



# Försättsblad till skriftlig tentamen vid Linköpings Universitet

(fylls i av ansvarig)

<b>Datum för tentamen</b>	2009-03-14
<b>Sal</b>	KÅRA
<b>Tid</b>	8-12
<b>Kurskod</b>	TDTS04
<b>Provkod</b>	TEN1
<b>Kursnamn/benämning</b>	Datornät och distribuerade system
<b>Institution</b>	IDA
<b>Antal uppgifter som ingår i tentamen</b>	10
<b>Antal sidor på tentamen (inkl. försättsbladet)</b>	9
<b>Jour/Kursansvarig</b>	Juha Takkinen
<b>Telefon under skrivtid</b>	0731-500 393
<b>Besöker salen ca kl.</b>	9.15
<b>Kursadministratör (namn + tfnr + mailadress)</b>	Madeleine Häger Dahlqvist 013-28 2360 madha@ida.liu.se
<b>Tillåtna hjälpmedel</b>	miniräknare fyra A4-sidor med handskrivna anteckningar
<b>Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)</b>	

## **Skriftlig tentamen i kursen TDTS04 Datornät och distribuerade system 2009-03-14 kl. 8–12**

### **Lokal**

KÅRA (Skrivsalen i Kårallen).

### **Tillåtna hjälpmedel**

Du får ha dels en *miniräknare*, med minnena tömda på kursrelaterad info, och dels *4 sidor linjerade A4-blad* med handskrivna anteckningar. inkl. figurer, med en rad text per linje inom en marginal på ca 3 cm från alla kanter.

### **Resultat**

Resultatet publiceras inom tolv arbetsdagar i LADOK.

### **Poänggränser**

Du kan få maximalt 40 poäng. För att få godkänt, betyg 3, krävs minst 20 poäng totalt och med minst 12 poäng i datornätsdelen och 8 poäng i distribuerade system-delen. För betygen 4 och 5 krävs 28 respektive 36 poäng totalt, förutom minst godkänt i respektive del.

### **Lärarjour**

Under tentamenstiden finns möjlighet att ställa frågor och få förtydliganden från Juha Takkinen, tel. 0731-500 393, som besöker tentamenslokalen *kl. 9.15* OBS! Enda besöket den här gången, så läs igenom frågorna till dess.

### **Instruktioner**

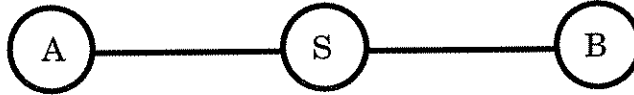
Om du gör antaganden utöver det som står i uppgiften, skriv ner dem tydligt, men du får inte förändra uppgiften. Du kan svara på svenska eller engelska. Ha samma ordning på dina svar som på frågorna. Skriv gärna svar direkt på tentamenssidorna men då måste du även skriva ditt AID på dem vid inlämning.

*"Wow! They've got the Internet on computers now". —Homer Simpson*

*Lycka till!*

**Del 1: Datornät**

1. Protokoll och nätgrunder. Antag att noderna A och B är sammankopplade via en switch S och en 10-Mbps-länk (se figur nedan). Vardera länk har en fördröjning på 20  $\mu$ s. Switchen S arbetar enligt store-and-forward-principen, d.v.s. den återsänder ett paket 35  $\mu$ s efter det att den har fått paketet.

