Computer Networking
Supervision Exercises

Andrew W. Moore*
Andrew.Moore@cl.cam.ac.uk

Compiled Friday 21st November, 2014 at 10:15

This and subsequent handout updates are intended to cover all the main points of understanding in the lecture course. There is more material here than can reasonably be expected to be covered in all your supervisions; this material is intended as a set of seed material from which supervisors can direct supervisions and material students may utilise in their exam preparation.

Alongside specific questions, these handouts suggest several end of chapter questions in
KR: *Computer Networking: A Top-Down Approach*, Kurose & Ross, Pearson, International Edition, 5th edition, 2010
OR
PD: *Computer Networks: A Systems Approach*, Peterson & Davie, Elsevier, International Edition, 5th edition, 2012

For each topic I have selected a subset of questions; beware many questions in KR are overly simplistic; also — annoyingly there is limited overlap of questions from either text book.

Unlike most Tripos questions, most of these can be answered quite briefly, so you should expect to spend more time reading and thinking than you spend writing answers. Be warned that exam questions may expect you to remember incidental facts from the lectures or lecture-notes which are not covered here.

The authors are grateful for input from many past students, supervisors and colleagues.

*This supervision handout is based upon work by Stephen Kell, Scott Shenker, Sylvia Ratnasamy, and Andy Rice, my thanks to them all.

Topic 01 - Introduction / Foundation

1. *Concepts in (computer) networking*

   Consider a communication network consisting of a room full of people, where one or more people are exchanging *thoughts* with one or more others by talking.

   (*a*) For each of the abstract terms *node*, *channel*, *entity*, *layer*, *transmission* (the act thereof), *coding*, *addressing* and *multiplexing*, identify one or more corresponding concrete components or activities within the system. If the correspondence is not exact, give the closest approximation you can, and explain why it is not exact. [If you're unfamiliar with any of the abstract terms, you may need to look ahead in the lecture notes, or in one of the course texts.]

   (*b*) Compare and contrast the network with a shared-media wireless Ethernet and a single Ethernet link, for the following channel criteria:

   - physical medium
   - total capacity
   - maximum user-to-user capacity
   - medium access control
   - geographical area
   - failure modes

   (*c*) Suggest one way in which *administration* (or "management") of a room-full-of-people network may be performed. State whether it is distributed, centralised or a mixture of the two. [Hint: consider the example of a private party.]

   (*d*) For one of the entities you identified above, describe

   (*i*) the abstract interface it demands of the lower-layer entity;

   (*ii*) the abstract interface it provides to a higher-layer entity;

   (*iii*) the *symbols* which the entity transmits to its *peer entity* across the corresponding *channel*;

   (*iv*) each of the other characteristics of the corresponding channel, as listed on page 16 of the notes.

   (*e*) Suggest a type of node in the room-full-of-people network which a suitable *horizontal abstraction* would remove from consideration.

(*f*) Give an example of a software system which performs

(*i*) abstraction without strict *layering*;

(*ii*) layering without *embedding*.

Your examples need not be typical (computer) communication systems.

(*g*) Explain what features of the following networking applications make them *unsuited* to shared media implementations:

(*i*) the Internet;

(*ii*) the telephone network within a single street (between houses and a kerbside box);

(*iii*) the rail network between Manchester and London.

2. *Multiplexing basics*

(*a*) Give an example of multiplexing in a real system. Using your example, explain the relationship between multiplexing and coding. Explain your example's *policy* (concurrency-control) on access to the lower-layer channel, and how this is agreed.

(*b*) Explain the similarity and distinction between *frequency division multiplexing* and *wave division multiplexing*, giving an example of each.

(*c*) What kind of traffic is suited to *synchronous time-division multiplexing*? Define the term *circuit*. Give an example of a circuit which is *not* implemented using time-division multiplexing.

(*d*) Give three ways in which *asynchronous* time-division multiplexing is more complex than synchronous time-division multiplexing. In what circumstances does asynchronous TDM make more efficient use of the lower-layer channel than synchronous TDM?

(*e*) Explain why *contention policies* are inherently more complex on a *shared media* link than a *point-to-point* link using asynchronous TDM. Give an example of each.

(*f*) Ethernet is an asynchronous TDM system with a random-access contention policy. With reference to *collision detection*, give one reason why shared media Ethernets have a maximum length, and suggest how this might be calculated from the bitrate of the link and the minimum packet length. Give an example of a link which might use asynchronous TDM but where collision detection is not feasible.

(*g*) Using the example of a two-way radio conversation, explain *token loss* and *token duplication* in token-based asynchronous TDM.

(*h*) Explain how *slotted systems* for asynchronous TDM are different from synchronous TDM. In what case are the two equivalent?

