

Tietoliikenteen perusteet, 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 21.11. - pe 23.11.2012

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

- a. 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 häiriöpurske tuhoaa datakehysten $D(n+1)$ ja sitä edeltävän kehysten $D(n)$ ACK-kuittauksen sekä vielä sitä seuraavan kuittauksen.

- a. Näytä kaaviokuvana, kuinka tilanteessa toimii
 - o paluu N:ään (Go-Back-N),
 - o valikoiva toisto (Selective Repeat)
- b. Olisiko NAK-kuittauksen käytöstä mitään hyötyä kummankaan protokollan tilanteessa? Mitä haittaa NAK-kuittauksesta voisi olla?
- c. 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 ns. yhden komplementin 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.
- b. Kuinka vastaanottaja havaitsee siirrossa tapahtuneen virheen?
- c. Onko mahdollista, että vastaanottaja ei havaitse yhden bitin virhettä (eli yhden bitin muuttumista nolasta 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? Linjan siirtonopeus (transmission rate) on niin suuri, että segmenttien lähettämiseen kuluvalle ajalle ei ole merkitystä.
- c. Kun täysi ikkunallinen on lähetetty, siihen ei saadakaan kuittausta ajoissa, vaan uudelleenlähetyksajastin (retransmission timer) ehtii laueta. Miten lähettämistä jatketaan tämän jälkeen? Onko tässä eroa sillä, käytetäänkö TCP:n versiota Tahoe vai Reno?
- d. Miten uudelleenlähetyksajastimen arvo määritetään? Miksi on tärkeää, että uudelleenlähetyksajastimen arvo on oikein? Mitä vaikutusta on sillä, että arvo on i) liian pieni tai ii) liian suuri?

6. Viime viikon harjoituksissa professori Sasu Tarkoma lähetti sähköpostia Berkeleyn yliopistoon. Mitä tapahtuu kuljetuskerroksella? Käy sanallisesti läpi tarvittavien protokollien toiminta ja luonnostele kuvana kuljetuskerroksen segmenttien rakenne. (Katso segmentin rakenne kalvoista tai kirjan 6. edition sivulta 260 (4ed. s. 270)).

☆ Ylimääräinen tehtävä TCP:n toiminnasta kiinnostuneille. Tätä ei käsitellä harjoituksissa:

Oleta, että TCP-lähteellä on aina lähetettäviä segmenttejä. Vastaanottaja lähettää ACKin heti segmentin saavuttua. Oleta, että TCP-yhteyden ikkunakoko on alussa 1 segmentti ja että RTT on 200 ms. Segmentit numero 15, 30 ja 52 hukkuvat, ja ne täytyy lähettää uudelleen. Tutki ja vertaa, mitä käy ensimmäisten 16 RTT:n ajan. Piirrä tapahtumien kulku (ruuhkaikkuna ja lähetetyt paketit), kun

(a) TCP-lähde käyttää slow start/congestion avoidance TCP-algoritmia, ja timeout-arvo on 400 ms. Exponentiaalinen peräytyminen on käytössä.

(b) TCP-algoritmina on ideaalisen additiivisen lisäyksen ja multiplikatiivisen vähennyksen

algoritmi, jossa ruuhkaikkunaa kasvatetaan yhdellä segmentillä joka RTT:tä kohden ja ruuhkaikkuna puolitetaan pakettihukan tapahduttua. Oleta, että käytössä on täydellinen timeout-mekanismi, joka huomaa pakettihukan, kun tasan 1 RTT on kulunut paketin lähettämisestä.

TKK:n tietokoneverkot kurssin tenttitehtävä 7 (10.5.2002) (linkki: <http://www.netlab.tkk.fi/opetus/tentit/188t2002-05.pdf>)