

Kuittaukset

- ACK
 - kumulatiivinen ACK
 - tähän saakka kaikki ok!
 - Go-Back N
 - yksittäinen ACK
 - vain tämä ok!
 - Valikoiva toisto
- NAK-kuittaus
 - sanoma virheellinen tai puuttuu

5.10.2001

40

Negatiiviset kuittaukset

- NAK-kuittauksilla voidaan nopeuttaa uudelleenlähettämistä
 - vastaanottaja ilmoittaa heti virheellisestä tai puuttuvasta kehuksesta
 - ei ole tarpeen odottaa ajastimen laukeamista
- hyödyllinen, jos kuittausten saapumisaika vaihtelee paljon
 - ajastinta vaikea asettaa oikein

5.10.2001

41

- NAK-kuittaukset voivat aiheuttaa turhia uudelleenlähetyksiä
 - lähetyksen ja kuittaus menevät ristiin
- NAK-kuittauksen katoaminen ei haittaa
- implisiittinen uudelleenlähetyksenä
 - ei NAK-kuittauksia
- explisiittinen uudelleenlähetyksenä
 - käytetään NAK-kuittauksia

5.10.2001

42

Ikkunankoko

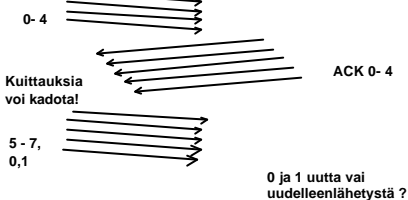
- Kun käytetty numeroavaruus on $0, 1, \dots, n$ ja eri numeroita siis käytettävissä $n+1$
 - yleensä jokin kakkosen potenssi
 - » koska numerokentän koko k bittinä \Rightarrow käytössä 2^k numeroa
- ikkunan koko 'go back n ':ssä voi olla korkeintaan n
 - eli ainakin yhtä pienempi kuin numeroavaruus
- ikkunan koko valikoivassa toistossa voi olla korkeintaan $(n+1)/2$
 - korkeintaan puolet numeroavaruudesta

5.10.2001

43

Miksi?

- Valikoiva toisto: ikkuna 5, numeroavaruus 8

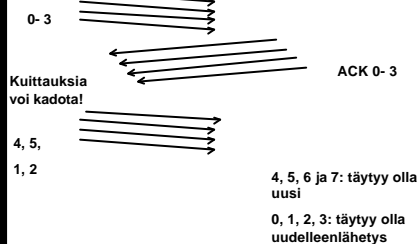


5.10.2001

44

Miksi?

- Valikoiva toisto: ikkuna 4, numeroavaruus 8



5.10.2001

45

Kaksisuuntainen liikenne

- datakehys ja kuittauskehys
- kehyksessä sekä data että kuittaus
 - 'piggybacking'
 - tehostaa lähetystä
- ongelma: kauanko kuittaja odottaa dataa ennen pelkän kuittauksen lähettämistä?

5.10.2001

46

3.5. TCP-protokolla

- yhteyden muodostus ja purku
- luotettavan tavuvirran toteuttaminen
- vuonvalvonta
- siirron optimointi
- TCP-segmentti
- ruuhkan valvonta
- TCP-palvelun käyttö

5.10.2001

47

6.2.2. Yhteyden muodostus ja purku TCP:ssä

- TCP käyttää yhteyden muodostamiseen ja purkuun ns. kolminkertaista kättelyä (three-way handshake)
 - välissä oleva verkko tekee yhteyden muodostamisen ja purun hankalaksi
 - viivästyneet sanomat => sanomille elinaika
 - sanomien numeroinnista sopiminen
 - Bysanttilainen ongelma (two-army problem)
 - "hyökkään, jos olen varma, että sinäkin hyökkäät"
 - symmetrinen yhteyden purku = molemmat osapuolet tietävät, että toinenkin on varmasti purkanut yhteyden

5.10.2001

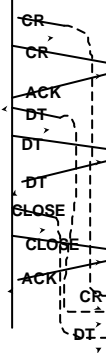
48

Yhteyden muodostus ruuhkaisessa verkossa

Jokainen paketti lähetetään kahteen kertaan

Kun yhteys on purettu, viivästyneet kaksoiskappaaleet saapuvat

Ne tulkitaan uudeksi yhteydeksi, ja data otetaan vastaan kahteen kertaan!



