

Tietoliikenne II (2 ov)

Syksy 2002

Liisa Marttinen

- Kurssikirja: **Kurose & Ross, Computer Networking** (2. edition)
 - ★ (kyllä 1. painoskin kelpaa, mutta siitä puuttuu mm. mobiiliverkot kokonaan)
- Lisämateriaalia: Aiheeseen liittyvät RFC:t

Tietoliikenne II

Täydennystä Tietoliikenne I -kurssin asioihin

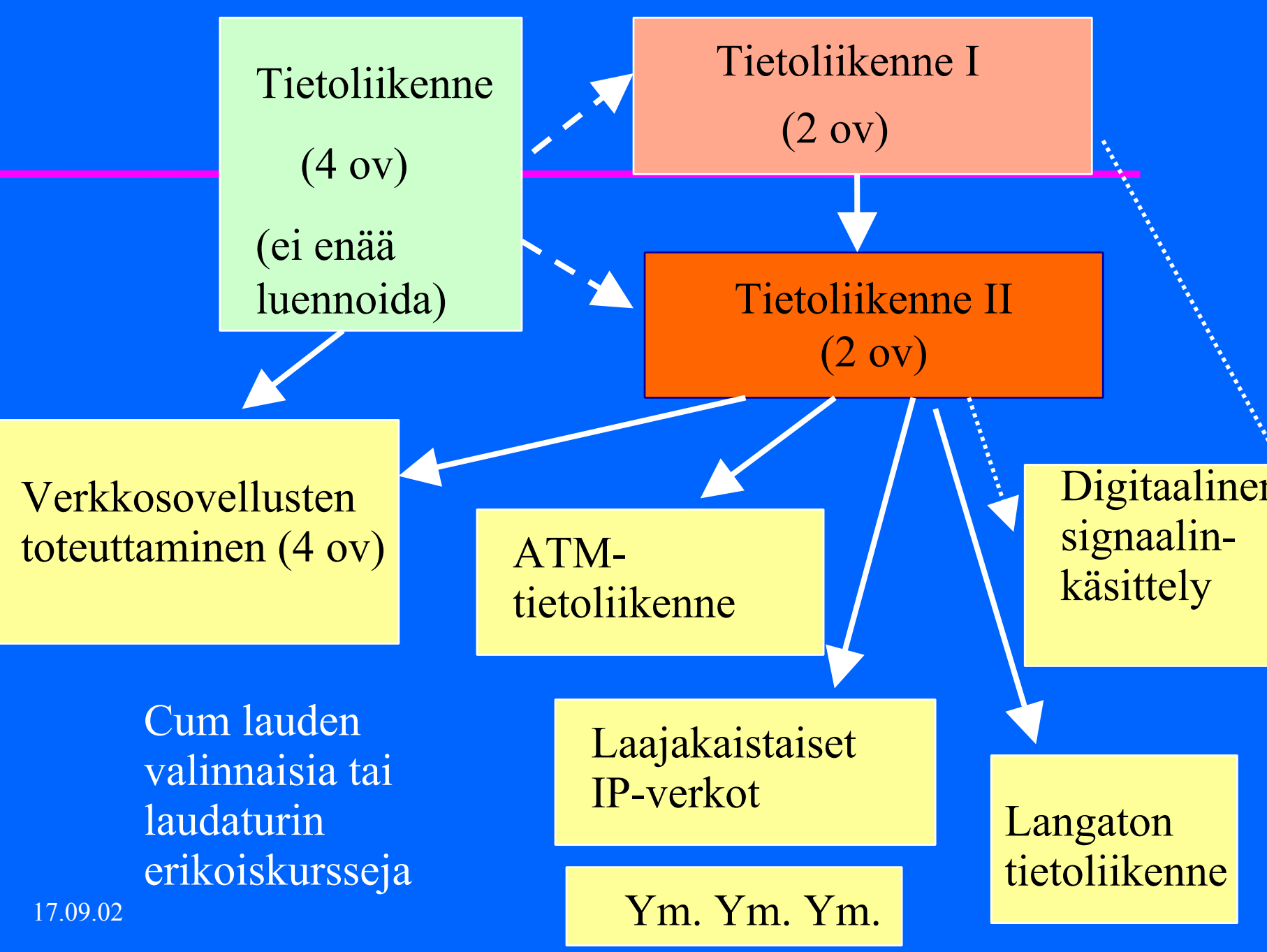
- perusteellisemmin
- laajemmin
- ‘teoreettisemmin’
- ◆ perus-, linkki- ja MAC-kerros
- ◆ reititys, IPv6
- ◆ TCP: suorituskyky ja uudet piirteet
- ◆ DNS, ..

Alustava sisällysluettelo

- ◆ 1. TCP:n suorituskyky
 - optiot
 - uudet piirteet ruuhkanvalvonnassa
- ◆ 2. IPv6, IPsec?
- ◆ ICMP
- ◆ Reititys
 - » OSPF, BGP, monilähetysreititys, mobiilireititys
- ◆ 3. muita verkkoja: atm, FDDI, ...

Sisällysluettelo jatkuu

- ◆ 4. Linkkikerroksen ja peruskerroksen asioita
 - » HDLC, PPP, SONET, ..
 - » Tiedonsiirron teoreettinen perusta (Shannon, Nyquist, ..)
- ◆ 5. Internetin palvelun laatu (QoS)
 - » integroidut palvelut
 - » eriytyneet palvelut
- ◆ 6. Turvallisuus (??)
- 7. Verkonhallinta (???)



