisen lähetysnopeuden. Ikkunan kokoa säädellään ylöspäin lineaarisesti. Yhteyden alkuvaiheessa sovelletaan yleensä ns. hitaan aloituksen (slow-start) periaatetta, jotta uusi yhteys löytäisi sopivan lähetysnopeuden ilman että verkko tukkeutuisi kuten kuvassa 7.9 on esitetty.



Kuva 7.9. TCP:n nopeuden säädön periaate (ACK-paketti sisältää viimeisen yhtenäisen sekvenssinumeron, jos numero on sama kuin edellisessä ACK-paketissa, niin lähettäjä olettaa, että vähintään yksi paketti on hukkunut).

[320] Jacobson, V. (1988). Congestion avoidance and control, ACM SIGCOMM Computer Communication Review (Vol. 18, No. 4, pp. 314-329). Katso myös https://flylib.com/books/en/4.245.1.75/1/

Käytännössä näiden periaatteiden avulla voidaan melko luotettavasti säätää Internetin käyttöastetta ilman, että verkon sisällä tarvitaan kuormituksen valvontamekanismeja. Olennaista on, että tässä tapauksessa kuormituksen hallinta hoidetaan hajautetusti päätelaitteissa. Lopputuloksena Internetissä voi joskus hukkua varsin paljon paketteja (parin prosentin hukkuneiden pakettien osuutta voidaan pitää normaalina), mutta uudelleenlähetysten avulla kaikille sellaisille sovelluksille, jotka eivät ole erityisen viiveherkkiä, voidaan taata varsin luotettava palvelu. Lisäksi korkean kuormituksen aikana kapasiteetti jakautuu suunnilleen tasan TCP:tä käyttävien yhteyksien välillä. Pullonkaulan kohdalla keskimääräinen kuormitus voi nousta noin 90 prosenttiin. Pääosa Internetistä toimii edelleen TCP:n avulla ns. best effort –periaatteella.

Reititys

Lopulta kun kaikki on periaatteessa kunnossa paketin siirtämistä varten, miten paketti löytää perille? Reititys on prosessi, jossa jonkin reititysalgoritmin avulla selvitetään sopivin reitti lähettäjän ja vastaanottajan välille ja talletetaan tämä tieto verkon solmupisteisiin. Reitti voidaan optimoida esimerkiksi reitin pituuden, viiveen tai kaistanleveyden suhteen. Sopivimman reitin valintakriteerit riippuvat verkon ylläpitäjän toiveista. Yleistäen reitityksen tavoitteena on verkon suorituskyvyn maksimointi ja kustannusten minimointi.

IP-protokolla sallii tietopakettien sekä suoran että epäsuoran reitittämisen. Suoraa reititystä käytetään samassa fyysisessä verkossa olevien koneitten välillä. Tällöin paketit eivät kulje reitittimen kautta. Esimerkiksi Ethernet-lähiverkossa olevissa tietokoneissa säilytetään verkossa olevia IP-osoitteita vastaava Ethernet-verkkokortin ns. MAC-osoite[321] ja muodostetaan lähtevät Ethernet-kehykset suoraan kohdekoneen MAC-osoitteen mukaan. MAC- ja IP-osoitteiden vastaavuuksien etsintä ja ylläpito on esimerkki OSI-kerroksien välisestä yhteistyöstä. MAC-osoitteita tarvitaan OSI-mallin 2. kerroksen toiminnassa ja IP-osoitteet vastaavasti toimivat OSI-mallin 3. kerroksella.

Reititysalgoritmit voidaan luokitella sen mukaan, miten algoritmi sopeutuu verkkotopologian muutoksiin. Tällöin algoritmit voidaan jakaa staattisiin ja adaptiivisiin. Kun lähettäjä ja vastaanottaja pysyvät kutakuinkin paikoillaan eikä verkon rakenne muutu jatkuvasti, sopivat reitit voidaan kirjata yksinkertaisimmillaan vaikkapa käsin reititystietokantaan. Tämän kaltaista reititystä kutsutaan staattiseksi reitit ykseksi, ja sitä voidaan käyttää pienissä lähiverkoissa tai kun vaihtoehtoisia reittejä ei ole.

Adaptiivinen reititys huomioi verkon tilan muutokset ja sopeutuu yhteyksien katkeamiseen. Verkon reitittimet vaihtavat jatkuvasti keskenään tietoja verkon tilasta, esimerkiksi havaitsemistaan uusista siirtoyhteyksistä tai entisten siirtoyhteyksien katkeamisesta.

[321] MAC = Media Access Control. MAC-osoite on verkkosovittimen ethernet-verkossa yksilöivä osoite.

Tämän kaltainen ratkaisu skaalautuu myös suurikokoisiin verkkoihin. Tietojen käsitteleminen ja reititystaulun ylläpitäminen vaatii reitittimeltä kuitenkin paljon prosessointitehoa. Reitityksen vaatima ohjausliikenne tulee myös pitää mahdollisimman pienenä niin, ettei se häiritse verkon varsinaista hyötyliikennettä.

Suurin osa adaptiivisista reititysalgoritmeista perustuu niin sanotun lyhimmän polun (shortest path algorithm) laskemiseen. Tässä menetelmässä jokaiselle siirtoyhteydelle määrätään laskennallinen pituus, joka puolestaan riippuu siitä, minkä kriteerin mukaan reitit halutaan optimoida. Tämän jälkeen etsitään lyhin tie lähettäjän ja vastaanottajan välille.

Tärkeimmät menetelmät lyhimmän polun laskemiselle ovat etäisyysvektorialgoritmit ja yhteystila-algoritmit. Etäisyysvektorialgoritmia käyttävässä verkossa reitittimet lähettävät naapureilleen reitti-ilmoituksia, jotka kertovat etäisyyksiä niihin verkkoihin, jotka kyseisen reitittimen kautta on tavoitettavissa. Reitti-ilmoituksissa etäisyys ilmoitetaan paljaana lukuna, eikä siitä käy ilmi, minkä muiden reitittimien kautta kyseinen reitti kulkee. Reititin valitsee saamiensa ilmoitusten perusteella lyhimmät reitit, eli niiden reitittimien kautta kulkevat polut, joilla on ollut pienin painokerroin.