5.10.2001

49

SYN =
tahdistus-
sanoma

SYN, Seqno=x

SYN+ACK, Seqno=y,
ack=x+1

ACK, Seqno=x+1,
ack=y+1

Kolminkertainen kättely

yhteyspyynnössä pyytäjän nro x

vahvistuksessa sekä pyytäjän järj.numero

ensimmäisessä datalähetyksessä molemmat numerot

5.10.2001

50

#1 #2

Hyökkään aamulla kello 5!

OK, siis kello 5!

OK!

#2 hyökkää vain, jos tietää minun saaneen vastaussanoman.

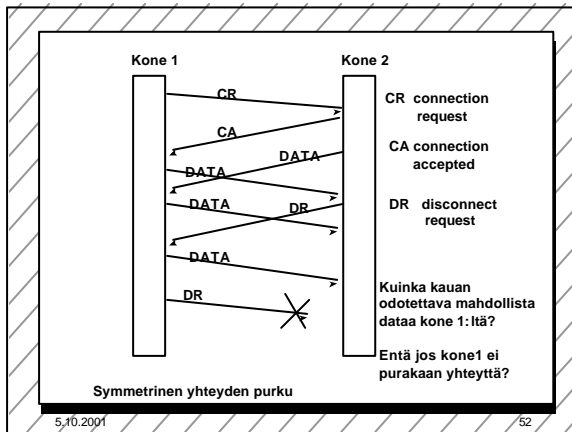
Entä jos vastaus ei mene perille? Silloin #1 ei hyökkää!

Loogisesti ratkeamaton ongelma. Kaikki riippuu aina viimeisestä sanomasta, jonka perillemeno ei voida taata!

Bysanttilainen ongelma (two-army problem)

5.10.2001

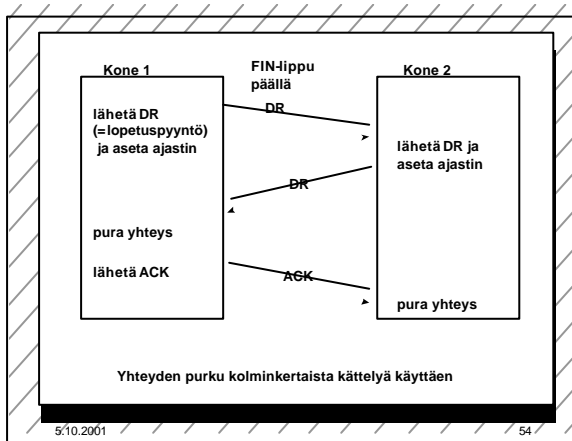
51



Yhteyden purku

- molemmat suunnat puretaan erikseen
- TCP-segmentti
 - FIN = 1
 - ei enää dataa lähetettävä
 - kun saadan kiittaus => yhteys tähän suuntaan purettu
 - yhteys kokonaan purettu, kun molemmat suunnat purettu
- purussa käytetään ajastimia
 - 2 * pakettin maksimaalinen elinikä

5.10.2001 53



Virheettömyys ja järjestys

- Järjestysnumerot
 - tavuvirta => tavunumerointi
 - segmentin 1. tavun järjestysnumero
 - yhteyden alussa satunnaiset numerot
- kiittaukset
 - kumulatiivinen ACK, ei NAK-kiittausta
 - kiittäuksessa seuraavaksi odotettava tavu
 - kuitataan 'tiheästi'
 - vähintään joka toinen

5.10.2001 55

- Go Back N -tyyppinen
 - virheellisiä tai väärässä järjestyksessä tulleita ei hyväksytä
 - ne voidaan myös tallettaa
 - mutta ei välttämättä lähetä kaikkia virheellisestä lähtien uudestaan
- Myös ehdotettu valikoivan toiston tyyppistä kuitaamista
 - SACK-kiittaus, joka kertoo, mitkä segmentit on vastaanotettu ok

5.10.2001 56

Toistokuittaukset

- Ensikuittaus
 - tähän saakka kaikki OK!
 - ensimmäisen kerran saatava
- toistokuittaus (duplicate ACK)
 - väärässä järjestyksessä saatu segmentti tai virheellinen segmentti => toistetaan uudestaan jo annettu kiittaus
 - NAK-kiittauksen korvike
 - 3 toistokuittausta => segmentti kadonnut tai

