

Tietoliikenteen perusteet 2021, syventäviä tehtäviä 4

Nämä tehtävät liittyvät tietoliikenteen-perusteet 2 osaan 3 kuljetuskerros.

1. Tarkastellaan vuorottelevan bitin protokollaa
 - a. Simuloi protokollan automaattien toimintaa. Käy läpi ainakin tilanne, jossa ensin katoaa juuri lähetetty datapaketti ja sen jälkeen lähetettävä ensimmäinen kuittaus katoaa myös. Kannattaa joko kirjata yhteistilat kurssimateriaalin tehtävän kaltaisesti tai piirtää aikajana, jossa automaattit lähettävät viestejä toisilleen. Tähän viestienvaihtosekvenssiin on hyvä merkitä automaattien tilat ja viesti, jotka ne lähettävät ja vastaanottavat.
 - b. Vuorottelevan bitin protokolla olettaa, että kanava voi vain kadottaa sanomia. Näytä, ettei protokolla toimi oikein, jos kanava voi kadottamisen lisäksi monistaa sanomia ja/tai muuttaa sanomien järjestyksiä. Sanomien järjestyksen muuttaminen tarkoittaa sitä, että sanomat saapuvat vastaanottajalle eri järjestyksessä kuin, missä lähettäjä ne lähetti. Tämä tapahtuu yleensä silloin, jos ensiksi lähetetyn sanoman siirto kestää poikkeuksellisen kauan ja lähettäjä ehtii tehdä uudelleenlähetyksen. Tämä on helpointa havainnollistaa itselle viestienvaihtosekvenssillä. Etsi siis sellainen tilanne, jossa joko automaattit lukkiutuvat eli molemmat vain odottavat saapuvaa viestiä tai tilanne, jossa numerointi menee sekaisin siten, että yksi viesti jää päätyttyä perille, mutta automaattien mielestä numerointi toimii oikein. On myös mahdollista saada tilanne, jossa sama viesti (=data sisältö) hyväksytään kahteen kertaan, ja numerointi toimii silti automaattien mielestä oikein.
2. Ovatko seuraavat TCP:n toimintaan liittyvät väittämät oikein vai väärin. Perustele vastauksesi.
 - a. Kone A lähettää koneelle B ison tiedoston TCP-yhteyden kautta. Koneella B ei ole mitään lähetettävää A:lle. B ei lähetä myöskään kiittäuksia, koska B ei voi liittää niitä datapaketien yhteyteen.
 - b. Vastaanottajan ikkunan (receive window, rcwd) koko ei koskaan muutu TCP-yhteyden aikana.
 - c. Oletetaan, että A lähettää B:lle suuren tiedoston TCP-yhteyden kautta. A:n lähettämien, mutta kiittaamattomien tavujen määrä ei voi ylittää vastaanottajan ikkunan (eli vastaanottopuskurin) kokoa.
 - d. Oletetaan, että A lähettää suuren tiedoston B:lle TCP-yhteyden kautta. Jos tietyssä vaiheessa segmentin järjestysnumero on m , niin seuraavan segmentin järjestysnumero on $m + 1$.
 - e. Oletetaan, että viimeisin SampleRTT:n arvo TCP-yhteydessä on 1 s. Tällöin TimeoutInterval:n nykyinen arvo on varmasti ≥ 1 s.
3. Kone A ja kone B kommunikoivat käyttäen TCP-yhteyttä. B on jo saanut A:lta kaikki tavut 0..126. Oletetaan, että A lähettää seuraavaksi kaksi segmenttiä perä jälkeen. Ensimmäisessä segmentissä on 80 tavua ja toisessa 40. Ensimmäisessä segmentissä segmenttinumero (segment number) on 127, lähdeportti (source port number) 302 ja kohdeportti (destination port number) 80. B lähettää kiittauksen aina kun se saa segmentin A:lta.
 - a. Mikä on toisessa segmentissä segmenttinumero, lähdeportti ja kohdeportti?

- b. Jos ensimmäinen segmentti saapuu ennen toista segmenttiä, niin mitkä ovat ensiksi saapuneen viestin kuittausviestissä kuittausnumero (acknowledgment number), lähdeportti ja kohdeportti?
- c. Jos toinen segmentti saapuu ennen ensimmäistä segmenttiä, niin mitkä ovat nyt ensiksi saapuneen viestin kuittausviestissä kuittausnumero, lähdeportti ja kohdeportti?
- d. Oletetaan, että A:n lähettämät segmentit saapuvat lähetysjärjestyksessä B:lle. Ensiksi saapuneen viestin kuittaus katoaa ja toisen viestin kuittaus saapuu A:lle vasta ensimmäisen kuittauksen odotukseen liittyneen ajastimen laukeamisen (timeout) jälkeen. Piirrä kaavio, josta käy ilmi nämä segmentit, niihin liittyvät kuittaukset sekä kaikki muut segmentit ja kuittaukset, jotka liittyvät tähän skenaarioon. Voit olettaa, että muut viestit eivät katoa. Merkitse huolellisesti kaaviossa jokaiselle segmentille sen järjestysnumero (segmenttinumero) ja kuinka monta tavua tässä segmentissä kulkee. Kuittausviestille on vastaavasti merkittävä kuittausnumero.