Tietoliikenne

(4 ov)

(ei enää luennoida)

Tietoliikenne I

(2 ov)

Tietoliikenne II

(2 ov)

Verkkosovellusten toteuttaminen (4 ov)

ATM-tietoliikenne

Digitaalinen signaalinkäsittely

Cum lauden valinnaisia tai laudaturin erikoiskursseja

Laajakaistaiset IP-verkot

Ym. Ym. Ym.

Langaton tietoliikenne

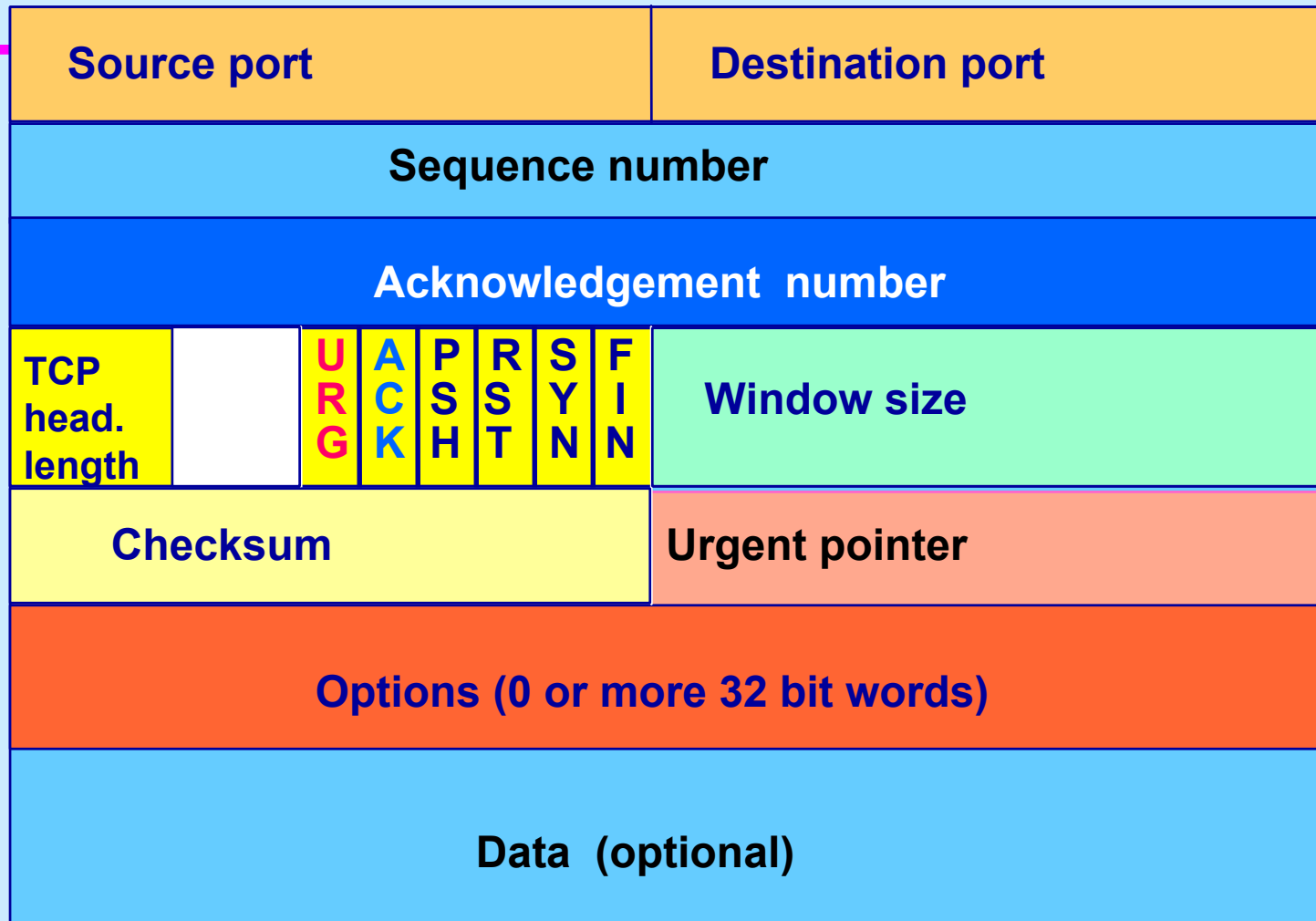
Suoritus

- ◆ kurssikoe maks. 50 p, min 25 p
 - ma 17.12. klo 9-13 sali 1 päärakennus ???
- ◆ kurssiaktiivisuus maks. 20 p
 - traditionaaliset harjoitukset => maks. 10 p
 - miniesseet (1-10 sivua) ja -esitelmät (10-15 min), keskusteluaktiivisuus yms. => maks. 10 p
- ◆ 30 p => 1-, 51 => 3
- ◆ tai sitten loppukokeella maks. 60 p
 - tammi- tai helmikuussa 2003

1. TCP ja suorituskyky

- ◆ TCP:n peruspiirteiden toiminta tarkemmin
 - osin harjoitustehtävissä
- ◆ TCP:n lisäpiirteitä
 - Ikkunaskaalaus (Window scaling)
 - Aikaleimaus (time stamping)
 - SACK (Selective Acknowledgement)
 - RED (Random Early Detection)
 - ECN (Explicit Congestion Notification)