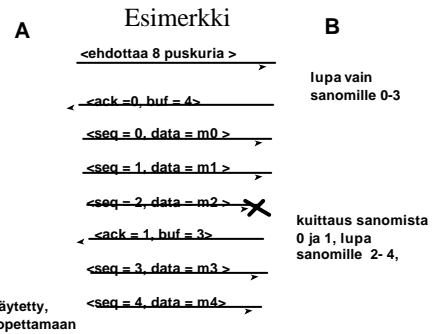
5.10.2001 57

TCP:n vuonvalvonta

- 'joustava' liukuva ikkuna (sliding window) (credit-vuonvalvonta)
- vastaanottaja kertoo, kuinka paljon suostuu vastaanottamaan
 - => kuittaus irroitettu vuonvalvonnasta
 - AdvertisedWindow-kenttä
 - paljonko saa lähettää = paljonko vastaanottajan puskureihin mahtuu
- myös ruuhkan valvonta rajoittaa lähettämistä

5.10.2001

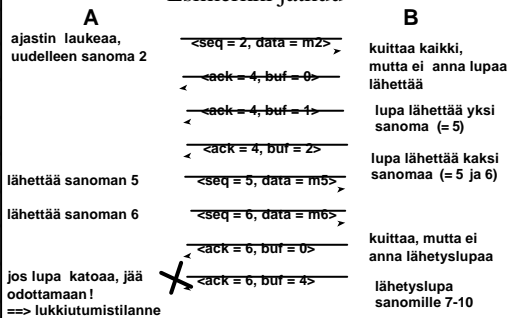
58



5.10.2001

59

Esimerkki jatkuu



5.10.2001

60

- jos ilmoitus lisäpuskureista katoaa, lähettäjä lukkiutuu odotustilaan
 - vastaanottaja voi luulla, ettei ole lähetettävää
- lukkiutumisen estämiseksi
 - kun ikkunankoko = 0 lähettäjä ei saa lähettää, paitsi
 - pikadataa (URG)
 - yhden tavun 'kyselyn', jonka vastaanottaja kuittaa ja samalla ilmoittaa ikkunan koon => estää turhat lukkiutumiset

5.10.2001

61

Siirron optimointi

- TCP saa optimoida lähettämisiään
 - ei tarvitse lähettää heti kun data on tullut
 - dataa kerätään puskuriiin ja lähetetään sopivassa tilanteessa
 - PUSH-lipun avulla sovellus ilmoittaa, että data on lähetettävä heti

5.10.2001

62

Optimointi on usein tarpeen:

- Interaktiivinen editori => merkki lähetetään heti
 - 21 tavun TCP-segmentti => 41 tavun IP-paketti
 - joka kuimitaan 40 tavun IP-paketilla
 - ilmoitus uudesta ikkunan koosta 40 tavun IP-paketilla
 - kaitetaan merkki vielä 41 tavun IP-paketilla
- yhden merkin käsittely=>
 - 162 tavun siirtäminen
 - ja neljän segmentin lähettäminen

5.10.2001

63

■ Ratkaisu: Naglen algoritmi

- jos data tulee tavuttain
 - lähetä 1. tavu
 - kerää sitä seuraavat tavut puskuriin ja lähetä vasta kun edellinen lähetys on kuitattu
 - paitsi jos lähetettävää on suurimman segmentin verran tai puolet ikkunan koosta
- hankala, jos hiirtä liikutellaan Internetin kautta!

5.10.2001

64

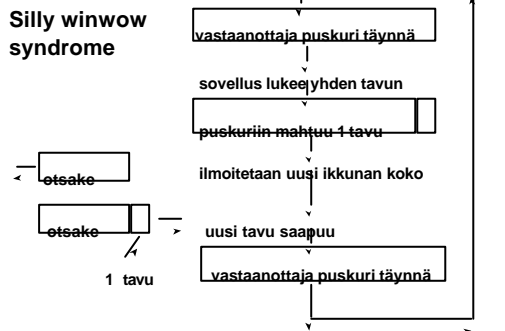
Silly window syndrome

- Tilanteessa, jossa
 - lähettäjältä dataa TCP:lle suurina lohkoina
 - vastaanottajalle mahtuu vain tavu kerrallaan
- voi tuhota TCP:n suorituskyvyn
 - koko data lähetetään tavu kerrallaan
 - joka tavun välissä ilmoitus ikkunan koon kasvattamisesta yhdellä
- Siis: ei ilmoitusta yhdestä tavusta, lähettäjä ei lähetä yhtä tavua
 - koko segmentti
 - puolet puskurin koosta

