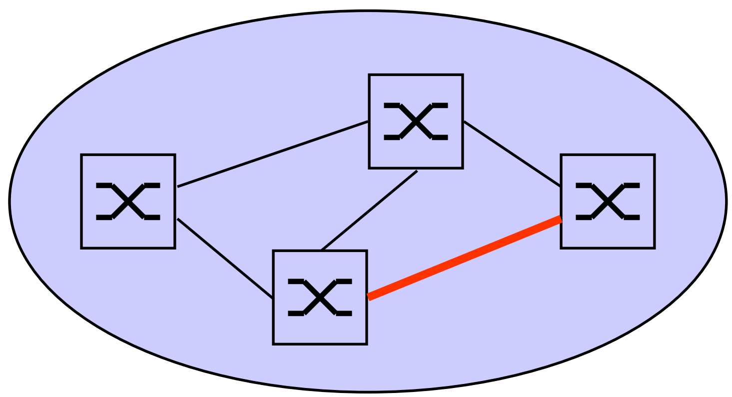
3. Esimerkkejä

Sisältö

- Puhelinliikenteen malli
- Pakettitason malli dataliikenteelle
- Vuotason malli elastiselle dataliikenteelle
- Vuotason malli virtaavalle dataliikenteelle

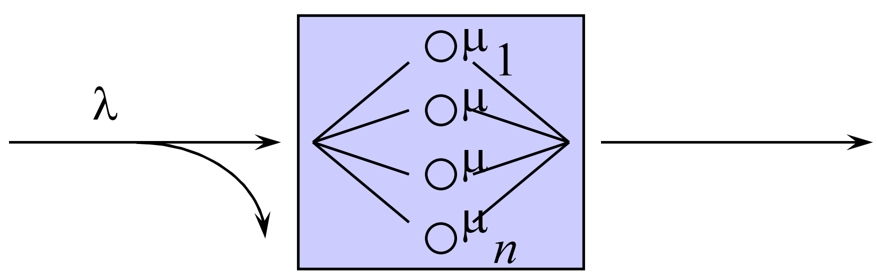
Klassinen puhelinliikenteen malli (1)

- **Menetysjärjestelmiä** on perinteisesti käytetty puhelinliikenteen kuvaamiseen kutsutasolla
  - Uranuurtajana oli tanskalainen matemaatikko *A.K. Erlang* (1878-1929).
- Tarkastellaan kahden keskuksen välisellä linkillä kulkevaa puhelinliikennettä (klassinen liikenneteoreettinen ongelma)
  - Liikenne koostuu käynnissä olevista puheluista, jotka käyttävät ko. linkkiä

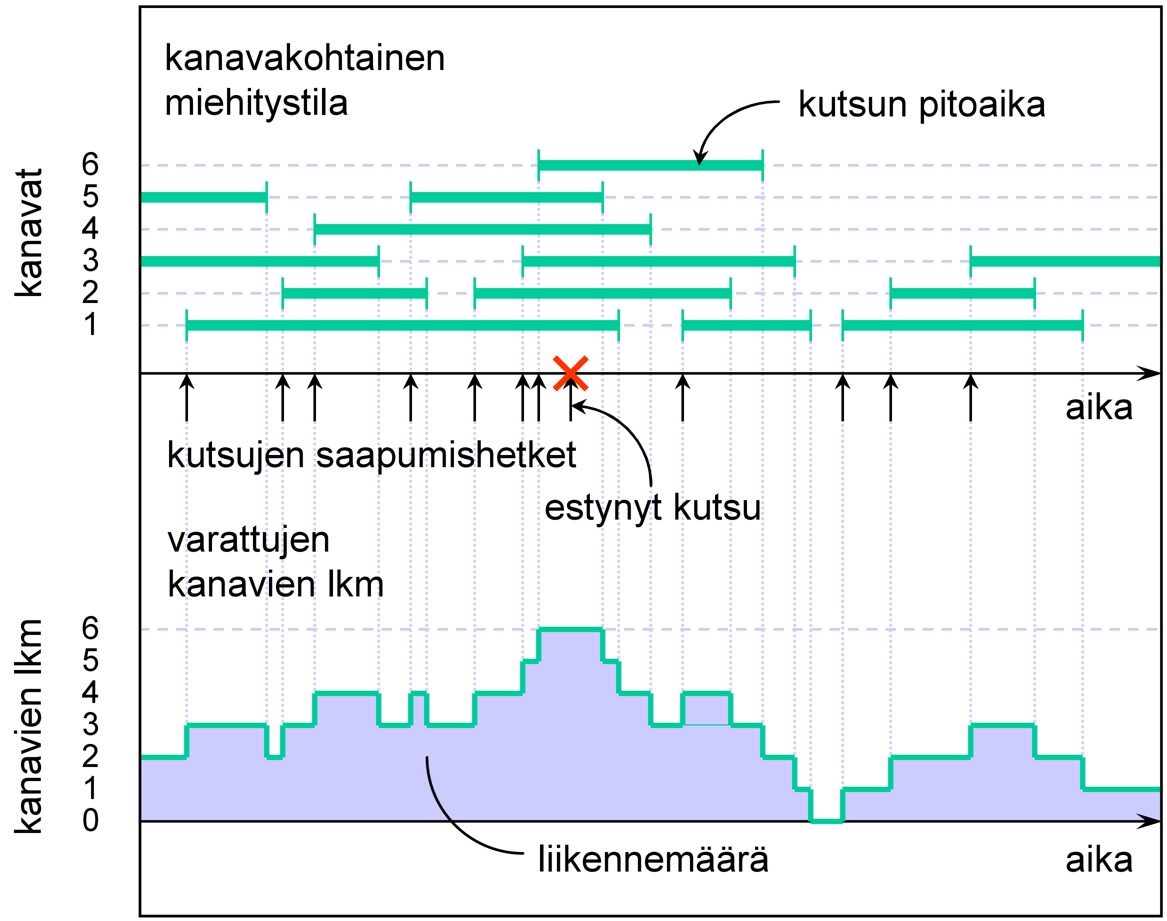


Klassinen puhelinliikenteen malli (2)

- Erlang käytti mallina **puhdasta menetysjärjestelmää** ($m = 0$)
  - asiakas = kutsu = puhelu
    - λ = uusien kutsujen saapumisintensiteetti (kutsua/aikayks.)
  - palveluaika = (kutsun) pitoaika
    - $h = 1/\mu$ = keskimääräinen pitoaika (aikayks.)
  - palvelija = yksittäinen linkin kanava
    - n = linkillä olevien rinnakkaisten kanavien lkm



Liikenneprosessi



Liikenneintensiteetti

- Tarjotun liikenteen voimakkuutta kuvaa liikenneintensiteetti a
- **Määritelmä**: **Liikenneintensiteetti** a on saapumisintensiteetin λ ja keskimääräisen pitoajan h tulo:

$$a = \lambda h$$

  - Liikenneintensiteetti on paljas luku, mutta asiayhteyden korostamiseksi sen "yksiköksi" usein merkitään **erlang** (**erl**)
  - Littlen kaavan nojalla: liikenneintensiteetti kertoo keskimäärin käynnissä olevien kutsujen lkm:n vastaavassa äärettömässä systeemissä
- Esimerkki:
  - Uusia puheluita tulee tunnissa keskimäärin 1800 kpl ja puhelun keskimääräinen pitoaika on 3 min. Tällöin liikenneintensiteetiksi tulee

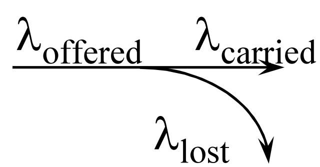
$$a = 1800 * 3/60 = 90 \text{ erlang}$$

Esto

- Menetysjärjestelmässä osa kutsuista menetetään:
  - Saapuva kutsu menetetään, jos kaikki kanavat on varattu (so. systeemi on täysi) ko. kutsun saapuessa
  - Termi **esto** (blocking) viittaa tähän tapahtumaan
- Menetysjärjestelmissä voidaan määritellä useita eri estosuureita:
  - **Kutsuesto** B_c = tn, että saapuva kutsu menetetään = niiden saapuvien kutsujen osuus, jotka menetetään
  - **Aikaesto** B_t = tn, että systeemi on täysi (mielivaltaisena ajanhetkenä) = se osuus ajasta, jolloin systeemi on täysi
- Nämä suureet eivät välttämättä ole samoja
  - Esimerkki: oma kännykkäsi
  - Mutta jos uudet kutsut saapuvat Poisson-prosessin mukaisesti, niin $B_\mathrm{c} = B_\mathrm{t}$
- Kutsuesto kuvaa paremmin käyttäjien kokemaa palvelun laatua
- Aikaesto taas on suoraviivaisemmin laskettavissa oleva suure

Kutsuintensiteetit

- Menetysjärjestelmässä voidaan erottaa seuraavat kutsuintensiteetit:
  - λ_{offered} = kaikkien saapuvien kutsujen saapumisintensiteetti
  - λ_{carried} = palveluun päässeiden kutsujen saapumisintensiteetti
  - λ_{lost} = menetettyjen kutsujen saapumisintensiteetti



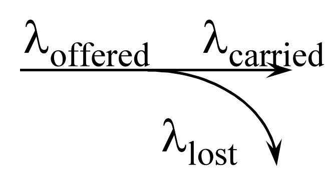
$$\lambda_{\text{offered}} = \lambda_{\text{carried}} + \lambda_{\text{lost}} = \lambda$$

$$\lambda_{\text{carried}} = \lambda(1 - B_{\text{c}})$$

$$\lambda_{\text{lost}} = \lambda B_{\text{c}}$$

Liikennevirrat

- Eri kutsuintensiteettien avulla voidaan määritellä seuraavat liikennevirrat:
  - **Tarjottu liikenne** $a_{\text{offered}} = \lambda_{\text{offered}} h$
  - **Kuljetettu liikenne** $a_{\text{carried}} = \lambda_{\text{carried}} h$
  - **Menetetty liikenne** $a_{\text{lost}} = \lambda_{\text{lost}} h$



$$a_{\text{offered}} = a_{\text{carried}} + a_{\text{lost}} = a$$

$$a_{\text{carried}} = a(1 - B_{\text{c}})$$

$$a_{\text{lost}} = aB_{\text{c}}$$

- Tarjottu ja menetetty liikenne ovat hypoteettisia suureita, mutta kuljetettu liikenne on mitattavissa, sillä Littlen kaavan mukaan se kertoo keskimäärin käynnissä olevien kutsujen lkm:n

