

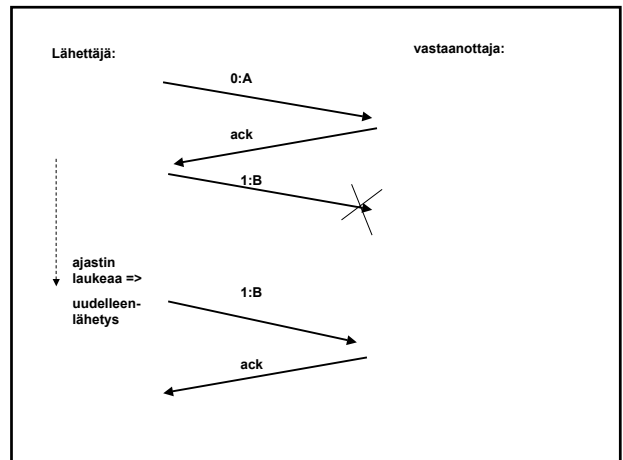
Monimutkaisempi "stop and wait"-protokolla

■ ajastin lähettäjälle

- jos kuittausta ei kuulu, sanoma lähetetään automaattisesti uudelleen
- kuittaus: ACK = 'ok, lähetä seuraava'
- uudelleenlähetyksen synnyttää kaksoiskappaleita!

■ Sanomanumerointi

- jotta vastaanottaja tunnistaa kaksoiskappaleet
- Miten paljon numeroita tarvitaan?
 - » Numero vie tilaa sanomassa!



Stop and wait -protokollan suorituskyky

■ Esim. satelliittiyhteydellä

- 50 kbps, kiertoviive ~520 ms, sanoma 1000 bittiä
- kanavan käyttöaste < 4%

■ => lähetetään useita sanomia ja sitten vasta odotetaan kuittauksia

- ideaali: lähetykset liukuhihnalla (pipeline)
 - lähetykset ja kuittaukset limittyvät
 - ei mitään odottelua
 - lähetyskanava koko ajan käytössä
- suorituskyky kasvaa

Liukuvan ikkunan protokolla

(Sliding Window)

■ Lähetyssikkuna

- ikkunan koko
 - montako sanomaa saa korkeintaan olla kuittaamatta
 - järkevä koko riippuu yhteyden tyypistä ja vastaanottajan kapasiteetista
 - kiinteä koko /vaihteleva koko
- sisältö = mitkä sanomat saa lähettää
 - sanomalla järjestysnumero
 - rajallinen, N bittiä => $2^{**}N$ arvoa
 - numerot käytettävä järjestyksessä

1/29/2003

34

■ Lähettäjä joutuu odottamaan vasta, kun kaikki ikkunan sanomat on lähetetty

- eli numerot käytetty

■ Kun kuittaus saapuu => ikkuna liikuu

- seuraavat numerot tulevat luvallisiksi

■ eli

- lähettäjä: tietyllä hetkellä sallittujen numeroiden joukko = lähettäjän ikkuna
 - mitkä sanomat saa lähettää "etukäteen" odottamatta kuittausta

1/29/2003

35

■ Vastaanottajan ikkuna

- kullakin hetkellä sallittujen numeroiden joukko
 - mitä sanomia suostuu vastaanottamaan

- kuittaus muuttaa myös vastaanottajan ikkunan

■ ikkuna pysäyttää sanomien lähetyksen

- seuraava sanomanumero ei ole lähetyssikkunassa

■ ikkuna estää sanoman vastaanoton

- saadun sanoman numero ei ole vastaanottoikkunassa

Kun ikkunan koko on 1

- Aina vain yksi sanoma kuittaamattomana
 - => One Bit Sliding Window -protokolla
 - ~ stop and wait -protokolla
- sanomanumerot 0 ja 1 riittävät
- ACK-sanoma identifioi viimeksi vastaanotetun virheettömän sanoman
 - jotta kuittausduplikaatti ei voi kuitata väärää sanomaa
 - ACK ilmoittaa joko
 - » seuraavaksi odotetun sanoman numeron
 - » viimeksi vastaanotetun sanoman numeron

1/29/2003

37

Entä kun tapahtuu virhe?

- kaksi eri tapaa hoitaa
 1. toisto virheestä lähtien (go back n) (tai paluu n:ään)
 2. valikoiva toisto (selective repeat)

1/29/2003

38

Toisto virheestä eli Paluu n:ään ('Go back n')

- virheellisen sanoman havaittuaan
 - vastaanottaja hylkää kaikkia sen jälkeiset sanomat eikä lähetä niistä kuittauksia
 - => sanomat hyväksytään vain oikeassa järjestyksessä
- kun lähettäjä ei saa kuittauksia,
 - sen lähetyksikkuna 'täyttyy'
 - eikä se voi enää lähettää
- lähettäjän ajastimet laukeavat aikanaan ja
 - virheellinen sanoma
 - sekä kaikki sen jälkeen lähetetyt sanomat lähetetään uudelleen
- tehoton, jos paljon virheitä ja iso ikkuna

Valikoiva toisto

- vastaanottaja hyväksyy kaikki kelvolliset sanomat
 - se kuittaa sanomat
 - puskuroi ne ja toimittaa eteenpäin oikeassa järjestyksessä
 - » tarvitaan puskuritilaa
- lähettäjä ei saa kuittauksia virheellisestä sanomasta
 - ajastin laukeaa ja sanoma lähetetään uudelleen
 - lähettää uudelleen vain virheellisen sanoman
 - ikkuna liukuu nytkin tasaisesti
 - » yksi puuttuva kuittaus voi pysäyttää lähetyksen