5.10.2001

65

Silly window syndrome



5.10.2001

66

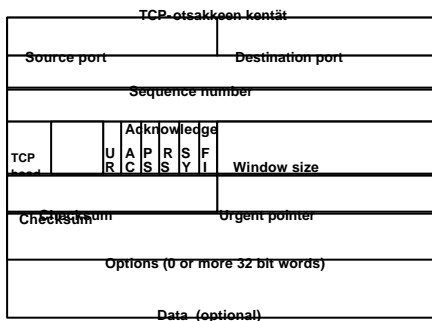
TCP-segmentti

- segmentti
 - 20 tavun otsake
 - + optionaalinen osa
 - dataosa
 - voi puuttua
- segmentin kokoa rajoittaa
 - MTU (Maximum transfer unit)
 - verkon rajoitus maksimikoolle (muutama tuhat tavua)
 - IP-paketin dataosa korkeintaan 65535 tavua
- liian isot segmentit paloittellaan
 - joka palalle IP-otsake => yleisrasite kasvaa

5.10.2001

67

TCP-otsakkeen kentät



5.10.2001

68

TPC-segmentin otsakekentät

- **Lähde- ja kohdeportit** (Source port, Destination port)
 - yhteyden päätepisteet
 - portti + koneen IP-osoite => 48 bittinen TSAP
- **Järjestysnumero** (Sequence number)
 - tavut numeroidaan => 32 bittiä
 - segmentin ensimmäisen tavun numero
- **Kuittausnumero** (Acknowledgement number)
 - seuraavaksi odotettu tavu
- **TCP-otsakkeen pituus** (TCP header length)
 - mahdollisten optiokenttien takia

5.10.2001

69

■ 6 lippubittii

- **URG** onko pikadataa
pikadatan sijainnin ilmoittaa
pikadatakenttä (Urgent pointer)
- **ACK** onko kuittauskenttä käytössä
- **PSH** onko hetilähetettävää (pushed) dataa
- **RST** yhteyden uudelleenalustuspyyntö (reset), yleensä ongelmatilanne
- **SYN** käytetään yhteyttä muodostettaessa
SYN = 1, ACK = 0 connection request
SYN = 1, ACK = 1 connection accepted
- **FIN** käytetään yhteyden purkuun
FIN = 1 ei enää lähetettävää

5.10.2001

70

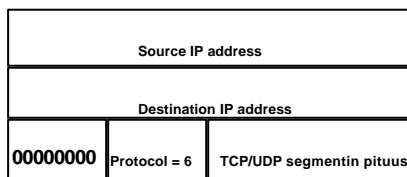
■ Ikkunan koko (window size)

- vaihteleva ikkunankoko
- kuittaus irroitettu lähetysluvasta
- **Tarkistussumma** (Checksum)
 - lasketaan otsakkeelle, datalle ja ns. pseudo-otsakkeelle

5.10.2001

71

pseudo-otsake



Auttaa havaitsemaan väärään osoitteeseen toimitetut paketit

Sisältää IP-otsakkeen tietoja!

5.10.2001

72

■ Optiokenttä (options)

- voidaan lisätä piirteitä, joita ei ole varsinaisessa otsakkeessa
 - suurin hyväksyttävä datakenttä
 - ikkunan koon moninkertaistaminen (window scale)
 - nopeille ja pitkän viipeen linjoille 64 tavun ikkunan koko on liian pieni
 - valikoivan toiston käyttö 'go back N':n tilalla
 - vähentää turhia uudelleenlähetystyksiä

5.10.2001

73

3.6. TCP:n ruuhkan valvonta

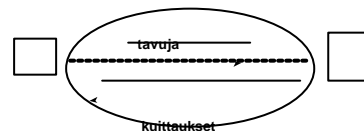
- Liikaa kuormitusta => verkko ruuhkautuu => hidastetaan lähettämistä
- Ruuhkan havaitseminen
 - nykyisin siirtovirheet harvinaisia
 - poikkeuksena langattomat verkot
 - => uudelleenlähetykset johtuvat ruuhkasta
 - uudelleenlähetysajastimen laukeaminen on merkki ruuhkasta

5.10.2001

74

■ ruuhkaikkuna

- "paljonko tavuja (segmenttejä) lähettäjällä saa korkeintaan olla verkossa liikkeellä"
- kuittaus => ko. tavut jo poistuneet verkosta



5.10.2001

75

■ Ruuhkaikkunan koko?