(*i*) The telephone network is an example of a *reservation* system. Explain what is happening when a caller dials a telephone number. Give two reasons why the delay between dialing a number and being connected (i.e. hearing the remote ringer) is variable.

Discussion questions

3. *Supervision discussion questions*

(*a*) Give an example of multiplexing at all five (seven?) layers in a networking stack.

Book Question Selection

KR	
⋆	Chap 1: P1 P3 P6 P8 P10 P16 P18 P21
⋆⋆	Chap 1: P25
PD	
⋆	Chap 1: 7 11 15 3 28 36 37 38
⋆⋆	Chap 1: 23

Topic 02 - Architecture and Internet

4. *Standards - so many to choose from and Internet Philosophy*

   (*a*) The Internet's standards body, the IETF, has a philosophy which was summarised by David Clark, one of the Internet's pioneers, as follows.

   > "We reject kings, presidents and voting. We believe in rough consensus and running code."

   This suggests an approach which is open, dynamic and led by implementation. By contrast, other standards bodies such as the ITU are closed, slow-moving and led by specification. Using examples, discuss ways in which the IETF's approach has enabled innovation in the Internet, and ways in which it has caused problems.

   (*b*) Prior to the Internet, wide-area networks were joined together at level of application protocols, using *gateways*. Explain, as fully as you can, why this approach limited application development.

   (*c*) Explain how the design of the *Internet protocol*, i.e. IP, addressed this problem of application development. You should explain how the term "hourglass model" describes IP's approach to network layering.

   (*d*) The design of IP makes explicit provision for *fragmentation*, i.e. the ability to split an individual packet into pieces during its journey across the network. By considering the hourglass model, suggest why this feature is essential.

   (*e*) TCP is a connection-oriented reliable byte-stream protocol designed to run over IP, an unreliable connectionless datagram protocol.

      (*i*) Where is the state of a TCP connection held?

      (*ii*) How does this constrain the set of guarantees offered by a TCP connection?

      (*iii*) In what ways is packet loss essential to TCP's methods of modeling the state of the channel (i.e. within the network)?

   (*f*) Comparing (data) delays in packet and circuit-switched networks. Compare the time it takes to transfer a file of data under circuit-switching and packet-switching. Consider a network consisting of n links in a row, each of B bandwidth and latency L.

Circuit-switching: At time $t = 0$, the first node sends out a circuit reservation packet (of size R) which is sent to the second node, which then receives the full packet and then forwards it to the next node. This is continued at each node, until the reservation packet arrives at the last node (after traversing n links). After this reservation message is processed at the last node (the destination), the last node sends back a reservation confirmation message (also of size R) back to the first hop. Because the circuit is established before this confirmation is sent, this packet need not be processed at each node; instead, the bits flow through the nodes without any delay. Once the confirmation message is received at the first node (the source), the source immediately starts sending the file (which is of size F) at the full bandwidth of the link.

Note that when the file is transferred, the data is not stored-and-forwarded at any of the intermediate nodes but is just passed through without delay.

Also, we ignore the teardown message, since it is only sent after the file arrives.

(i) Assuming no problems in transmission along the way, at what time does the last bit of the file arrive at the last node (the destination)?

Packet-switching: Here the file is broken into Q packets of size D, each with header size H and payload size P. Since the entire file must be carried, $Q \times P = F$. At time $t = 0$, the source (the first node) sends the first packet, which is stored-and-forwarded at each of the subsequent nodes until it reaches the destination (the last node). As soon as the source finishes sending the first packet, it sends the second packet (at full link bandwidth). Note that the source does not wait until the first packet arrives at the next node before starting the next transmission, it starts sending the next packet as soon as it has finished transmitting the previous packet. We assume that a node can immediately send a packet out on the next link as soon as the last bit has arrived from the previous link (i.e., there is no time required to process the packet before sending it on the next link)

(ii) Assuming no packet drops or other errors, at what time does the last bit of the file arrive at the destination?

In the following questions, we refer to cases where some quantities are big. By that we mean consider the limit where that quantity becomes infinitely large or infinitesimally small. Note that some quantities are linked (i.e., if the payload P gets smaller, the number of packets Q must get larger to keep $Q \times P = F$). For each

question, the answer could be either: circuit-switched is faster, or packet-switched is faster.

Even if you didn't get the formulae above completely correct, you should understand how these perform relative to each other in the limit. Use this as a way to check your answers for the two previous questions.

(*iii*) If the file size F is very large, which is faster? (Assume that the header size H has not changed.)

(*iv*) If the payloads become small (but the header size remains constant), which is faster?

