*Q1*

*(a) [4].*

Provide 4 reasons why layered protocols help in comms network design.

1. *Place all marks in the LH margin.*

*Totals for each section should be clearly marked with square brackets [ ].*

*(b) [2]*

Provide 2 reasons why layered protocol architectures can be disadvantageous.

(c) [8]

Provide the 7 layers of the ISO/OSI layered model, in the correct order with an example function for each.

(d) [3]

Provide 3 key differences in the structural description between the 2 models.

(e) [3]

[1] Adherence gives a clear demarcation between the protocols and parameters in different layers.

[1] Lack of adherence can cause problems e.g. use of Port numbers (transport layer parameters) in network layer can mean interchange of transport or network layer protocols is difficult.

[1] Alternatively a lack of adherence can mean that a quick fix is possible which may prove very effective e.g. NAT addressing.

*Q2.*

1. [1] An interface
2. [1] 10000000 01010101 10101010 00000011
3. [1] [(16x0s): (16x0s): (16x0s) :(16 x 1s): (16x0s): 0101000001010000 : 1010000010100000 : (16x0s)] Note that there are 8 groups of 16 bits.
4. *[1 mark for each of:]*

1. Increased address space.

2. Hierachical addressing.

3. Simplified header.

4. Improved security.

5. Autoconfiguration facility.

6. QoS guarantees.

7. Support of mobile computing

*(e) [1]* Through the contents of the (4-bit) version field.

*(f) [2]* The server is set up to run both IPv4 & IPv6 and according to version field selects which version of IP to use on a particular packet.

*(g) [2]* The IPv6 packets are contained within the IPv4 packets. This is provided by having IPv6/IPv4 gateways to generate the IPv4 packets and the packets then ‘tunnel’ through the IPv4 network.

*(h) [5]*

Conversion/translation can take place either at the network layer, but then for example addresses contained inside the data field would be problematic so conversion at this layer would not always be satisfactory OR at the application layer in which case a very complicated translation function would need to be provided but would need to be designed to deal with all cases.