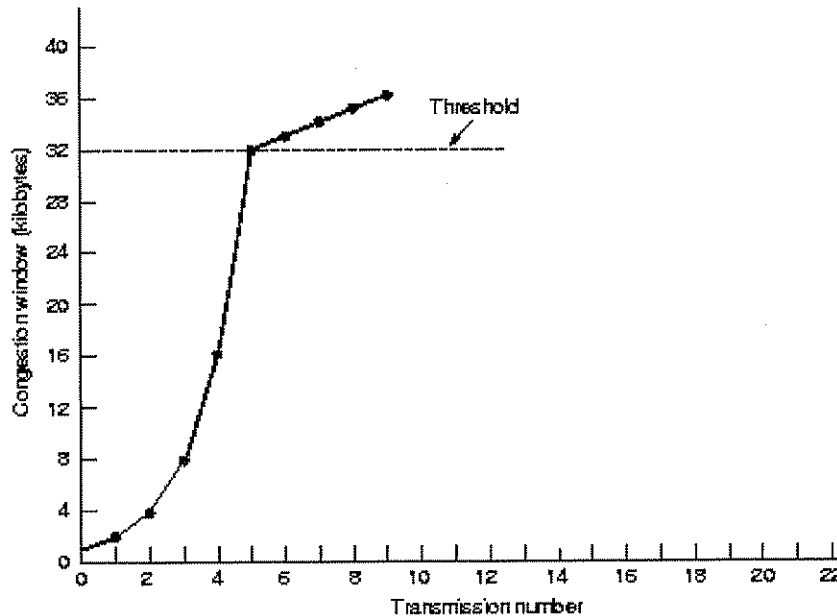


- Beräkna den totala tiden för att skicka 10 000 bitar från A till B som ett enda paket. (1,5 p.)
  - Beräkna den totala tiden för att skicka 10 000 bitar från A till B som två lika stora paket direkt efter varandra. (1,5 p.)
  - Hur stor är pipe:n, d.v.s. hur många bitar får plats i en länk? (1 p.)
2. Applikationsprotokoll. Nedanstående sekvens av ASCII-tecken erhöles vid en spårning av kommunikationen från en webbläsare som gjorde ett HTTP GET-anrop till en server. Tecknen <cr> och <lf> är symboler för radbrytnings- och radmatningstecken.

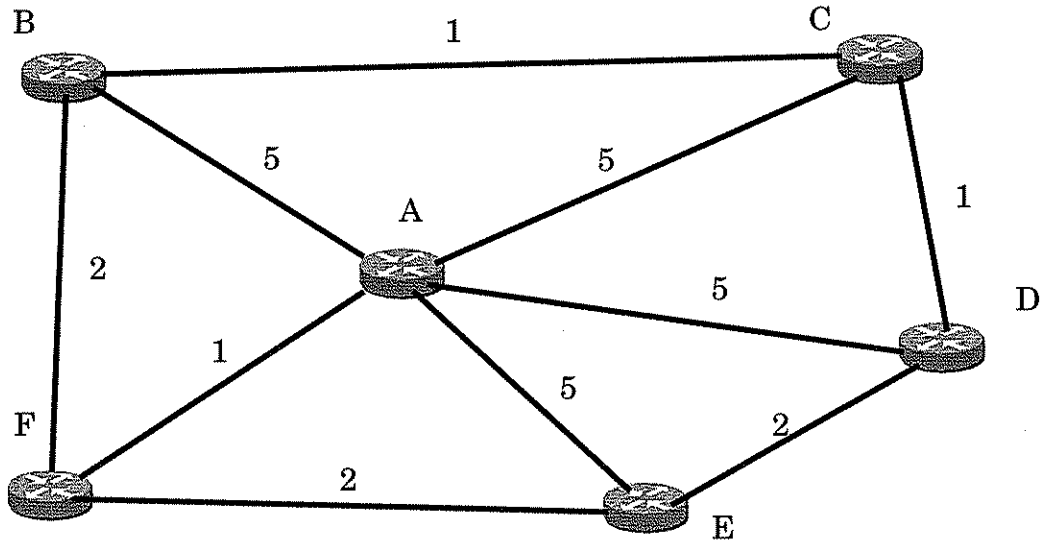
```

GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml,application/xml,application/xhtml+xml,text
/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-
Encoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
  
```

- Vilken URL var det webbläsaren efterfrågade? Förklara hur du kom fram till svaret. (1 p.)
  - Begärde webbläsaren en persistent uppkoppling av servern? Motivera ditt svar och definiera även begreppet *persistent uppkoppling*. (2 p.)
  - Vad är IP-adressen för klienten? (1 p.)
3. TCP
- Antag att stockningsfönstret (eng. congestion window) är 35 KB stort och att threshold är 32 KB vid sändningsomgång 9 (se figuren överst på nästa sida). Visa vad som sker när TCP:s stockningskontroll (eng. congestion control) upptäcker tre dubletter av en ACK. (1 p.)