Liikenneteoreettinen analyysi (1)

- Järjestelmän kapasiteetti
  - n = linkissä olevien rinnakkaisten kanavien lkm
- Liikenne
  - a = (tarjottu) liikenneintensiteetti
- Palvelun laatu (käyttäjän näkökulmasta)
  - B_{c} = kutsuesto = tn, että saapuva kutsu menetetään
- Tarkastellaan tyyppiä **M/G/n/n** olevaa **puhdasta menetys-järjestelmää**, ts. oletetaan, että
  - uudet kutsut saapuvat **Poisson-prosessin** mukaisesti (intensiteetillä λ) ja
  - kutsujen pitoajat ovat riippumattomia ja samoin jakautuneita noudattaen **mitä tahansa jakaumaa**, jonka odotusarvo on h

Liikenneteoreettinen analyysi (2)

- Tällöin eri tekijöiden (järjestelmä, liikenne ja palvelun laatu) välisen yhteyden kertoo ns. **Erlangin kaava**

$$B_{\mathrm{c}} = \mathrm{Erl}(n,a) := \frac{\frac{a^n}{n!}}{\sum_{i=0}^{n} \frac{a^i}{i!}}$$

$$n! = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1, \quad 0! = 1$$

- Vaihtoehtoisia nimiä:
  - Erlangin B-kaava
  - Erlangin estokaava (blocking formula)
  - Erlangin menetyskaava (loss formula)
  - Erlangin ensimmäinen kaava

Esimerkki

- Tarkastellaan esimerkkinä hyvin pientä systeemiä. Oletetaan, että rinnakkaisten kanavien lkm on $n = 4$ ja liikenneintensiteeetti $a = 2.0$ erlang. Tällöin kutsuestoksi B_c tulee

$$B_\mathrm{c} = \mathrm{Erl}(4,2) = \frac{\frac{2^4}{4!}}{1 + 2 + \frac{2^2}{2!} + \frac{2^3}{3!} + \frac{2^4}{4!}} = \frac{\frac{16}{24}}{1 + 2 + \frac{4}{2} + \frac{8}{6} + \frac{16}{24}} = \frac{2}{21} \approx 9.5\%$$

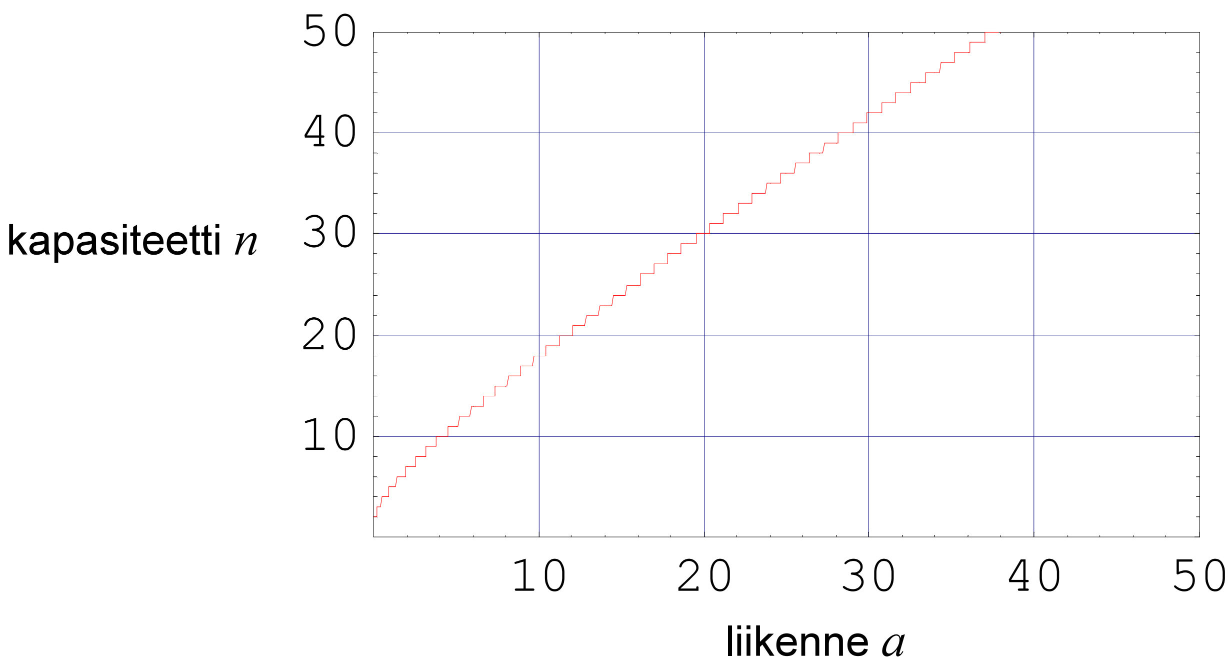
- Jos linkin kapasiteetti kasvatetaan $n = 6$ kanavaan, niin B_c pienenee arvoon

$$B_\mathrm{c} = \mathrm{Erl}(6,2) = \frac{\frac{2^6}{6!}}{1 + 2 + \frac{2^2}{2!} + \frac{2^3}{3!} + \frac{2^4}{4!} + \frac{2^5}{5!} + \frac{2^6}{6!}} \approx 1.2\%$$

Kapasiteetti liikenteen funktiona

- Asetetaan palvelun laatuvaatimukseksi, että kutsuesto $B_c < 1\%$
- Tarvittava kapasiteetti n liikenteen a funktiona saadaan kaavalla:

$$n(a) = \min\{i = 1,2,\ldots \mid \mathrm{Erl}(i,a) < 0.01\}$$

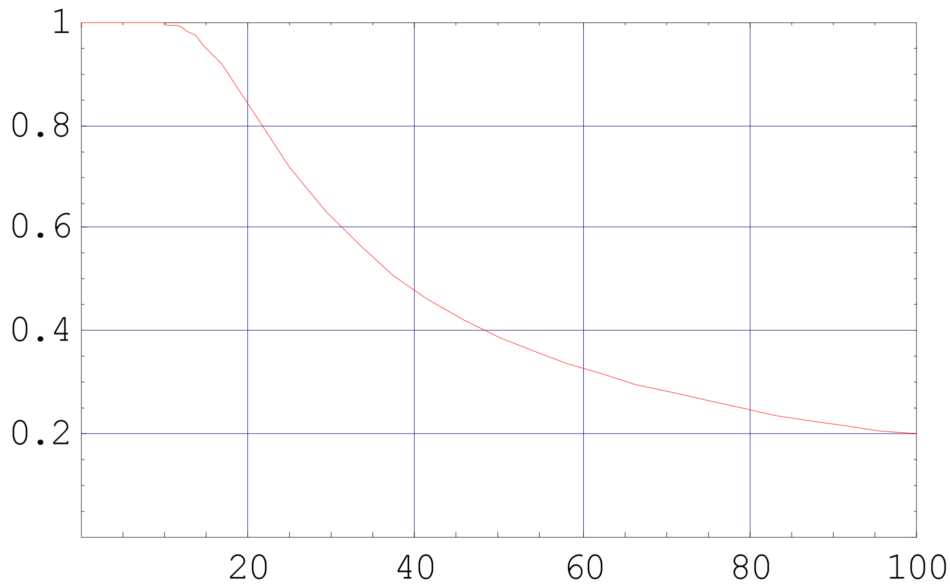


Palvelun laatu liikenteen funktiona

- Oletetaan sitten, että rinnakkaisten kanavien lkm eli kapasiteetti $n = 20$
- Palvelun laatu $1 - B_c$ liikenteen a funktiona saadaan kaavalla:

$$1 - B_c(a) = 1 - \mathrm{Erl}(20, a)$$

palvelun laatu
$1 - B_c$

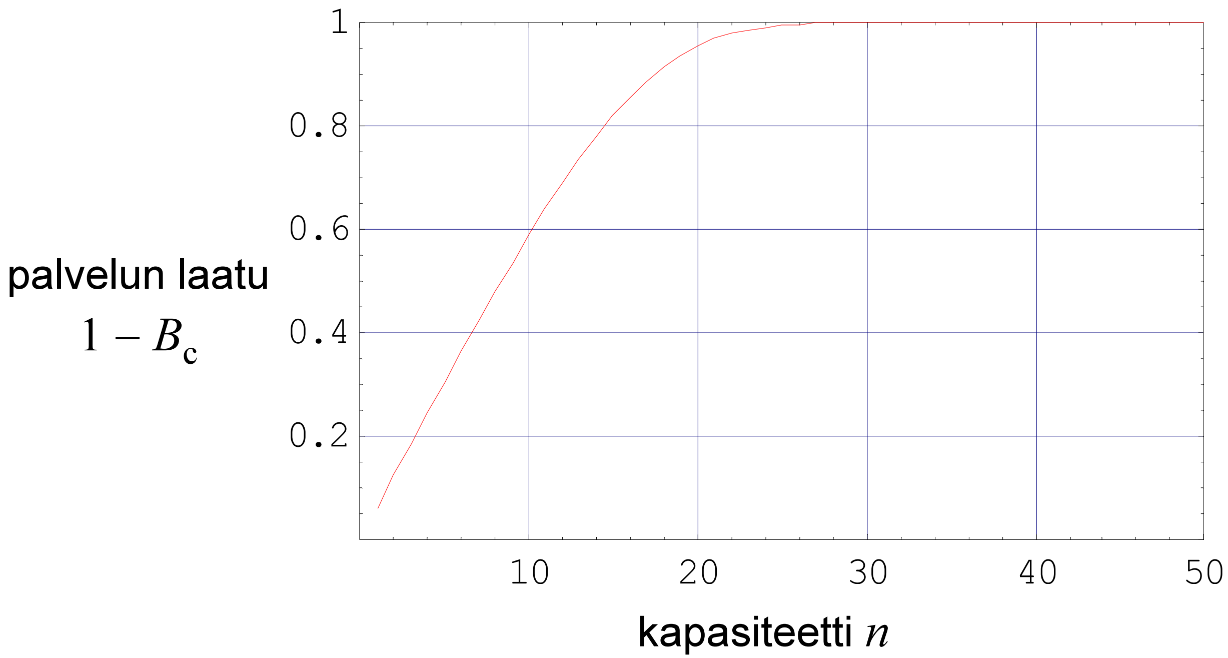


liikenne a

Palvelun laatu kapasiteetin funktiona

- Oletetaan lopuksi, että tarjotun liikenteen intensiteetti $a = 15.0$ erlang
- Palvelun laatu $1 - B_c$ kapasiteetin n funktiona saadaan kaavalla:

$$1 - B_c(n) = 1 - \mathrm{Erl}(n, 15.0)$$

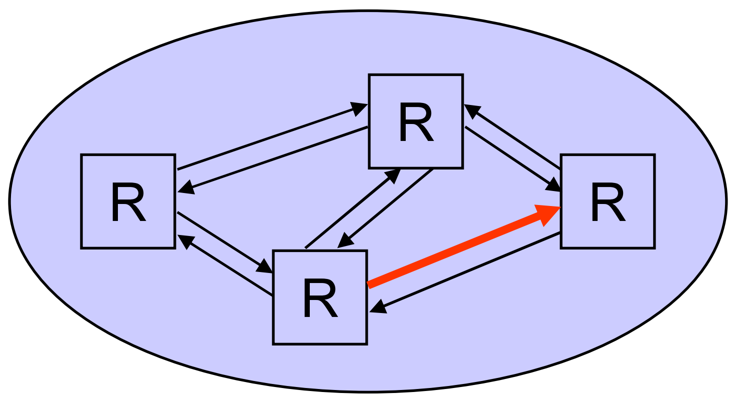


Sisältö

- Puhelinliikenteen malli
- Pakettitason malli dataliikenteelle
- Vuotason malli elastiselle dataliikenteelle
- Vuotason malli virtaavalle dataliikenteelle

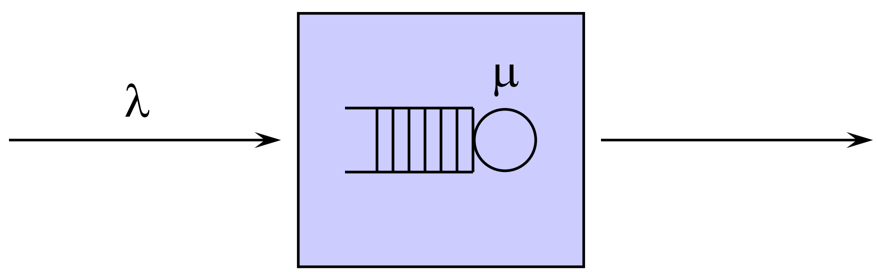
Pakettitason malli dataliikenteelle (1)

- **Jonotusjärjestelmät** soveltuvat dataliikenteen kuvaamiseen pakettitasolla
  - Uranuurtajina 60- ja 70-luvuilla ARPANET:in tutkijat, eritoten *L. Kleinrock* (`http://www.lk.cs.ucla.edu/`)
- Tarkastellaan yhtä IP-reitittimen ulostulolinkkiä
  - Liikenne koostuu linkkiä pitkin lähetetyistä datapaketeista

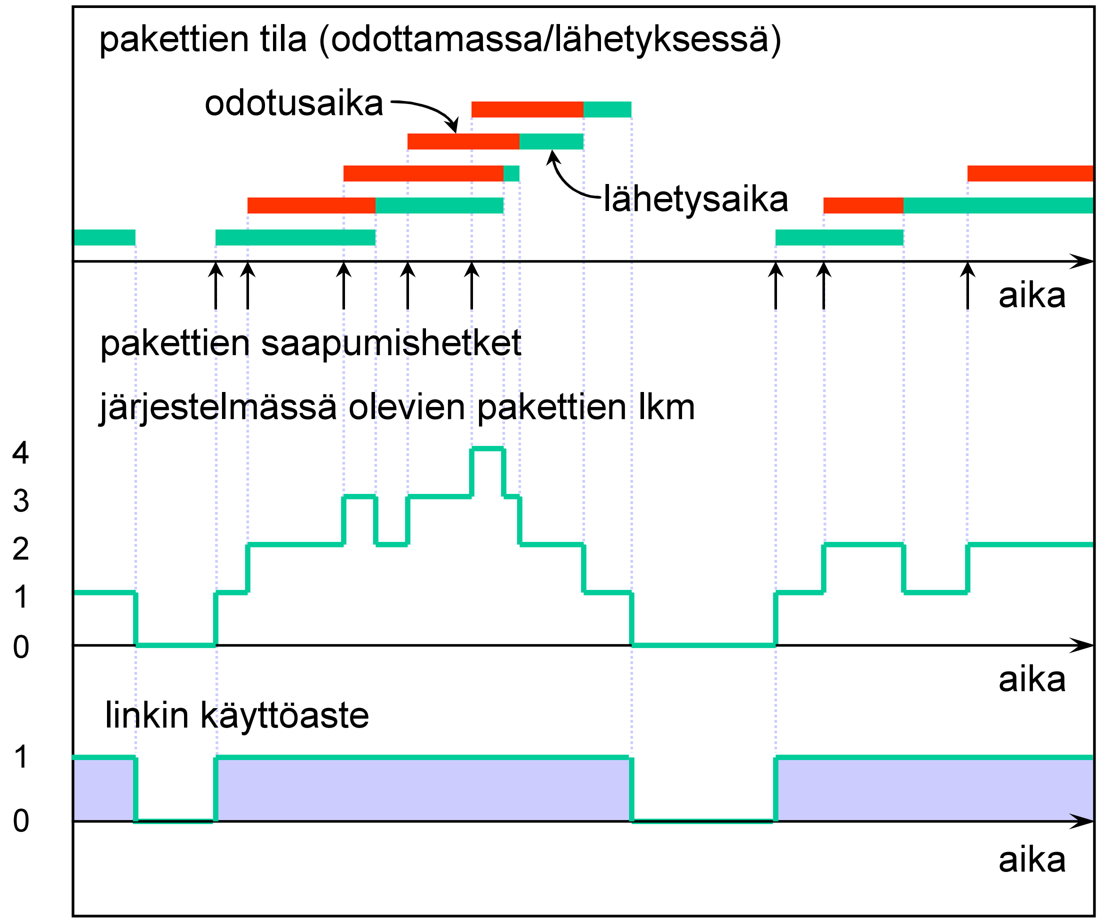


Pakettitason malli dataliikenteelle (2)

- Klassisena mallina on yhden palvelijan ($n = 1$) **puhdas jonotus-järjestelmä**, jossa on siis ääretön määrä odotuspaikkoja ($m = \infty$)
  - asiakas = paketti
    - λ = uusien pakettien saapumisintensiteetti (pakettia per aikayks.)
    - L = keskim. paketin pituus (datayks.)
  - palvelija = linkki, odotuspaikat = puskuri
    - C = linkin kapasiteetti (datayks. per aikayks.)
  - palveluaika = paketin lähetysaika
    - $1/\mu$ = L/C = keskim. paketin lähetysaika (aikayks.)



Liikenneprosessi



Liikennekuorma

- Tarjotun liikenteen voimakkuutta kuvataan liikennekuormalla (load).
- **Määritelmä**: **Liikennekuorma** ρ on pakettien saapumisintensiteetin λ suhde pakettien palveluintensiteettiin $\mu = C/L$:

$$\rho = \frac{\lambda}{\mu} = \frac{\lambda L}{C}$$

  - Liikennekuorma on paljas luku (kuten menetysjärjestelmän liikenneintensiteettikin)
  - Littlen kaavan nojalla: liikennekuorma kertoo keskimäärin palvelussa olevien asiakkaiden lkm:n.
  - Se voidaan myös tulkita tn:ksi, että palvelija on mielivaltaisella ajanhetkellä käytössä. Näin ollen se kertoo järjestelmän **käyttöasteen** (utilization).

Esimerkki

- Tarkastellaan reitittimen ulostulolinkkiä. Oletetaan, että
  - lähetettäviä paketteja saapuu keskimäärin 50,000 kpl sekunnissa,
  - yhden paketin keskimääräinen pituus on 1500 tavua, ja
  - linkin kapasiteetti on 1 Gbps.
- Tällöin linkin kuormaksi (ja samalla käyttöasteeksi) tulee

$$\rho = 50{,}000 * 1500 * 8 / 1{,}000{,}000{,}000 = 0.60 = 60\%$$

Viive

- Jonotusjärjestelmässä osa paketeista joutuu odottamaan lähetykseen pääsyä:
  - Saapuva paketti jää odottamaan puskuriin, jos ko. paketin saapuessa linkki on jo varattu
- Paketin **viive** (delay) reitittimen ulostulolinkillä koostuukin
  - **odotusajasta** (jonotusviive), joka riippuu systeemin tilasta paketin saapuessa, sekä
  - **lähetysajasta**, joka riippuu paketin pituudesta ja linkin kapasiteetista
- Esim.
  - paketin pituus = 1500 tavua
  - linkin kapasiteetti = 1 Gbps
  - paketin lähetysaika = 1500*8/1,000,000,000 = 0.000012 s = 12 μs

Liikenneteoreettinen analyysi (1)

- Järjestelmän kapasiteetti
  - C = linkin kapasiteetti (kbps)
- Liikenne
  - λ = pakettien saapumisintensiteetti (pakettia sekunnissa)
  - L = keskimääräinen paketin pituus. Oletetaan tässä: L = 1 kbit
- Palvelun laatu (käyttäjän näkökulmasta)
  - P_z = tn, että paketin täytyy odottaa “liian kauan”, so. kauemmin kuin annettu referenssiviive z. Oletetaan tässä: z = 0.00001 s = 10 μs
- Tarkastellaan tyyppiä **M/M/1** olevaa **puhdasta jonotusjärjestelmää**, ts. oletetaan, että
  - uudet paketit saapuvat **Poisson-prosessin** mukaisesti (intensiteetillä λ) ja
  - pakettien pituudet ovat riippumattomia ja samoin jakautuneita noudattaen **eksponenttijakaumaa** odotusarvolla L

Liikenneteoreettinen analyysi (2)

- Tällöin eri tekijöiden (järjestelmä, liikenne ja palvelun laatu) välisen yhteyden kertoo seuraava kaava:

$$P_z = \mathrm{Wait}(C, \lambda; L, z) := \begin{cases} \frac{\lambda L}{C} \exp(-(\frac{C}{L} - \lambda) z) = \rho \exp(-\mu(1-\rho) z), & \text{if } \lambda L < C \ (\rho < 1) \\ 1, & \text{if } \lambda L \geq C \ (\rho \geq 1) \end{cases}$$

- Huom:
  - Järjestelmä on **stabiili** vain tapauksessa $\rho < 1$. Muutoin odottavien pakettien jono kasvaa lopulta äärettömän pitkäksi.

Esimerkki

- Oletetaan, että paketteja saapuu intensiteetillä $\lambda = 600{,}000$ pps = 0.6 pakettia/μs ja linkin kapasiteetti on $C = 1.0$ Gbps = 1.0 kbit/μs.
- Järjestelmä on stabiili, sillä

$$\rho = \frac{\lambda L}{C} = 0.6 < 1$$

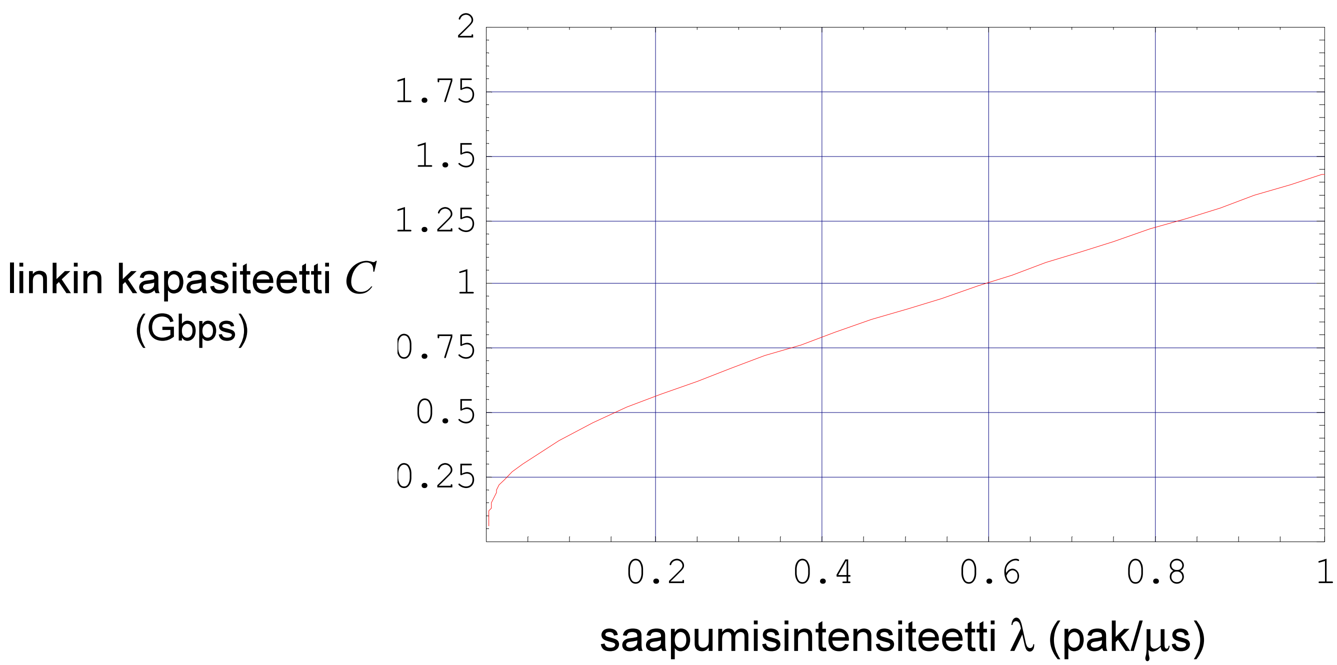
- Liian pitkän viiveen tn:ksi P_z (missä siis z = 10 μs) tulee

$$P_z = \mathrm{Wait}(1.0, 0.6; 1, 10) = 0.6 \exp(-4.0) \approx 1\%$$

Kapasiteetti saapumisintensiteetin funktiona

- Asetetaan palvelun laatuvaatimukseksi, että $P_z < 1\%$
- Tarvittava kapasiteetti C saapumisintensiteetin λ funktiona saadaan kaavalla:

$$C(\lambda) = \min\{c > \lambda L \mid \text{Wait}(c, \lambda; 1, 10) < 0.01\}$$

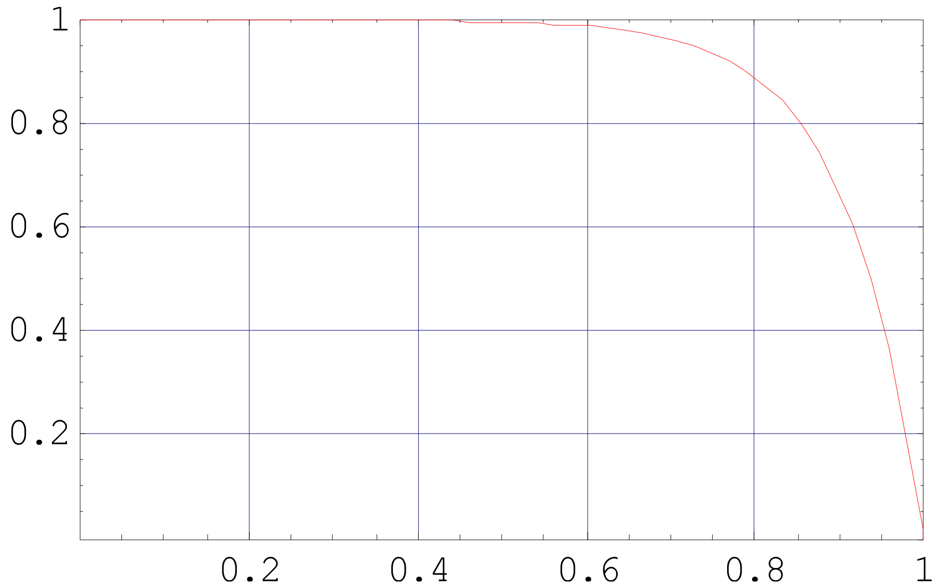


Palvelun laatu saapumisintensiteetin funktiona

- Oletetaan sitten, että linkin kapasiteetti on $C = 1.0$ Gbps $= 1.0$ kbit/μs
- Palvelun laatu $1-P_z$ saapumisintensiteetin λ funktiona saadaan kaavalla:

$$1-P_Z(\lambda) = 1-\mathrm{Wait}(1.0,\lambda;1,10)$$

palvelun laatu $1 - P_z$

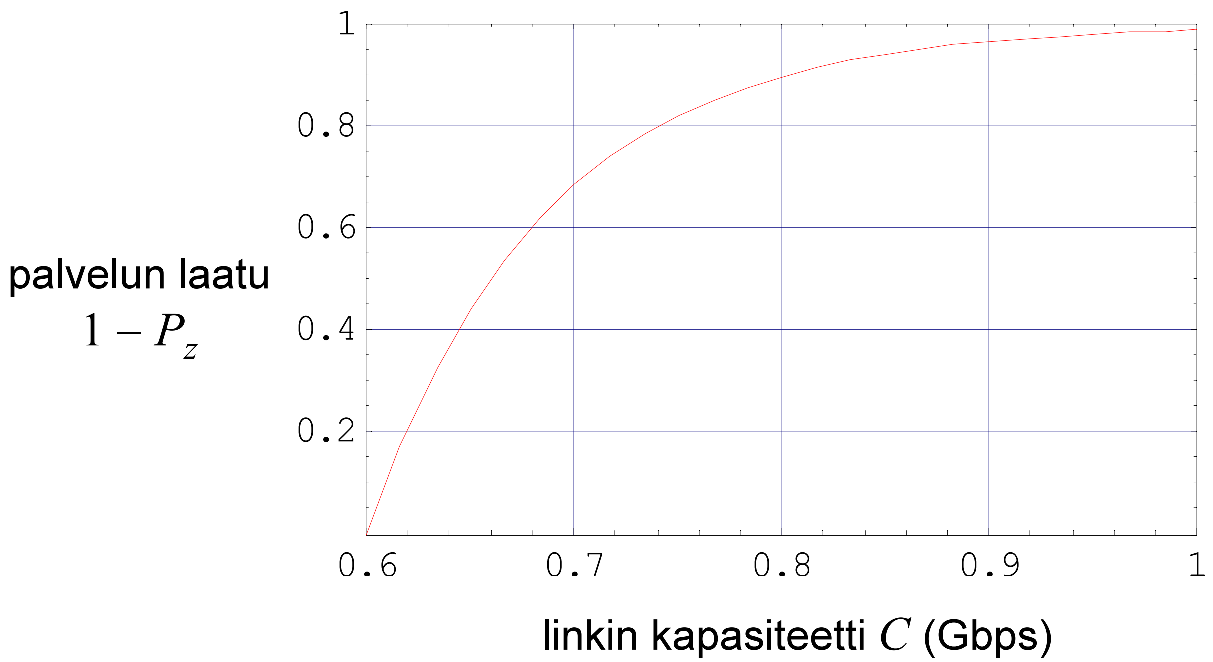


saapumisintensiteetti λ (pak/μs)

