INT20CN Computer Networks

Tutorial #2

1. What is a *protocol*? How does the use of the word "protocol" in computer networking differ from its meaning in other contexts?
2. What is a *protocol data unit* (PDU)?
3. What tasks are performed by the *transport service module* (or layer)? Name at least two.
4. What is the major function of the *network service module* (or layer)?
5. In the example from the lecture (slide 6), the transport service layer (or module) split the application data unit into two TPDUs. Give reasons why it might do this.
6. In the lecture, it was claimed (slide 6) that the network service module did not have to guarantee reliable delivery of packets. Why not?
7. **Harder**: Is it possible that the network layer could additionally split a TPDU into two or more NPDUs? Give reasons why it might do this, and suggest the likely format of the resulting NPDUs.

Extra Notes

This tutorial (and the accompanying lecture) were originally developed from material in Chapter 10 of Stallings and Van Slyke.

Tutorial #3

1. What are the characteristics of the service delivered to application processes by TCP? Give three important aspects.
2. The TCP protocol is commonly used in the context of a *client-server* model of communication. Describe briefly what is meant by this term. In particular, what characterises a *server* process? How does a *client* process use the facilities provided by a server?
3. The TCP segment header contains both a sequence number and an acknowledgement number. Discuss the significance of this.
4. Lost TCP acknowledgements do not necessarily force retransmissions. Explain why.
5. It is (usually) possible, by examining the contents of the two "port" fields of a TCP segment to discover whether this particular segment came from a server process or a client. How?
6. The TCP disconection mechanism (not covered in the lecture) is particularly complex. This complexity is due, in part, to what is sometimes referred to as the "Red Army - Blue Army" problem. Discuss this problem in the tutorial session, and explain why TCP takes the approach it does. NB: part of the problem is that TCP must not (under any circumstances) lose application data.
7. When a segment (or its corresponding acknowledgement) is lost, the sender will time out and resend. Participate in a discussion in the tutorial about how long the sender should wait before resending.
8. The following gives the contents of a TCP segment, obtained using a network analyser. The values are given in hexadecimal.

```
09 6c 00 19 45 6d 70 01 3c 32 28 7d 50 18 10 00
a6 bd 00 00 45 48 4c 4f 20 72 65 64 67 75 6d 2e
62 65 6e 64 69 67 6f 2e 6c 61 74 72 6f 62 65 2e
65 64 75 2e 61 75 0d 0a
```

What are the source and destination port numbers, the sequence and acknowledgement numbers and the contents of the data area? Did this segment probably originate from a client or a server process?
Note that if you are unfamiliar with hexadecimal notation you may safely skip this question.

Tutorial #4

1. What is meant by the term "**remote login**" in the context of the **telnet** protocol?

2. The idea of **plain ASCII text** is fundamental in IT. For example, computer programs (ie, source code) are always plain text files. And, as we will see in later lectures, many Internet application protocols are based on the exchange of ASCII text messages.

   a) What do we mean by the term "printable ASCII"? How many printable ASCII characters are there?

   b) What is the conventional way to generate the ASCII control codes from a computer keyboard?

   c) What are the particular advantages of "plain ASCII text" compared to other "character sets"?

   d) What are some of the obvious disadvantages of ASCII text?

   e) All computers support ASCII text files. However, the convention for indicating the end of a text line is different in each. Unix (and Linux) systems use a single Linefeed (LF, decimal 13), Macs use a single Carriage Return (CR, decimal 10) and Microsoft systems use both CR and LF, the same as the telnet NVT. What kinds of problems could occur when text files are shared between each of these computer types?

3. What are some of the characteristics of the telnet **Network Virtual Terminal** (NVT)?

4. The telnet NVT "end-of-line" convention delimits lines of text using the two character ASCII control code sequence <CR><LF> (in English: Carriage Return followed by Line Feed). Why do you suppose the designers of the protocol adopted this two character sequence instead of just a single character <CR>, or a single <LF>, or something else entirely?

5. (Philosophical question) The NVT approach used by the telnet protocol means that servers and clients don't need to know the details of the actual terminal or host (if any) at the other end of the connection. The alternative is to perform **terminal emulation**. What is meant by terminal emulation?

6. (Deeply philosophical question) The notion of how the NVT is used in telnet leads on to a (moderately) serious philosophical issue about how to map the requirements of different kinds of computer systems to one another when you need to perform some kind of Networked Computing function. Start by comparing the telnet NVT approach to terminal emulation, and extend the discussion to the more general case.

7. When the **`telnet`** program starts up, it informs the user as follows:

   ```
   ironbark 27> telnet greybox
   Trying 149.144.20.62...
   Connected to greybox.bendigo.latrobe.edu.au.
   Escape character is '^]'.
   ```

   What is the "escape character" used for?