- Lähettäjän on itse pääteltävä ja arvioitava sopiva ruuhkaikkunan koko
 - kukaan muu ei sitä kerro!
 - timeout => on ruuhkaa
 - kuittaukset tulevat tasaisesti => ei ole ruuhkaa

■ Dynaaminen ruuhkaikkunan koko:

- ruuhkaikkunaa kasvatetaan kunnes törmätään ruuhkaan
- sen jälkeen ruuhkaikkunaa pienennetään reilusti
- ja aletaan uudestaan kasvattaa ruuhkaikkunaa

5.10.2001

76

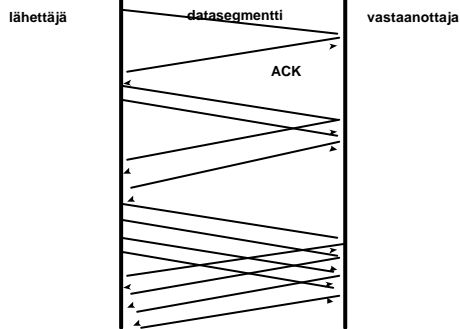
Hitaan aloituksen algoritmi (slow start)

- Algoritmi pyrkii löytämään sopivan ikkunan koon yhteyden alussa tai ruuhkatilanteen jälkeen mahdollisimman nopeasti

- ei ole niin kovin hidas, vaan alussa eksponentiaalinen!
- alussa ruuhkaikkuna = yksi segmentti
- kuitattu ruuhkaikkunallinen kasvattaa ruuhkaikkunan kaksinkertaiseksi

5.10.2001

77



5.10.2001

78

■ kynnysarvo (threshold)

- 'varoitussarvo' = tästä lähtien syytä varoa ruuhkaa
- aluksi 64 K

- kynnysarvoon saakka voidaan kasvattaa ruuhkaikkunaa eksponentiaalisesti
- kynnysarvon saavuttamisen jälkeen kasvatetaan ruuhkaikkunaa vain lineaarisesti
 - = kasvatetaan kuittausten jälkeen vain yhdellä
 - edetään hyvin varovaisesti!

5.10.2001

79

■ jos ajastin ehtii laueta => ruuhkatilanne

- kynnysarvoksi puolet nykyisestä ruuhkaikkunan arvosta
- hitaalla aloituksella etsitään taas uusi sopiva ruuhkaikkunan arvo
 - ruuhkaikkunan arvoksi 1 segmentti
 - ruuhkaikkunaa kasvatetaan aluksi eksponentiaalisesti eli kaksinkertaistetaan kun ikkunallinen on kuitattu
- kynnysarvon saavuttamisen jälkeen kasvatetaan vain segmentti kerrallaan
- kunnes taas havaitaan ruuhka ja aloitetaan ruuhkaikkunan uuden arvon etsiminen

5.10.2001

80

Uudelleenlähetyssajastimen hallinta

- uudelleenlähetyssajastin (retransmission timer)

- asetetaan aina kun segmentti lähetetään
- ruuhkaa, jos kuittaus ei saavu ajoissa

- mikä on sopiva ajastimen aika?

- kuittaus aika vaihtelee suuresti
- vaihtelu on myös nopeaa

- dynaaminen arvo

- saadaan jatkuvien verkon suorituskykymittauksien perusteella

5.10.2001

81

■ RTT

- arvio kiertoviiveelle (round-trip time)
- mitataan jokaisen lähetetyn segmentin kiertoviive M
- RTT = α RTT + (1- α)M, tyypillisesti $\alpha = 7/8$

■ uudelleenlähetyksajastimen arvo β RTT

- aluksi β oli aina 2
- parannus: otetaan huomioon myös poikkeama D (deviation) oletetun ja saadun kiertoviiveen välillä $|RTT-M|$
- $D = \alpha D + (1-\alpha)|RTT-M|$
- ajastimen arvo = RTT + 4*D

5.10.2001

82

■ uudelleenlähetyksen vaikutus ajastimeen

- kumpaan segmenttiin kiittäus kohdistuu?

■ Karnin algoritmi

- ei oteta huomioon uudelleenlähetyksen segmenttien kiittäuksia RTT:n laskemisessa

5.10.2001

83