(*v*) If the bandwidth B is very large, which is faster? And by what ratio (in the limit)?

5. *Internet philosophy*

(*a*) What is the different between an architectural principle and an architectural design (choice)? What other examples can you think of?

(*b*) A NAT stores state about the flows (connections) that pass through it.

(*i*) Why does a NAT break the *end-to-end* principle?

(*ii*) Why might the NAT violation of end-to-end principles not actually matter?

Discussion questions

6. *Supervision discussion questions*

(*a*) What is a de-facto standard?

Topic 03 - Data-Link (Physical)

7. *Shared media multiplexing in local area networks*

   (*a*) Define the term *shared media network*.

   (*b*) Explain how Ethernet performs *carrier sense*, *collision detection*, how it aims to minimise the probability of collision on retransmission and how this is adapted to handle varying load.

   (*c*) Explain why token ring does not share media at the physical level, but is still analyzed as a shared media system.

   (*d*) What is the role of a token monitor in token ring? Why does the monitor *not* prevent failure when one of the nodes in the ring suffers a hardware failure? Suggest how a token ring system might be designed to handle failure of a computer attached to the ring.

   (*e*) Explain the meaning of *destination delete* and *source delete* as used in ring-based networks.

   (*f*) Explain the difference between conventional token rings and *slotted rings*. What are the advantages of a slotted ring?

8. *Multiplexing redux*

   (*a*) Several real-time video streams are to share the same lower-layer channel.

      (*i*) Give one example of a lower-layer channel in which the flows might be *scheduled*, and one in which scheduling is not possible.

      (*ii*) A lecturer remarks that "centralised multiplexing" offers potential gains in efficiency over non-centralised multiplexing. Give *two* reasons why this can improve efficiency. What, in general terms, is the "centralised" facility necessary for these gains to be possible?

      (*iii*) Using an example, describe why specifying a scheduling *policy* in terms of *priority* may cause problems, even where it is safe to use priority within the scheduling *mechanism*. [Hint: consider CPU scheduling in an operating system.]

   (*b*) Code-division multiple access (CDMA) is a code-division multiplexing system, used for mobile telephony.

(*i*) What is a *code*? What property of codes causes them to be "nearly orthogonal" to each other?

(*ii*) Two transmitters, A and B, both want to transmit a four-bit message at the same time using CDMA. Transmitter A has code 10010111 and message 1001. Transmitter B has code 00111101 and message 0011. Write down the bit sequences transmitted by A and B. Write down the bit sequence seen by a receiver, stating any assumption you make. Show that the original messages of both A and B may be recovered. [Each bit is transmitted as the exclusive OR of the code sequence with the bit value.]

9. *Coding, digitisation, error detection and error correction*

   (*a*) Give, with examples, *three* advantages of digitising audio, and *three* corresponding disadvantages. (Compare with storing and processing it exclusively on analogue media and equipment.)

   (*b*) Explain *quantisation* and *sampling* of analogue signals, and the distinction between these. State an upper bound for the signal-to-noise ratio of a signal quantised at b bits resolution, assuming the analogue original to be noiseless and the quantisation process completely accurate.

   (*c*) Outline encode and decode procedures for a simple (m, k) block code. Show that the minimum distance of any simple checksum code is always 2.

10. *CRCs*

    (*a*) Explain, giving an example, how to write a binary message (i.e. a sequence of binary digits) as a polynomial.

    (*b*) Outline send and receive procedures for CRC-based message coding and error detection. What information must be agreed in advance by the sender and receiver?

    (*c*) Draw a shift register which will compute the remainder on division of an input polynomial by the CRC-8 polynomial $x^8 + x^2 + x^1 + 1$.

11. *Physical layer transmission and channel characteristics*

    (*a*) (*i*) Explain the distinction between capacity and bandwidth, and the relationship between the two.

(*ii*) Explain the term latency. How is it related to capacity?

(*b*) For each of the following statements, state with explanation whether it is true or false.

(*i*) "Restricting an analogue channel's bandwidth does not restrict its information capacity."

(*ii*) "White noise can never be completely removed from a transmitted signal."

(*iii*) "Attenuation necessarily decreases a signal's signal-to-noise ratio."

(*iv*) "Because thermal noise attenuates along with the signal, doubling the length of a wire does not affect its effective signal-to-noise ratio."

(*c*) You have the option of doubling a channel's bandwidth or doubling its signal-to-noise ratio. Explain how you decide which maximizes the channel's capacity.

12. *Digital channels, modulation and transmission*

(*a*) Explain the distinctions between *baud rate* and *bit rate* and between *baseband* and *broadband*.

(*b*) Why is synchronisation not an issue for transmission of analogue signals?