8. The[1] text file for RFC 854 (the telnet protocol specification) contains exactly 854 lines. Do you think there is cosmic significance in this?

9. (Research and discussion question) Once upon a time, The ASCII control codes (ie, the ASCII characters less than 32_{decimal}) had important functions. Do some research, or just infer from the names of some of the control characters, what these functions may have been. Which of the control characters still have important meanings?

10. (Research question) How does telnet perfom option negotiation?

Tutorial #5

1. What is your preferred electronic mail address? In general, what is the format of an Internet email address?

2. In his book Computer Networks 3/e, P646, Tanenbaum quotes a study by Perry and Adam (1992) which reported that *"Some companies have estimated that email has improved their productivity by as much as 30 percent."* Give some reasons why such a spectacular increase could occur from the use of email in a business environment.

3. What is the general format of an RFC822 electronic mail message?

4. The SMTP protocol, in common with many other Internet application protocols, uses a 3-digit code in all communications from the server to the client. From the example given in the lecture, can you generalise about the significance of the first digit of this code? What about the second digit?

5. What is MIME? Explain briefly how MIME works to send a file (eg, a GIF image) as an attachment to an email message.

6. A MIME-encoded email message is still a valid RFC822 message. How is this achieved, and why was it it regarded as important?

7. The MIME Type of an email attachment provides a hint to the "user agent" software as to how to handle the data in the attachment. What do you think the user agent software should be expected to do with attachments of the following types
   a) **`text/plain`**
   b) **`text/html`**
   c) **`audio/basic`**
   d) **`application/octet-stream`**?

8. What is the Post Office Protocol (POP) used for? Why is it important that POP uses **authentication**, whereas SMTP does not? What sorts of commands do you think POP would implement?

9. A file which has been encoded using Base64 is bigger than the unencoded data. How much bigger, on average, would you expect it to be?

10. Philosophical question: in the last slide of today's lecture, some security/privacy considerations of Web-based email systems were mentioned. Do you believe these should be of concern to an average user? Is it possible that privacy may actually be *enhanced* by using a Web-based email system? How? Can you think of any other advantages (and disadvantages of Web-based email services?

11. Deep thinking question: In today's lecture, the slides which discussed RFC821 (SMTP) and RFC822 both referred to email addresses. In SMTP, the **`MAIL FROM:`** and **`RCPT TO:`** protocol messages both specify email addresses --- these are called **envelope addresses**. The header of the RFC822 message which is being delivered also contains **`From:`** and **`To:`** lines -- these are **header addresses**. The interesting question to contemplate is: "what happens if the **`To:`** header address and the **`RCPT TO:`** envelope address are **different**" in an SMTP delivery operation? Where does the mail get delivered?

Tutorial #6

1. Before coming to the tutorial, use your browser to "view source" for this tute sheet. Can you understand the HTML markups used? If you don't, ask for it to be explained in the tutorial

2. Show[1] the <A> tag that is needed to make the string "ACM" be a hyperlink to *`http://www.acm.org`*.

3. In the Web materials for INT20CN, your lecturer (usually) prefers **logical** markups for text emphasis. The alternative is to use **physical** markups. Why do you think purists recommend the use of the logical markup?

4. In the lecture notes, mention was made of the philosophical difference between designing for the "structure" of a Web page, versus designing for its "appearance". Explain the difference between each of these, and comment briefly on their advantages and disadvantages.

5. In this lecture, we discussed the authoring of what we termed **Web pages**. You will also commonly hear people talk **Home** pages and Web **sites**.
   a) What is a Home Page, and how is it different to any other Web page?

b) What is a Web site, and how is it different to a Web page?

6. Philosophical question: Why do I have to deal with this stuff? In general, most Web-page authors do not write HTML code "by hand" -- they either use a Web Authoring package such as DreamWeaver, MS FrontPage, Claris HomePage, etc, or alternatively they use the "Export as HTML" option in their favourite word processor. Give at least one reason why your lecturer regards it as useful for a student in this subject to have a rudimentary knowledge of HTML.

7. Research 1: In the lecture notes, the "hyperlink" example showed the value of the HREF as a full URL. At least two other variations exist for the value of the HREF. Discover what they are, and how to use them. Hint: they're all used in this tute...

8. Research 2: One interesting aspect of the Web is that the Netscape people have defined a *de facto* standard set of 216 "Web Colours" which their browser will render correctly. Other colours will (often) simply be mapped by Netscape browsers to the "closest" of the 216 available. Other browsers have adopted this set of colours as well.
   a) Why do you think Netscape specified 216 colours, when almost all modern computers have at least "8-bit" video, and can therefore display 256 colours?
   b) Find out how you specify "Netscape compatible" colours in the standard hexadecimal RGB format (where, eg, white is "**`#FFFFFF`**").
   c) How should this affect your choice of colours in Web backgrounds and images?

