

# Tietoliikenteen perusteet 2014, viikko 4

Viikon teemat: kuljetuskerros (transport layer), protokollien "ajatusmaailmaa" ja teoriaa, TCP-yhteyden vaiheet (aloitus, datan siirtäminen, lopetus), TCP:n ruuhkanhallinta ja vuonvalvonta

Harjoitukset ke 19.11. - to 20.11.2014

## 1. Vuorottelevan bitin protokolla (stop-and-wait eli rdt3.0)

- a. Jos vuorottelevan bitin protokollassa kuittaukset ACK0 ja ACK1 korvataankin ACK ja NAK kuittauksilla, onko myös kuittaukset ACK ja NAK numeroitava vuorottelevan bitin protokollassa? Tutki taas protokollan toimintaa eri tilanteissa ja selvitä, löytyykö tilannetta, jossa numeroimaton ACK tai NAK aiheuttaa ongelmia.
- b. Toimiiko vuorottelevan bitin protokolla, jos välissä oleva linkkiyhteys voi muuttaa sanomien järjestystä? Perustele vastauksesi osoittamalla kaaviona tilanne, jossa protokolla toimii väärin tai selvitä, miksi tällaista tilannetta ei voi syntyä. [Ch3P13]

## 2. Liukuvan ikkunan protokollien toiminta

Oletetaan, että ikkunan koko on 3 eli lähetettynä saa olla 3 sanomaa ( $n$ ,  $n+1$  ja  $n+2$ ), joita vastaanottaja ei ole vielä kuitannut. Oletetaan, että verkkoon tulee häiriöpurske, joka tuhoaa kolme viestiä: sanoman  $D(n+1)$ , sitä edeltävän sanoman  $D(n)$  ACK-kuittauksen ja vielä sitä seuraavan sanoman  $D(n+2)$  ACK-kuittauksen.

- a. Simuloi sanomien vaihtoa sanoman  $D(n)$  lähetyksestä sanoman  $D(n+2)$  ACK-kuittauksen vastaanottoon, kun toipumisessa käytetään
  - o paluu N:ään (Go-Back-N),
  - o valikoiva toisto (Selective Repeat)Esitä sanomien vaihto lähettäjän ja vastaanottajan välillä kaaviona, jossa aika kulkee ylhäältä alaspäin (tai vasemmalta oikealle).
- a. Olisiko NAK-kuittauksen käytöstä mitään hyötyä kummankaan protokollan tilanteessa? Mitä haittaa NAK-kuittauksesta voisi olla?
- b. Onko ajastin ihan välttämätön molemmissa protokollissa?

## 3. Tarkistussummista [Ch3P3, mutta eri arvot]

UDP ja TCP käyttävät ns. yhden komplementtia tarkistussummissaan.

- a. Laske 8 bitin mittainen UDP-tarkistussumma seuraavalle kolmen tavun mittaiselle sanomalle: 0101 0101 0111 0000 1100 1100. Yhteenlasku suoritetaan täysin normaalina yhteenlaskuna, jossa bitit lasketaan normaalin kaksijärjestelmän mukaisesti yhteen oikealta vasemmalle (kts. TiTo), Huomaa, että tarkistussumman laskennassa mahdollinen ylivuotanut muistibitti (tai bitit, jos yhteenlaskettavia ykkösiä on paljon) lasketaan lopuksi yhteen jo saadun summan kanssa. Tarkistussumma lähetetään sitten komplementtinaan osana viestiä.
- b. Kuinka vastaanottaja havaitsee siirrossa tapahtuneen virheen?
- c. Onko mahdollista, että vastaanottaja ei havaitse yhden bitin virhettä (eli yhden bitin muuttumista nollasta ykköseksi tai päinvastoin)? Entä kahden tai useamman bitin virhettä?