TCP-otsakkeen kentät



TCP-optiot

◆ Optio-kenttä

- erilaisia valinnaisia piirteitä varten
- Option pituus on 40 tavua
 - » TCP header length -kenttä = 4 bittiä kertoo otsakkeen pituuden 32 bitin sanoina => $15 * 4$ tavua = 60 tavua
 - » 60 tavua - 20 tavua vakio-otsaketta => enintään 40 tavua optioita varten

Option tyyppi	Option pituus	Option merkitys
1 tavu	1 tavu	pituus - 2 tavua

Optioita:

◆ **MSS (Maximum Segment Size)**

– käytetään ilmoittamaan vastaanottajan yhteydellä hyväksymä suurin segmentin koko

» eri suuntiin voi olla eri koko

» voi olla suurempi tai pienempi kuin oletus MSS

◆ **MTU (Maximum Transfer Unit)** = suurin yhdessä verkon kehyksessä kulkeva datamäärä

– Eri verkoissa eri kokoja, mutta minimi MTU= 576 B

◆ $MSS = MTU - IP\text{-otsake} - TCP\text{-otsake}$

– oletus $MSS = 576 - 20 - 20 = 536$

– Otsakkeet voivat olla suurempia!

◆ MSS ilmoitetaan yhteyttä muodostettaessa eli SYN-segmenteissä

– kumpikin osapuoli voi ilmoittaa oman MSS-arvonsa

» jos ei ilmoita, niin suostuu vastaanottamaan minkä tahansa kokoisia segmenttejä.

2	4	maksimi segmentin koko
---	---	------------------------

Muita ehdotettuja optioita (RFC 1323):

- ◆ ikkunaskaalaus (window scaling factor)
 - kasvattaa TCP-otsakkeen 16-bitin ikkunan koon 32-bitin ikkunan kooksi



- ◆ aikaleimaus (timestamp)
 - segmentin aikaleima palautetaan kuittauksessa



Erilaisia suorituskykyongelmia

- ◆ TCP-protokolla käytössä hyvin erilaisissa ympäristöissä

- » pitkän viipeen satelliittiyhteyksillä
- » erittäin nopeilla yhteyksillä
- » langattomilla yhteyksillä

- ◆ => suorituskykyongelmia

- otsakkeen kentät liian pieniä

- » **ikkunankoko 16 bittiä => 65536 tavua**

- ◆ rajoittaa lähetysnopeutta mm.satelliittiyhteyksillä

- » **järjestysnumero 32 bittiä**

- ◆ rajoittaa lähetysnopeutta erittäin nopeilla yhteyksillä

Kaistan ja kiertoviiveen tulo (bandwidth * delay product)

TCP:n suorituskyky

- Riippuu siirtonopeudesta (kaista, bandwidth) ja
- kiertoviiveestä (RTD, round-trip delay, 1 ms - 100 s)

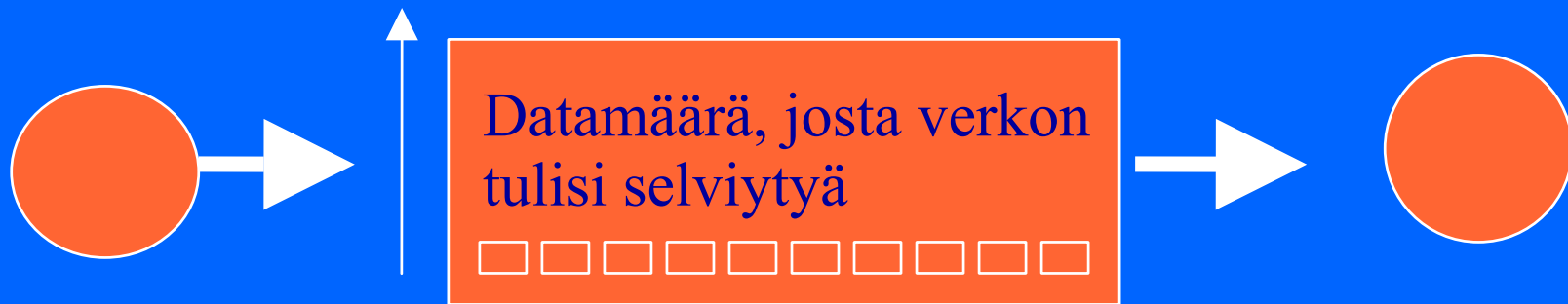
Tulo siirtonopeus * kiertoviive kertoo sen datamäärän, joka TCP:n täytyy pystyä käsittelemään, jotta lähettäjä ja vastaanottaja voisivat toimia täydellä vauhdilla

- paljonko kuittaamatonta dataa verkon täytyy pystyä välittämään

◆ Ongelmia syntyy, jos tulo on hyvin suuri!

Ideaalitulanteessa 'lähetysputki' on koko ajan täynnä!

Lähetysnopeus (bps)



RTD (s) →

aika lähetyksen alusta
kuittauksen saapumiseen

Ongelmia aiheuttavat LFN-verkot (long, fat pipe network):
pitkä viive ja suuri lähetysnopeus, esim. satelliittiyhteydet
ja nopeat runkolinjat => tulo > 1 Mb ~ 100 segmenttiä

Ongelmia LFN-verkoissa

- **Ikkunan koko -kenttä liian pieni** => ‘putki’ ei täyty (suurin mahdollinen ikkuna $2^{16} = 65\text{ KB}$)
 - ikkunan skaalaus -optio
- **pakettien katoaminen** => hidas aloitus eli ‘putki’ joudutaan tyjentämään
 - parannukset ruuhkanhallintakäytäntöön
 - » Fast Retransmit, Fast Recovery (*yksi paketti katoaa*)
 - » SACK (*mitkä saapuneet, mitkä kadonneet*)
- **Uudelleenlähetyssajastimen tarkempi ajastus**
 - » Timestamp -optio

Lisää ongelmia:

- Järjestysnumeroiden uudelleenkäyttö
 - Samannumeroinen segmentti voi yhä olla verkossa => se hyväksytään 'uutena'
 - Samannumeroinen ACK-kuittaus voi yhä olla verkossa => lukkiutunut tilanne, josta toivutaan vain RST:llä
 - » Lähettäjä saa kuittauksen ja siirtyy seuraavaan segmenttiin
 - » Vastaanottaja jää odottamaan puuttuvia tavuja ja hylkää muut
 - » Ikkkuna täyttyy ja lähettäjä alkaa lähettää uudelleen, mutta

MSL (Maximum Segment Lifetime)

- MSL = 2 minuuttia
 - IP-kerroksen elinaikakenttä (time-to-live) rajoittaa
- Järjestysnumerokenttä on 32 bittiä ($\sim 4.3 \cdot 10^9$)
- Esimerkkejä eri verkkojen numeroiden kiertoajasta
 - ARPANET 56 kbps 7 KBps ~ 3.6 päivää
 - Ethernet 10 Mbps 1.25 MBps ~ 30 min
 - FDDI 100 Mbps 12.5 MBps ~ 3 min
 - **Gigabit 1 Gbps 125 MBps 17 sec**
- **Nopeissa verkoissa aiheuttaa ongelmia!**

Ikkunan skaalaus (Window scale factor)

- ◆ ikkunakoko = 16 bittiä \Rightarrow 65536 tavua
 - » kertoo vastaanottajan ikkunan = kuinka monta tavua voi lähettää ennenkuin täytyy jäädä odottamaan kuittausta
 - » Jos RTT (Round-trip-time) on suuri, niin joudutaan odottelemaan
 - » Efektiivinen nopeus $B = 2^{16}/RTT$
- ◆ jos käytössä ikkunan skaalaus -optio, ikkunakentän arvo kerrotaan 2^F , jossa F on skaalausoption arvo.
 - » Suurin F:n arvo on 14.

Miksi uudelleenlähetyssajastimen arvo on tärkeä!

- ◆ Ruuhkan oikea havaitseminen riippuu uudelleenlähetyssajastimen ‘oikeasta’ arvosta.
 - Liian suuri arvo => alkavaa ruuhkaa ei huomata ajoissa => verkkoa ylikuormitetaan => syntyy ruuhkatilanne => resurssien hukkakäyttöä
 - Liian pieni arvo => luullaan ruuhkaksi, vaikka ei olekaan => hidastetaan turhaan lähetystä => resurssien hukkakäyttöä

Uudelleenlähetyksajastimen arvo

- ◆ mitataan paketin kiertoviive M ja viiveen poikkeama odotetusta eli $|RTT-M|$
 - $RTT = aRTT + (1-a)M$
 - $D = bD + (1-b)|RTT-M|$
 - ajastimen arvo = $RTT + 4D$
- ◆ Ongelmia aiheuttavat uudelleenlähetykset
 - ◆ Mikä sanomista kuitataan?
 - ◆ Karn: uudelleenlähetyksiä ei oteta mukaan, vaan ajastimen arvo kaksinkertaistetaan aina uudelleenlähetyksessä, kunnes saadaan onnistuneesti kuittaus.
- ◆ Useat toteutukset mittaavat vain yhden paketin ikkunasta!
 - ◆ Ongelmia, jos ikkuna suuri.

Aikaleima (timestamp)

- ◆ Kaksi eri optiota
 - Timestamp Value
 - ◆ lähtevissä segmenteissä,
 - Timestamp Echo Reply
 - ◆ kuittauksessa
 - ◆ sama kuin kuitatun segmentin Timestamp-arvo
- ◆ Voidaan käyttää missä tahansa datasegmentissä
- ◆ \Rightarrow Voidaan laskea kiertoviive jokaiselle segmentille, myös uudelleenlähetyksille.

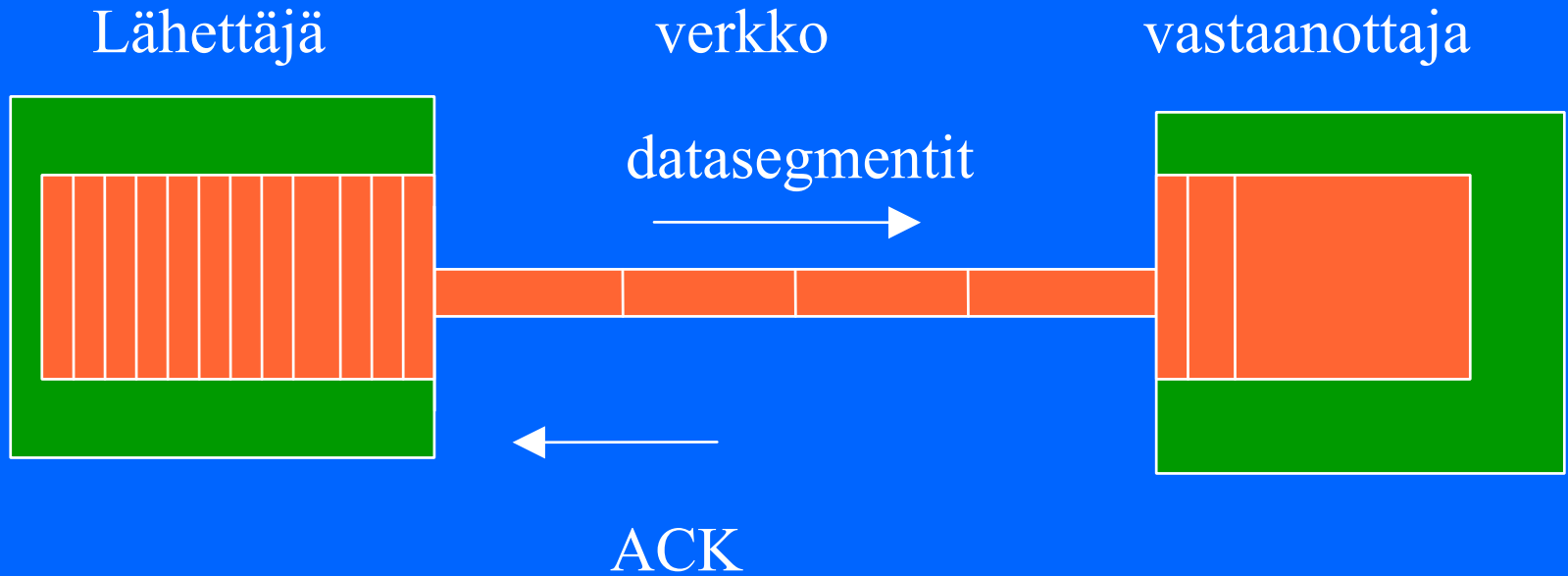
RTTM (Round-Trip Time Measurement)

- ◆ lähetettävään sanomaan liitetään aikaleima-
optioon aikaleima
 - ◆ aikaleimakello, joka tikittää riittävän nopeasti
- ◆ sama aikaleima palautetaan sanoman
kuittauksessa
- ◆ ongelmatilanteita:
 - viivästyneet kuittaukset: aikaisin kuittaamaton
 - ◆ TCP:n ei tarvitse kuitata jokaista segmenttiä
 - puuttuva segmentti: viimeisin hyväksyty
 - puuttuvan segmentin saapuminen: viimeisin
puuttuva

Viivästetty ACK (Delayed ACK)

- Ei tarvitse välttämättä kuitata jokaista segmenttiä
 - kuitenkin kuitattava ainakin joka toinen ja viive saa olla korkeintaan 500 ms,
 - ◆ usein noin 200ms
- Hyöty: kuittaus kulkee datan mukana
 - samalla kertaa ikkunan muotos, kuittaus ja kaiutus
- Haitta: kiertoviiveen laskeminen, pakettien kellotus

TCP-ruuhkanvalvonta



toimii lähetyksen tahdistajana

putkesta poistunut dataa, joten voidaan lähettää sama määrä lisää

TCP self-clocking

- ◆ TCP tahdistaa itse oman lähetyksensä ACK:ien avulla
 - nopeutta voi rajoittaa
 - » verkko
 - ◆ ruuhkan takia syytä vielä pienentää lähetyksnopeutta
 - » vastaanottaja
 - ◆ lähetyksnopeus ok
 - lähettäjä ei voi tietää kumpi

Ruuhkanvalvonta on hankalaa!

- ◆ Sitä varten on koko ajan kehitetty yhä parempia menetelmiä
 - uudelleenlähetysajastimen arvo
 - » RTT:n varianssin arviointi
 - » Karnin algoritmi
 - » exponential retransmission timer backoff
 - lähetysikkunan hallinta
 - » slow start
 - » congestion avoidance
 - » fast retransmit
 - » fast recovery

Lähetettynä voi olla vain rajallinen määrä kuittaamatonta dataa ('Flight size')

- ◆ vastaanottoikkuna (receiver window, **rwnd**)
 - vastaanottaja ilmoittaa lähettämiensä segmenttien ikkunakentässä
 - vastaanottaja voi vapaasti kasvattaa tai pienentää
 - vuonvalvontaa varten
- ◆ ruuhkaikkuna (congestion window, **cwnd**)
 - lähettäjä saa korkeintaan lähettää verkkoon, jotta verkko ei tukkeutuisi
 - ruuhkanhallintaa varten
- ◆ **$\min(\text{rwnd}, \text{cwnd})$** rajoittaa lähettämistä

Ruuhkaikkunan arvo eri tilanteissa

– initial window (IW)

» ruuhkaikkunan arvo heti kolminkertaisen kättelyn jälkeen

◆ korkeintaan kaksi segmenttiä tai 2^* suurin määrä tavuja, jonka lähettäjä voi kerralla lähettää (SMSS)

– loss window (LW)

» ikkunan arvo, kun TCP on havainnut, uudelleenlähetyksajastimen lauettua, segmentin kadonneeksi