9. Research 3: The Web Accessibility Initiative is a project of the World Wide Web consortium. Find out what you can about this initiative, and its relevance to good design of Web pages.

Tutorial #7

1. In the lecture, the statement was made that the various versions of HTTP are *backwards compatible*. What does this mean?

2. (Slightly tricky question) In HTTP/0.9, there was no way to specify the content type in objects returned from the server. How did the browser know whether it was receiving a Web page (in HTML), a GIF image, a sound sample or whatever?

3. In HTTP/1.0 the `GET` request (and other types as well, but we didn't elaborate in the lecture) is terminated by **two** newlines. Why are two necessary? Wouldn't one newline be sufficient, as in HTTP 0.9?

4. The response from HTTP/1.0 and HTTP1.1 servers is "MIME-compatible".
   a) What does this mean?
   b) What is the MIME type for ordinary Web pages in HTTP/1.0?
   c) Why is the "`Content-length:` " header required in HTTP/1.0?
   d) Why isn't an SMTP-style MIME "`Content-encoding:` " header required in HTTP/1.0 as it is for email?

5. A browser makes the following request to a Web server:

   ```
   GET /Fig1.gif HTTP/1.0<newline><newline>
   ```

   What would the server return? Explain in some detail.

6. What is the HTTP/1.0 `HEAD` request method used for?

7. A client can optionally include a `GET` request method header of the form `If-Modified-Since:`
   a) Why is this used, and what is it called?
   b) Why is the date/time specified in GMT (UTC) instead of local time?
   c) This header takes a date/time value. Is there an Internet standard format for this value? Hint: look at section 3.3 of RFC1945
   d) The HTTP/1.1 **Etag** provides a better mechanism for achieving the same result as an "`If-Modified-Since:`" request. Explain, *very briefly* how this works.

8. The HTTP/1.0 specification permits a `GET` request method to include a "`Referer:` " header in the request. Why is this considered to be a potential privacy issue?

9. Give some of the reasons why HTTP/1.0 is not a highly regarded protocol in the Internet technical community. How does HTTP/1.1 address these problems? At least two points required.

10. Proxy servers can reduce network (download) costs for an organisation by caching recently requested Web documents (pages, images, etc), however they are not altogether successful in practice. Give two reasons why proxy caches are not able to satisfy the majority of Web requests in The Real World(tm).

11. Contemplate this: the specification for the original version of HTTP (0.9) was approximately 6 Kbytes in size. The RFC for HTTP/1.0 (RFC1945) was 134 Kbytes. The RFC for the current version of HTTP/1.1 (RFC2616) is 412 Kbytes. What conclusions can you draw?

Tutorial #8

1. In the RR examples given in the lecture, the TTL field is set to 86400. What is the significance of this strange number?

2. The DNS is described as a "distributed database" of RRs.
   a) What does this mean?
   b) What is the alternative, and why is it generally regarded as unworkable? (Optional philosophical discussion question: How does this second alternative compare (conceptually) with the various Web "search engines" such as Google and AltaVista?)

3. A nameserver acts not only as a server, but also as a client under certain circumstances. What are these circumstances?

4. Why[1] should each nameserver know the IP address of its parent instead of its domain name? Similarly -- when configuring an Internet-connected computer, why is the nameserver always specified as an IP address, not as a domain name?

5. Nameservers are usually (always?) configured to know the IP address of at least one root nameserver, as well as that of their parent nameserver. Why is this?

6. Why do you suppose the rules for nameservers in the Internet are so stringent in the matter of off-site "replication" servers?

7. What is a **reverse lookup** in the DNS, and why is it regarded as a significantly harder problem than normal lookups?

8. What is the significance of the fact that machine **`luga.latrobe.edu.au`** appears in an **`MX`** RR (Resource Record) for machine **`ironbark.bendigo.latrobe.edu.au`**?

9. A nameserver query contains a parameter bit which is set to **`1`** if recursion is desired at the server and **`0`** otherwise. What would you expect to be the result of queries in each of these situations?

10. Research & discussion question: Most (all?) implementations of the domain name system appear to allow **abbreviations** of names so that, for example, the name **`ironbark`** resolves to a correct address for machines at located at the Bendigo campus. How is this handled by the DNS, and whereabouts is it implemented (ie, in the nameserver/s or in the resolvers)? What about **`ironbark.bendigo`** -- can this be handled?

11. Implementation question[2]: The standard suggests that when a program needs to find the domain name associated with an IP address, it should send an inverse query to the local server first and domain `in-addr.arpa` only if that fails. Why?

