

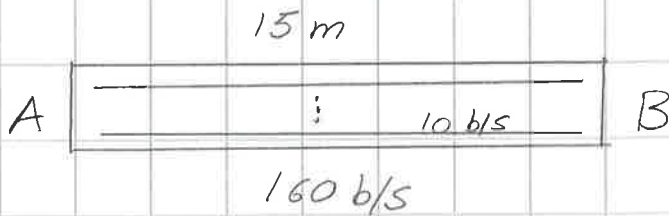
# Harj. 3



Osoitteiden saaminen kestää  
 ~~$kD_1$~~   $kD_1 + (n-k)D_2$ .

Tämän jälkeen yhden objektin haku ja siihen liittyvä TCP-yhteyden perustaminen vie ajan  $2RTT_0$ . Eli yhteensä aikaa kuluu  $kD_1 + (n-k)D_2 + 20RTT_0$ .

2.



Data  $L = 200\,000\text{ b}$ , kontroll  $K = 100\text{ b}$   
 Objekti  $O = 200\,000\text{ b}$ .

Jokainen objekti mahtuu yhteen datapakettiin. Olkoon  $T_p$  etenemisviive yhteen suuntaan. Tarkastellaan ensin ~~riippuvuudesta~~ haku ~~...~~ säilyttävin yhteyksin.

Ensiksi täytyy päättää kontrollipaketin lkm. Sen jälkeen lasketaan aika, joka kuluu pääsivun siirtoon, ja sen jälkeen aika, joka kuluu 10 muun objektin siirtoon.

Kontrollipaketteja on 3 kpl (tcp-yhteys + http-pyyntö). Siis kokonaisaika yhtä ~~kanava~~ yhteyttä käyttäen on

$$100/160 + T_p + 100/160 + T_p + 100/160 + T_p + 200\,000/160 + T_p + 10 * (100/160 + T_p + 200\,000/160 + T_p) = 13\,758 + 24 * T_p.$$

↑

http-request,

tcp on jo muodostettu

Lasketaan seuraavaksi rinnakkainen siirto ei-säilyttävin yhteyksin. Objektit saapuvat nyt rinnakkain, joten riittää laskea pääsivun ja yhden objektin hakemiseen kuluva aika. Saadaan

$$100/160 + T_p + 100/160 + T_p + 100/160 + T_p + 200\,000/160 + T_p + 100/16 + T_p + 100/16 + T_p + 100/16 + T_p + 200\,000/16 + T_p = 13\,771 + 8 T_p.$$

$$T_p \approx 0.05 \mu s, \text{ jos } s = 300 \times 10^6 \text{ m/s.}$$

Eli menetelmillä ei ole eroa.

- 4.
- i) Käyttäjän kone aloittaa DNS-kyselyn.
  - ii) Selain poimii nimen "www.somesite.com" URL:stä ja luovuttaa sen DNS:lle.
  - iii) Asiakas-DNS lähettää kyselyn DNS-palvelimelle.
  - iv) Asiakas-DNS saa jossain vaiheessa IP-osoitteen.
  - v) Kun IP-osoite on tullut, asiakas voi aloittaa TCP-yhteyden.

5. Kontrolliyhteyttä käytetään lähettämään kontrollitietoa, kuten käyttäjätunnus, salasana, hakomistokomennot, get, put. Datayhteys on varsinaista tiedostonsiirtoa varten.

6. Käytetään seuraavia kaavoja:

$$D_{cs} = \max \{ NF/\mu_s, F/d_{min} \},$$

$$D_{p2p} = \max \{ F/\mu_s, F/d_{min}, NF/(\mu_s + \sum_{i=1}^N \mu_i) \}.$$

Nyt  $F = 15 \text{ Gb}$ ,  $\mu_s = 30 \text{ Mb/s}$ ,  
 $d_{min} = d_i = 2 \text{ Mb/s}$ .

Saadon taulukko:

	N			
	10	100	1000	
$\mu$ 300	7680	51200	512000	Asiakas-palvelin
700	7680	51200	512000	
2	7680	51200	512000	

7680	25904	47550	Peer-to-peer
7680	15616	21525	
7680	7680	7680	

Laskuissa on käytetty arvoja

$$1 \text{ Gb} = 1024 \text{ Mb}$$

$$1 \text{ Mb} = 1024 \text{ Kb}$$

7. a)  $\mu_s/N \leq d_{\min}$ . Aika  $NF/\mu_s$ ?

Palvelin lähettää yhtäaikaan nopeudella  $\mu_s/N$ . Ehdon perusteella asiakkaat voivat ottaa vastaan tällä nopeudella. Siten koko tiedosto on asiakkaalla ajan  $F/(\mu_s/N) = NF/\mu_s$  kuluttua.

7.

b)  $u_s/N \geq d_{\min}$ .

Nyt palvelin ei voi lähettää nopeammin kuin  $d_{\min}$  hitaimmalle asiakkaalle. Siten hitain saa tiedoston ajassa  $F/d_{\min}$ .

c) Tämä seuraa kohdista a ja b sekä yleisestä kirjan kaavasta  $D_{CS} \geq \max\{NF/u_s, F/d_{\min}\}$ .

30. Selvitetään ensin juuripalvelimet. Nämä saadaan selville unix-komennoilla

```
nslookup  
> type=ns  
> .
```

(tai verhoista)

saadaan lista nimipalvelimista. Otetaan esim. ~~a.fi~~ seuraavaksi d.root-servers.net. Seuraavaksi etsitään tietty juuripalvelin:

```
dig +norecurse @d.root-servers.net any  
melkkari@melkki.cs.helsinki.fi
```

Otetaan vastauksista esim. a.fi.

jonka jälkeen uudestaan

dig +norecurse @ a.fi any melkki.cs.helsinki.fi

Myös komennolla

dig +trace

saa selville, ympäristöstä riippuen, nimi-  
palvelimia.