#### 4. TCP-yhteyden kulku ja TCP:n vuonvalvonta

- a. Esitä TCP-yhteyden muodostussanomien, kun A aloittaa segmenttien järjestysnumerot numerosta 56 ja B numerosta 182.
- b. A siirtää ensin 1000 tavun segmentin B:lle, B kuittaa tämän ja lähettää A:lle 800 tavun segmentin, jonka A kuittaa. Esitä vaihdettujen TCP-segmenttien järjestysnumero- ja kuittauskenttien sisällöt.
- c. Esitä TCP-yhteyden purkusanomat, kun A lopettaa lähettämisen.
- d. Koneet C ja D on yhdistetty 200 Mbps:n linkillä. Koneiden välille on muodostettu TCP-yhteys, jolla kone C siirtää hyvin suurta tiedostoa koneelle D. Koneen C lähetysnopeus on 100 Mbps ja kone D kykenee tyhjentämään vastaanottopuskuriaan korkeintaan nopeudella 50 Mbps. Miten tässä voidaan hyödyntää TCP:n vuonvalvontaa?

#### 5. TCP-ruuhkanhallinta

- a. Mihin hidasta aloitusta (slow start) käytetään? Mitä hyötyä siitä on? Voiko siitä olla haittaa? Jos, niin missä tilanteissa? Mitä tarkoitetaan ruuhkan välttämällä (congestion avoidance)?
- b. Käytetään hidasta aloitusta linjalla, jonka kiertoviive (round-trip time) on 200 ms. Verkossa ei ole ruuhkaa ja kuittaukset saapuvat ajoissa. Vastaanottajan ikkuna on 18 KB ja yhdessä segmentissä voidaan lähettää korkeintaan 2 KB (= maximum segment size). Kynnysarvo (threshold) on aluksi 30 KB. Kauanko kestää, ennen kuin voidaan lähettää ensimmäinen täysi ikkunallinen eli 9 segmenttiä ilman kuittauksen odottelua? Kuinka monta segmenttiä siihen mennessä on kaikkiaan lähetetty? Linjan siirtonopeus (transmission rate) on niin suuri, että segmenttien lähettämiseen kuluvalle ajalle ei ole merkitystä.
- c. Kuinka kauan b-kohdassa kestäisi lähetyksikkunalla saavuttaa vastaanottajan ikkunan koko, jos kynnysarvo olisikin aluksi 16 KB? Kuinka monta segmenttiä tässä lähetettiin ennen kuin lähetyksikkunan koko on tuo 18 KB?

#### 6. TCP-ruuhkanhallinta jatkuu

- d. Miten ruuhkanhallinta toimii, jos edellisen tehtävän lähetyksessä 9. lähetetty segmentti katoaa? Miten lähettämistä jatketaan tämän jälkeen? Simuloi pakettien ja kuittausten lähettämiset! Miten ruuhkaikkunan koko käyttäytyy?
- e. Entä jos täysi ikkunallinen on lähetetty, eikä mitään kuittausta saada ajoissa, vaan uudelleenlähetyksajastin (retransmission timer) ehtii laueta. Miten lähettämistä jatketaan tämän jälkeen? Miten ruuhkaikkunan koko käyttäytyy?

☆ Ylimääräinen tehtävä: Viime viikon harjoituksissa professori Sasu Tarkoma lähetti sähköpostia Berkeleyyn yliopistoon. Mitä tapahtuu kuljetuskerroksella? Käy läpi tarvittavien protokollien toiminta ja luonnostelee kuvana kuljetuskerroksen segmenttien rakenne.

☆ WIRESHARK tehtävä: Viime viikolla tarkastellussa kaappauksessa smtp.cap (<http://wiki.wireshark.org/SampleCaptures?action=AttachFile&do=get&target=smtp.pcap>) tapahtuu yhden sähköpostiviestin siirto sähköpostipalvelimelta toiselle. Miten TCP:n kuittaus- ja segmenttinumerot tässä kaappauksessa toimivat? Minkä kokoisia siirretyt segmentit olivat?