Tutorial #9

1. Assume you're writing a TCP (socket) based client/server application, using Java code similar to that given in the lecture. Why is important that the server program be started up (ie, executed) before the client program?

2. How does the **`accept()`** socket method in Java indicate that a TCP connection has been established?

3. The **`getInetAddress()`** and, to a lesser extent **`getPort()`**, methods are probably more useful for a socket in a server program than in a client. Why?

4. What does it mean to say that the simple Java server program given in the lecture is **single threaded**? What is the disadvantage of a single-threaded server, and how is the alternative implemented? Can you think of any particular advantages of a single-threaded server?

5. In the lecture it was claimed that the 4-tuple which identifies a TCP connection will always be unique. Suppose you have two telnet sessions running on your local desktop machine, both logged in to **`ironbark`**. How are each of these connections uniquely identified?

6. Suppose you were to run the example server program from the lecture on machine ironbark, without changing the port number from 7277, as given there. Someone else has the same idea, and simultaneously runs a copy of the program on ironbark. What will happen?

7. In the "traditional" Unix/C implementation of sockets, a socket is first **created**, and then becomes either a client or server socket depending on subsequent operations performed on it -- eg, **`connect()`** for a client socket and **`bind()`** followed by **`accept()`** for a server. How does this compare with the Java approach?

8. Look at the Perl socket client given in the lecture. What comments can you make on the code, compared to the Java implementation?

Tutorial #10

1. The following are some (possibly hypothetical) IP addresses:

   ```
   205.184.10.20   139.130.17.42   138.80.128.18
   10.170.45.56    149.144.20.82   192.54.252.7
   ```

   Extract the **network number** and the **host number** from each of these, stating what **class** of network it is.

2. An IP address with all zeros in the host part is said to be a **network address**, and is not available to be allocated to any host on the network. What does this mean? Give an example of a Class B network address.

3. Calculate *exactly* how many hosts each of the traditional three classes of IP address (class A, B & C) can support. Remember that some addresses are reserved for special purposes.

4. What[1] is the chief difference between the IP addressing scheme and the International (and Australian) telephone numbering scheme?

5. When a computer is being configured to connect to the Internet, in addition to setting its IP address the system administrator also usually configures an **address mask** or **netmask** parameter. Why is this needed? Is it likely to have a (sensible) default value?

6. Why was it necessary to introduce **subnetting** into the IP address structure?

7. The IP address **`127.0.0.0`** is reserved as the Internet **loopback address**. What do you think this means?

8. In the lecture, it was stated that the broadcast address is an IP address with all 1's in the **host part** of the address. However, many systems in practice use a broadcast address format whereby **all of the bits** are 1, ie `255.255.255.255`. How should this address be interpreted? Optional harder question: a broadcast address of the form discussed in the lecture is sometimes called a **directed broadcast** address. What do you think this means?[2]

9. One of the weaknesses of the IP addressing scheme is that when a machine is physically moved from one network to another, its address must change. Why is this so, and why is it a problem?

10. A certain company desires to give the computers in (eg) its sales office full-time connections to the Internet. It needs 25 unique IP addresses. In the modern CIDR address allocation system, what will be the "**`/x`**" specifier at the end of the address block which is allocated?

Tutorial #11

1. What do you think is the origin of the term **datagram**?

2. Why must a router always have at least two different IP addresses? Can a router have more than two IP addresses?

3. An IP datagram encapsulates a TCP segment. The TCP segment, in turn, encapsulates portion of an HTTP GET request. Draw a diagram showing the boundaries between the various headers, and the location of the application data in the datagram. Assume minimum-sized headers.

4. The three characteristics of datagram delivery in the Internet are:
   a) Unreliable
   b) Connectionless
   c) Best Effort

   Explain how each of these terms is interpreted in the context of datagram delivery.

5. What is UDP? In the lecture, some examples where UDP is useful were given. However, one of the appropriate uses of UDP occurs in **multi-player (networked) games**. Why is it likely to be particularly appropriate in this application?

6. Consider[1] a machine with two physical network connections, and two IP addresses, I(1) and I(2). Is it possible for that machine to receive a datagram destined for I(2) over the network with address I(1)? Explain. Hint: consider the algorithm for datagram delivery given in the lecture.