4. Tarkastellaan TCP Reno -menetelmää ruuhkanhallintaan. Alla olevassa kuviossa näkyy menetelmän eräs käyttäytyminen TCP-yhteyden aikana. Vastaa perustellusti seuraaviin kysymyksiin.

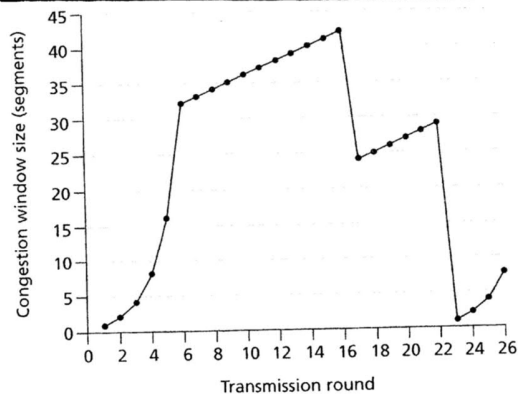


Figure 3.58 ♦ TCP window size as a function of time

- a. Millä aikaväleillä TCP:n hidas aloitus on toiminnassa?
 - b. Millä aikaväleillä TCP:n ruuhkanvälttely on toiminnassa?
 - c. Huomaako TCP segmentin kadonneen 16.kierroksella ajastimen vai kolmen toistokuittauksen perusteella?
 - d. Huomaako TCP segmentin kadonneen 22.kierroksella ajastimen vai kolmen toistokuittauksen perusteella?
5. Jatketaan äskeisen tehtävän kuva käsittelyä, mutta tarkastellaan nyt TCP:n kahden tilan hitaan aloituksen ja ruuhkavälttelyn välisen kynnyсарvon (slow start threshold, ssthresh) muutoksia. Muistathan, että kynnyсарvo ei läheskään aina ole ruuhkaikkunan sen hetkinen koko, vaan vain raja-arvo hitaan aloituksen ja ruuhkavälttelytilojen välissä. Arvioi tämän arvon muutoksia edellisen tehtävän kuvan perusteella.
- a. Mikä on kynnyсарvo ensimmäisellä kierroksella?
 - b. Mikä on kynnyсарvo 18. kierroksella?
 - c. Mikä on kynnyсарvo 24. kierroksella?
 - d. Millä kierroksella lähetetään 70. segmentti?

- e. Oletetaan, että paketin katoaminen huomataan 26. kierroksella kolminkertaisen toistokuittauksen perusteella. Mikä on tällöin ruuhkaikkunan koko ja kynnyсарvo?
6. UDP ja TCP käyttävät ns. yhden komplementtia tarkistussummissaan.
- a. Laske 8 bitin mittainen UDP-tarkistussumma seuraavalle kolmen tavun mittaiselle sanomalle: 01010011, 01100110, 01110100. Mikä on yhden komplementti näiden kolmen tavun summalle. Näytä kaikki laskennan välivaiheet.
 - b. Miksi käytetään nimenomaan yhden komplementtia eikä summaa sellaisenaan?
 - c. Kuinka vastaanottaja havaitsee siirrossa tapahtuneen virheen?
 - d. Onko mahdollista, että vastaanottaja ei havaitse yhden bitin virhettä (eli yhden bitin muuttumista nolasta ykköseksi tai päinvastoin)? Entä kahden tai useamman bitin virhettä?

☆ Ylimääräinen WIRESHARK: 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?

☆ **Kiinnostuneille tiedoksi.** Helsingin Sanomissa on 16.2.2013 on julkaistu hyvä artikkeli siitä, miten googlen hakukoneelle annettu haku seikkailee tietoverkoissa. Bibliografiset tiedot: Olavi Koistinen: Uskomaton hakukone. Helsingin sanomat, 16.2.2013, D6-D9. Huomaa, että artikkeli on jo useita vuosia vanha, joten palvelimien sijainnit ynnä muut yksityiskohdat ovat jo ehtineet muuttua, mutta periaatteet ovat edelleen samat.