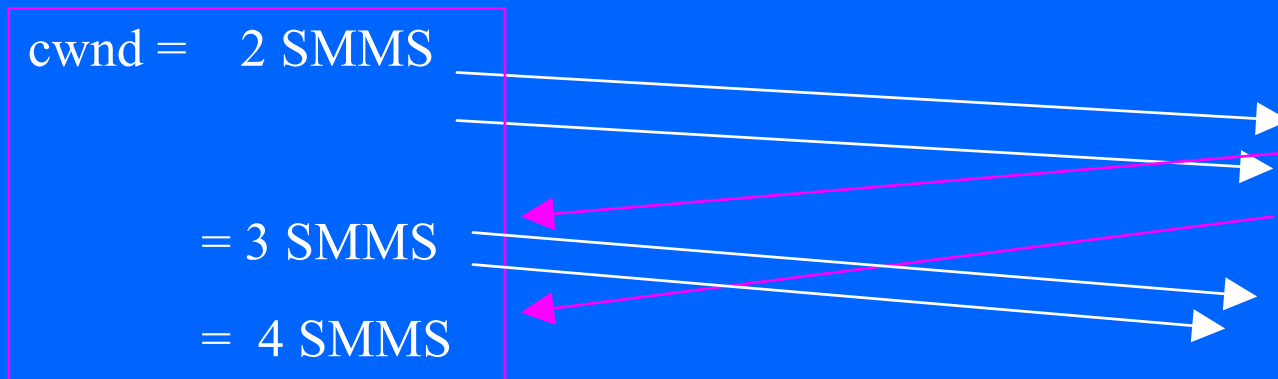
– restart window (RW)

» kun lähetys käynnistetään uudelleen joutilaana olon jälkeen

Slow start

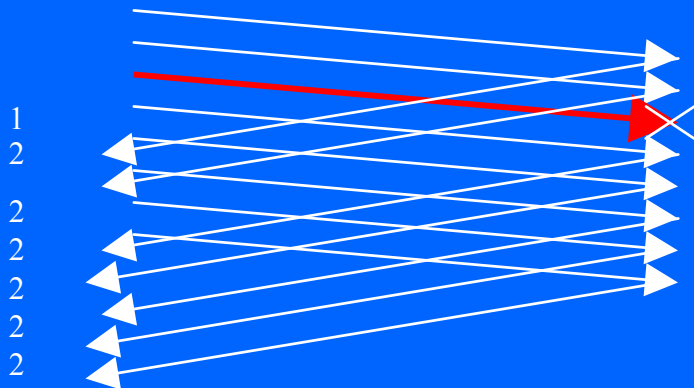
◆ Hitaan aloituksen aikana

- Ruuhkaikkunaa $cwnd$ kasvatetaan korkeintaan maksimilähetysmäärällä (SMSS) jokaista uutta dataa kuittaavaa ACKia kohden



Hidas aloitus

- ◆ Aina yhteyden alussa
- ◆ kun kuittausta ei tule ajoissa (paketti kadonnut!)



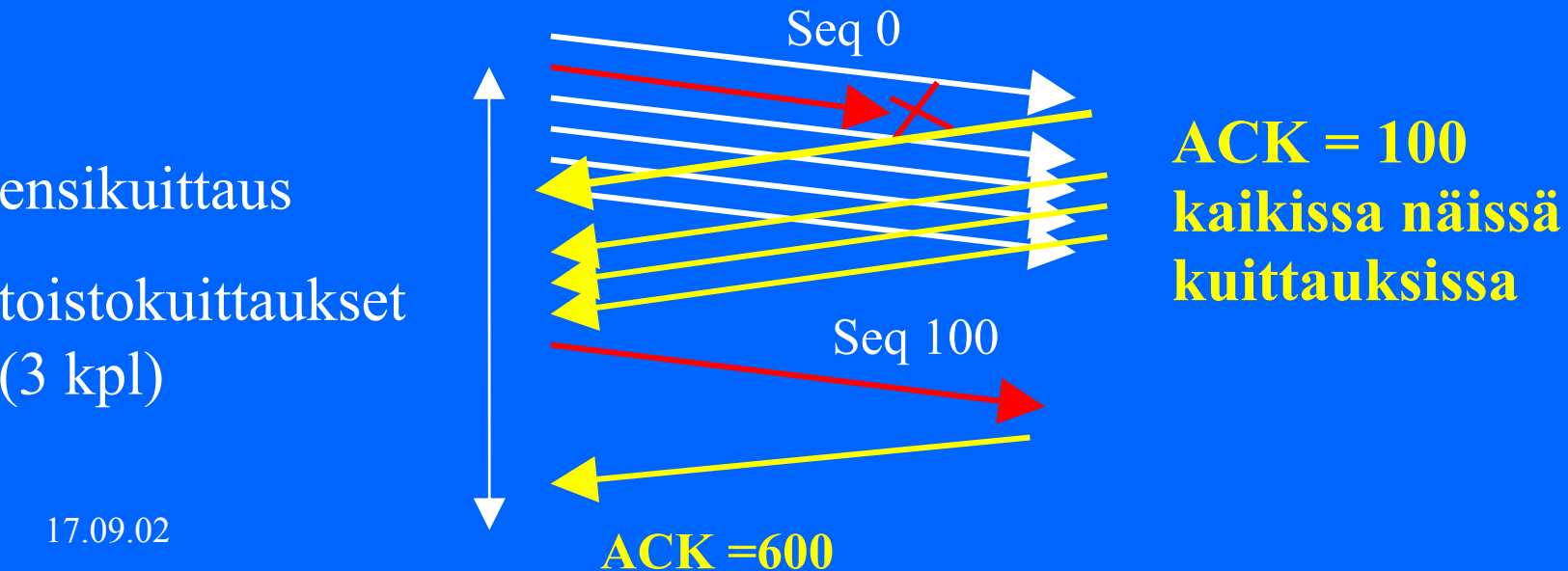
Ajastin laukeaa noin 400 ms kuluttua, jonka jälkeen aloitetaan hidaskäynnitys!