7. Is[2] it possible to address a datagram to a router's (IP gateway's) IP address? Does it make sense to do so?

8. The following is the output of a run of the `traceroute` command, "looking from" outside La Trobe back towards Bendigo. It was, in fact, run on a Unix system called `morinda` in the School of IT at NTU in Darwin.

```
morinda> traceroute ironbark.bendigo.latrobe.edu.au
    [....six lines deleted]
 6  nsw-vic.atm.net.aarnet.edu.au (192.12.76.2)  79 ms  79 ms  79 ms
 7  vic-gw.vrn.EDU.AU (203.21.130.162)  80 ms  80 ms  87 ms
 8  latrobe-gw.vrn.EDU.AU (203.21.130.133)  81 ms  83 ms  80 ms
 9  r-elt-fddi.latrobe.edu.au (131.172.20.8)  80 ms  81 ms  80 ms
10  r-bgoatm34-atm.latrobe.edu.au (131.172.239.5)  81 ms  328 ms  84 ms
11  busfddi0.bendigo.latrobe.edu.au (149.144.10.1)  81 ms  81 ms  81 ms
12  ironbark.bendigo.latrobe.edu.au (149.144.21.60)  81 ms  81 ms  81 ms
morinda>
```

   Use the information contained in this `traceroute` output to fill in the missing IP addresses in the "Internet Structure" diagram in the lecture.

Tutorial #12

1. What does CSMA/CD mean? CSMA/CD is sometimes referred to as the "polite dinner table" algorithm? Can you think of a reason for this?

2. Ethernet/802.3 is a **shared medium**, **multi-access** network technology.
   a) What do these terms mean?
   b) Does this pose any potential security risks? Explain. What about a network based on a switching hub?

3. What is interesting about the Ethernet/802.3 "MAC address"? Discuss.

4. In the lecture, it was stated that an Ethernet/802.3 collision occurs when two stations start to transmit at the same time. Discuss the meaning of the term **at the same time** as used in this context.

5. The 10baseT configuration has captured the Ethernet market from thin wire. Discuss reasons why network planners and managers might prefer this technology.

6. **Switching Hubs** build a table which maps destination MAC addresses to port numbers. How do they build this table?

7. An IP packet is "encapsulated" into an Ethernet frame for delivery in the network -- this is IP "local delivery". How will this encapsulation differ in the case where the packet is sent to a router for "Internet Delivery"?

8. In slide 6 of today's lecture, it is stated that the key difference between Ethernet and IEEE 802.3 LANs is the meaning of the 16 bit "type" field (used as a "length" field in 802.3). In many LANS, Ethernet and 802.3 frames co-exist perfectly happily. How can a receiving station know whether to treat the field as a **length** or as a **type**?

9. Assume[1] a one megabyte file must be transferred across a network. Ignoring delays caused by waiting for access and other overhead (ie, counting only the data transferred), how long would it take to send the file across an Ethernet? Across a Fast Ethernet?

10. Research question: It's not required knowledge for this subject, but you might care to investigate the Internet **Address Resolution Protocol** (ARP), which provides a mapping between the IP address of a host and its MAC address -- the question is: *How does a host discover the MAC-level (Ethernet) address of another host on the same network, if all it knows is its IP address?* This mapping is obviously needed to enable "local delivery" of a datagram.

11. Engineering research question[2]: A shared medium (non-switched) Ethernet is generally regarded as heavily loaded (approaching overloaded, in fact) if the network utilisation goes over (approximately) 20%. This, on the face of it, seems a low value. What do you think is going on?

12. Opinion question: ATM is **probably** the dominant high-speed networking technology at present. What do you think is the particular attraction of ATM networks over other high-speed technologies, on the basis of the material presented in the lecture?

13. Serious Research question #1:In the lecture, it was mentioned that Gigabit Ethernet is **compatible** with 10/100Mbps Ethernet. What does this mean? It's OK to guess...

14. Serious Research question #2: What is the historical origin of the difference between the frame formats of Ethernet and IEEE 802.3?

Tutorial #13

1. Why is a modem needed for data communications over the telephone system?

2. The fastest external modems which can be purchased at present operate at 56Kbps, yet the serial ports on most home computers are set up to operate at 115Kbps or faster.
   a) Why the disparity?
   b) Why does the serial port operate at a weird speed like 115200bps?

3. What is a **null modem** and why is it sometimes needed where RS232 interfaces are used to built point-to-point data links? Describe briefly the connections required in a minimum RS232 null modem (ie: one which uses only pins 2, 3 & 7).

4. What is the link efficiency (or utilisation) in an asynchronous system which sends 8 bits of data with one start bit and one stop bit? What if the data was only 7 bits, as in ancient ASCII data links? How many 8-bit bytes per second can be transmitted using a 28.8kbps modem (ignoring the possibility of compression)?

5. The Department of Information Technology at Bendigo has a dial-in router (for staff use only, sorry!) which is connected to subnet 20, ie **`149.144.20.0/24`**. When registered users dial in to this router, what would you expect the network/subnet part of their home machines IP address to be?