- b. Antag att en TCP-uppkoppling ska användas för att överföra 5000 byte med data. Det första TCP-segmentet har vid handskakningen fått sekvensnummer 501. Vad blir sekvensnumret för resterande segment, om varje segment är 1000 byte stort? Vilket nummer har ACK:en för det sista segmentet? (1 p.)
- c. En egenskap som UDP saknar är *rättvisa* (eng. fairness), som TCP i allra högsta grad har. Förklara vad som menas med detta begrepp. Redogör även för hur en modern webbläsare kan missbruka denna egenskap hos TCP. (2 p.)
4. IP
- a. Förklara i detalj hur traceroute-programmet utnyttjar IP- och ICMP-protokollen för att mäta och kartlägga vägen från A till B genom ett nät. (2 p.)
- b. Antag att det finns tre routrar mellan källa och destination och att ingen fragmentering sker. Hur många gränssnitt (eng. interface) kommer ett IP-paket att passera? Hur många dirigeringstabeller (eng. forwarding tables) behöver skapas av en routningsalgoritm för att förflytta paketet från källa till destination? (1 p.)
- c. Definiera begreppen *nätprefix* (eng. prefix) och *subnät* (eng. subnet). (1 p.)
5. Routning
- a. Fyll i nedanstående tabell med hjälp av Dijkstras algoritm och beräkna den kortaste vägen från nod A till alla övriga noder i nätverket i figuren nedan. Om flera kolumner skulle ha samma värde på en rad, välj det värde som finns längst till vänster på raden. (3 p.)

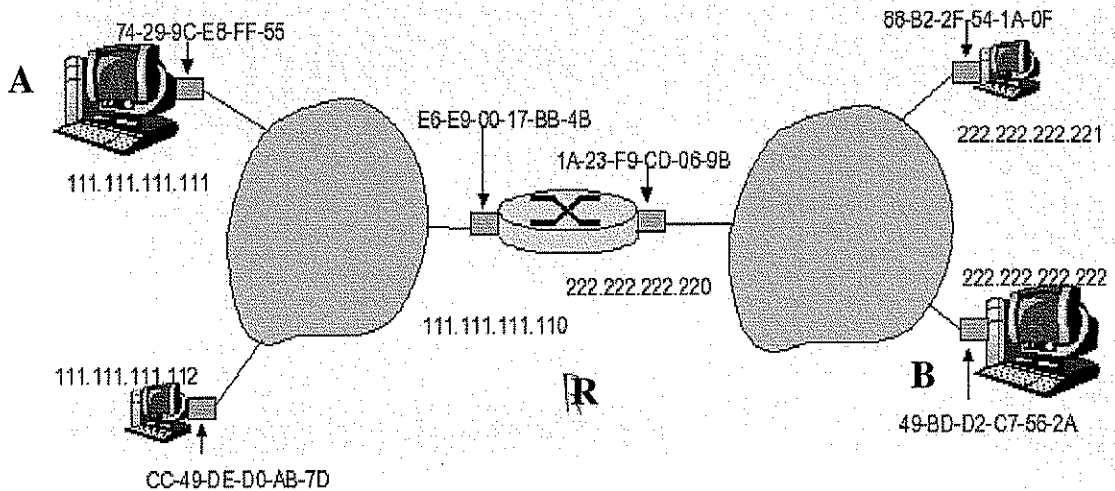


Steg	N'	D(B), p(B)	D(C), p(C)	D(D), p(D)	D(E), p(E)	D(F), p(F)
0	A					
1						
2						
3						
4						

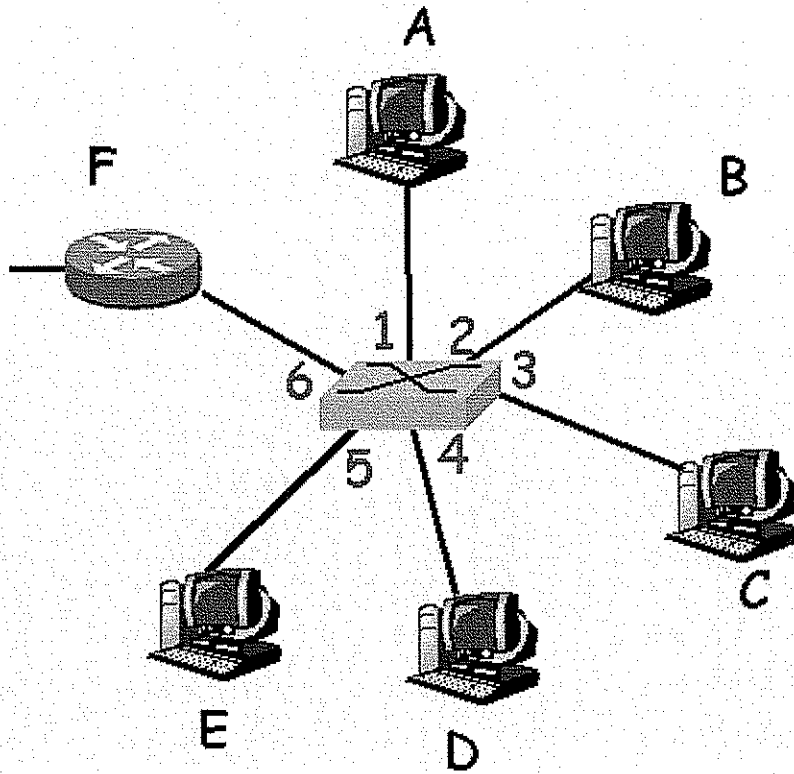
b. Definiera begreppet *BGP-rutt* (eng. BGP route) och ange hur det är relaterat till nätprefix (som omnämndes i fråga 4c ovan). (1 p.)

6. Lokala nät

a. Antag att nod A ska skicka ett IP-paket till nod B via routern R (se figur nedan). Antag också att samtliga noders ARP-tabeller är uppdaterade. Beskriv vilka steg som behöver utföras för att paketet ska kunna levereras till rätt MAC-adress. (2 p.)



- b. Antag att switchen i nedanstående figur är självlärande (eng. self-learning). Antag också att switchtabellen är tom från början innan följande sker i nätet: (i) A börjar med att skicka en ram adresserad till D. (ii) D svarar med en ram adresserad till A. (iii) C skickar sedan en ram adresserad till D. (iv) D svarar slutligen med en ram till C. Beskriv vilka länkar som används i respektive fall (i)–(iv). Motivera dina svar. (2 p.)



**Del 2: Distribuerade system****7. Distribuerade systemgrunder och kommunikationsmodeller**

- a. Napster använde sig av en centraliserad server med en katalog över filer och användare. Denna arkitektur har i nyare p2p-system ersatts av andra arkitekturer. Redogör för två nackdelar som en Napsterliknande, centraliserad arkitektur har för lokalisering av filer. (2 p.)
- b. En TCP-socket använder sig av följande primitiver: accept, bind, close, connect, listen, receive, send och socket. Beskriv hur en klient och en server använder dessa primitiver för att skapa en uppkoppling via socketen, sända data och sedan avsluta uppkopplingen. (2 p.)

**8. Objektbaserade distribuerade system**

- a. Du har fått i uppdrag att utveckla ett Corbaprogram som kan användas för att kontrollera om kursböckerna finns att köpa i bokaffärerna på Campus Valla. Programmet ska användas av studenter (klienter) för att se efter i den centrala databasen (servern) om kursböckerna har kommit och i så fall i vilka bokaffärer som de finns att köpa.
- Servern har en lista på boktitlar som är möjliga att beställa.
  - För varje boktitel så lagrar servern hur många kopior som finns och i vilka bokaffärer dessa finns (Bokakademien, Akademibokhandeln).
  - När en klient anropar servern via lookup() så måste den ange ISBN-numret, t.ex. "ISBN 0-13-088893-1", för den bok som den vill ha reda på.
  - Servern svarar på anrop med informationen att boken ifråga antingen finns tillgänglig eller inte. Om boken finns att köpa så specificerar servern även hur många bokaffärer som har boken samt en lista med namnen på bokaffärerna.