Palvelun laatu kapasiteetin funktiona

- Oletetaan lopuksi, että $\lambda = 600{,}000$ pakettia/s $= 0.6$ pakettia/μs
- Palvelun laatu $1 - P_z$ linkin kapasiteetin C funktiona saadaan kaavalla:

$$1 - P_z(C) = 1 - \mathrm{Wait}(C, 0.6; 1, 10)$$

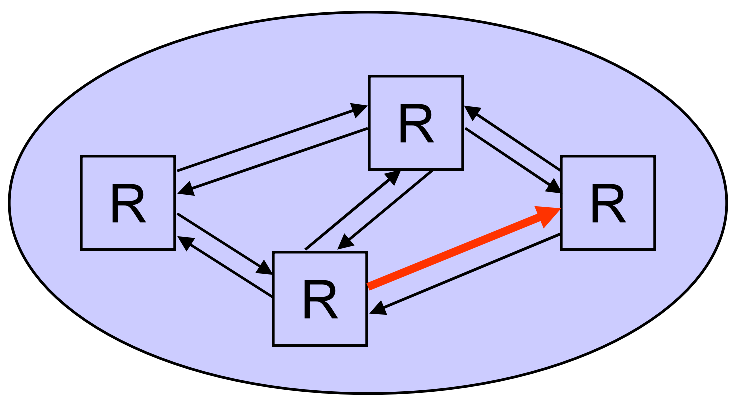


Sisältö

- Puhelinliikenteen malli
- Pakettitason malli dataliikenteelle
- Vuotason malli elastiselle dataliikenteelle
- Vuotason malli virtaavalle dataliikenteelle

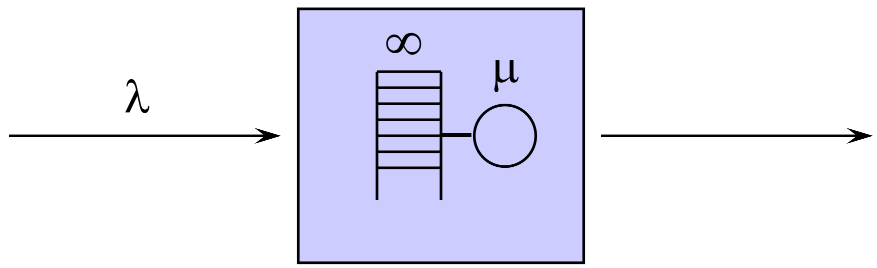
Vuotason malli elastiselle dataliikenteelle (1)

- **Jakojärjestelmät** soveltuvat **elastisen** dataliikenteen kuvaamiseen vuotasolla
  - Elastisuus tarkoittaa, että voiden lähetysnopeus sopeutuu vallitsevaan liikennetilanteeseen: ruuhka pudottaa kaikkien voiden lähetysnopeuksia
  - Tätä koulukuntaa edustaa esim. *J. Roberts* tutkijoineen (**`http://perso.rd.francetelecom.fr/roberts/`**)
- Tarkastellaan yhtä reitittimen ulostulolinkkiä
  - Liikenne koostuu linkkiä pitkin kulkevista TCP-voista, joita käytetään erilaisten digitaalisten dokumenttien (tiedostojen, www-sivujen, …) siirtoon

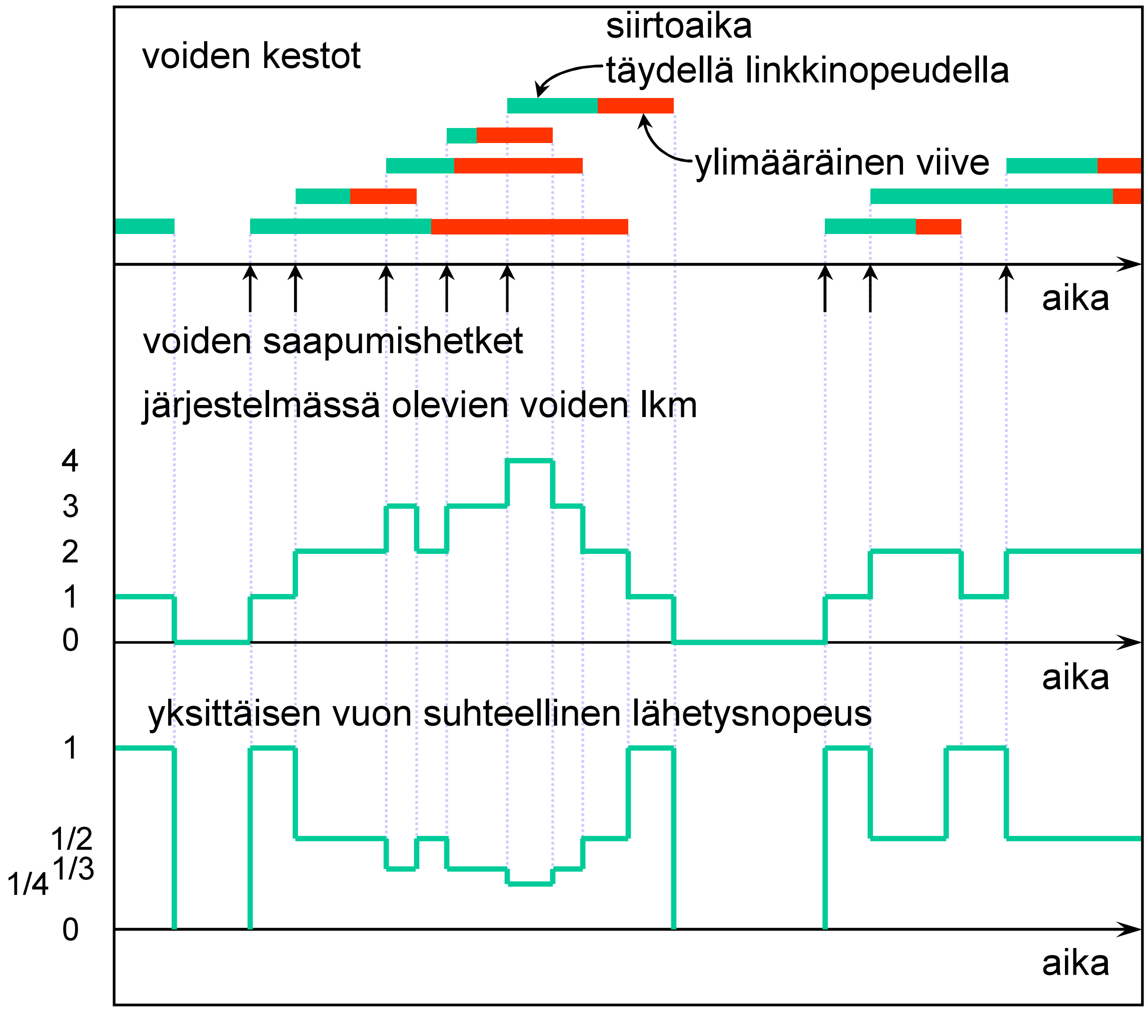


Vuotason malli elastiselle dataliikenteelle (2)

- Yksinkertaisimpana mallina on yhden palvelijan $(n = 1)$ **puhdas jakojärjestelmä**, jossa kokonaispalvelunopeus on kiinteä μ
  - asiakas = TCP-vuo = siirrettävä "tiedosto"
    - λ = uusien voiden saapumisintensiteetti (vuota per aikayks.)
    - S = keskim. vuon pituus = keskim. siirrettävän tiedoston koko (datayks.)
  - palvelija = linkki
    - C = linkin kapasiteetti (datayks. per aikayks.)
  - palveluaika = tiedoston siirtoaika täydellä linkkinopeudella
    - $1/\mu$ = S/C = keskim. tiedoston siirtoaika täydellä nopeudella (aikayks.)



Liikenneprosessi



Liikennekuorma

- Tarjotun liikenteen voimakkuutta kuvataan liikennekuormalla ρ
- **Määritelmä**: **Liikennekuorma** ρ on voiden saapumisintensiteetin λ suhde voiden kokonaispalveluintensiteettiin $\mu = C/S$:

$$\rho = \frac{\lambda}{\mu} = \frac{\lambda S}{C}$$

  - Liikennekuorma on tässäkin tapauksessa paljas luku
  - Jakojärjestelmissä liikennekuorma ei kerro keskimäärin palvelussa olevien asiakkaiden lukumäärää. Miksei?
  - Se voidaan kuitenkin edelleen tulkita tn:ksi, että palvelija on mielivaltaisella ajanhetkellä käytössä. Näin ollen se kertoo järjestelmän **käyttöasteen** (utilization).