(*c*) What problem in synchronous transmission is Manchester coding designed to solve? Suggest a simpler but possibly more expensive way of solving the same problem. Explain why a system which has a slow but accurate oscillator would be more suited to asynchronous transmission than to synchronous transmission using Manchester coding.

(*d*) List three properties of a sinusoidal waveform which admit *modulation*. Explain the relationship between *modulation* and *shift keying*.

Discussion questions

13. *Supervision discussion questions*

(*a*) Is physical line coding a challenge or an aid for the NSA?

(*b*) Why can a row/column parity check correct single bit errors or detect two bit errors, but not both.?

Book Question Selection

KR	
⋆	Chap5 P1 P4 P14 P19
⋆⋆	Chap5 P13 D1
PD	
⋆	Chap 2: 4 5 11 18 22 24 32 42 43 45
⋆⋆	Chap 2: 12 14 29 54 55

Topic 04 - Network

14. *Switching*

    (*a*) Why is switching a practical necessity in large networks? In what way is switching a form of multiplexing?

    (*b*) The switching process consists roughly of a *demultiplexing* stage, a *routing* stage and a *remultiplexing* stage. For each of the following examples of switching, explain what is being demultiplexed, what routing decisions are made, and how remultiplexing is performed:

        (*i*) packet switching in the postal network;

        (*ii*) packet switching in an Ethernet switch;

        (*iii*) packet switching in an IP router;

        (*iv*) circuit switching in the telephone network;

        (*v*) wave-division switching in an optical switch.

    (*c*) Switching can improve the efficiency of a network's link utilisation, but may also cause problems. In a packet-switched network, two particular problems are *increased latency* and *data loss*.

        (*i*) For one of the packet-switched examples above, explain how latency and loss might occur.

        (*ii*) Using the same example, suggest one way in which latency might be improved, and one way in which loss might be reduced.

        (*iii*) To what extent are the problems of latency and loss less significant in circuit-switched networks? Give *two* disadvantages of circuit-switched networks over packet-switched networks.

15. *IP addressing, forwarding and routing*

    (*a*) Explain in outline what is meant by a *class* of IP addresses. What problem did class-based addressing suffer? Suggest one modification to the IPv4 class system which relieves this problem *without* moving to classless routing.

    (*b*) In addition to an IP address, what extra information does *classless* routing require to represent a network address? Explain how a suitable address allocation policy can minimize the resulting increase in the size of forwarding tables.

(*c*) A router has the following forwarding table.

Destination	**Gateway**	**Interface**
62.24.128.0/17	62.24.128.1	1
128.232.0.0/16	195.9.190.9	2
80.2.192.0/18	80.2.224.200	3
default	195.9.190.9	2

(*i*) What is the relationship between the "gateway" and "interface" fields?

(*ii*) After selecting the correct outgoing interface, the router must generate a packet on the underlying data-link network (perhaps Ethernet). Explain how the destination field of the Ethernet frame is calculated.

(*iii*) A packet arrives at the router with destination IP 62.24.192.12. On which interface is it sent out, and what is the destination address in the IP header of the forwarded packet?

(*iv*) In the early days of the Internet, forwarding tables were maintained by hand. Give *two* reasons why the size of today's Internet makes this no longer feasible (except for routers close to the edge of the network).

(*v*) A *routing protocol* is a system used by routers to automatically maintain their forwarding tables. Outline a simple routing protocol which might be used to maintain the table above under a *shortest path* routing policy. Mention any additional information that you must store in the router, and any problems you notice.

(*vi*) Routes are often chosen to provide the shortest path across the network, but in many cases they are chosen for other reasons. Give *two* other factors which might affect the choice of route.

16. *General routing concerns*

(*a*) Following are some examples of routing strategies and real systems which use them. For each one, suggest one reason why the strategy is a good choice, and another in which it might cause problems.

(*i*) flood routing in an Ethernet hub

(*ii*) random routing in a peer-to-peer file-sharing network

(*iii*) source routing in the road network

(*iv*) hot potato routing in crowded supermarket aisles (when heading for a target grocery shelf)

(*b*) As well as benefits of bounded latency and assured capacity, circuit switching allows routing to be performed only once per connection, at set-up time. This contrasts with datagram-based routing, as commonly used in packet-switched networks, where routing is done for each datagram individually.

(*i*) Outline a set of additions or modifications to TCP/IP which would allow routing decisions to be made only once per TCP connection. Identify what (if any) other benefits of circuit switching your modifications provide, and which ones they do not.

(*ii*) Do your modifications preserve the *reliability* properties of datagram-based routing? Specifically, the property in question is that end-to-end connections can be maintained across a catastrophic failure of a router or link, assuming that an alternative path through the network exists. If they do, explain why. If not, suggest how this could be achieved, or explain why your modifications expressly preclude it.

(*c*) You are required to design a topology discovery protocol for a network of switching nodes interconnected by links. There are n nodes, l links, the maximum degree of any node is k and there is a path between any two nodes of not more than d hops. All links are bi-directional.

Each node has a unique identifier of four bytes which it knows.

(*i*) Outline a protocol (including message formats) for a node to learn about its immediate neighbours

(*ii*) Design a protocol (including message formats) for distributing this information across the network.

(*iii*) Give a bound on the total amount of information which is transmitted to ensure that every node acquires complete topology information.

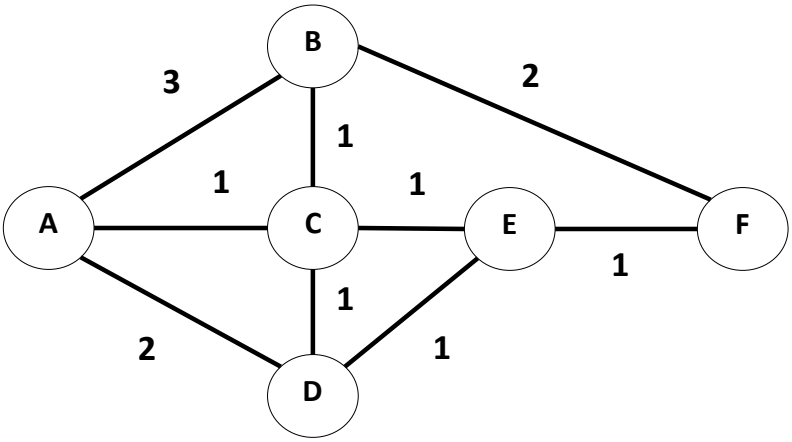


Figure 1: A network topology. Numbers refer to the link-cost of indicated link

(d) Consider a Distance-Vector Routing protocol

The routing table for noda A is expresed in a format where th eros indicate the destination and the colums indicate the first hop. That is, the number in (row C column D) denotes the cost of the best currently known path to C that starts with A and sending to D. The initial table for Noda A, before A has exchanged any routing information with any other node, takes the form:

Table for A	B	C	D
B	3	∞	∞
C	4	1	∞
D	∞	∞	2
E	∞	∞	∞
F	∞	∞	∞

Note that in this initial routing table, A does not know paths to any destinations except its immediate neighbours.

(i) After A receives a route update from B (B sends its initial table), what are the entries in the routing table for A?

(ii) After A receives a route update from C (C sends its initial table), what are the entries in the routing table for A?

(*iii*) Assume now that all nodes exchange tables in an iterative process until steady-state is achieved. What are the steady-state entries in the table for node A?

Now consider the network of Figure 1 and a Link-State Routing algorithm.

(*iv*) If A sends a packet to B, what path does the packet take?

(*v*) If A sends a packet to F, what path does the packet take?

(*vi*) Now assume that the link cost for the links C-E and E-F both change to 6. E announces these changes and all nodes but C get the update (that is, C still thinks C-E and E-F are link-cost 1). Now A sends to F, what path does the packet take?

(*vii*) Finally, C gets the new link-weight information and now knows that C-E and E-F are link-cost 6. When A sends to F, what path does it take?

Discussion questions

17. *Supervision discussion questions*

(*a*) Compare Forwarding versus Routing

(*b*) Compare and contrast Link-State Routing with Distance-Vector Routing. What are the (information) consistency models of each? What information is exchanged?

(*c*) What makes a fast LPM algorithm? (Longest-Prefix-Match)?

(*d*) What happens when (packet) fragment loss occurs?

(*e*) What is the state held by a NAT box?
How does a NAT box work out its state?

(*f*) Why do packets tend towards the same path(route) through the network?

(*g*) How does ARP work?

(*h*) Compare DNS and ARP

(*i*) Why would I (or wouldn't I) want to broadcast an ARP response?

(*j*) How does Gateway ARP do its thing?

Book Question Selection

KR	
$\star$	Chap4 P1 P8 P9 P17
$\star\star$	Chap4 P18
PD	
$\star$	Chap 3: 4 26 32 41 42 51 55
$\star\star$	Chap 1: 45 74 75

Topic 05 - Transport

18. *Transport flavours*

    (*a*) The User Datagram Protocol (UDP) is sometimes used instead of TCP. It provides very few features above those provided by IP.

        (*i*) Give one feature provided by UDP but *not* by IP, and one provided by UDP but *not* by TCP.

        (*ii*) Explain the role of a *port number* in UDP. Why isn't it the Process ID? Explain the relation between TCP and UDP port numbers.