6. In a dial-in situation, IP addresses are usually "dynamically-allocated", and therefore different for each dial-in session. It's usually possible to pay a somewhat higher rate but have a "static" IP address, which is the same for every dial-in. Why do you think dial-in accounts with dynamically-allocated addresses are cheaper?

7. How are IP addresses normally allocated in the situation where a point-to-point link is used to connect two routers together? Can you imagine a more address-space-efficient way of allocating these addresses?

8. What is the function of PPP in a data link using modems?

9. What are the characteristics of a "Basic Rate" ISDN service?

10. Research Question: when a dial-in Internet user connects to an ISP using PPP, their machine has to somehow discover its own IP address. How do this think this happens? Harder research question: in the olden days, when SLIP was commonly used for dial-in access, the protocol provided no support for allocating an IP address to a remote user. How did the dial-in machine discover its own IP address using SLIP?

Tutorial #14

1. What is a **leased line**? Why is this considered an outmoded term nowadays?

2. What is meant by the term **peering** in the context of Internet Network Providers?

3. What is meant by the term **Basic Carriage Service** (sometimes called a **Telecommunications Service**) in the context of a full-time Internet connection?

4. Two aspects of a generic ISDN service which have been emphasised by the marketing people are **dial on demand** routing, and **bandwidth on demand** data transfer. What do you think these terms refer to? A Telstra local ISDN data call in Australia has a flagfall cost (day rate) of 20 cents (which includes the first 180 seconds of the call) plus 0.05 cents per second after 180 seconds has elapsed. Under what conditions do you think it would be reasonable to configure a local-call ISDN-based system to use "dial on demand" and/or "bandwidth on demand"?

5. What is the attraction of a **frame relay** telecommunications service over an ISDN service at the same "port speed"?

6. Imagine that you (as a graduate of this subject) have been asked to advise a small Australian business on establishing a permanent connection to the Internet. As part of your work, you decide to investigate the performance and price options available from Telstra, the dominant telecommunications (and Internet Service) provider in Australia. Before you attend this tutorial class, check out Telstra's Bigpond Direct Web pages, and be prepared to answer the following questions:

   a) Telstra splits Bigpond Direct (see also here) pricing into four components. What are they? You're excused if you can't immediately see what the fourth one is...!
   b) What are some of the options for the "Basic Carriage Service"?
   c) You should have discovered that one of the options for Basic Carriage Service is a dial-in modem link. Why is this a particularly attractive option?
   d) Some (all?) of Telstra's pricing options for Bigpond Direct have a data volume component. What does this mean, and which ones have this "feature"? What is the particular disadvantage of this pricing model in a business context? Does it have any advantages? Do any of the pricing options not have a volume component?
   e) **VERY Hard question:** Telstra's OnRamp XPress has some attraction as a Basic Carriage Service for a full-time Internet service. What would the minimum monthly cost for a business using this service for its Internet link, including the Bigpond Direct charges? Don't even **try** to incorporate the Back Channel Tariff stuff into your calculations! NB: I don't *seriously* expect you to do this -- I've tried and failed myself! However, if you're an Accounting Major you might find it fun to try.
   f) **Research and extension question:** What other issues will your client business have to have in place in order to establish a full-time "Internet Presence"?

7. In Australia, it's possible to purchase (in some capital cities) a full-time ("always on") Internet service from (eg) Optus@Home which doesn't separate the Basic Carrier service from the Internet service in the same way as Telstra does, and which doesn't have a data volume component -- there's simply a single monthly charge. How can they do this? Discuss. Note: this service is also not terribly useful for business-style permanent Internet connection. Why not?

Tutorial #15

1. In the lecture, the concept of *abstract* versus *concrete* representation of data types was introduced. Explain briefly, with examples, the meaning of each of these terms.

2. Why are the Basic Encoding Rules (BER) needed for ASN.1?

3. Express the following ASN.1 specification in terms of an programming language such as C, C++, Java, etc, with which you are familiar:

```
name ::= OCTETSTRING -- or IA5String, see subranges, later.
married ::= BOOLEAN
yrsWithCompany ::= INTEGER
```

4. Describe how each of the specifications in the previous question would be encoded using BER. Note that we don't have actual values for the data items, so you'll have to make some up.

5. The following is a record declaration in the Pascal[1] programming language. Express this in terms of ASN.1

```
personnelRecord = record
    name : array[1..100] of char;
    yrswithCompany : integer;
    married : boolean
end ;
```

6. Which of the following ASN.1 syntactic elements would you expect to be keywords, types and variables?

```
AP-Title
ipInReceives
ENUMERATED
ObjectDescriptor
OCTET STRING
sysDescr
```

7. You receive the following string of BER-encoded octets, values given in hexadecimal. What does it mean?

```
30 08 02 01 03 04 03 48 69 21
```

[1] Yes, I know you don't study Pascal in any of our courses, but the declaration given here is pretty obvious, and you really shouldn't have any trouble with it! If you prefer, you could pretend it was Delphi, because the syntax is (almost?) the same...

