

Övning 3, Datorkommunikation

Problem 1

Är följande påståenden sanna eller falska?

- Processer på två olika datorer kan kommunicera med varandra genom att utbyta meddelanden (messages) över nätverket.
- En klient-server-tjänst är fullständigt säker.
- En socket är ett hårdvarugränssnitt genom vilket en process sänder meddelanden till nätverket och också tar emot meddelanden från nätverket.
- I tjänster som internettelefoni och multimedia kan man aldrig tolerera att data inte kommer fram.
- När man utvecklar en ny applikation måste man bestämma sig för om den ska använda TCP eller UDP.

Problem 2

Nedan finns ett svar på en http-request som har skickats från en server som svar på ett http GET-meddelande. Besvara frågorna nedan.

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)
<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46
GMT<cr><lf>ETag: "526c3-f22-a88a4c80"<cr><lf>Accept-
Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>
Keep-Alive: timeout=max=100<cr><lf>Connection:
Keep-Alive<cr><lf>Content-Type: text/html; charset=
ISO-8859-1<cr><lf><cr><lf><!doctype html public "-
//w3c//dtd html 4.0 transitional//en"><lf><html><lf>
<head><lf> <meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"><lf> <meta
name="GENERATOR" content="Mozilla/4.79 [en] (Windows NT
5.0; U) Netscape]"><lf> <title>CMPSCI 453 / 591 /
NTU-ST550A Spring 2005 homepage</title><lf></head><lf>
<much more document text following here (not shown)>
```

- Kunde servern hitta filen som klienten ville ha?
- Vid vilken tid skickades svaret från servern?
- När ändrades dokumentet som efterfrågades senast?
- Hur många oktetter finns det i det returnerade dokumentet?
- Vilka är de fem första oktetterna (oktett = byte) i filen som returneras?

Problem 3

Du klickar på en länk i en browser. IP-adressen för servern finns inte cachad i din dator så en DNS-slagning måste göras. n stycken DNS-servrar måste besökas innan din dator får IP-adressen till servern. k av dessa besök ger en RTT (Round Trip Time) på D_1 per besök och att de övriga har en RTT på D_2 per besök. Webbsidan innehåller m mycket små bilder. RTT för att hämta HTML-filen och var och en av dessa små bilder (som ryms i ett TCP-paket) är RTT_0 . Man använder inte persistent http, och parallella TCP-förbindelser är inte tillåtna. Om transmissionstiden för alla paket som skickas kan försummas, hur lång tid tar det då innan webbsidan kan visas?

Problem 4

Antag att tre DNS-servrar besöks i problem 3 och att $k = 2$ och $m = 5$. HTML-filer och de fem bilderna finns alla på samma server. Hur lång tid tar det att hämta hela webbsidan om vi använder:

- Non-persistent http som inte tillåter parallella TCP-förbindelser
- Non-persistent http som tillåter parallella TCP-förbindelser
- Persistent http där man tillåter parallella hämtningar

Problem 5

Vi har en 15 meter lång länk med kapaciteten 160 kbps. N stycken parallella förbindelser får var och en $1/N$:tedel av länkens kapacitet. Vi laddar ner en webbsida över denna länk som förutom HTML-filen består av tio objekt. Både HTML-filen och filerna som innehåller objekten ryms i paket som är 4000 bitar långa. Paket utan data (t ex ACK:ar) innehåller 160 bitar och paket med GET-request innehåller 640 bitar.

- Vi laddar ner webbsidan med non-persistent http som använder parallella TCP-förbindelser. Hur lång tid tar det?
- Nu använder vi stället persistent http med parallella hämtningar. Hur lång tid tar det?
- Jämför svaren i deluppgift a och b. Gör man någon vinst med att använda persistent http i stället för non-persistent?

Problem 6

En browser ska visa sidan www.somesite.com/index.html. Då måste först browsern med hjälp av DNS-applikationen översätta denna adress till en IP-adress. Skriv ner alla steg som browsern och DNS-applikationen i datorn måste gå igenom innan browsern har fått IP-adressen.