        (*iii*) Give *three* characteristics which might make an application protocol better suited to implementation over UDP than over TCP.

19. *Error control, ARQ*

    (*a*) Why do error control protocols for packet switched networks use error detecting codes but not error correcting codes?

    (*b*) A transport protocol for packet-switched networks uses a "sliding window" Automatic Repeat reQuest (ARQ) scheme for error control and flow control.

        (*i*) As well as error detecting codes, ARQ protocols use *acknowledgments* and *timeouts* to achieve error control. Briefly explain what these are, and how they are combined to achieve reliable transmission.

        (*ii*) What *two* error cases might cause a receiver to send a negative acknowledgment (NACK)? How are they detected? What happens if the NACK is lost?

        (*iii*) In what circumstances will a receiver receive a packet with the same *sequence number* twice? What should it do in these circumstances?

        (*iv*) Given that the protocol provides bidirectional communication, what optimisation can be made in the implementation of acknowledgments to reduce the total number of packets sent?

        (*v*) If two hosts are connected by a 100Mbps link with a round-trip time of 20ms, how big (in bytes) should the *sliding window* be to maximise link usage?

        (*vi*) Give *two* reasons why, at a given time, the window size might be set to a smaller value.

(*c*) Consider a sliding window protocol with a window size of 5 using cumulative ACKs.

Retransmissions: retransmissions occur under two conditions:
Reception of three duplicate ACKs
(that is, three identical ACKs after the initial ACK)
Time out after 100msec
(timer starts at the beginning of the packet transmission)

Timing:
Data packets have a transmission time of 1 msec
ACK packets have zero transmission time
The link has a latency of 10msec.
The source A starts off by sending its first packet at time t=0.

(*i*) Assume all packets are successfully delivered except the following:
The first transmission of data packet #3
The ACK sent in response to the receipt of data packet #6
When is data packet #3 first retransmitted (expressed in terms of msec after t=0)?

(*ii*) Consider the same scenario, but with everything successfully delivered except the following:
The first transmission of data packet #3
The first transmission of data packet #5
The ACK sent in response to the receipt of data packet #6
When is data packet #3 first retransmitted (expressed in terms of msec after t=0)?

(*iii*) Assume we can only observe the ACK packets arriving at the sender.
The same sliding window algorithm is used, with the same timings and retransmission policies apply.
Notation (read carefully): The notation Ax is used to mean that the ACK packet is acknowledging the receipt of all packets up to and including data packet x.
That is, A5 is acknowledging the receipt of packet 5; to be clear, the notation does not mean that the receiver is expecting packet 5 as the next data packet.
Assume that the following ACK packets arrive (just the ordering is shown, no timing information is provided):
A1
A2
A3
A3
A4

A5
A6

Which of the following five scenarios (described only by the unusual events that occurred; assume all else functioned normally) would have produced such a series of ACKs? (consider all that apply)

(a) Data packet number 4 was dropped.

(b) Data packet number 4 was delayed, arrived immediately after data packet 5

(c) Data packet 3 was duplicated by the network

(d) ACK packet A3 was duplicated by the network

(e) ACK packet A4 was delayed, arriving after A5

(*iv*) With the same set up as in the previous problem, consider the following stream of ACK packets

A1
A2
A3
A5
A4
A6

Which scenarios (described only by the unusual events that occurred; assume all else functioned normally) would have produced such a series of ACKs? (consider all that apply)

(a) Data packet number 4 was dropped.

(b) Data packet number 4 was delayed, arrived immediately after data packet 5

(c) Data packet 3 was duplicated by the network

(d) ACK packet A3 was duplicated by the network

(e) ACK packet A4 was delayed, arriving after A5

(*v*) With the same set up as in the previous problem, consider the following stream of ACK packets

A1
A2
A3
A3
A5
A6

Which scenarios (described only by the unusual events that occurred; assume all else functioned normally) would have produced such a series of ACKs? (consider all that apply)

(a) Data packet number 4 was dropped.

(b) Data packet number 4 was delayed, arrived immediately after data packet 5

(c) Data packet 3 was duplicated by the network
(d) ACK packet A3 was duplicated by the network
(e) ACK packet A4 was delayed, arriving after A5

20. *TCP specifics*

(*a*) Use the approximate equation for throughput as a function of drop rate:

$$\texttt{throughput} = \frac{\sqrt{1.5} \times \texttt{MSS}}{\texttt{RTT}\sqrt{p}}$$

Assume an RTT of 40msec and an MSS of 1000bytes. In the following questions ignore IP and TCP headers in your calculations.

(*i*) What drop rate p would lead to a throughput of 1Gbps?

(*ii*) What drop rate p would lead to a throughput of 10 Gbps?

(*iii*) If the connection is sending data at a rate of 10Gbps, how long on average is the time interval between drops?

(*iv*) What window size W (measured in terms of MSSes) would be required to maintain a sending rate of 10Gbps? (rounded down to the nearest integer)

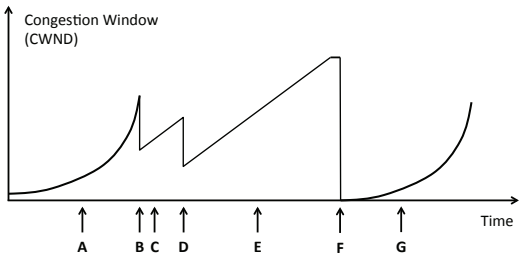
(*v*) If a connection suffered a drop upon reaching 10Gbps, how long would it take for it to return to 10Gbps (after undergoing a fast retransmit)? (in seconds, rounded down to the nearest second)

(*vi*) Consider two TCP connections whose throughput obeys the TCP throughput equation listed above.
The first TCP connection has the following parameters:
MSS = 1000 bytes, RTT = .2msec, drop rate = .5%
The second TCP connection has the following parameters:
MSS = 2000 bytes, RTT = .1msec, drop rate = 8%
What is the ratio of throughputs (the throughput of the first TCP connection divided by the throughput of the second TCP connection)? Why?

(*b*) Consider the plot of CWND versus time for a TCP connection.

(*i*) At each of marked marked points along the timeline in the figure on the next page, indicate what event has happened, or what phase of congestion control TCP is in (as appropriate), from the following set: Slow-Start, Congestion-Avoidance, Fast-Retransmit, and Timeout.

(*ii*) Assume CWND is 10,000 right before F, what is the value of SSTHRESH at G?



Discussion questions

21. *Supervision discussion questions*

    (*a*) What are four of the timers in the TCP ARQ?

    (*b*) How do I recover from silly window syndrome?

    (*c*) Why does UDP use port number and not process id?

    (*d*) What is the effect of out-of-order delivery on Go-Back-N?

    (*e*) What does congestion cause?

    (*f*) Why does doubling the window-size cause exponential growth?

    (*g*) How might duplicates occur in RDT 3.0?

    (*h*) When/Why can you cache window-size information between connections?

    (*i*) Why is reordering bad (for fast-transmit)?

    (*j*) How does multiple routing paths screwup TCP?

Book Question Selection

KR	
$\star$	Chap3 P2 P11 P13 P18 P23 P26 P27
$\star\star$	Chap3 P19 P32 P38
PD	
$\star$	Chap 5: 6 9 12 20 30 39 43 - Chap6: 13 16 19
$\star\star$	Chap 5: 25 37 48 54 - Chap6: 27

Topic 06 - Network Applications

22. *Internet Applications*

(*a*) HTTP is an application protocol, built on TCP, which was originally designed to allow retrieval of hypertext documents. Firewalls are application-level gateways (or routers) which aim to filter out malicious, illegal or unauthorised IP traffic based on the contents of packet payloads.

(*i*) Explain why network firewalls traditionally accept inbound HTTP traffic, but may not accept traffic for other services such as network filesystems or e-mail transfer (SMTP).

(*ii*) Describe the similarities and differences between an e-mail gateway (as might have been found joining two wide-area networks before the advent of IP) and an application-level firewall.

(*iii*) In modern networks, HTTP is used as a transport for remote procedure call, e-mail and even networked filesystems. A network engineer proposes that, to counter security concerns regarding some types of HTTP traffic, there is a need for a higher-level firewall which can selectively filter these. Suggest why this is more difficult than a firewall operating at the levels of TCP and UDP, and explain why this does not solve the security problem.

(*b*) Consider a case where a client A is retrieving files F and G from web site B. F and G are both 125KB (i.e., one megabit).

(*i*) The RTT between A and B is 10msec (note, these are round-trip-times, not one-way latencies), and the bandwidth between the sites is 10Mbps. Assume all TCP SYN/ACK packets and HTTP request packets are negligible in size. How long does it take A to retrieve both files under the following circumstances:
Sequential (one-at-a-time) requests with non-persistent TCP connections?
Concurrent requests with non-persistent TCP connections?
Sequential requests within a single persistent TCP connection?
Pipelined requests within a single persistent TCP connection?

