WINTER 2007

PROGRAMME CODE/TITLE
SCHDF0018 / HIGHER DIPLOMA IN COMPUTER SCIENCE EXAMINATION
ARBDF0015 /THIRD YEAR ARTS EXAMINATION
SCBDF003 / SCBDF0015 / THIRD YEAR SCIENCE & B.Sc. (GENERAL) DEGREE EXAMINATION

Networks and Internet Systems
COMP30040

Head of School Dr Joe Carthy
Examiner Dr John Murphy

Time Allowed: 2 Hours

Instructions for Candidates

Answer Question 1 (COMPULSORY) and any TWO of Questions 2, 3, 4 or 5.

All questions carry equal marks.

READ EACH QUESTION CAREFULLY.

Instructions for Invigilators

Loose Rough Work sheets are not to be distributed or used.

Question 1

(1-a) *Connection-oriented:* modelled on the telephone system.

1. establish a connection
2. use the connection (for data transfer)
3. release the connection

Essential feature: sender pushes objects (e.g. bits, packets) in at one end of the connection, and receiver takes them out *in the same order* at the other end.

Connectionless: each message is sent independently of any other messages going from the same sender to the same receiver: modelled on the postal service.

Essential features: each message must *include the receiver's address*, and messages can be received in a *different order* to the order in which they were transmitted.

(1-b) Address Resolution Protocol (ARP):

- each host maintains an *ARP cache* of mappings between IP addresses and physical addresses
  - ARP cache entries ***time out*** after (typically) 15 minutes
- if destination's IP address not in sending host's ARP cache, sending host broadcasts *ARP query* on local network
  - ARP query asks for physical address for destination's IP address
  - ARP query includes sending host's IP address and physical address (so other hosts can enter this in their ARP caches)
- if destination is a host on local network, it sends its physical address to sending host in an *ARP reply*if destination not on the local network, ARP server replies with its own physical address: *proxy ARP*

(1-c) Ethernet CSMA/CD – try to reduce the ***likelihood*** and ***effects*** of a collision:

- CSMA: a node wishing to transmit must first listen to the channel (e.g. by measuring the channel's voltage level). If the channel is busy, some other node is transmitting and our node must wait until it detects that the channel is idle. When the channel is determined to be idle, our node can transmit.
- CD: during transmission, our node listens to the channel and if another transmission is detected (e.g. higher voltage level than expected for one transmission), all nodes involved in the collision stop transmitting immediately. Each node then computes a randomly sized time interval, waits for that amount of time, and begins the transmission attempt again.
- Problem: even with Carrier Sensing before transmission, collisions can still occur because it takes non-zero time for signal to propagate along the channel. If the propagation time along the length of the channel is denoted by τ, can show that the ***worst-case collision detection time*** is (approximately) 2τ.
- this shows why there is a ***minimum frame length*** in CSMA/CD systems: all frames must take more than 2τ to transmit (by "padding" the information field, if necessary).

(1-d) propagation delay = (300)/(300,000,000) = 1 millisec
packet transmission time = (2,500)/(100,000,000) = 0.025 millisec

therefore number of packets in transit = (1)/(0.025) = **40 packets**.

Question 2

(2-a) Using the formula **`throughput = 1/(TRANSF + 5.4×PROP)`**,

1. If the length of the channel is increased, then PROP decreases, therefore throughput **decreases**.

2. If the average frame length is decreased, then TRANSF increases, therefore throughput **increases**.

(2-b) ***Datagram packet-switching:***

- Each packet is treated individually within the network, so successive packets may follow different routes through the network.Each packet contains the receiver's address and a sequence number (so that receiver can put them into correct order).Network nodes are routers, which have routing tables telling them which output link to use for each possible destination.No connection set-up needed.
- Flexible routing possible (e.g. if a router crashes).
- Network resources are *not shared at the same time:* each packet monopolises a link during its transmission, after which the link is available for other packet transmissions.
- Ideal for *short-lived* bursty traffic; less suitable for *long-lived* &/or *interactive* bursty traffic.

Virtual circuit packet-switching:

- A route is set up in the network between sender and receiver (by making appropriate entries in the routing tables).
    - Resources may or may not be reserved for this route. If resources need to be reserved and are not available, the connection request is blocked.
    - Each packet contains its virtual circuit identifier.
    - Routers have routing tables telling them which output link to use for each established virtual circuit.
- Connection set-up required, which may involve significant delay.
- Network resources are *not shared at the same time:* each packet monopolises a link during its transmission, after which the link is available for other packet transmissions.