Esimerkki

- Tarkastellaan reitittimen ulostulolinkkiä. Oletetaan, että
  - uusia voita saapuu keskimäärin 50 kpl sekunnissa,
  - yhden vuon keskimääräinen pituus on 1,500,000 tavua, ja
  - linkin kapasiteetti on 1 Gbps.
- Tällöin linkin kuormaksi (ja samalla käyttöasteeksi) tulee

$$\rho = 50 * 1{,}500{,}000 * 8 / 1{,}000{,}000{,}000 = 0.60 = 60\%$$

Läpimeno

- Jakojärjestelmässä palvelukapasiteetti jaetaan tasan kaikkien aktiivisten voiden kesken. Tästä taas seuraa, että **kaikki** vuot viivästyvät, ts. kokonaisviive ylittää pelkän lähetysajan (ellei vuo sitten satu olemaan yksinään järjestelmässä).
- **Määritelmä**: Vuon keskimääräisen koon S suhdetta sen kokemaan keskimääräiseen kokonaisviiveeseen D sanotaan **läpimenoksi** θ (throughput) eli keskimääräiseksi lähetysnopeudeksi,

$$\theta = S / D$$

- Esimerkki:
  - $S = 1$ Mbit
  - $D = 5$ s
  - $\theta = S/D = 0.2$ Mbps

Liikenneteoreettinen analyysi (1)

- Järjestelmän kapasiteetti
  - C = linkin kapasiteetti (Mbps)
- Liikenne
  - λ = voiden saapumisintensiteetti (vuota sekunnissa)
  - S = keskimäär. vuon pituus. Oletetaan tässä: S = 1 Mbit
- Palvelun laatu (käyttäjän näkökulmasta)
  - θ = vuon läpimeno eli keskimääräinen lähetysnopeus
- Tarkastellaan tyyppiä **M/G/1-PS** olevaa **jakojärjestelmää**, ts. oletetaan, että
  - uudet vuot saapuvat **Poisson-prosessin** mukaisesti (intensiteetillä λ) ja
  - voiden pituudet ovat riippumattomia ja samoin jakautuneita noudattaen **mitä tahansa jakaumaa**, jonka odotusarvo on S

Liikenneteoreettinen analyysi (2)

- Tällöin eri tekijöiden (järjestelmä, liikenne ja palvelun laatu) välisen yhteyden kertoo seuraava kaava:

$$\theta = \mathrm{Xput}(C,\lambda;S) := \begin{cases} C - \lambda S = C(1-\rho), & \text{if } \lambda S < C \ (\rho < 1) \\ 0, & \text{if } \lambda S \geq C \ (\rho \geq 1) \end{cases}$$

  - Tulkinta: Jokaisen vuon kokema läpäisy vastaa systeemin ”vapaana olevaa kapasiteettia” $C(1 - \rho)$.

- Huom:
  - Järjestelmä on **stabiili** vain tapauksessa $\rho < 1$. Muutoin voiden lukumäärä ja keskimääräinen läpimenoaika kasvaa rajatta, ja vuon kokema läpimeno lähestyy nollaa.

Esimerkki

- Oletetaan, että voita saapuu intensiteetillä $\lambda = 600$ vuota sekunnissa ja linkin kapasiteetti on $C = 1000$ Mbps $= 1.0$ Gbps.
- Järjestelmä on stabiili, sillä

$$\rho = \frac{\lambda S}{C} = \frac{600}{1000} = 0.6 < 1$$

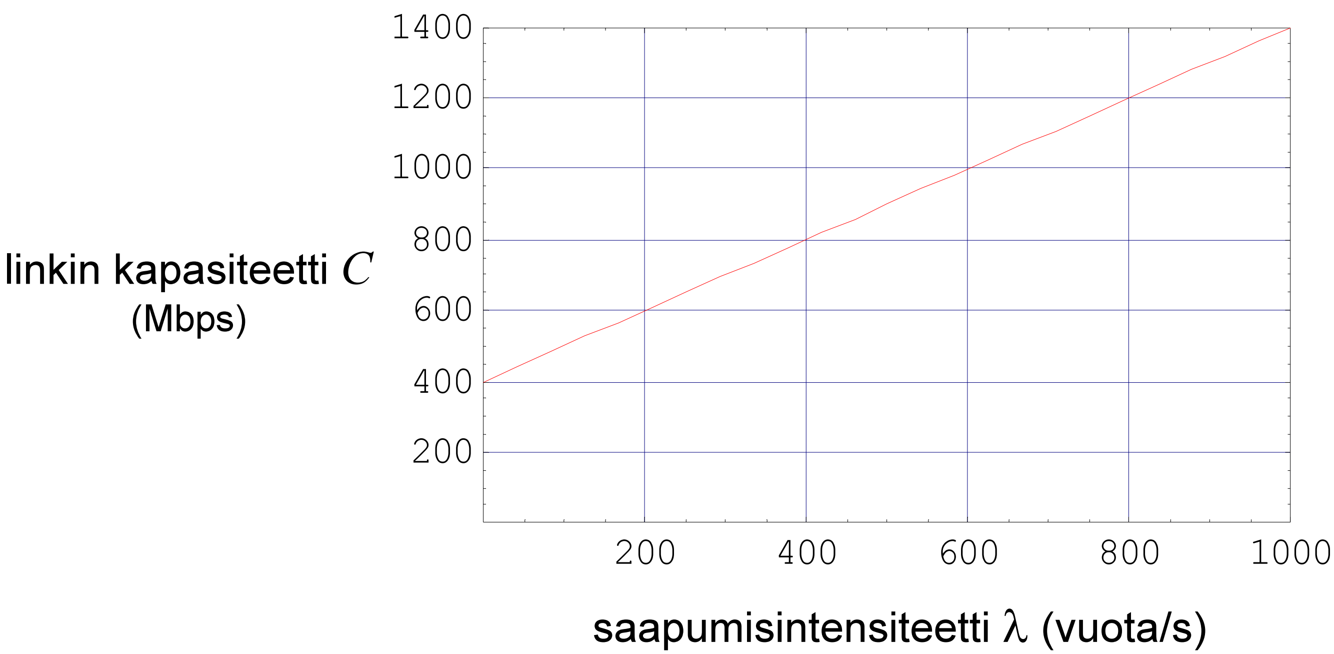
- Läpimenoksi tulee

$$\theta = \mathrm{Xput}(1000{,}600;1) = 1000 - 600 = 400\ \mathrm{Mbps} = 0.4\ \mathrm{Gbps}$$

Kapasiteetti saapumisintensiteetin funktiona

- Asetetaan palvelun laatuvaatimukseksi, että $\theta \geq 400$ Mbps.
- Tarvittava kapasiteetti C saapumisintensiteetin λ funktiona saadaan kaavalla:

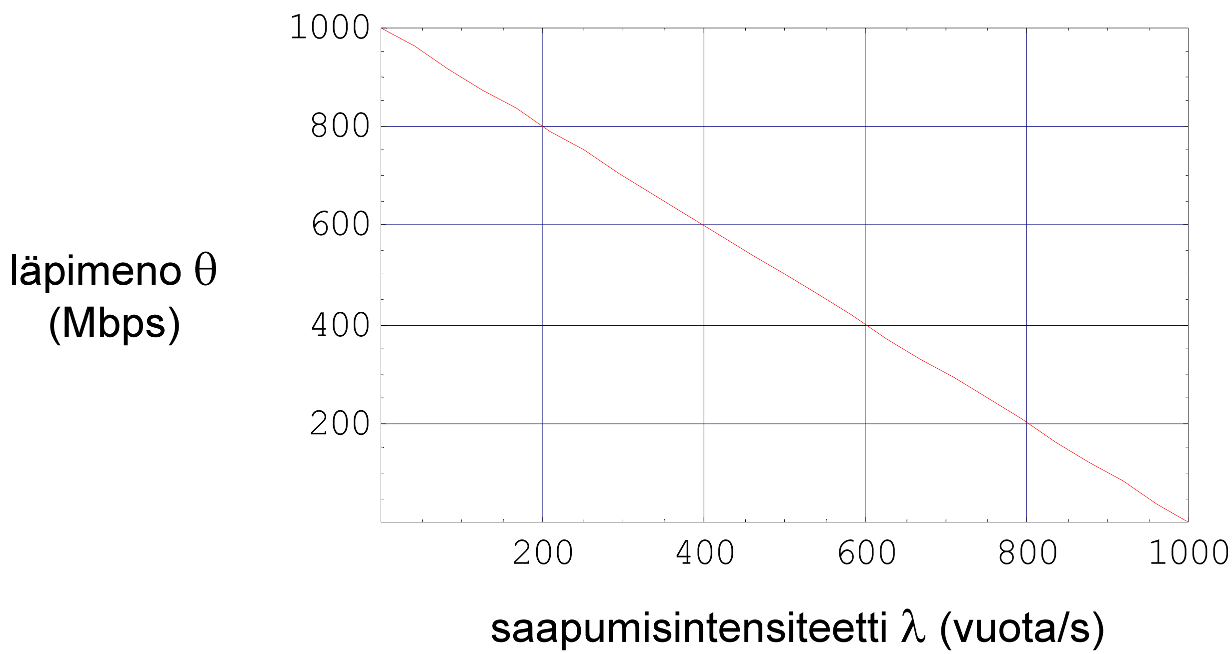
$$C(\lambda) = \min\{c > \lambda S \mid \mathrm{Xput}(c, \lambda; 1) \geq 400\} = \lambda S + 400$$



Palvelun laatu saapumisintensiteetin funktiona

- Oletetaan sitten, että linkin kapasiteetti on $C = 1000$ Mbps
- Palvelun laatu θ saapumisintensiteetin λ funktiona saadaan kaavalla:

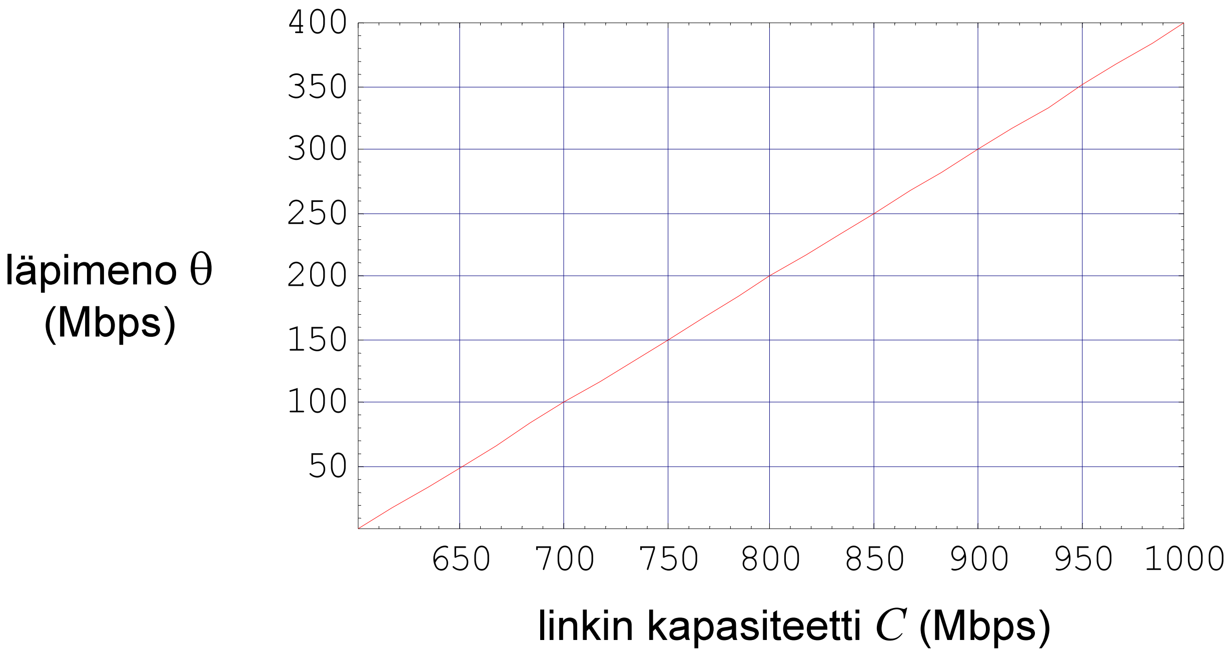
$$\theta(\lambda) = \mathrm{Xput}(1000, \lambda; 1) = 1000 - \lambda S, \quad \lambda < 1000/\mathrm{S}$$



Palvelun laatu kapasiteetin funktiona

- Oletetaan lopuksi, että saapumisintensiteetti on $\lambda = 600$ vuota/s
- Palvelun laatu θ linkin kapasiteetin C funktiona saadaan kaavalla:

$$\theta(C) = \mathrm{Xput}(C{,}600;1) = C - 600S, \quad C > 600S$$

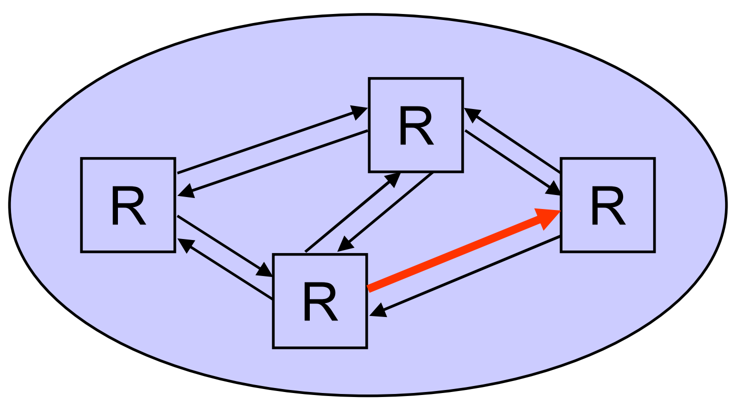


Sisältö

- Puhelinliikenteen malli
- Pakettitason malli dataliikenteelle
- Vuotason malli elastiselle dataliikenteelle
- Vuotason malli virtaavalle dataliikenteelle

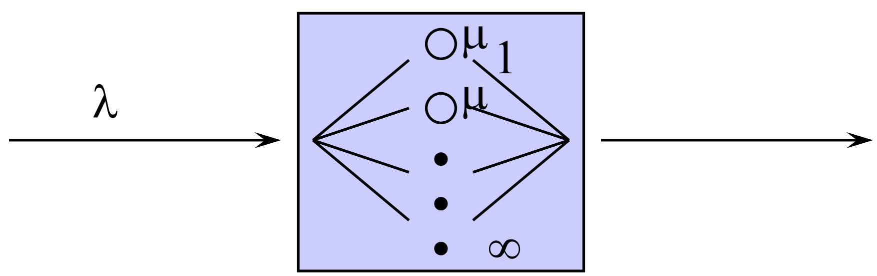
Vuotason malli virtaavalle CBR-liikenteelle (1)

- **Ääretön järjestelmä** soveltuu **virtaavan vakionopeuksisen** dataliikenteen kuvaamiseen vuotasolla
  - Virtaavan vuon lähetysnopeus ei reagoi verkon tilaan, eikä verkon tila myöskään vaikuta vuon kestoon
  - Tällaisia malleja sovellettiin 90-luvulla ATM-verkkojen CBR-liikenteen liikenneteoreettiseen analyysiin
- Tarkastellaan yhtä reitittimen ulostulolinkkiä
  - Liikenne koostuu linkkiä pitkin kulkevista UDP-voista, joita käytetään virtaavan vakionopeuksisen liikenteen (esim. VoIP) siirtoon

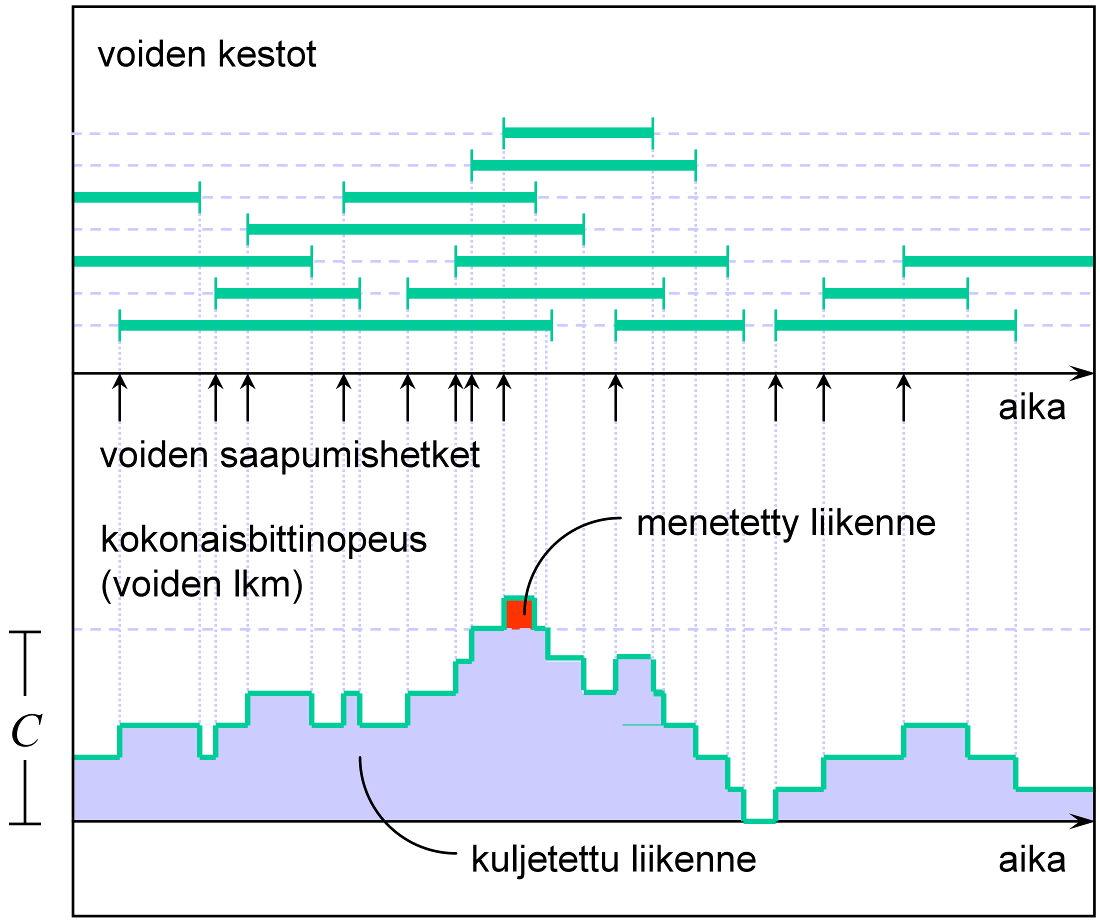


Vuotason malli virtaavalle CBR-liikenteelle (2)

- Mallina on siis **ääretön järjestelmä** ($n = \infty$)
  - asiakas = UDP-vuo = vakionopeuksinen bittivirta
    - λ = uusien voiden saapumisintensiteetti (vuota per aikayks.)
  - palveluaika = vuon kesto
    - h = $1/\mu$ = keskimääräinen vuon kesto (aikayks.)
- **Puskuriton** vuotason malli:
  - kun voiden yhteinen lähetysnopeus ylittää linkin nopeuden, bittejä katoaa (tasaisesti kaikilta voilta)



Liikenneprosessi



Tarjottu liikenne

- Merkitään r:llä yksittäisen vuon bittinopeutta
- Tarjotun liikenteen voimakkuutta kuvaa keskimääräinen voiden yhteenlaskettu bittinopeus R
  - Littlen kaavan nojalla keskimääräinen aktiivisten voiden lkm on

$$a = \lambda h$$

  - Tätä voidaan kutsua **liikenneintensiteetiksi** (vrt. puhelinliikenne)
  - Tästä seuraa, että

$$R = ar = \lambda hr$$

Häviösuhde

- Merkitään N:llä systeemissä olevien voiden lukumäärää
- Aina kun voiden yhteinen lähetysnopeus Nr ylittää linkin kapasiteetin C, bittejä katoaa nopeudella

$$Nr - C$$

- Keskimääräinen katoamisnopeus on siis

$$E[(Nr - C)^{+}] = E[\max\{Nr - C, 0\}]$$

- **Määritelmä**: **Häviösuhde** p_{loss} kertoo kadonneen liikenteen osuuden koko liikenteestä:

$$p_{\mathrm{loss}} = \frac{E[(Nr-C)^{+}]}{E[Nr]} = \frac{1}{ar} E[(Nr - C)^{+}]$$

Liikenneteoreettinen analyysi (1)