*Q3.*

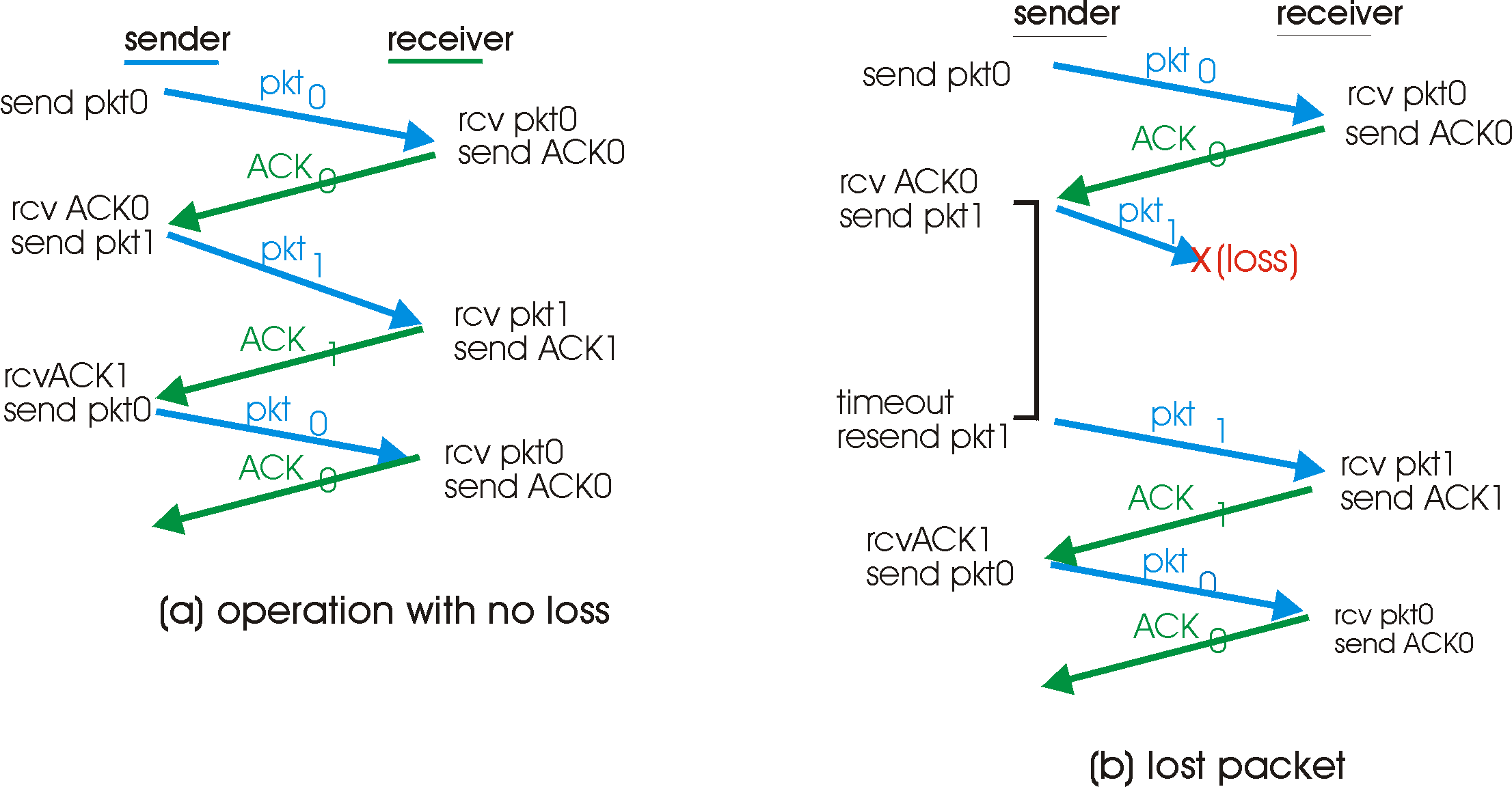
1. *[3]*

TCP uses acknowledgements of correctly received segments (i.e. no detected errors and in order delivery of all) and sequence numbers for identification. The objective of flow control in TCP is to avoid packets overwhelming the transport layer receive buffer.