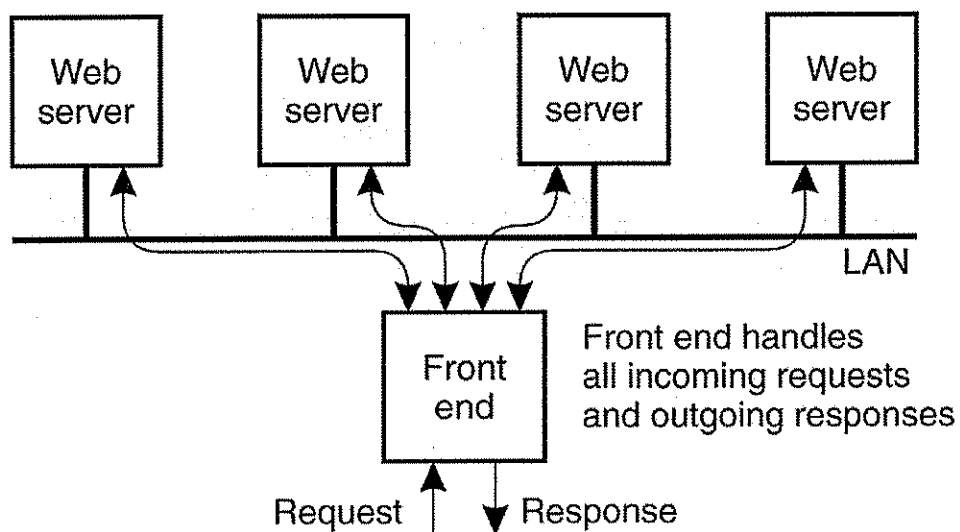
Specificera ett gränssnitt ISBNLookup.idl för ovanstående distribuerade system genom att fylla i nedanstående rader med hjälp av IDL (interface description language). Använd den datatyp som mest överensstämmer med kraven ovan och använd parameternamn som tydligt anger vad du menar. (2 p.)

```
module ISBNLookupApp {
    interface ISBNLookup {
        _____ lookup(_____,
            _____,
            _____);
    };
};
```

IDL Construct	Java Construct
module	package
interface (non-abstract)	signature interface and an operations interface, helper class, holder class
interface (abstract)	signature interface, helper class, holder class
constant (not within an interface)	public interface
boolean	boolean
char, wchar	char
octet	byte
string, wstring	java.lang.String
short, unsigned short	short
long, unsigned long	int
long long, unsigned long long	long
float	float
double	double
fixed	java.math.BigDecimal
enum, struct, union	class
sequence, array	array
exception	class
readonly attribute	accessor method
readwrite attribute	accessor and modifier methods
operation	method

## 9. Webbaserade distribuerade system

- a. Nämn två fördelar med att använda en webbproxycache (hierarkisk eller distribuerad). (2 p.)
- b. Du har blivit kontaktad av en populär webbplats för att lösa ett problem med en flaskhals de har hos frontend-servern (se figur nedan). Redogör för *en* möjlig lösning av problemet. (2 p.)





## 10. Koordinationsbaserade distribuerade system

- a. Fyll i det som saknas i nedanstående klassificering av olika koordinationsmodeller. Ange sedan ett exempel på vardera typ av system, inklusive de redan ifyllda Direct och Mailbox. (2 p.)

		Temporal	
		Coupled	Decoupled
Referential	Coupled	Direct	Mailbox
	Decoupled		

- b. Antag ett koordinationsbaserat distribuerat system av typen Jini/ Javaspace som är replikerat över tre noder i ett tillförlitligt broadcastnät (se figur nedan). Redogör för stegen som nod A utför när denna ska göra WRITE på en tupel A. Redogör sedan för stegen som nod B utför när denna ska göra READ på en tupel B. (2 p.)