“Duplicate Ack”

- ◆ ensikuittaus (first-time ACK)
 - segmentin ensimmäinen kuittaus
 - tähän saakka kaikki on kunnossa
- ◆ toistokuittaus (duplicate ACK)
 - vastaanottaja kuittaa viimeksi saatua hyväksytyä segmenttiä aina kun saa virheellisen tai väärässä järjestyksessä tulevan segmentin
 - NAKin korvike, jolla ilmoitetaan ongelmista lähettäjälle

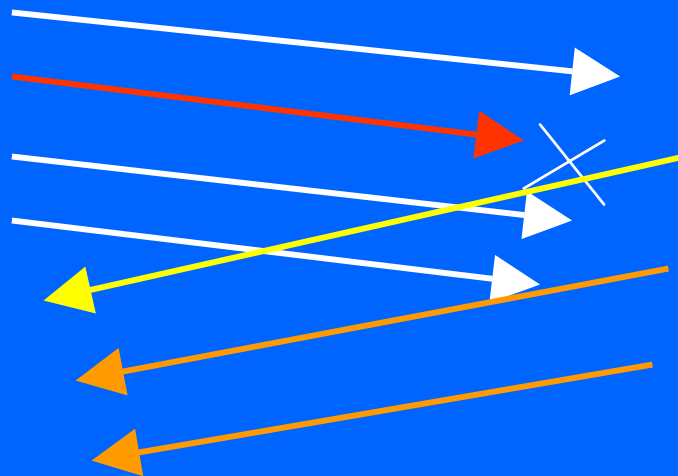
Nopea uudelleenlähetytys (Fast retransmit)

- ◆ Kun lähettäjä vastaanottaa 3 toistokuittausta samalle segmentille, se lähettää heti puuttuvan segmentin uudestaan
 - eikä odota segmentin ajastimen laukeamista



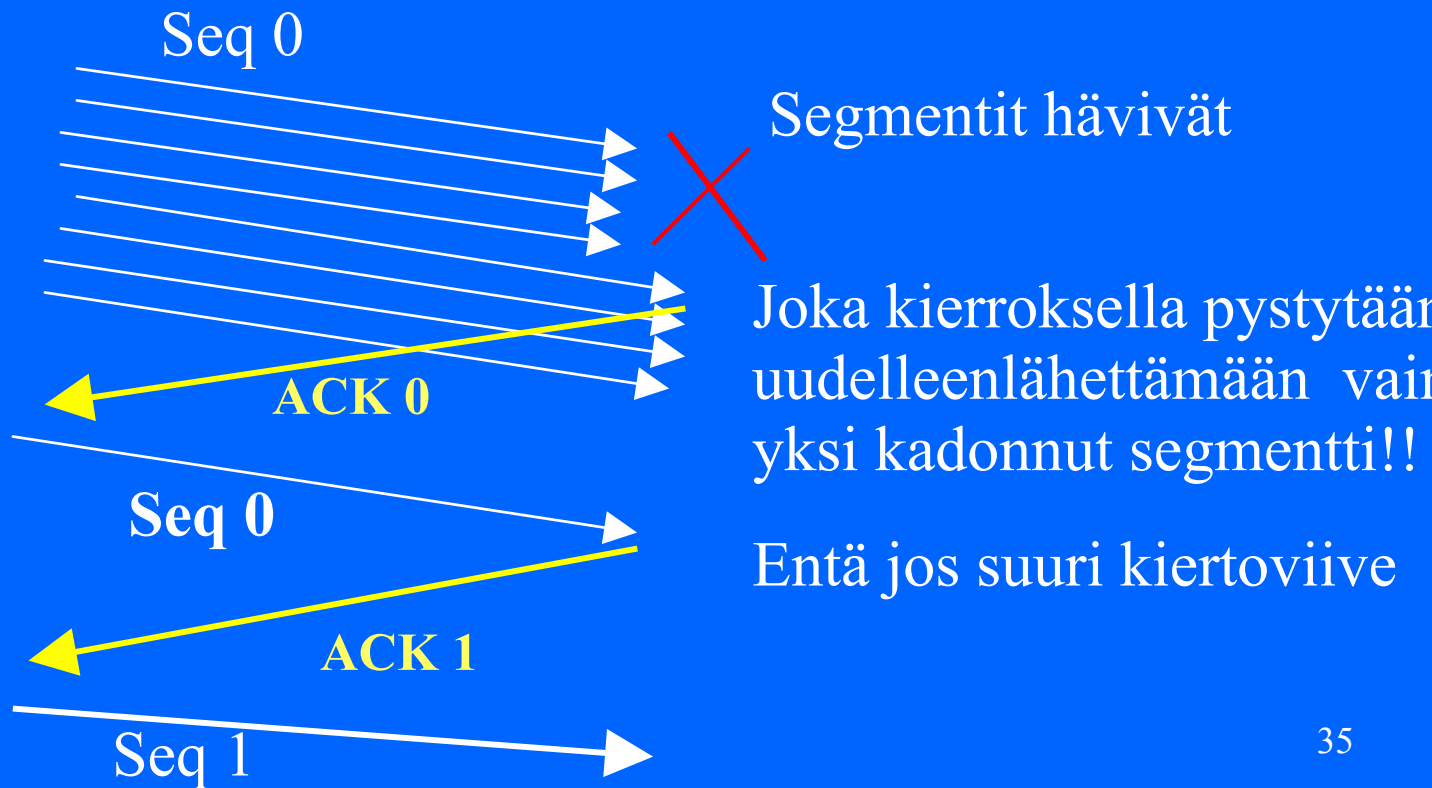
Ongelma 1: Ei saada kolmea toistokuittausta