Kuittaukset

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

1/29/2003

41

Negatiiviset kuittaukset

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

1/29/2003

42

- **NAK-kuittaukset voivat aiheuttaa turhia uudelleenlähetystyksiä**
 - lähetys ja kuittaus menevät ristiin
- **NAK-kuittauksen katoaminen ei haittaa**
- **implisiittinen uudelleenlähetys**
 - ei NAK-kuittauksia
- **explisiittinen uudelleenlähetys**
 - käytetään NAK-kuittauksia

1/29/2003

43

Ikkunankoko

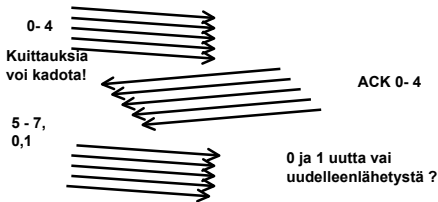
- Kun käytetty numeroavaruus on $0, 1, .. n$ ja eri numeroita siis käytettävissä $n+1$
 - yleensä jokin kakkosen potenssi
 - » koska numerokentän koko k bittiä => käytössä 2^{**k} numeroa
- ikkunan koko 'go back n':ssä voi olla korkeintaan n
 - eli oltava ainakin yhtä pienempi kuin numeroavaruus
- ikkunan koko valikoivassa toistossa voi olla korkeintaan $(n+1)/2$
 - saa olla korkeintaan puolet numeroavaruudesta

1/29/2003

44

Miksi?

Valikoiva toisto: ikkuna 5, numeroavaruus 8

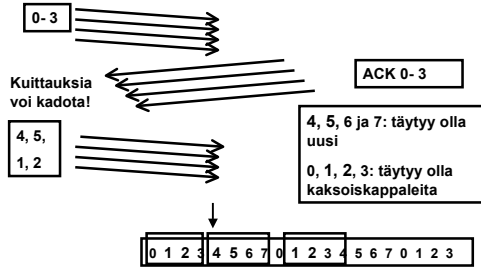


1/29/2003

45

Miksi?

Valikoiva toisto: ikkuna 4, numeroavaruus 8



1/29/2003

46

Kaksisuuntainen liikenne

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

1/29/2003

47

3.5. TCP-protokolla

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

1/29/2003

48

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 (max 3 minuuttia)
 - sanomien numeroinnista sopiminen
- Kahden armeijan ongelma (two-army problem)
 - "hyökkään, jos olen varma, että sinäkin hyökkäät"
 - symmetrinen yhteyden purku = molemmat osapuolet tietävät, että toinenkin on varmasti purkanut yhteyden

1/29/2003

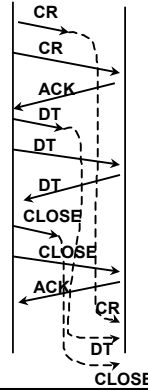
49

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!



1/29/2003

50

SYN =
tahdistus-
sanoma

SYN, Seqnro=x

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

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

Kolminkertainen kättely

yhteyspyynnössä
pyytäjän nro x
vahvistuksessa
sekä pyytäjän
järj.numero

ensimmäisessä
datalähetyksessä
molemmat
numerot

Yhteyden muodostus

1/29/2003

51

#1 Hyökkätään aamulla kello 5!

OK, siis kello 5!

OK!

#2 hyökkää vain, jos tietää minun saaneen vastaussanomana.

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

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

Kahden armeijan ongelma (two-army problem)

1/29/2003

52

Kone 1

Kone 2

CR

CA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

DATA

CR connection request

CA connection accepted

DR disconnect request

Kuinka kauan odotettava mahdollista dataa kone 1:ltä?
Entä jos kone 1 ei purakaan yhteyttä?

Sama ongelma: Symmetrinen yhteyden purku

1/29/2003

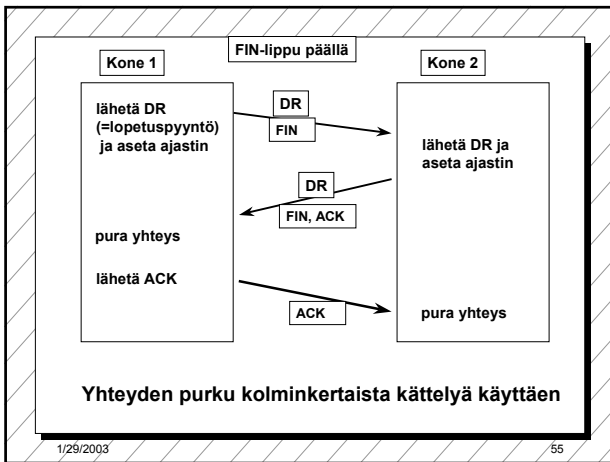
53

Yhteyden purku

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

1/29/2003

54



TCP: Virheettömyys ja järjestys

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

1/29/2003 56

- **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 kuittaamista**
 - SACK-kuittaus, joka kertoo, mitkä segmentit on vastaanotettu ok

1/29/2003 57

Toistokuittaukset

- **Ensikuittaus**
 - ensimmäinen vastaanotettu sanoman kuittaus
 - ACK(i): sanomaan i saakka kaikki OK!
- **toistokuittaus (duplicate ACK)**
 - väärässä järjestyksessä saatu segmentti tai virheellinen segmentti => toistetaan uudestaan jo annettu kuittaus
 - NAK-kuittauksen korvike
 - 3 toistokuittausta => segmentti kadonnut tai virheellinen

1/29/2003 58

TCP:n vuonvalvonta

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

1/29/2003 59