Koska viesteistä ei käy ilmi kuin reitin pituus ja se, minkä naapurin kautta reitti kulkee, mahdollisten reitityssilmukoiden havaitseminen on vaikeaa. Topologian muuttuessa saattaa kestää kauan, ennen kuin verkko stabiloituu ja oikeasti lyhimmät reitit onnistutaan ottamaan käyttöön. Tästä syystä etäisyysvektorialgoritmi ei sovellu suuriin verkkoihin yhtä hyvin kuin yhteystila-algoritmi. Etäisyysvektorialgoritmiin perustuvia reititysprotokollia ovat muun muassa RIP (Routing Information Protocol) ja IGRP (Interior Gateway Routing Protocol).

Yhteystila-algoritmia käyttävissä verkoissa kaikki reitittimet tuntevat koko alueen topologian. Jokainen reititin ylläpitää tietokantaa, jonka perusteella se laskee itse oman reititystaulunsa. Aina topologian muuttuessa reititystaulu lasketaan uudelleen. Reitittimet välittävät toisilleen tietoa verkon topologiasta säännöllisin yhteystilailmoituksin. Muutostilanteissa, esimerkiksi jonkin siirtoyhteyden katketessa, muutoksen havainnut reititin kertoo siitä välittömästi naapureilleen, jotka välittävät viestin edelleen omille naapureilleen. Lyhyet muutosviestit kulkevat koko verkon halki, joten verkko sopeutuu nopeasti uuteen topologiaan. Vastapainoksi menetelmä vaatii reitittimiltä runsaasti laskentakapasiteettia ja muistia.

Yhteystila-algoritmiin perustuvia reititysprotokollia ovat muun muassa OSPF (Open Shortest Path First) ja IS-IS (Intermediate System to Intermediate System). Toisinaan verkon ylläpitäjä haluaa esimerkiksi sopimusteknisistä syistä suosia sellaista verkkoa, joka tarjoamat polut eivät ole kaikkein lyhyimpiä. Tästä syystä suurempien verkkokokonaisuuk-

sien, ns. autonomisten alueitten, väliseen reititykseen on kehitetty polkuvektoreihin perustuvia reititysprotokollia, joista merkittävin on BGP (Border Gateway Protocol). BGP-viesteistä käy ilmi se, minkä autonomisten alueiden kautta tieto kustakin tunnetusta verkosta on saatu. Näitä välitysketjuja tulkitsemalla voidaan selvittää verkon topologia autonomisten alueiden tasolla ja havaita mahdolliset reitityssilmukat.

IP-verkon rakenneosat ja toiminta

Edellä oli kuvattu Internetin toimintaa periaatteellisella tasolla. Käytännössä tarvitaan tietysti laitteita toteuttamaan haluttu verkon toiminnallisuus ja lopuksi vielä organisaatiot ylläpitämään verkkojen toimintaa. Laitteita voidaan nimetä sen mukaan miten ja mihin niitä käytetään ja erityisesti millä OSI-tasolla ne pääosin toimivat kuten kuvassa 7.10 on esitetty:[322]

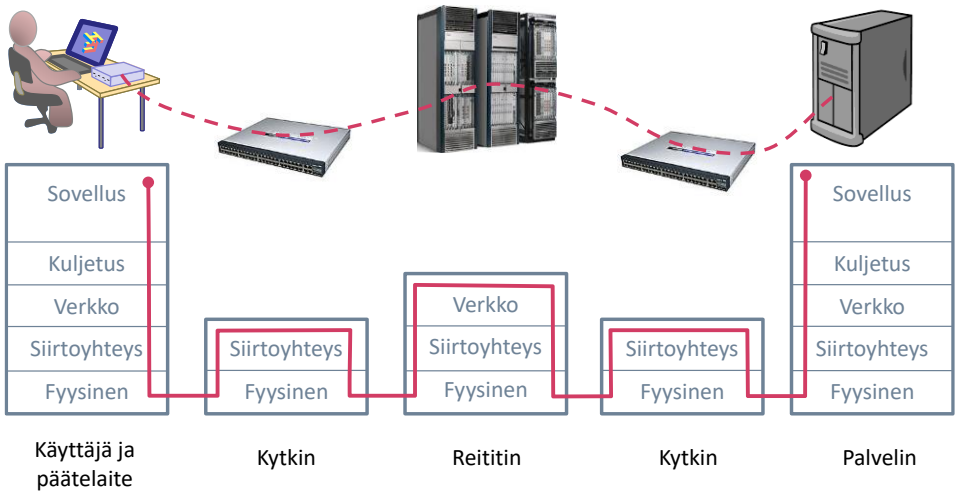
- Toistin (repeater): laite, joka vahvistaa vastaanottamansa signaalin ja lähettää sen sitten eteenpäin.
- Keskitin (hub): tietoliikenneverkon laite, jolla voidaan kytkeä useita pääte- ja verkkolaitteita samaan fyysiseen verkkoon.
- Silta (bridge): tietoliikenneverkon laite, joka yhdistää verkon osia OSI-mallin toisella eli siirtoyhteyskerroksella.
- Kytkin (switch): laite tai laitteen osa, johon muut tietoliikenneverkon laitteet on kytketty ja joka välittää yhdestä laitteesta tulevan tietoliikenteen niihin laitteisiin, joihin se on tarkoitettu.
- Reititin (router): laite tai ohjelmisto, joka ohjaa tietoliikennettä sopivalle reitille kohti määränpäätä.
- Yhdyskäytävä (gateway): tietoliikenneverkossa oleva verkon solmu, joka mahdollistaa erilaisten tietoliikenneverkkojen yhteensovittamisen.
- Palvelin (server): tietokone tai ohjelmisto, joka hoitaa tehtäviä muiden samaan verkkoon kytkettyjen tietokoneiden pyyntöjen ohjaamana tai niiden puolesta.
- Palomuuri (firewall): tekninen järjestely, jolla hallitaan liikennettä tietoliikenneverkosta toiseen tai tietoliikenneverkon ja yksittäisen järjestelmän välillä.

Tähtimäinen Ethernet-verkko rakennettiin aluksi yksinkertaisesti yhdistämällä työasemat keskittimellä. Tavallinen, ei-kytkentäinen keskitin välittää saamansa sanomat kaikille verkon asemille. Kytkentäinen keskitin eli kytkin vähentää verkon kuormitusta tavalliseen

[322] Termit pääosin Sanastokeskuksen terminpankin http://www.tsk.fi.määritelmien mukaisia.

keskittimeen verrattuna, ohjaamalla sanomat ainoastaan yhteen suuntaan kerrallaan. Lähiverkko voidaan periaatteessa rakentaa toistimien ja siltojen avulla. Niitä ei kuitenkaan nykyverkoissa käytetä vaan ne on korvattu kytkimillä, joissa on sillan toiminnallisuus jokaisen verkkoliitännän välillä.