Tutorial #16

1. What information can the network manager obtain from the `ping` command? Give at least 3 answers. What does `ping` actually do?

2. What does the `traceroute` command tell the network manager? Why is `traceroute` not recommended for use in regular network monitoring? (**Hard research question**:How does traceroute work?)

3. The ASN.1 APPLICATION-specific data types of SNMP mostly use the *sub range* feature of ASN.1, which we didn't actually mention in our introductory ASN.1 lecture. What is the significance of the sub range values for the SNMP ASN.1 APPLICATION types *Counter* and *Gauge*?

4. The following are some (highly edited) entries from the standard Unix MIB definition, in file `/etc/mib.txt`.

```
ipForwarding OBJECT-TYPE ::= { ip 1 }
icmpInEchos OBJECT-TYPE ::= { icmp 8 }
tcpMaxConn OBJECT-TYPE ::= { tcp 4 }
```

   What are the full numeric `OBJECT IDENTIFIERS` of the objects `ipForwarding`, `icmpInEchoes` and `tcpMaxConn`.

5. (Extension Question) One interesting aspect of the ASN.1/BER is the way in which `OBJECT IDENTIFIERS` are encoded for transmission. In general, the integers which specify the `OBJECT IDENTIFIER` are simply encoded in BER as a `SEQUENCE` of single byte values. However, the first two integers (let's call them *a* and *b*) are encoded in a compact form, taking only a single byte, of the form $40a + b$. So, for the Internet, the first two integers are 1.3, therefore they are encoded as the single byte value 43. Can you imagine a reason why this is done? What does it say about the values of these first two integers?

6. (Extension Question) Marshall T. Rose[1] espouses the fundamental axiom of network management, which is: *The impact of adding management to managed nodes must be minimal, reflecting a lowest common denominator*. Discuss.

[1] Internet and network management demigod.

Tutorial #17

1. How would you request a router to return the actual values of the objects **`ipForwarding`**, **`icmpInEchoes`** and **`tcpMaxConn`**? Give solutions using each of the **get** and **get-next** commands. Use a command syntax of your choice, although the CMU SNMPlib syntax given in the lecture notes would be the most appropriate.

2. The following is a diagrammatic view of portion of a table (**`ifTable`**) in the interfaces portion of the standard MIB, edited to fit the page. The table consists of a sequence of **`ifEntry`** elements. Values shown are from the router **`r-bgowan`** at Bendigo, which is nowadays used as a backup (over ISDN) to the microwave link to Bundoora.

	ifIndex	ifDescr	ifType	ifMtu	ifSpeed	[illegible]	ifOperStatus
Interface 1	1	Ethernet/0	ethernet-csmacd(6)	1500	10000000	00 00 0C 42 45 ID	up(1)
Interface 2	2	Serial/0	Prop Poin To Poin Serial (22)	1500	1544000		down(2)
Interface 3	3	Serial/1	Prop Poin To Poin Serial (22)	1500	252000		up(1)

   a) Describe interface 3 on this router.
   b) Draw the OBJECT IDENTIFIER subtree, in "tree" format, in the region which defines the various **`ifSpeed`** entries. You might find it helpful to refer to your lecture notes for some useful information here.
   c) What would be the structure of an SNMP **`get-request`** to discover the speed, in bps, of interface 1.
   d) What value would be returned by **`get-next(...ifSpeed.2)`** ?

3. Why do SNMP proponents use the expression "**powerful get-next**"? In other words, what problem does the **get-next** operation solve very elegantly?

4. (Philosophical, but still practical, question) In this section of the unit, we have played pretty "fast and loose" with abbreviations of OBJECT IDENTIFIERs, eg **`{ system sysDescr }`**. Of course, in the "Real World", we would normally have to be more careful to ensure that the software we were using was able to unambiguously fetch the variable we desired. How should SNMP software resolve such abbreviations, especially with numeric OBJECT IDENTIFIER strings? Hint: think of how Fully Qualified Domain Names (FQDNs) are specified, compared to abbreviated versions.

5. (Philosophical question) What is the purpose of the SNMP portion of the MIB?

Tutorial #18

1. What are some of the standard security attacks which The Bad Guys can make against an Internet-connected computer system? What are the implications of "springboard" attacks for security of so-called "unimportant" systems?

2. What is meant by the term "packet filtering firewall"? Why would such a device be used? What are some of its limitations?

3. In the lecture, a diagram was presented showing a **DMZ & Bastion Host** firewall structure. Describe in detail how each of the two packet-filtering (firewall) routers would be configured in this structure.