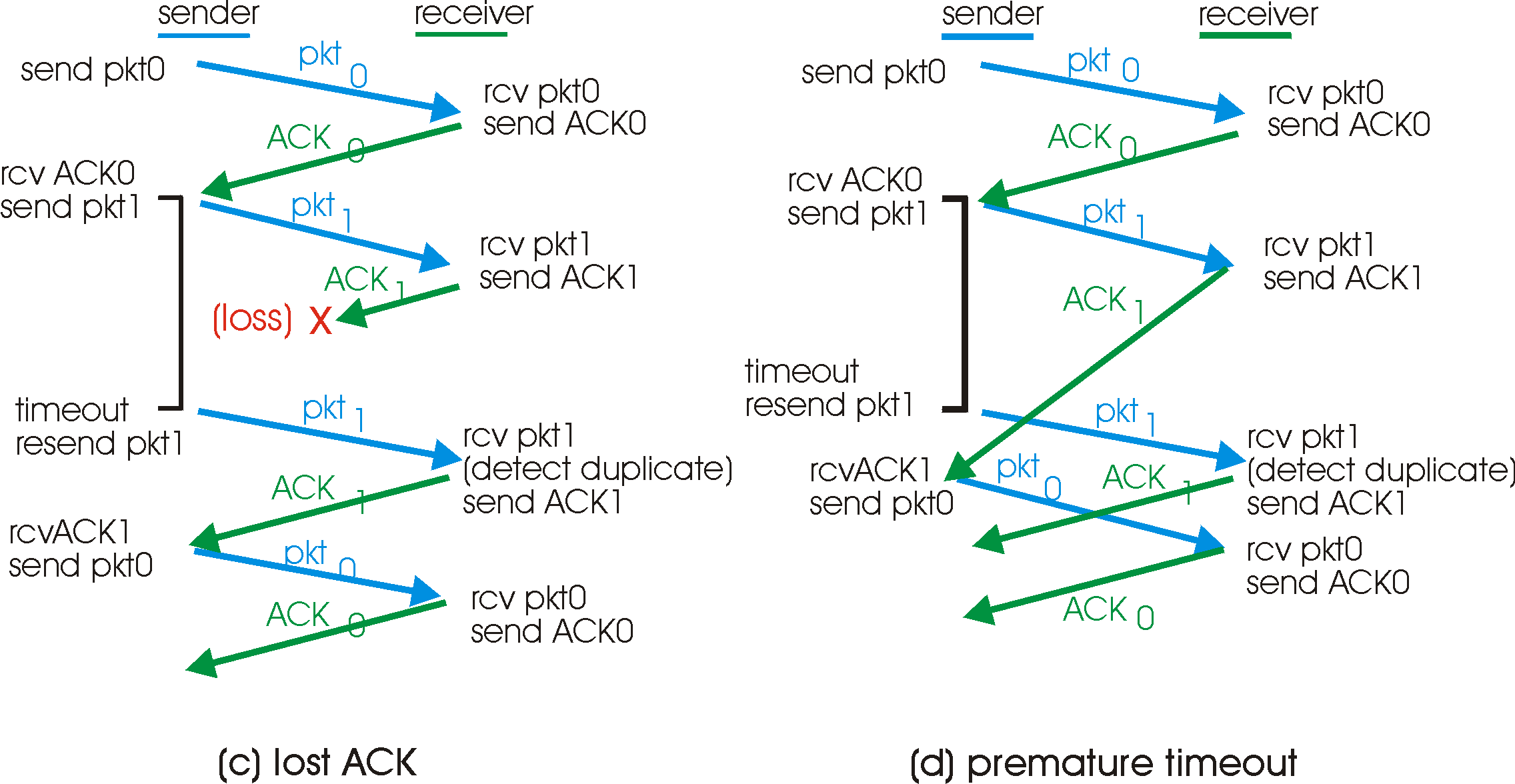
1. *[4]*

Flow control aims to avoid a transmitter overwhelming a slower receiver i.e. data being lost because it is arriving at a receiver buffer more quickly than an application is extracting it, i.e. fundamentally across a single end-to-end link. Whereas, congestion control is to avoid the traffic arriving at routers overwhelming the input buffers and being lost through discard. This is done by reducing the rate at which traffic is transmitted. This aims to ensure the total aggregate traffic entering a network is less than the traffic leaving the network, i.e. fundamentally a network wide process.

1. *[2]*



1. *[2]*



1. *.*
   1. *..*

Link Efficiency, U = {L/R}/ [RTT+ {L/R}]

L = 125B = 1000b, d=2x106m

{L/R} = 1x10-6s, RTT={2x106 /2x108} = 1x10-2.

U = {1x10-6}/{1x10-2 + 1x10-6} = ~1x10-4 ~ 0.01%]

*ii.*

L =125B =1000b, d = 20m, R = 1x106bps,

{L/R} = 1 x 10-3, RTT = 20/2x108 = 1x10-7

U = 1x10-3/{1x10-3+1x10-7} = ~1 ~100%

iii.

Fundamentally for short links where the passage of the data is filling the link stop and wait is very efficient e.g. Bluetooth.

However for larger and faster networks where the RTT is significant and the data are not continuous across the link stop and wait is highly inefficient.

In these cases go back n or selective repeat are better but each have particular storage requirements.

*Q4.*

1. *[3]*

TCP provides a reliable data transfer service i.e. in order delivery of all data.

1. *[2]*

UDP provides a best effort service, i.e. it simply provides unenhanced the service of the network layer below, i.e. it provides multiplexing and limited error detection.

1. *[5]*

Both TCP and UDP are using IP [1] but TCP uses sequence numbers [1] (so segments can be reordered is necessary [1]), acknowledgements [1] and retransmission of missing segments [1].

1. *[6]*

1. The transport layer (TCP) entity within the host machine trying to establish a connection sends a segment with the SYN bit high and a random initial sequence number (X) to the host which the connection is to be made with.

2. The TCP entity within the destination host acknowledges the received SYN packet whilst holding SYN high, random initial sequence number (Y) ACK high and Acknowledgement number X+1.

3. The origin of the request then acknowledges the returned packed and the connection is established. ACK=1, Ack = Y+1.

i.e. a 3 way handshake.

1. *[4]*

Any applications which are more sensitive to time delays than errors might choose to use UDP since the congestion control and flow control mechanisms of TCP can lead to increased and variable delays of packets but data carried by UDP is likely to have more errors.