- ◆ Jos ruuhkaikkuna on hyvin pieni,
 - ei voi tulla kolmea toistokuittausta, jos ruuhkaikkuna sallii vain kolme kuittaamatonta lähetystä



Ongelma 2: Virheryöppy tuhoaa monta segmenttiä

- ◆ Kun useita segmenttejä katoaa ‘samasta ikkunasta’



Limited Transmit

◆ RFC 3042: Enhancing TCP's Loss Recovery Using Limited Transmit.

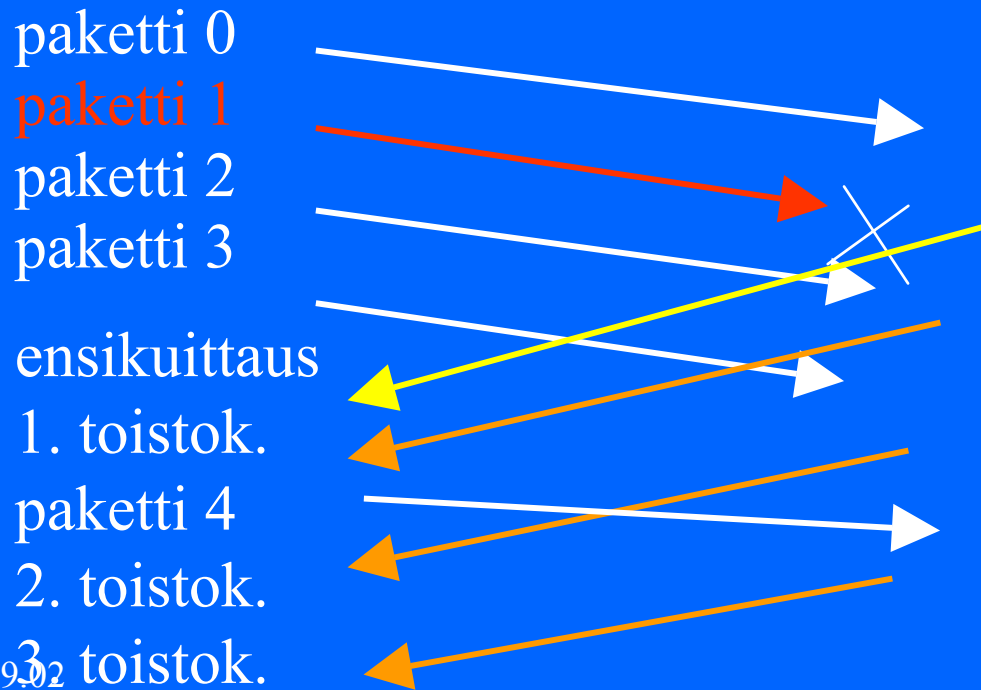
M. Allman, H. Balakrishnan, S. Floyd. January 2001
(Status: PROPOSED STANDARD)

- ◆ Lähettäjä ei saa kolmea toistokuittausta =>
 - odotettava aina ajastimen laukeamista ja
 - suoritettava hidas aloitus
 - => hidastaa usein turhaan lähettämistä

Ratkaisu:

- ◆ Lähettäjä saa lähettää yhden **uuden paketin** verkkoon vastaanotettuaan **1. ja 2. toistokuittauksen**.
 - » kuittaus kertoo, että verkosta poistettu paketti, joten verkkoon siis mahtuu!
 - » Tilapäisesti ruuhkaikkunan koko ylitetään kahdella MSS:llä
- ◆ Kun saman paketin toistokuittauksia tulee kolme, niin suoritetaan nopea uudelleenlähetyks ja nopea toipuminen (fast recovery)

- ◆ Vaikka ruuhkaikkuna on pieni, niin rajoitetulla lähetyksellä saadaan tarvittaessa syntymään kolme toistokuittausta



Miksi lähetetään uusi paketti?

- ◆ Miksi ei heti ensimmäisen toistokuittauksen jälkeen lähetä uudestaan sitä jo lähetettyä kuittaamatonta pakettia?
- ◆ Koska ei vielä olla varmoja siitä, että paketti on todella kadonnut.
 - Se voi olla vain viivästynyt
 - tai paketit ovat matkalla joutuneet väärään järjestykseen
- ◆ => näin vältetään turhia uudelleenlähetystyksiä