

## Tietoliikenne II (2 ov)

Syksy 2001  
Liisa Marttinen

Kurssikirja:  
Kurose & Ross, Computer Networking  
Lisämateriaalia: Aiheeseen liittyvät RFC:t

24.10.2001

1

## Tietoliikenne II

Täydennystä Tietoliikenne I -kurssin asioihin

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

24.10.2001

2

## Alustava sisällysluettelo

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

24.10.2001

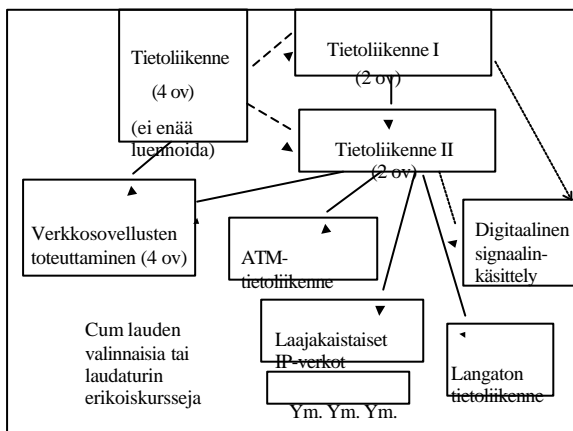
3

## Sisällysluettelo jatkuu

- ◆ Linkkikerroksen ja peruserroksen asioita
  - HDLC, PPP, SONET, ..
  - Tiedonsiirron teoreettinen perusta (Shannon, Nyquist, ...)
- ◆ Internetin palvelun laatu (QoS)
  - integroidut palvelut
  - eriytyneet palvelut
- ◆ ???

24.10.2001

4



## Suoritus

- ◆ kurssikoe maks. 50 p, min 20 p
  - ma 17.12. klo 9-13 sali 1 päärakennus
- ◆ kurssiaktiivisuus maks. 20 p
  - traditionaaliset harjoitukset => maks. 10 p
  - miniesseet (1-5 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 2002

24.10.2001

6

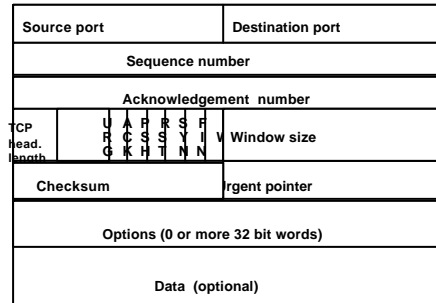
## TCP

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

24.10.2001

7

### 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 =>  $15 \cdot 4$  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

24.10.2001

9

## Optioita:

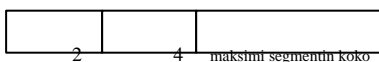
- ◆ MSS (Maximum Segment Size)
  - käytetään ilmoittamaan vastaanottajan yhteydellä hyväksymä suurin segmentin koko
    - » eri suuntiin voi olla eri koko
    - » voi olla suurempi tai pienempi kuin oletus MSS
      - ◆ MTU (Maximum Transfer Unit) = suurin yhdessä verkon kehyksessä kulkeva datamäärä
      - ◆ eri verkoissa eri kokoja, mutta minimi MTU= 576 B
      - ◆ MSS = MTU - IP-otsake - TCP-otsake
      - ◆ oletus MSS =  $576 - 20 - 20 = 536$
      - ◆ otsakkeet voivat olla suurempia!

24.10.2001

10

- ◆ MSS ilmoitetaan yhteyttä muodostettaessa eli SYN-segmenteissä

- kumpikin osapuoli voi ilmoittaa oman MSS-arvonsa
  - » jos ei ilmoita, niin suostuu vastaanottamaan minkä tahansa kokoisia segmenttejä.

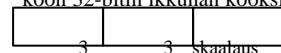


24.10.2001

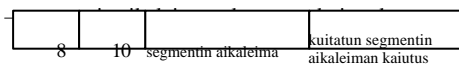
11

## Muita ehdotettuja optioita (RFC 1323):

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



- ◆ aikaleimaus (timestamp)



24.10.2001

12

## 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 lähetysopeutta mm. satelliittiyhteyksillä
    - » järjestysnumero 32 bittiä
      - ◆ rajoittaa lähetysopeutta erittäin nopeilla yhteyksillä

24.10.2001

13

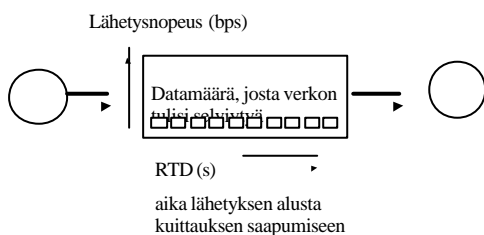
## Kaistan ja kiertoviiveen tulo (bandwidth \* delay product)

- ◆ 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!