- Järjestelmän kapasiteetti
  - $C = nr$ = linkin kapasiteetti (kbps)
- Liikenne
  - $R = ar$ = tarjottu liikenne (kbps)
  - r = vuon bittinopeus (kbps).
- Palvelun laatu (käyttäjän näkökulmasta)
  - p_{loss} = häviösuhde
- Tarkastellaan tyyppiä **M/G/∞** olevaa **ääretöntä järjestelmää**, ts. oletetaan, että
  - uudet vuot saapuvat **Poisson-prosessin** mukaisesti (intensiteetillä λ) ja
  - voiden kestot ovat riippumattomia ja samoin jakautuneita noudattaen **mitä tahansa jakaumaa**, jonka odotusarvo on h

Liikenneteoreettinen analyysi (2)

- Tällöin eri tekijöiden (järjestelmä, liikenne ja palvelun laatu) välisen yhteyden kertoo seuraava kaava:

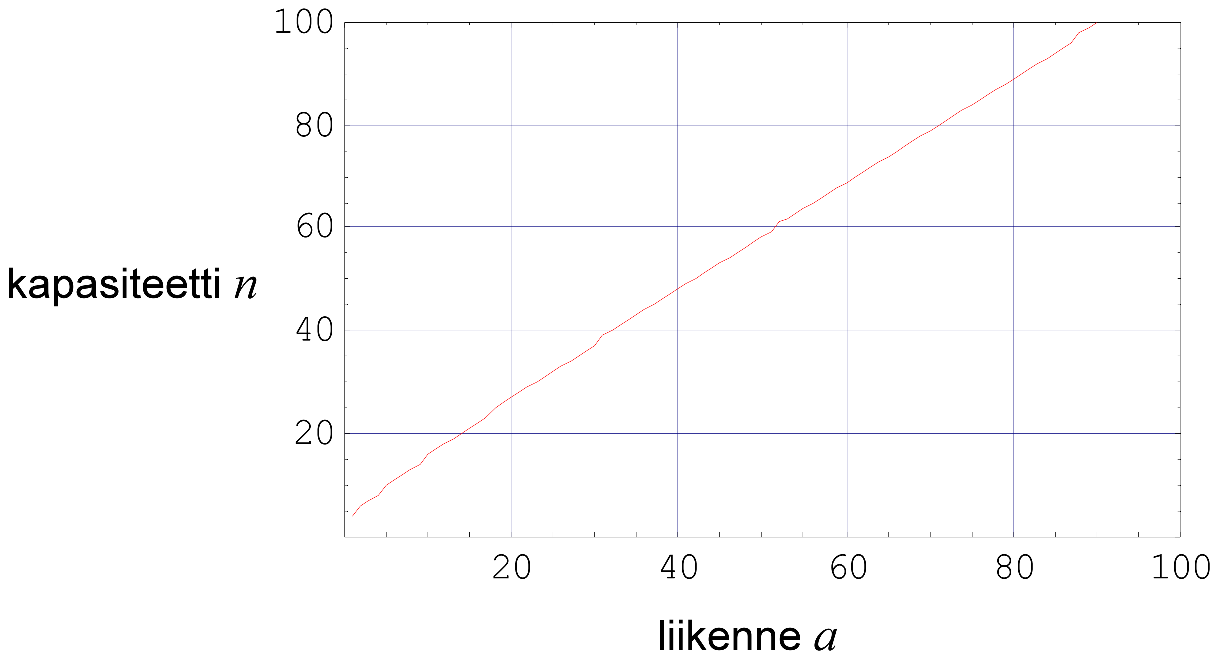
$$p_{\text{loss}} = \mathrm{LR}(n,a) := \frac{1}{a} \sum_{i=n+1}^{\infty} (i-n) \frac{a^i}{i!} e^{-a}$$

- Esimerkki:
  - $n = 20$
  - $a = 14.36$
  - $p_{\text{loss}} = 0.01$

Kapasiteetti liikenteen funktiona

- Asetetaan palvelun laatuvaatimukseksi, että häviösuhde $p_{\text{loss}} < 1\%$
- Tarvittava kapasiteetti n liikenteen a funktiona saadaan kaavalla:

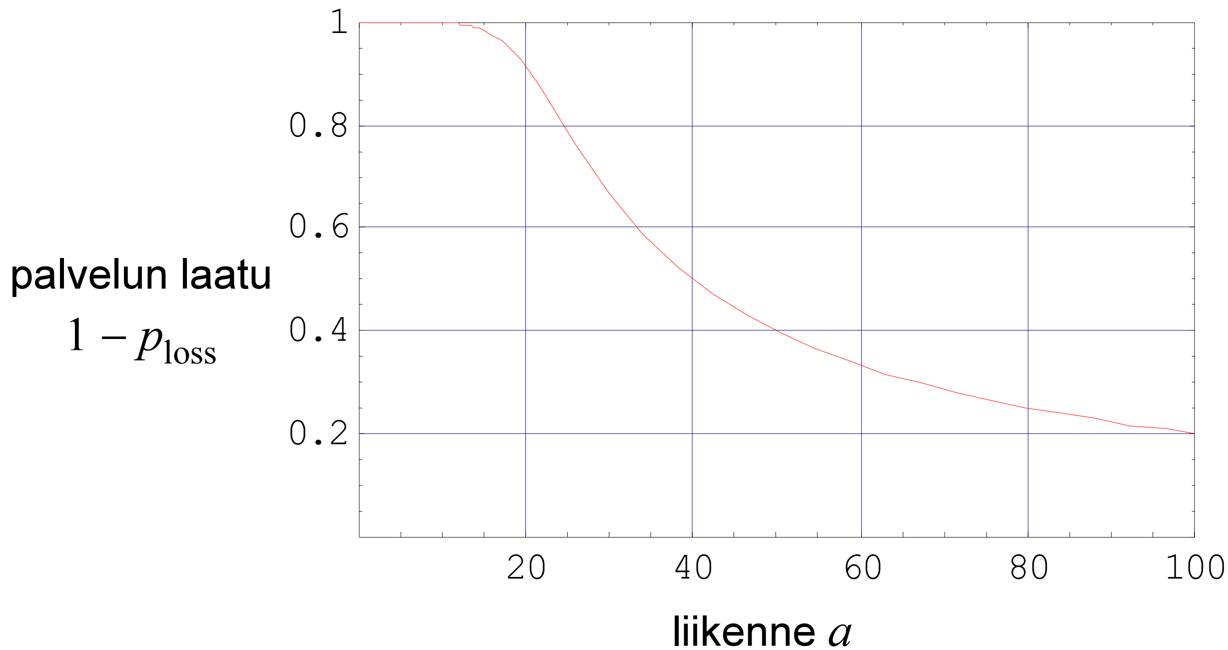
$$n(a) = \min\{i = 1,2,\ldots \mid \mathrm{LR}(i,a) < 0.01\}$$



Palvelun laatu liikenteen funktiona

- Oletetaan sitten, että kapasiteetti $n = 20$
- Palvelun laatu $1 - p_{\text{loss}}$ liikenteen a funktiona saadaan kaavalla:

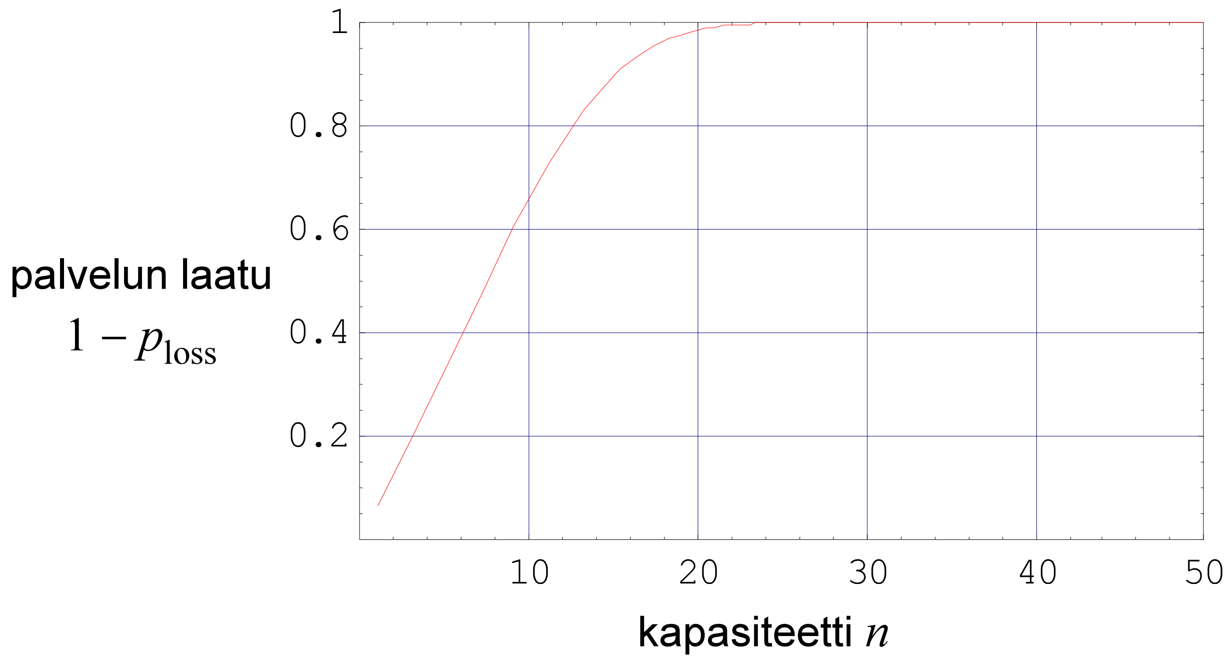
$$1 - p_{\text{loss}}(a) = 1 - \text{LR}(20, a)$$



Palvelun laatu kapasiteetin funktiona

- Oletetaan lopuksi, että tarjotun liikenteen intensiteetti $a = 15.0$ erlang
- Palvelun laatu $1 - p_{\text{loss}}$ kapasiteetin n funktiona saadaan kaavalla:

$$1 - p_{\text{loss}}(n) = 1 - \text{LR}(n, 15.0)$$



THE END