Reititin on Internetin IP-protokollia käyttävien tietokoneverkkojen vastine puhelinverkon puhelinkeskukselle. Se reitittää sanomia yleisiin verkkoihin tai toisiin lähiverkkoihin. Se suodattaa, valvoo ja rajoittaa hyvinkin tarkkaan läpikulkevaa liikennettä. Reititin voi toimia myös palomuurina ja voi siten estää murtautumisyritykset. Reititin toimii OSI-mallin verkkokerroksella, eli se ohjaa paketteja eteenpäin pelkän IP-osoitteen perusteella.

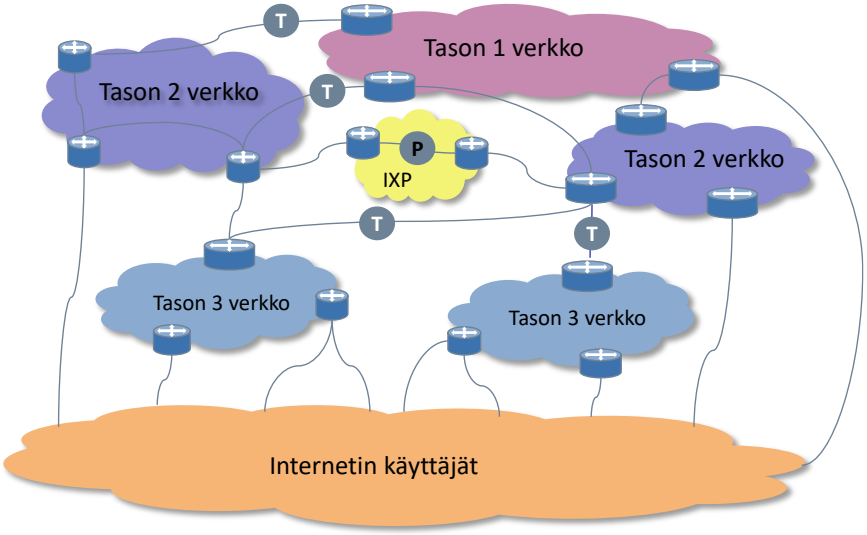


Kuva 7.10. Yhteyden muodostuminen IP-verkon läpi eri OSI-mallin kerroksilla.

Yhdyskäytävä on protokollamuunnin eli käytännössä laite tai ohjelmisto, jonka avulla voidaan yhdistää täysin erilaisia protokollia noudattavat verkot. Yhdyskäytävän kautta lähiverkkoon liitetyt koneet voivat olla yhteydessä ulkoisiin palveluihin. Yhdyskäytävää tarvitaan myös silloin, kun sähköposti lähetetään edelleen tekstiviestinä vastaanottajan kännykkään. Monissa järjestelmissä termiä ”gateway” käytetään viittaamaan reitittimeen, jolle lähetetään kaikki saman aliverkon ulkopuolelle suuntautuva IP-liikenne. Tällöin ei välttämättä tehdä mitään protokollamuunnosta. Yhdyskäytävää voidaan käyttää myös tietoturvatasojen erottamiseen.

Laitteiden lisäksi tarvitaan joku organisaatio pitämään huolta verkkojen toiminnasta. Vaikka Internet on toimintaperiaatteeltaan varsin epähierarkkinen, globaalilla tasolla verkot ja niitä hallinnoivat operaattorit muodostava hierarkkisen rakenteen. Ylimmällä tasolla (tier 1) toimivat suurimmat kansainväliset verkkopalveluja tarjoavat yritykset, jotka ovat kaikki suoraan toisiinsa yhteydessä eivätkä siten tarvitse muiden operaattoreiden apua kattavan kansainvälisen palvelun toteuttamiseen.

Alemman tason operaattorit ostavat välityspalveluita (transit) ylemmän tason operaattoreilta, jotta niiden asiakkaat voisivat olla yhteydessä Internetiin kokonaisuudessaan. Tasoja voi olla useita, mutta kaikilta tasoilta voidaan tarjota palveluita loppukäyttäjille. Samalla tasolla olevat operaattorit voivat vaihtaa keskenään liikennettä tasavertaisesti ilman operaattoreiden välistä rahallista korvausta ns. peering–periaatteella. Merkittävä osa peering-yhteyksistä tapahtuu keskitetyissä välityspisteissä (IXP, Internet Exchange Points).



Kuva 7.11. Internetin verkkojen ja operaattoreiden hierarkia. T = transit-yhteys, P = peering-yhteys.[323]

Internetin turvallisuudesta

Internetin negatiivisista lieveilmiöistä kohutaan tasaisin välein jopa iltapäivälehtien palstoilla. Milloin otsikoissa ovat pomminrakennusohjeet, milloin anonyymit seksipalvelimet, tietokonevirukset, kansainvälinen rikollisuus, tai eri maiden tiedusteluelimien suorittama laajamittainen vakoilu. Ihmisillä on jyrkkiä mielipiteitä Internetin hyvyydestä ja pahuudesta, vaikka käytännön todellisuudesta olisi enintään hämäriä käsityksiä. Kiistatta Internetissä on myös riskejä ja sudenkuoppia. Jokainen (kansainväliseen) tietoverkkoon liitetty kone on jossain määrin altis ulkopuoliselle hyökkäykselle. Hyökkäyksiä tekevät niin (teollisuus)vakoojat, jännitystä etsivät harrastelijat kuin rahanahneet rikollisetkin.

Nimitys hakkeri (hacker) syntyi 1960-luvun alussa MIT:n (Massachusets Institute of Technology) tekoälylaboratoriossa. Hakkerin merkityksen vivahteet ovat jonkin verran vaihdelleet vuosien saatossa. Aluksi hakkerointi (hacking) oli suhteellisen viatonta modernien laitteiden kanssa pelaamista, mutta 1960-luvulla hakkerit käyttivät jo selkeästi väärin

[323] Kuva perustuu Wikipedian Internetiä käsittelevään artikkeliin, http://en.wikipedia.org/wiki/Internet katso myös http://arstechnica.com/security/2016/04/flashback-declassified-1970-dod-cybersecurity-document-still-relevant/

mm. puhelinverkon yhteyksiä.[324] Historiallisesta taustasta johtuen lienee parempi, että termiä hakkeri käytetäänkin vain sellaisissa yhteyksissä, joissa on vähintään väärinkäytön mahdollisuus olemassa. Lisäksi on käytössä termi krakkeri (cracker), joka yksiselitteisesti liittyy järjestelmien väärinkäyttämiseen.

