**Question 1.**

a) False, because sliding window is a protocol for flow control and reliable, in-order packet delivery—not for accessing a medium.

b) False, because congestion control is a mechanism concerned with the current network capacity, while flow control is a (different) mechanism concerned with the current receiver capacity.

c) False, because routers work in the network layer and exchange routing table information using a routing protocol situated in the network layer, not the application layer where the file transfer protocol can be found.

**Question 2.**

a) We note that the netmask is 255.255.248.0 (twenty-one 1s, followed by eleven 0s). Subsequently, the network number is the left-hand part of the IP address, determined by the netmask: 135.104.192.0 (calculated by doing AND of the IP address and the netmask). The host number is the right-hand part of the IP address, also determined by the netmask: 0.0.0.100 (calculated by doing AND of the IP address and the inverse of the netmask).

The number of networks and hosts that can be addressed with the netmask 255.255.248.0 is $2^{21}$ (2,097,152) and $2^{11}$ (2,048), respectively (including possible broadcast and reserved addresses).

b) The fields TTL and Checksum are updated by the routers; TTL is decremented by one for each hop, until it reaches zero and the packet is discarded by the router; the Checksum is recalculated by the router before retransmitting the packet to the next hop. (By the way, “UDP address”, which has a format identical to TCP’s, is not found in the IP header at all)

**Question 3.**

(The “R1 – B” is a typo and should be “R2 – B”)

We assume the routers have enough buffers for queuing packets.

a) The time for the whole packet, including the last bit, to reach host B is:

$$
\text{transmissionDelayAtA} + \text{linkDelayAtR1} + \text{transmissionDelayAtR1} + \\
+ \text{linkDelayR1toR2} + \text{transmissionDelayAtR2} + \text{linkDelayR2toB} = 
$$
Answer: the time for the packet to reach host B is 47 ms.

The round-trip time is:
47 ms + the same calculation as above, but for a 1,000-bit packet

\[ = \frac{1}{10 \times 10^3} + 0.005 + \frac{1}{2 \times 10^3} + 0.030 + \frac{1}{10 \times 10^3} + 0.005 = 0.0001 + 0.005 + 0.0005 + 0.030 + 0.0001 + 0.005 = 0.0407 = 40.7 \text{ ms} \]

Answer: the round-trip time is 87.7 ms

b) (“Round-trip delay” is the same as “Round-trip time”)

We assume that pipelining is allowed, that is, no ACK is sent between fragments.
Dividing the packet into 1,000-bit fragments gives us 10 fragments to be sent to host B, and also enables parallelisation of retransmissions at the routers. The new round-trip time becomes:

10 fragments to send A–R1: \( 10 \times \left( \frac{1,000}{10 \times 10^6} + 5 \times 10^{-3} \right) = 10 \times (0.0001 + 0.005) = 0.051 = 51 \text{ ms} \)

(each previous fragment after the second fragment are retransmitted by routers R1 and R2 at the same as a new fragment is output by A)

wait for 1 fragment no. 10 being sent R1–R2: \( \frac{1,000}{2 \times 10^6} + 30 \times 10^{-3} = 0.0005 + 0.030 = 0.0305 = 30.5 \text{ ms} \)

(fragment no. 9 is at the same time retransmitted R2–B)

wait for 1 fragment no. 10 being sent R2–B: \( \frac{1,000}{10 \times 10^6} + 5 \times 10^{-3} = 0.0001 + 0.005 = 0.0051 = 5.1 \text{ ms} \)

ACK to send B–A (see a) above): 40.7 ms

The new round-trip time is: 51 + 30.5 + 5.1 + 40.7 = 127.3 ms

(Receiver R1 will be very busy, because of the slow link R1–R2 …)

**Question 4.**

a) RSVP (ReSource reserVation Protocol) is a signalling protocol for the Internet, used to reserve bandwidth in routers, for example in the IntServ framework for Quality of Service.

b) Scheduling is either a mechanism or a discipline. The discipline is the manner in which queued packets in a router are selected for transmission over a link. An example scheduling mechanism is Weighted Fair Queuing.

Policing is the regulation of the rate at which a flow is allowed to inject packets into the network. An example policing mechanism is the leaky bucket.
Question 5.

a) A diagram for CSMA/CD (cf. the textual description on pp. 460–461 in Kurose & Ross, 2nd Ed.). It is also all right to describe the algorithm without the jam signal end the exponential backoff (as on p. 460 in the textbook), but if you include one of them then you need to have the other in your description too; these are the two different levels of granularity in the description.

![Ethernet Transmission Algorithm Diagram]

b) This is because the transmitting station must be sure that the previous frame has arrived safely before trying to transmit the next packet.

Question 6.

We use Dijkstra’s algorithm. The routing table at D is (with the notation found in the textbook):

<table>
<thead>
<tr>
<th>N</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>8, D</td>
<td>*</td>
<td>*</td>
<td>2,D</td>
<td>*</td>
</tr>
<tr>
<td>DE</td>
<td>8, D</td>
<td>4, E</td>
<td>3, E</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>DEC</td>
<td>6, E</td>
<td>4, E</td>
<td>9, C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECB</td>
<td>6, E</td>
<td>9, C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECBA</td>
<td></td>
<td>9, C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECBAF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
or with another notation (from a previous exam):

<table>
<thead>
<tr>
<th>Step</th>
<th>Current shortest path</th>
<th>Candidate set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(D, 0, -)</td>
<td>(A, 8), (E, 2)</td>
</tr>
<tr>
<td>2.</td>
<td>(D, 0, -), (E, 2, E)</td>
<td>(A, 8), (B, 4), (C, 3)</td>
</tr>
<tr>
<td>3.</td>
<td>(D, 0, -), (E, 2, E), (C, 3, E)</td>
<td>(A, 6), (B, 4), (F, 9)</td>
</tr>
<tr>
<td>4.</td>
<td>(D, 0, -), (E, 2, E), (C, 3, E), (B, 4, E)</td>
<td>(A, 6), (F, 9)</td>
</tr>
<tr>
<td>5.</td>
<td>(D, 0, -), (E, 2, E), (C, 3, E), (B, 4, E), (A, 6, E)</td>
<td>(F, 9)</td>
</tr>
<tr>
<td>6.</td>
<td>(D, 0, -), (E, 2, E), (C, 3, E), (B, 4, E), (A, 6, E), (F, 9, E)</td>
<td></td>
</tr>
</tbody>
</table>

The routing table consists of the (node, cost, nextHop) entries listed in step 6 above.

b)

We assume that each node only knows the distances to its immediate neighbours.

The sequence of distance-vector exchanges is listed in the table below, with one step of exchange listed per table row.

Notation:

- “*” equals “infinity” or “unknown value”
- “(0, *, 3, 8, *, *)” in the first row under A in the table shows that node A knows the distance to itself (“0” in the first position between the parentheses), to C (“3” in the third position) and to D (“8” in the fourth position) after the first exchange of distance vectors.

(always remeber to explain your notation if you are not uding the notation in the textbook)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, *, 3, 8, *, *)</td>
<td>(*, 0, *, 2, *)</td>
<td>(3, *, 0, 1, 6)</td>
<td>(8, *, *, 0, *)</td>
<td>(2, *, 1, *)</td>
<td>(*, *, *, 0, 6)</td>
</tr>
<tr>
<td>(0, *, 3, 8, 4, 9)</td>
<td>(*, 0, 3, 4, 2, *)</td>
<td>(3, 3, 0, 3, 1, 6)</td>
<td>(8, 4, 3, 0, 2, *)</td>
<td>(4, 2, 1, 0, 7)</td>
<td>(9, *, 6, 7, 0)</td>
</tr>
<tr>
<td>(0, 6, 3, 6, 4, 9)</td>
<td>(6, 0, 3, 4, 2, *)</td>
<td>(3, 3, 0, 3, 1, 6)</td>
<td>(6, 4, 3, 0, 2, 9)</td>
<td>(4, 2, 1, 0, 7)</td>
<td>(9, 9, 6, 7, 0)</td>
</tr>
<tr>
<td>(0, 6, 3, 6, 4, 9)</td>
<td>(6, 0, 3, 4, 2, *)</td>
<td>(3, 3, 0, 3, 1, 6)</td>
<td>(6, 4, 3, 0, 2, 9)</td>
<td>(4, 2, 1, 0, 7)</td>
<td>(9, 9, 6, 7, 0)</td>
</tr>
</tbody>
</table>

There is no change in the vectors in the fourth iteration, so we stop.

The routing tables are (same notation as in a) above:

A: (A, 0 -), (B, 6, C), (C, 3, C), (D, 6, C), (E, 4, 3), (F, 9, C)
B: (A, 6, E), (B, 0 -), (C, 3, E), (D, 4, E), (E, 2, E), (F, 9, E)
C: (A, 3, A), (B, 3, E), (C, 0 -), (D, 3, E), (E, 1, E), (F, 9, F)
D: (A, 6, E), (B, 4, E), (C, 3, E), (D, 0 -), (E, 2, E), (F, 9, E)
E: (A, 4, C), (B, 2, B), (C, 1, C), (D, 2, D), (E, 0 -), (F, 7, C)
F: (A, 9, C), (B, 9, C), (C, 6, C), (D, 9, C), (E, 7, C), (F, 0, -)

Note that you must let the algorithm run all the way to end in order to be sure that there are no changes in the values for the distances to E!

Question 7.

a) Marshalling is the encoding of data so that it can be transported over the network (unmarshalling is the reverse process done at the receiving end).
In CORBA, marshalling is typically done in *stub* on the client side, while unmarshalling is done in the *skeleton* on the server side.

b) The role of IDL in CORBA is to be an *interface definition language*, used to specify objects and services (expressing methods and their parameters).

**Question 8.**

a) A two-tiered architecture is one where a distinction is made between two kinds of machines: clients and servers.

A three-tiered architecture is when the server also needs to act as a client, for example, an application server that needs to access a separate database server between queries received from a user interface (the client).

b) The purpose of *transparency* is to make a distributed system transparent to a user, that is, make the system able to present itself to both users and applications as if it were only a single computer system.

Two examples of transparency are

i) *location* transparency, which means that users cannot tell where a resource is physically located in the system, and

ii) *access* transparency, which deals with hiding differences in data representation and the way that resources can be accessed by users.

**Question 9.**

a) The naming service in DCOM in Windows 2000 is implemented with support from Windows Active Directory. The client uses DNS to find the address of the Domain controller, which is one of many local directory servers that keep track of users and resources in a domain. Then the client can make an LDAP query to the Domain controller and request the name of an resource.

b) DCOM has transient objects, which means that when an object is no longer referred, it is destroyed.