Major differences:

- Virtual circuits require less work at intermediate routers than datagrams: given a packet's input link and virtual circuit identifier, the router can look up its routing table to find the output link.
- Virtual circuits are not as robust to network problems as datagrams
- Virtual circuits have to be set up in advance and torn down
- Resources can be reserved for virtual circuits

Question 3

(3-a) **X = 2, Y = 2, Z = 3, W = 2**

(3-b) TRANSF = 400 microseconds;
TIMEOUT = TRANSA + 2×PROP, since TIMEOUT chosen optimally and PROC=0
= 100 + 2×20 = 140 microseconds;

therefore throughput = $(1 - 0.05)/((400+140) \times 10^{-6})$ = **1,759.26 packets/second** (*not* frames/second)

If Go-back-n ARQ used instead of Stop-and-wait, throughput will be higher (in this case, $\text{throughput}_{GBN} = (1 - 0.05)/((400 + 0.05\times140) \times 10^{-6})$ = 2334.15 packets/second). But this comes at the cost of higher complexity and (slightly) higher overhead to indicate packet sequence numbers.

Question 4

(4-a) **A:**

Network ID	**Cost**	**Next Hop**
12	**1**	**--**
10	**1**	**--**

B:

Network ID	**Cost**	**Next Hop**
12	**1**	**--**
90	**1**	**--**
7	**1**	**--**

C:

Network ID	**Cost**	**Next Hop**
10	**1**	**--**
90	**1**	**--**
6	**1**	**--**

D:

Network ID	**Cost**	**Next Hop**
7	**1**	**--**

When A receives B's initial routing table, since B is 1 hop from A, A modifies B's table as follows:

Modified B:

Network ID	Cost	Next Hop
12	2	B
90	2	B
7	2	B

The only entry in common is for Network 12, and A's original entry is lower-cost, so A's new table is

A:

Network ID	**Cost**	**Next Hop**
12	**1**	**--**
10	**1**	**--**
90	**2**	**B**
7	**2**	**B**

If A then receives C's initial table, since C is 1 hop from A, A modifies C's table as follows:

Modified C:

Network ID	Cost	Next Hop
10	2	C
90	2	C
6	2	C

The entries in common are for Networks 10 and 90. In both cases the information from C is not lower-cost, so A's new table is

A:

Network ID	**Cost**	**Next Hop**
12	**1**	**--**
10	**1**	**--**
90	**2**	**B**
7	**2**	**B**
6	**2**	**C**

(4-b) **A:**

Advertiser	**Network**	**Cost**	**Neighbour**
A	**12**	**1**	**B**

	A	**10**	**4**	**C**

B:	**Advertiser**	**Network**	**Cost**	**Neighbour**
	B	**12**	**3**	**A**
	B	**90**	**1**	**C**
	B	**7**	**1**	**D**

C:	**Advertiser**	**Network**	**Cost**	**Neighbour**
	C	**10**	**1**	**A**
	C	**90**	**2**	**B**
	C	**6**	**1**	**--**

D:	**Advertiser**	**Network**	**Cost**	**Neighbour**
	D	**7**	**3**	**B**

A uses Dijkstra's shortest-path algorithm to determine its shortest-path spanning tree. Final result is shown in attached Figure (alternative sequences of steps exist due to details of how ties are broken).

Question 5

(5-a)

Transmission number	Sender's Congestion Window (kB)	Threshold (kB)
0	**2**	64
1	**4**	64
2	**8**	64
3	**16**	64
4	**32**	64
5	**2**	16
6	**4**	16
7	**8**	16
8	**16**	16
9	**18**	16
10	**20**	16
11	**2**	10
12	**4**	10
13	**8**	10
14	**10**	10
15	**12**	10
16	**12**	10

(5-b) IP address: 154.7.7.220
Mask: 255.255.255.0 (255 means all-1's)

Subnetwork address is Boolean-AND of these: <u>**154.7.7.0**</u>

Since this is a Class B address, the <u>**Netid is 154.7**</u>, the <u>**Subnetid is 7**</u>, and the <u>**Hostid is 220**</u>.

Figure for Question 4: