Digital Communication I
Supervision Exercises

Stephen Kell
Stephen.Kell@cl.cam.ac.uk

February 4, 2010

These exercises are intended to cover all the main points of understanding in the lecture course. There are roughly $1\frac{1}{2}$ questions per lecture, and each question is supposed to take roughly the same amount of time to complete. Don't expect to be able to answer everything. You are advised not to spend more than an hour on any one question – unless you really want to.

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 lecture notes which are not covered here.

Criticisms or comments about any aspect of these questions would be very gratefully received, however large or small. This version has benefited from comments and corrections by David Miller, Andrew Moore and Gareth Williams.

1. *Concepts in (digital) communication*

   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*, [the act of] *transmission*, *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*) 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*;

   (*c*) Imagine that the people in the room do not all speak the same language, and that some are present simply as interpreters for the others. Considering different ways of *abstracting* networks, what kind of abstraction would remove these interpreters from view?

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

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

   (*ii*) layering without *embedding*.

   Your examples need not be typical communication systems.

2. *Internet philosophy, design of IP, transport protocols*

   (*a*) 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.

   (*b*) 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.

   (*c*) 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.

   (*d*) 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 modelling the state of the channel (i.e. within the network)?

   (*e*) 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. 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.

3. *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*) Taking the IP packet-switching example 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.

4. *Names, addresses, routes*

A *name* is a piece of information which denotes something else. A name's denotation (i.e. the identity of the other thing) depends both on the name itself, and on some *context* or *environment* in which it is used. The context consists of a set of associations mapping from names to their referents, and these associations are called *bindings*.

The following are all names used in everyday networks. The *scope* of the name, i.e. the most general context in which it can successfully be resolved, is given in brackets after the name. In each case, say

(*i*) whether the name is from a *flat* or *hierarchical* namespace, or from a space of some other structure;

(*ii*) what type of object the name denotes;

(*iii*) whether the name is interpreted as an *address*, and if so, between which two entities it denotes a *service access point*;

(*iv*) whether the name is interpreted a *route*, and if so, what process is used to deduce the next step along the route.

(*a*) `Stephen.Kell@cl.cam.ac.uk` (Internet mail)

(*b*) `01223 736709` (UK voice telephone network)

(*c*) `00:11:11:4A:A1:36` (an Ethernet)

(*d*) `128.232.9.85` (the Internet)

(*e*) `/var/spool/mail/srk31` (a Unix filesystem)

(*f*) `Edinburgh Waverley via Peterborough` (the UK's rail network)

5. *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 *acknowledgements* 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 acknowledgement (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 acknowledgements 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.

6. *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 minimise 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	81.5.6.6	1
128.232.0.0/16	203.20.203.1	2
80.2.192.0/18	41.230.5.250	3
default	203.20.203.203	2
81.5.0.0/17	none	1
203.20.192.0/18	none	2
41.230.5.0/24	none	3

(*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.

7. *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 why 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.

8. *Compression, encryption, and other higher-level coding*

   (*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*) Symmetric-key cryptographic coding is useful for *authentication*, *integrity* and *confidentiality*. Give one example of a real system which uses each, specifying what the message (or challenge) is in each case.

9. *Higher-layer Internet protocols; standards bodies*

(*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.

(*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*) 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.

10. *Multiplexing basics*

(*a*) Give an example of multiplexing in a real system. Using your example, explain the relation between multiplexing and coding. Explain your example's *policy* 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 dialling a number and being connected (i.e. hearing the remote ringer) is variable.

11. *Shared media multiplexing in local area networks*

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

(*b*) 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.

(*c*) 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.

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

(*e*) 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.

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

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

12. *Multiplexing finalé*

(*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.]

13. *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 is the better option. What external factors might influence your decision?

14. *Digital channels, modulation and transmission*

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

(*b*) Why is synchronisation usually not an issue for transmission of analogue signals? [Bonus: can you think of a scenario where it *would* be an issue?]

(*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*.

15. *Coding, digitization, error detection and error correction*

(*a*) For each kind of coding mentioned on slide 5-3 of the notes, give one example of how and where it is used in a real system.

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

(*c*) 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.

(*d*) 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.

16. *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*) The Bluetooth CRC algorithm uses the polynomial
$x^{16} + x^{12} + x^5 + 1$
to produce a sequence of 16 check digits for each outgoing (unencoded) message.

(*i*) Explain why this polynomial will detect all one-bit errors.

(*ii*) Explain why this polynomial will detect all errors consisting of an odd number of incorrect bits.

[You need not give full proofs, but be as convincing as you can.]

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