24.10.2001

14

Ideaalitilanteessa 'lähetyspotki' on koko ajan täynnä!



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

## Ongelmia LFN-verkoissa

- ◆ Ikkunan koko -kenttä liian pieni => 'putki' ei täyty
  - ikkunan skaalaus -optio
- ◆ pakettien katoaminen => hidas aloitus eli 'putki' joudutaan tyjentämään
  - parannukset ruuhkanhallintakäytäntöön
    - » Fast Retransmit, Fast Recovery, SACK, ...
    - » uudelleenlähetyssajastimen tarkempi ajastus
      - ◆ Timestamp -optio

24.10.2001

16

## Ikkunan skaalaus (Window scale factor)

- ◆ ikkunakoko = 16 bittiä => 65536 tavua
  - » vuonvalvonnassa
  - » kertoo vastaanottajan ikkunan = kuinka monta tavua voi lähettää ennenkuin täytyy jäädä odottamaan kuittausta
- ◆ jos käytössä ikkunan skaalaus -optio, ikkunakentän arvo kerrotaan  $2^*F$ , jossa F on skaalausoption arvo.
  - » Suurin F:n arvo on 14.
- ◆ käytetään vain yhteyden aloituspyynnössä

24.10.2001

17

## 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

24.10.2001

18

## Uudelleenlähetyssajastimen 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.

24.10.2001

19

## Aikaleima (timestamp)

- ♦ Kaksi eri optiota
  - Timestamp Value
    - ♦ lähteissä segmenteissä,
  - Timestamp Echo Reply
    - ♦ kuitauksessa
    - ♦ sama kuin kuitatun segmentin Timestamp-arvo
- ♦ Voidaan käyttää missä tahansa datasegmentissä
- ♦ => Voidaan laskea kiertoviive jokaiselle segmentille, myös uudelleenlähetyksille.

24.10.2001

20

## RTTM (Round-Trip Time Measurement)

- ♦ lähetettävään sanomaan liitetään aikaleima-option aikaleima
  - ♦ aikaleimakello, joka tikittää riittävän nopeasti
- ♦ sama aikaleima palautetaan sanoman kuitauksessa
- ♦ ongelmatilanteita:
  - viivästyneet kuitaukset: aikaisin kuitaamaton
    - ♦ TCP:n ei tarvitse kuitata jokaista segmenttiä
  - puuttuva segmentti: viimeisin hyväksytty
  - puuttuvan segmentin saapuminen: viimeisin puuttuva

24.10.2001

21

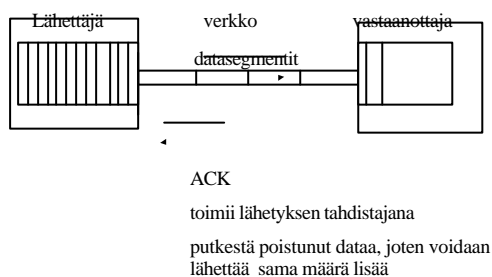
## Viivästetty ACK (Delayed ACK)

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

24.10.2001

22

## TCP ruuhkanvalvonta



24.10.2001

23

## TCP self-clocking

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

24.10.2001

24

## Ruuhkanvalvonta on hankalaa!

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

24.10.2001

25

## 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(rwnd, cwnd)** rajoittaa lähettämistä

24.10.2001

26

## Ruuhkaikkunan arvo eri tilanteissa

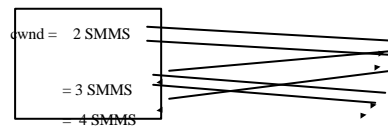
- initial window (IW)
  - » ruuhkaikkunan arvo heti kolminkertaisen kättelyn jälkeen
    - ◆ korkeintaan kaksi segmenttiä tai 2\* suurin määrä tavuja, jonka lähettäjä voi kerralla lähettää (SMSS)
- loss window (LW)
  - » ikkunan arvo, kun TCP on havainnut, uudelleenlähetyksajastimen lauettua, segmentin kadonneeksi
- restart window (RW)
  - » kun lähetyks käynnistetään uudelleen joutilaana olon jälkeen

24.10.2001

27

## Slow start

- ◆ Hitaan aloituksen aikana
  - Ruuhkaikkunaa cwnd kasvatetaan korkeintaan maksimilähetyksmäärällä (SMSS) jokaista uutta dataa kuittaavaa ACKia kohden

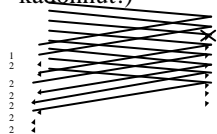


24.10.2001

28

## Hidas aloitus

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



Ajastin laukeaa noin 400 ms kuluttua, jonka jälkeen aloitetaan hidas aloitus!