(*ii*) Consider the same situation as above, but assume that rather than a dedicated link there is a large shared link with many flows traversing it, and each TCP connection gets 10Mbps

(adding additional flows does not significantly change the bandwidth per TCP connection, because there are thousands of flows on the link).
Now, how long does it take A to retrieve both files under the following circumstances:
Sequential (one-at-a-time) requests with non-persistent TCP connections?
Concurrent requests with non-persistent TCP connections?
Sequential requests within a single persistent TCP connection?
Pipelined requests within a single persistent TCP connection?

(*iii*) Consider the first situation, except that A is only downloading file F and there is now a cache C between A and B. All requests from A to B go through cache C, and assume the bandwidth along the path from A to C is 1Gbps and the RTT between A and C is negligible, while the bandwidth along the path from C to B is 10Mbps with an RTT of 10msec. Note, these are roundtriptimes, not oneway latencies. As above, assume that the file is 125KB (i.e., one megabit) and that all TCP SYN/ACK packets and HTTP request packets are negligible in size.
Assume the cache operates as follows: (where the origin server refers to the site named in the URL)

- If the object is not in the cache, the request is forwarded to the origin server
- If the object is in the cache, and the cache entry has not timed out (i.e., the cache TTL has not expired), the object is returned to the client
- If the object is in the cache, but the cache entry has timed out, the cache issues a conditional-GET to the origin server, asking if the object has changed since this object was cached: if the origin server responds that it hasnt, the cache returns the cached object, otherwise the origin server responds with the updated object which the cache forwards to the client.

How long does it take for A to receive the file under the following circumstances:
The file is not in the cache?
The file is in the cache and the TTL has not expired?
The file is in the cache, the TTL has expired, but the file has not been changed?
The file is in the cache, the TTL has expired, and the file has

changed?

(*iv*) Why is this question answerable *before* you have even discussed TCP?

23. *DNS as an Internet application*

(*a*) Consider the host `here.eye.am`, with a local DNS server `nameserver.eye.am`. `here.eye.am` asks `nameserver.eye.am` to resolve the hostname `there.you.ar` Assume there are no cached entries relevant to this request and that `nameserver.eye.am` utilises recursive resolution by default.

(*i*) Write down the steps taken to resolve `there.you.ar` and respond to *here.eye.am*

(*ii*) Describe the differences between this solution and one achieved using iterative an iterative DNS enquiry.

24. *Compression*

(*a*) Give one example of *perfect* compression coding, one of *stable imperfect* compression coding and another of *unstable imperfect* compression coding. For each example, outline one application for which it is a good choice.

(*b*) Suggest why inexpensive digital video-capable cameras typically perform Motion JPEG encoding rather than MPEG. What trade-off is being exploited here?

(*c*) Logically speaking, MPEG is tolerant of loss and a WiFi (Wireless Ethernet) network may have some unpredictable loss. How might you design an appliance that MPEG encodes Video for transmission over WiFi?

(*d*) Consider the misconfigured Microsoft Windows box, with a DNS domain of DOMAIN.

(*i*) Will DNS ask the root servers to resolve the server address for every DOMAIN enquiry?

(*ii*) Why does DNS use UDP?
Why are there only thirteen root servers?
Implications for the future?

25. *P2P*

(*a*) Peer-to-Peer systems might typically do some combination of three tasks, searching (e.g., keyword search), lookup (mapping name to location), and download.

Of the following phrases pick the most appropriate for each question

Some form of flooding
Distributed Hash Tables
Chunking
Number of participating peers
Asymmetry of bandwidth
Lack of centralised control
Self-scaling

(*i*) Which approach is often used for download?

(*ii*) Which approach is typically used for search?

(*iii*) Which approach is typically used for lookup?

(*iv*) Which factor is most responsible for making chunking advantageous?

(*b*) Skype and iPlayer both have a Peer-2-Peer heritage. Skype - a P2P success, iPlayer, less so — at least for now.

(*i*) Considering Metclafe's law, why has Skype been a success?

(*ii*) Why did iPlayer P2P not succeed whereas iPlayer most definitely has succeeded.

(*iii*) Skype is described as a hybrid application — both P2P and (traditional) client-server. Explain this.

Discussion questions

26. *Supervision discussion questions*

(*a*) What is the relationship between methods in Java (or syscalls in C) and the opening and closing of a TCP connection? (SYN, SYN-ACK, etc.)??

(*b*) Itemize every activity (packet) that your machine sends and receives between being attached to the network and going to a popular webpage.

(*c*) How does slow-start interact with HTTP?

(*d*) How do multiple connections speed web-pages?

Book Question Selection

KR	
⋆	Chap2: P7 P10 P16 P20
⋆⋆	Chap2: P18
PD	
⋆	Chap 9: 1. 12. 39. 40.
⋆⋆	Chap 9: 14. 42.

Topic 07 - Advanced Topic - DataCenters

Non-examinable.