Eräs keino turvata Internetin käyttöä ovat palomuurit, jotka valvovat kahden verkon, yleensä luotettavan verkon ja epäluotettavan verkon, välistä liikennettä. Palomuurin tehtävänä on estää ulkopuolisia tunkeutumasta suojattuun verkkoon ja samalla taata omille käyttäjille turvalliset yhteydet oman verkon ulkopuolelle. Palomuuri voi rajoittaa liikennettä muun muassa lähettäjän tai vastaanottajan IP-osoitteen, protokollan tai portin perusteella. Kevin Mitnick huiputti palomuureja niin sanotulla IP-huijauksella (IP spoofing).[325] IP-osoite on melko helppo väärentää ja murtautuja voi väärän IP-osoitteen avulla tekeytyä luotettavaksi. Mikäli luvaton palvelin onnistuu vakuuttamaan palomuurin siitä, että se on kotoisin turvallisesta osoitteesta, se voi saada samat oikeudet kuin osoitteen todelliset käyttäjätkin. Kehittyneemmät palomuurit eivät mene tähän halpaan, vaan hylkäävät paketit, jotka väittävät olevansa lähiverkosta, mutta tulevat kuitenkin ulkopuolisesta verkosta.

Palomuurin ongelmana on rajanveto: salliako kaikki, mikä ei erikseen ole kiellettyä, vai kieltääkö kaikki, mitä ei erikseen sallita? Turvallisempi järjestelmä on usein hankala sekä ylläpitäjille että käyttäjille, toisaalta heikommalla suojauksella ei välttämättä saavuteta riittävää turvatasoa. On silti muistettava, että palomuuri voi valvoa vain kauttansa kulkevaa liikennettä. Hyväkään palomuuri ei anna täydellistä suojaa millekään tietoverkolle. Täysin turvallista järjestelmää ei ole olemassa, vaan käyttäjät ovat itse vastuussa verkon turvallisuudesta. Tietoturvaan liittyviä hyödyllisiä ohjeita löytyy mm. Viestintäviraston sivuilta.[326]

Pakettiverkkojen analysointi jonoteorian avulla

Pakettikytkentäinen liikenne muodostuu siis paketeista. Tämä ajatus alkoi kehittyä 1960-luvun alkupuolella. Pakettikytkentäisyys oli vaativa teknologinen haaste ottaen huomioon silloin käytettävissä olleet tietokoneet ja niiden (nykymittapuun mukaan äärimmäisen) rajallinen tietojenkäsittelykapasiteetti. Toisaalta haasteena oli ymmärtää ja analysoida miten laaja pakettipohjainen verkko toimisi erilaisissa kuormitustilanteissa ja

[324] Katso esimerkiksi B. Yagoda, A Short History of "Hack", The New Yorker, 6.3.2014, http://www.newyorker.com/tech/elements/a-short-history-of-hack

[325] Mitnick vangittiin vuonna 1995 ja pitkien oikeuskäsittelyjen jälkeen hän sai viiden vuoden vankeustuomion. Toisaalta hän kirjoitti yhdessä W.L. Simonin kanssa kirjan *The art of deception: Controlling the human element of security*, johon Google Scholarin mukaan on 1264 tieteellistä viittausta (26.11.2018). Konna vai sankari?

[326] https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvaohjeet.html

olosuhteissa. Tähän kysymykseen pyrittiin vastaamaan kehittämällä jonoteoriaa. Jonoteoria ei kuitenkaan ollut sinänsä uusi ala, vaan sitä oli kehitetty jo 1900-luvun alusta alkaen. Jonoteorian alkukohtana voidaan pitää Andrei Markovin vuonna 1906 julkaisemaa artikkelia, jossa hän esittää Markovin ketjun periaatteet.[327]

Verkon solmupisteeseen tulevaa pakettia voidaan käsitellä periaatteessa kolmella tavalla: se voidaan lähettää välittömästi eteenpäin, se voidaan hylätä tai se voidaan laittaa jonoon. Käytännössä tulevaa pakettia käsitellään aina sen verran, että tiedetään, minkälaisesta paketista on kysymys, joten ainakin osa paketista joudutaan laittamaan muistiin odottamaan jatkokäsittelyä. Tässä yhteydessä keskitytään vain jonotukseen ja sen vaikutukseen paketin saamaan palvelun laatuun eli viiveeseen ja paketin hylkäämisen todennäköisyyteen.

Tässä yhteydessä jonotusta käsitellään suppeasti yksinkertaisten mallien avulla, jotka usein ovat arkielämästä otettuja. Perusyksikkönä käytetään usein nimitystä *asiakas*, mutta täsmälleen samat periaatteet soveltuvat mihin tahansa järjestelmään, johon sisältyy jonotusmahdollisuus, myös silloin kun kyseessä on informaatiota sisältävä datapaketti, fyysinen paketti tai ihminen. Asiakas-termiä käytetään tässä yhteydessä, koska sitä on usein helpompi hahmottaa mielikuvana: asiakas saapuu ja jonottaa, häntä palvellaan ja sitten hän poistuu. Ensisijaisena tavoitteena on ajatusmallien, laskentamenetelmien ja keskeisimpien kaavojen esitteleminen.

Jonojen analysoinnin kannalta olennaisinta ovat seuraavat neljä asiaa:[328]

1. Minkälaisella prosessilla asiakkaat saapuvat
2. Palveluaikojen jakauma
3. Palvelupaikkojen määrä (*S*)
4. Odotuspaikkojen määrä (*N*)

Näihin pohjautuen yksinkertaisimmat jonot voidaan määritellä X/Y/S/N notaatiolla, jossa X määrittelee saapumisprosessin ja Y määrittelee palveluprosessin. Esimerkiksi M/M/S/N tarkoittaa:

M/M/S/N kutsut saapuvat Poisson-prosessin mukaisesti,

M/**M**/S/N palveluajat ovat eksponentiaalisesti jakautuneita,

M/M/**S**/N palvelupaikkojen määrä on *S* ja

[327] Markov julkaisi paperinsa ennen kuin Erlang julkaisi kaavansa. Erlangin kaava perustuu Markovin prosessiin.
[328] Perusteellinen kirja jonottamisen mallinnuksesta: Hassin, R. (2016). Rational queueing. https://content.taylorfrancis.com/books/download?dac=C2015-0-61989-6&isbn=9781498745284&format=googlePreviewPdf

M/M/S/N odotuspaikkojen määrä on *N*. Jos merkintää ei ole, odotuspaikkoja on ääretön määrä (ei siis nolla!).