24.10.2001

29

## “Duplicate Ack”

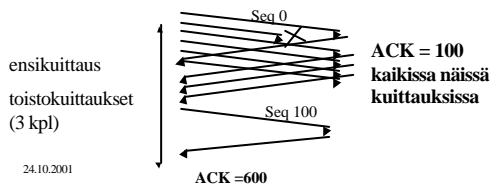
- ◆ ensikuittaus (first-time ACK)
  - segmentin ensimmäinen kuittaus
  - tähän saakka kaikki on kunnossa
- ◆ toistokuittaus (duplicate ACK)
  - vastaanottaja kuittaa viimeksi saatua hyväksytyä segmenttiä aina kun saa virheellisen tai väärässä järjestyksessä tulevan segmentin
  - NAKin korvike, jolla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

24.10.2001

30

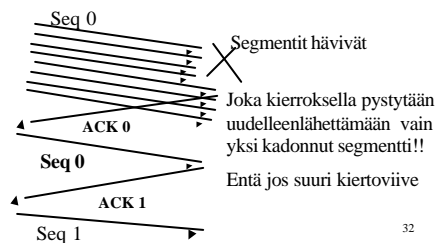
## Nopea uudelleenlähetys (Fast retransmit)

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



## Suorituskykyongelmia!

- ◆ 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)

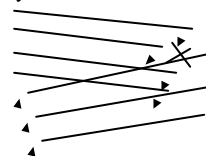
- ◆ Lähettäjä ei saa kolmea toistokuittausta =>
  - odotettava aina ajastimen laukeamista ja
  - suoritettava hidas aloitus
  - => hidastaa usein turhaan lähettämistä

24.10.2001

33

## Lähettäjä ei saa kuittauksia

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



## Ratkaisu:

- ◆ Kun lähettäjä saa toistokuittauksen, se saa aina lähettää yhden **uuden paketin** verkkoon
  - » kuittaus kertoo, että verkosta poistettu paketti, joten verkkoon siis mahtuu!
- ◆ Kun saman paketin toistokuittauksia tulee kolme, niin suoritetaan nopea uudelleenlähetys ja nopea toipuminen (fast recovery)

24.10.2001

35

## SACK (Selective Acknowledgement)

- RFC 2018  
TCP Selective Acknowledgement Options.  
M. Mathis, J. Mahdavi, S. Floyd, A. Romanow. October 1996.  
(Status: **PROPOSED STANDARD**)

INTERNET DRAFT  
Mark Allman, Ethan Blanton. "A Conservative SACK-based Loss Recovery Algorithm for TCP".  
(draft-allman-tcp-sack-02.txt), January, 2001

24.10.2001

36

- ◆ Kumulatiivinen kuittaus paljastaa aina vain yhden puuttuvan kerrallaan

- ◆ SACK paljastaa kaikki puuttuvat
  - » ilmoittamalla, mitkä segmenttivälit on jo vastaanotettu

- ◆ Esim. Segmentin koko 1000 tavua
  - 1. segmentti katoaa ja muut tulevat perille
    - ◆ segmentin 2 kuittaus: ACK 0, 1000: 2000
    - ◆ segmentin 10 kuittaus: ACK 0, 1000: 10000
  - 1. ja 3. segmentti katoavat
    - ◆ segmentin 10 kuittaus:
    - ◆ ACK 0, 1000:2000 3000:10000

24.10.2001

37

## SACK-optiot

- ◆ SACK- permitted yhteyden muodostuksessa eli vain SYN-segmentissä ilmoittamaan, että yhteydellä voidaan käyttää SACK-kuittauksia

- ◆ (type = 4, length = 2)

- ◆ SACK-optio
  - kuljettaa lisäinformaatiota saapuneista segmenteistä eli kertoo, mitkä 'tavupätkät' ovat jo valmiina vastaanottajan puskurissa
  - kuljetetaan TCP-segmentin optio-osassa

24.10.2001

38

## TCP:n SACK-optio



4 lohkoa mahtuu yhteen TCP-segmenttiin, jossa optiolle on varattu 40 tavua, jos ei käytetä muita optioita kuten aikaleimaa (timestamp).

24.10.2001

39

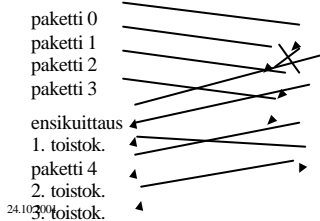
## Vain neuvoo-antava!

- ◆ Ohjeellista tietoa lähettäjälle
  - vastaanottaja voi tarvittaessa poistaa SACK-optiossa ilmoittamia tavuja puskureistaan
- ◆ Jos vastaanottaja käyttää SACK-optiota, niin sitä on käytettävä aina kun vastaanottajalla on puskureissaan epäjärjestyksessä olevaa dataa
  - tällöin kaikissa ACK:ssa on oltava ajantasalla oleva tieto siitä, mitkä tavut on jo puskureissa

24.10.2001

40

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



24.10.2001

41

## 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

24.10.2001

42