SCHDF0018 - HIGHER DIPLOMA IN COMPUTER SCIENCE EXAMINATION
ARBDF0015 – THIRD YEAR ARTS EXAMINATION

COMPUTER SCIENCE

COMPP303: Networks and Internet Systems
COMP3616: Networks and Internet Systems

OUTLINE SOLUTIONS AUTUMN 2005
Dr. J. Murphy

Question 1

(1-a) TCP: retransmissions can lead to high delay and delay jitter; doesn't support multicast; slow start congestion control mechanism not suitable for continuous media.
UDP: traditionally, no retransmissions or congestion control; supports multicast.

(1-b) Hierarchical modularity: computer network software is structured as a hierarchy of layers. Each layer offers certain *services* to the higher layers, while hiding from the higher layers the *details* of how those services are implemented – this is hierarchical modularity.
Distributed scripts: although actual communication is "vertical" (except in the physical medium), peer entities – at same layer, but in 2 different computers – are programmed as if data transmission were "horizontal". Together, these peer entities execute *distributed scripts.*

(1-c) propagation delay = (23)/(230,000) = 0.1 millisec, or 100 microsec
packet transmission time = (10,000)/(100,000,000) = 0.1 millisec, or 100 microsec
therefore number of packets in transit = (0.1)/(0.1) = (100)/(100) = **1 packet**

(1-d) in the Lecture Notes (or any computer networks textbook)

Question 2

(2-a) TRANSF = 400 microseconds;
TIMEOUT = TRANSA + 2×(PROP+PROC), since TIMEOUT chosen optimally
= 40 + 2×(20+10) = 100 microseconds;

therefore $\textbf{throughput}_{SW}$ = $(1 – 0.01)/((400+100) \times 10^{-6})$ = **1,980 packets/second** (*not* frames/second)

and $\textbf{throughput}_{GBN}$ = $(1 – 0.01)/((400+(0.01\times 100)) \times 10^{-6})$ = **2,468.83 packets/second**

(2-b) see lecture notes.

Question 3

(3-a)

1. In Ethernet, each node's physical address is guaranteed to be globally unique: TRUE.

2. The General Parity Check error-handling scheme, in which the receiver takes the closest valid codeword (in Hamming distance) to the received word to be the transmitted codeword, can detect any combination of bit errors: FALSE.

3. In any flow control scheme, if the receiver cannot handle the sender's current transmission rate it must send an explicit "slow down" signal to the sender: FALSE.

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

1. If the length of the channel is increased, then PROP increases, therefore throughput **<u>decreases</u>**.

2. If the average frame length is decreased, then TRANSF decreases, therefore throughput **<u>increases</u>**

<u>Question 4</u>

(4-a) desirable properties of a routing algorithm:

- correctness, simplicity, efficiency – obviously
- robustness – since usually the entire network can't be "re-booted" !!!
- stability – routing algorithm reaches equilibrium in a reasonable time
- fairness, optimality (often in conflict)
    - optimality – with respect to what ? What are we trying to optimise ?!
      -- average Packet delay ? total Packet throughput ?
      -- but these goals are also in conflict: operating any network near capacity implies long queueing delays in node buffers
      -- compromise – minimise number of relays (or hops) a Packet needs

(4-b) ***distance-vector:*** each router exchanges information about the entire network with neighbouring routers at regular intervals. Neighbouring routers = connected by a direct link (e.g. a LAN); regular intervals: e.g. every 30 seconds. Information exchanged = routing tables (details in lecture notes).

link-state: each router exchanges information about its neighbourhood with all routers in the network when there is a change. Neighbourhood of a router = set of neighbour routers for this router; each router's neighbourhood information is <u>flooded</u> through the network; change: e.g. if a neighbouring router does not reply to a status message. Information exchanged = link-state packets (details in lecture notes).

<u>Question 5</u>

(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

(5-b) see lecture notes.