Merkinnän M lisäksi yleisimpiä merkintöjä ovat:

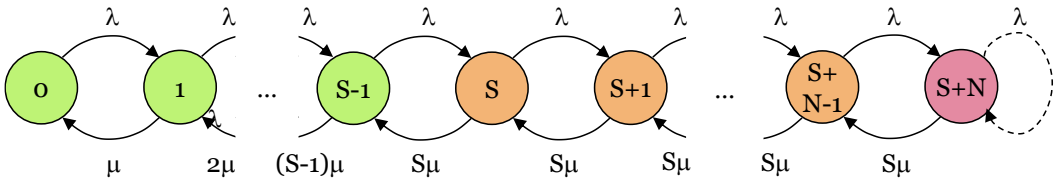
D deterministinen prosessi eli saapumisaikavälit tai palveluajat ovat vakiopituisia ja

G yleinen aikajakauma.

Vaikeammin käsiteltäviä D ja G-jonomalleja ei tarkastella tässä kurssissa. Pääsääntönä voidaan sanoa, että jonotusjärjestelmissä keskimääräiset odotusajat ja jonojen pituudet ovat sitä pidempiä, mitä suurempia saapumis- ja palveluaikojen vaihtelut ovat.

Vertaamalla liikenneteoreettisia laskelmia (esim. Erlangin kaavan johto luvussa 4) ja edellä esitettyjä laskelmia, voidaan havaita selviä yhtäläisyyksiä. Itse asiassa Erlangin malli on M/M/S/0 -järjestelmä jonoteorian kannalta.

M/M/S/N-järjestelmän tasapainotilan kaavio on esitetty kuvassa 7.12. Merkintä μ tarkoittaa yhden palvelupaikan palveluintensiteettiä, eli sitä miten monta asiakasta yksi palvelupaikka saa palveltua keskimäärin aikayksikössä. Kuten 4. luvussa esitettiin eksponentiaalisesti jakautuneet palveluajat tarkoittavat sitä, että päättymisen todennäköisyys ei riipu siitä, miten kauan palvelu on jo kestänyt. Keskimääräinen palveluaika (niille asiakkaille, jotka palveluun pääsevät) on tällöin $h = 1/\mu$.



Kuva 7.12. M/M/S/N-järjestelmän tasapainotilan kaavio.

Tilojen todennäköisyydet voidaan laskea kuvan mukaisesti rekursiivisesti seuraavasti:

$$\begin{aligned} i\mu P(i) &= \lambda P(i-1) \quad kun\ \ 0 < i \le S \\ S\mu P(i) &= \lambda P(i-1) \quad kun\ \ S < i \le S+N \\ P(i) &= 0 \qquad\qquad kun\ \ i > S+N, tai\ i < 0 \end{aligned} \tag{7.1}$$

Näistä saadaan tilojen todennäköisyydet tilan 0 todennäköisyyteen suhteutettuna:

$$\begin{aligned} P(i) &= \frac{A^i}{i!}P(0) \qquad kun\ \ 1 \le i \le S \\ P(i) &= \left(\frac{A}{S}\right)^{i-S} P(S) \quad kun\ \ S < i \le S+N \end{aligned} \tag{7.2}$$

jossa $A = \lambda/\mu$. Jonoteoriassa A:n sijasta käytetään yleensä merkintää $\rho = A/S$, joka kuvaa yhden palvelupaikan keskimääräistä kuormitusta. Todennäköisyys P(0) voidaan ratkaista merkitsemällä eri tilojen todennäköisyyksien summa ykköseksi.

Tässä vaiheessa (siis peruskurssin puitteissa) on olennaista ymmärtää periaate, jolla tilojen todennäköisyydet voidaan laskea, kun oletetaan, että asiakkaita tulee Poisson-prosessin mukaisesti ja palveluajat ovat eksponentiaalisesti jakautuneita. Lisäksi kun tilojen todennäköisyydet tunnetaan, voidaan Littlen lauseen (kaava 4.7 eli $A = \lambda h$) avulla laskea keskimääräinen odotusaika.

M/M/1 -jono

Käytännön kannalta on hyvä luoda itselle kuva siitä, miten yksinkertaiset jonot tyypillisesti käyttäytyvät. Tähän tarkoitukseen on paras tarkastella yksinkertaisinta M/M/1-jonomallia, jossa siis on yksi palvelupaikka ja ääretön määrä odotuspaikkoja ja tarjottu liikenne on Poisson-prosessin mukaista. Tällöin saadaan varsin yksinkertaisesti (kun A < 1):[329]

$$P(i) = (1 - A)A^i \qquad (7.3)$$

Todennäköisyydet muodostavat siis geometrisen sarjan. Asiakkaiden määrän keskiarvoksi saadaan tällöin:

$$E[i] = \frac{A}{1 - A} \qquad (7.4)$$

Tässä asiakkaiden määrän keskiarvossa ovat siis mukana sekä odottavat että palveltavana oleva asiakas. Kun A < 1 niin kaikki asiakkaat saavat lopulta palvelua, joten heitä on keskimäärin palveltavana A kappaletta. Keskimääräinen odottavien asiakkaiden määrä on siten:

$$E[w] = \frac{A}{1 - A} - \mathrm{A} = \frac{A^2}{1 - A} \qquad (7.5)$$

Useimmiten ollaan kiinnostuneita keskimääräisestä odotusajasta (h_w). Se saadaan soveltamalla Littlen lausetta kaikkiin asiakkaisiin, jolloin lopputulemana on:

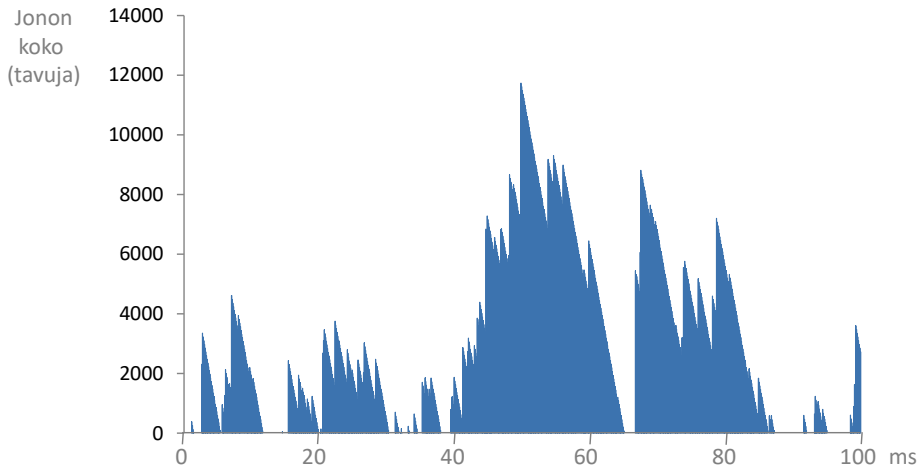
$$h_w = \frac{A}{1 - A}\, h \qquad (7.6)$$

Tässä on huomattava, että keskimääräinen odotusaika sisältää myös ne asiakkaat, jotka pääsevät välittömästi palveltavaksi.

[329] Jos A ≥ 1, niin mitään tasapainotilaa ei synny.

Nyt voi näyttää siltä, että jonoteoria on pelkkiä kaavoja. Toki kaavoja riittää varsinkin, kun mennään ilmiöihin syvemmälle ja monimutkaisempiin malleihin. Toisaalta kaavat eivät ole sillä tavalla oleellisia, että kovinkaan montaa niistä kannattaa opetella ulkoa. Jos jotain on hyvä muistaa, niin kaava 7.6 eli keskimääräinen odotusaika M/M/1-järjestelmässä. Sen mukaan, kun kuormitus on puolet palvelukapasiteetista (eli A = 0,5), odottamiseen menee keskimäärin yhtä paljon aikaa kuin itse palveluun. Kun kuormitusaste on 90 prosenttia, odottamiseen menee keskimäärin 9 kertaa enemmän aikaa kuin palveluun.

Kuvassa 7.13 on esitetty jonon käyttäytyminen satunnaiselle dataliikenteelle, siten että järjestelmän kapasiteetti on 10 Mbit/s ja keskimääräinen kuormitus on 0,8. Tällä nopeudella keskimääräinen palveluaika (h) on 0,8 ms. Tässä siis oletetaan, että paketin palveluaika on suorassa suhteessa paketin kokoon, mikä on oletettavaa silloin kun järjestelmän suorituskykyä rajoittaa lähtevän linkin kapasiteetti. Jonon pituuden vaihtelut voivat olla hyvinkin suuria jo 80 prosentin kuormituksella, kuten edellä esitetty teoreettinen tarkastelu osoitti. Samoin kuin aikaisemmassa liikennesimulaatiossa (kuva 4.13), tässäkin hahmottaa helposti liikenteen käyttäytymisessä erilaisia vaiheita. Esimerkiksi kuvassa 7.13 jonon nopealle kasvamiselle noin 40 ja 50 millisekunnin välillä helposti olettaa löytyvän jonkun ymmärrettävän syyn. Mitään erityistä syytä ei kuitenkaan ole, vaan liikenteen vaihtelut ovat simuloinnin vuoksi täysin satunnaisia.



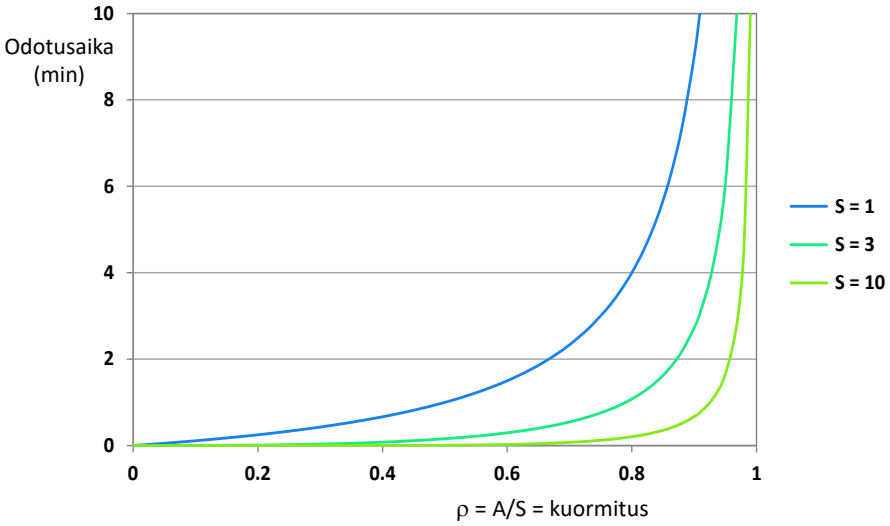
Kuva 7.13. Simuloitua liikennettä M/M/1-järjestelmässä, y-akselilla datan määrä jonossa tavuina, x-akselilla aika, yhteensä 100 ms. paketin keskikoko on 1000 tavua, pakettien välisen ajan keskiarvo on 1 ms.

M/M/S -jono

Entä sitten hieman monimutkaisemmat jonot, esimerkiksi M/M/S? Kuvassa 7.14 on esitetty keskimääräinen odotusaika kuormituksen (A/S) funktiona, kun palvelupaikoilla on yhteinen jono (ilman että jonottajien määrää rajoitetaan). Kun palvelupaikkojen määrä kas-

vaa, päästään yhä lähemmäs 100 prosentin kuormitusta ilman merkittävää odotusajan kasvua. Jos odotusaika halutaan pitää alle keskimääräisen palveluajan, kolmen palvelupaikan tapauksessa kuormitus voidaan nostaa noin 79 prosenttiin ja kymmenellä palvelupaikalla noin 92,5 prosenttiin. Käytännössä täytyy lisäksi huomioida se, että edes nykyistä tarjottua liikennettä ei yleensä tunneta tarkasti, saati tulevaa. Käytännössä jonon kuormitusaste on syytä pitää alle 80 prosentin, jotta vältetään kohtuuttoman pitkät jonot ja viiveet.

Käytännössä pakettiverkkojen mitoituksessa täytyy ottaa huomioon kaikki yhteyden laatuun vaikuttavat tekijät. Kun yhteyden bittinopeus on suuri (esim. 1 Gbit/s), jono voi olla paketeilla mitattuna pitkä ilman, että loppukäyttäjän havaitsema viive kasvaa merkittävästi. Pitää myös muistaa, että odotuspaikkojen lisääminen ei auta silloin kun palvelukapasiteetti ei riitä kaikkien asiakkaiden palvelemiseen tai pakettien välittämiseen. Tällöin suuret puskurit ovat pikemminkin haitallisia, koska ne väistämättä kasvattavat viiveitä.



Kuva 7.14. Keskimääräinen odotusaika kuormituksen funktiona M/M/S-järjestelmillä, kun keskimääräinen palveluaika on 1 min.

Esimerkki 7.1. M/M/1 -jonojärjestelmien analyysi

Reitittimellä on yksi jono linkille, jonka nopeus on 10 Mbit/s. Paketteja tulee keskimäärin 750 kappaletta sekunnissa ja pakettien keskimääräinen koko on 1 kB. Laske keskimääräinen odotusaika niille paketeille, jotka joutuvat jonottamaan, kun oletetaan että pakettien koko on eksponentiaalisesti jakautunut. Lisäksi oletetaan, että paketit saapuvat eksponentiaalisin väliajoin ja että puskurin koko riittävä, jotta käytännössä kaikki tulevat paketit mahtuvat puskuriin.

Ratkaisu

Voimme soveltaa tässä suoraan M/M/1 jonon kaavaa asiakkaiden eli pakettien määrälle (kaava 7.4). Ensin pitää laskea mikä on keskimääräinen kuormitus. Vastaus on (750 × 8 kbit/s)/ (10 Mbit/s) = 0,6 (yksiköiden kanssa on syytä olla tarkkana). Onneksi keskimääräinen kuormitus on pienempi

kuin yksi, koska vain tällöin voimme soveltaa aiemmin esitettyjä kaavoja. Keskimäärin järjestelmässä on siten

$$E[i] = \frac{0{,}6}{1-0{,}6} = 1{,}5\ pakettia.$$

Näistä siis keskimäärin 0,6 on palveltavana (koska jokainen lopulta pääsee linkille), joten jonossa on keskimäärin 0,9 pakettia. Se kuinka moni paketti joutuu jonoon aikayksikössä, voidaan laskea sen tiedon pohjalta, että uuden paketin tullessa todennäköisyys, että paketti on palveltavana (eli sen lähetys on käynnissä), on 0,60 (miksi näin, se kannattaa selvittää itselleen huolellisesti!). Paketteja tulee yhteensä 750 sekunnissa, joten jonoon joutuu keskimäärin 450 pakettia sekunnissa. Soveltamalla Littlen lausetta jonottajiin saadaan:

$$t = \frac{E[w]}{\lambda} = \frac{0{,}9}{450/s} = 0{,}002\ s\ = 2\ ms$$

Huomaa myös se, että kaavalla 7.6 saataisiin arvo: $h_w = \frac{A}{1-A} \cdot h = \frac{0{,}6}{1-0{,}6} \cdot 0{,}8\ ms = 1{,}2\ ms$, sillä keskimääräinen paketin lähetysaika h = 8 kbit / 10 Mbit/s = 0,8 ms. Tämä viive on pienempi kuin edellä laskettu arvo, koska kaavan 7.6 laskelmassa mukana ovat kaikki paketit, eli myös ne jotka pääsevät suoraan lähetettäväksi, kun taas tehtävässä pyydettiin ottamaan huomioon vain jonottamaan joutuvat paketit.

Pakettien koot eivät tietenkään voi olla eksponentiaalisesti jakautuneita, koska tavu on pienin yksikkö mitä paketissa käsitellään. M/M/1-malli antaa kuitenkin kohtuullisen hyvän arvion jonon käyttäytymisestä. M/M/1-mallin perusteella 30 paketin puskurilla tuleva paketti jouduttaisiin hylkäämään todennäköisyydellä $5\ 10^{-8}$.

Esimerkki 7.2. Jonojärjestelmän suorituskyvyn parantaminen

Tarkastellaan tapausta, jossa palvelupaikkaan (vaikkapa valintamyymälän kassalle) tulee keskimäärin yksi asiakas minuutissa, palveluaika on eksponentiaalisesti jakautunut ja jonon pituutta ei rajoiteta. Oletetaan aluksi, että asiakkaan palvelu kestää keskimäärin yhden minuutin, joten yksi palvelupaikka ei riitä, koska jonon keskipituus kasvaisi kohti äärettömyyttä. Tehtävänä on vertailla kolmea vaihtoehtoa, joilla tilannetta voidaan parantaa:

a) Lyhennetään palveluaikaa 30 sekuntiin.

b) Lisätään toinen palvelupiste siten, että palvelupaikoilla on yhteinen jono.

c) Jaetaan asiakkaat satunnaisesti kahteen palvelupaikkaan, joilla on kummallakin oma jononsa (eli asiakas ei valitse jonoansa, vaan valinta tapahtuu satunnaisesti, eikä asiakas myöskään vaihda jonoa, vaikka toinen palvelupaikka sattuisi olemaan vapaana).

Ratkaisu

Kohdat *a* ja *c* ovat selvästi M/M/1-järjestelmiä, jolloin keskimääräinen odotusaika voidaan laskea kaavalla 7.6. eli a-kohdassa h = 0,5 ja λ = 1 ja *c*-kohdassa kummallakin palvelupaikalla h = 1 ja λ = 0,5, joten kummassakin tapauksessa kuormitus on sama eli 0,5. Keskimääräinen odotusaika on kuitenkin *c*-kohdassa kaksinkertainen.

Kohta *b* muodostaa M/M/2-järjestelmän, jolloin yleensä selvin tapa odotusajan laskemiseksi on ratkaista tilojen todennäköisyydet kaavalla 7.1. Saatujen todennäköisyyksien avulla voidaan sitten laskea keskimääräinen odotusaika. Esimerkin tapauksessa keskimääräiseksi odotusajaksi saadaan 20 s, joka ehkä hieman yllättäenkin on lyhempi kuin *a*-kohdan tapauksessa. Kuitenkin, jos otetaan huomioon myös asiakkaan palveluaika, on *a*-kohta asiakkaan kannalta edullisempi. Tulokset on koottu taulukkoon 7.1. Lukija voi miettiä, miksi toisaalta *b*-kohdassa odotusaika on lyhempi kuin a-kohdassa ja toisaalta miksi kokonaisaika on pitempi *b*-kohdassa.

Taulukko 7.1. Odotus- ja palveluajat esimerkin tapauksessa.

		Keskimääräinen odotusaika	Keskimääräinen palveluaika	Kokonaisaika
a)	M/M/1, h = 0,5 min	30 s	30 s	60 s
b)	M/M/2, h = 1 min	20 s	60 s	80 s
c)	2 × M/M/1, h = 1 min	60 s	60 s	120 s

Esimerkki 7.3. Pankkiautomaatin suorituskyvyn analyysi

Toiseksi esimerkiksi aikanaan useampaan kertaan tenttitehtävänä ollut ongelma. Pankkiautomaatilla käy keskimäärin 8 asiakasta tunnissa. Asiointi kestää keskimäärin 3 minuuttia siten että aika on eksponentiaalisesti jakautunut. Asiakas jää odottamaan, jos automaatilla on yhteensä korkeintaan kolme asiakasta, muutoin asiakas siirtyy viereisen pankin tiskille asioimaan (silloin kun pankeissa vielä oli palvelutiskejä).

a) Kuinka monta asiakasta tunnissa siirtyy pankkiin?

b) Kuinka monta asiakasta on keskimäärin odottamassa pääsyä automaatille?

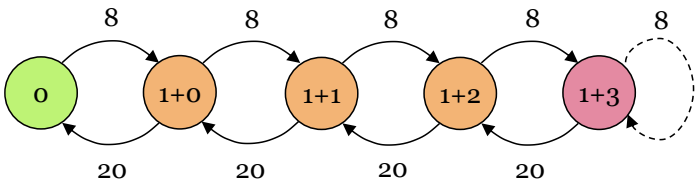
c) Kuinka pitkään asiakas joutuu keskimäärin odottamaan ennen kuin hän pääsee automaatille, kun asiakkaan tullessa täsmälleen yksi henkilö on automaatilla ja yksi odottamassa?

Ratkaisu

Vaikka kyseessä on ilmeisen jonoteoreettinen ongelma, ei sen ratkaisemiseen tarvitse muistaa kaavoja (toisin kuin varsin monet tenttiin osallistuneet olettavat). Tässä riittää, kun ymmärtää mallintamisen perusteet ja joitakin laskentamenetelmiä. Ensiksi täytyy selvittää järjestelmän mahdolliset tilat. Automaatilla voi uuden asiakkaan tullessa olla nollasta neljään asiakasta: jos automaatilla on kolme asiakasta, tuleva asiakas jää neljänneksi, jonka jälkeen tulevat asiakkaat siirtyvät pankkiin, kunnes yksi asiakas poistuu (aika moni erehtyy jo tässä vaiheessa).

Asiakkaita tulee 8 tunnissa, jolloin saapumisintensiteetti $\lambda = 8$, kun aikayksikkönä on tunti. Käytettävällä aikayksiköllä ei sinänsä ole väliä, kunhan käyttää koko ajan samaa yksikköä (aikayksiköiden kanssa erehtyy noin neljäsosa vastaajista). Keskimääräinen palveluaika on 3 min eli palveluintensiteetti on $60/3 = 20$ asiakasta tunnissa. Oletuksena on, että asiakkaat tulevat toisistaan riippumatta eli yksi kerrallaan, jolloin järjestelmän tilassa tapahtuu siirtymisiä vain peräkkäisten tilojen välillä. Näin saadaan kuvan 7.15 mukainen järjestelmä.

On huomattava, että esimerkiksi järjestelmän tila 1+2 tarkoittaa tässä tilannetta, jossa yksi asiakas on automaatilla ja kaksi odottamassa (muunkinlaiset merkintätavat ovat mahdollisia, kunhan merkinnät ovat johdonmukaisia). Yläpuolella oleva 8 tarkoittaa, että tunnissa tulee keskimäärin 8 asiakasta (tämä on useimmissa vastauksissa oikein) ja alapuolella oleva 20 tarkoittaa, että jos järjestelmässä on vähintään yksi asiakas, asiakkaita poistuu keskimäärin 20 kappaletta tunnissa (tähän asti oikein vastanneista ehkä noin kolmanneksen mielestä poistuvien asiakkaiden määrä aikayksikössä riippuu jonon pituudesta, joka tässä tehtävässä on siis väärä oletus).



Kuva 7.15. Jonotusjärjestelmän tilat ja niiden välisten siirtymisten intensiteetit.

Tämän jälkeen lasketaan tilojen todennäköisyydet esimerkiksi siten, että otetaan lähtökohdaksi tilan 0 todennäköisyys, jolloin muiden tilojen todennäköisyydet voidaan ratkaista rekursiivisesti kaavoista $8\,P(0) = 20\,P(1+0)$, $8\,P(1+0) = 20\,P(1+1)$ jne. Kun vielä tiedetään, että järjestelmä on täsmälleen yhdessä tilassa, voidaan todennäköisyydet laskea suhteellisen helposti (jos kuva on osattu johtaa oikein, useimmat osaavat kyllä suorittaa laskelmatkin oikein). Tilojen todennäköisyyksiksi saadaan:

$P(0) = 0{,}6062$

$P(1+0) = 0{,}2425$

$P(1+1) = 0{,}0970$

$P(1+2) = 0{,}0388$

$P(1+3) = 0{,}0155$

Tämän jälkeen kohtien *a* ja *b* ratkaisut ovat melko yksinkertaisia. Kohdan *a* vastaus on tilan P(1+3) todennäköisyys (0,0155) kertaa tulevien asiakkaiden määrä tunnissa (20) eli 0,31. Kohdassa *b* vastaus saadaan yhteenlaskulla $1\,P(1+1) + 2\,P(1+2) + 3\,P(1+3) = 0{,}221$ (tässäkin kohdassa on varsin helppo erehtyä ottamalla mukaan myös automaattia käyttävä asiakas).

Kohdan *c* laskemiseen ei edellä esitettyjä laskelmia tarvita ollenkaan. Ainoa mitä tarvitsee tietää, on eksponentiaalisen jakauman niin sanottu muistamattomuusominaisuus, joka tässä tapauksessa merkitsee sitä, että asiakkaan jäljellä oleva keskimääräinen palveluaika on aina vakio. Tässä tapauksessa automaatilla olevan asiakkaan jäljellä oleva aika on keskimäärin 3 min ja ensimmäisenä jonossa olevan automaatilla käyttämä aika on keskimäärin myös 3 min eli yhteensä 6 min.