4. The firewall examples given in the lecture all assumed a single point of connection between a business's internal network (or Intranet) and the outside Internet. How would the situation be complicated if there were multiple connections?

5. You have been asked to configure the Bendigo "gateway" router `r-bgoatm34` to prohibit traffic from subnet 8 (ie, `149.144.8.0`) from crossing the microwave link to Bundoora. Define an access list (address and mask pair) which will do this, using the syntax from the lecture.

6. The La Trobe "gateway" router blocks connections made to TCP port 80, except under certain conditions. What are these conditions?

7. The "Firewall and DMZ" configuration discussed in the lecture protects the "internal" hosts from most types of security attacks, but **not all**. For example, internal hosts could still be vulnerable to **virus** (various forms), **worm** and **trojan horse** attacks. Discuss these issues.

8. In the lecture, a minimal firewall structure was suggested whereby the "gateway router" (or host) for an organisation serves as in a similar function to a combined firewall and bastion host. This type of structure is sold by several vendors as an economical solution to Internet security. How would you expect the firewall/host system to be configured?

9. (Philosophical Question) Discuss some of the legal and ethical questions alluded to in the last slide of today's lecture.

Tutorial #19

1. The following ciphertext was created using a **Caesar Cipher**: *FUBSWRJUDSKB LV IXQ*. Discover the plain text message.

2. Why are monoalphabetic substitution ciphers not regarded as being very secure? Discuss briefly some known vulnerabilities of these cryptosystems.

3. Encode the plaintext string "*where will we meet*" using the transposition cipher described in the lecture with a key consisting of the word "bendigo".

4. The DES (in its original form) used a 56 bit key.
   1. How does this compare with the number of "bits" in the key for a typical ATM card? NB: think about how many bits are needed to represent the known key space size.
   2. What is the key size in bits for typical Unix passwords chosen from the 96 character printable ASCII character set?
   3. What if the Unix password was only chosen from the set of upper and lowercase letters and the 10 digits?
   4. (Practical question) The security of a PIN system, as used in auto teller machines, is actually higher than it might at first seem. Why?
   5. What is the key size in the **XOR**-based monoalpabetic cryptosystem described in the lecture?

5. For each of the keys discussed in the previous question, *how long* would it take to search the entire key space if one key can be tried every 0.1 microseconds (ie, 10^7 keys tried per seconed)? This is called a **brute force** attack on a cryptosystem.

6. The following string of bits is ciphertext which has been encrypted using a *one-time pad*, to which you have (through your well-paid spies) discovered the key. Use your cryptographic knowledge to crack the code and discover the plaintext message. Some (possibly useful) ASCII codes are given below to convert the resulting plaintext bit string into English text.

   ```
   Ciphertext: 0001010 0001001 0000010

          Key: 1001011 1000101 1000111
   ```

   Some useful ASCII character codes:

   ```
   A:  1000001 B:  1000010 C:  1000011 D:  1000100 E:  1000101
   F:  1000110 G:  1000111 H:  1001000 I:  1001001 J:  1001010
   K:  1001011 L:  1001100 M:  1001101 N:  1001110 O:  1001111
   ```

7. Explain briefly the difference between the **electronic code book** and **cipher block chaining** modes of DES. Of these, why is cipher block chaining normally used?

8. Why would you not use Vernam Cipher for large messages?

9. More usable one-time pad systems (of the kind that **Real Spies**(tm) might use) can use a variety of encryption functions. For example, one system uses a sequence of random numbers in the range of 0 to 25 as the key. How would this work? Is it secure?

10. One of the biggest problems with single key encryption is to do with key management. Propose some methods of distributing keys for single-key encryption. Discuss their advantages and disadvantages.

11. (Research, do this in your own time) Modern security systems normally use **Triple DES**, briefly mentioned in the lecture. Discover how Triple DES works in practice.

Tutorial #20

1. What are the advantages and disadvantges of a public key cryptosystem compared to a single key system? Draw a table giving at least one advantage and one disadvantage of each system, mentioning such problems as key management and encryption/decryption speed.

2. In the RSA example given in the lecture, what aspect of the system makes it difficult to discover K_s (the decryption, or private key) given that you know K_p, the public encryption key?

3. One of the major disadvantages of a public key cryptosystem based on RSA is its slow speed of encryption and decryption. Explain briefly why the RSA algorithm is slow compared to a single-key encryption system such as DES. How can public key and single key encryption be combined to give a secure, but fast, encryption system?

4. In the public-key authentication protocol given in the lecture notes, in message 3 (sent from A to B), R_B is encrypted with K_S. Is this encryption necessary, or would it have been adequate to send it back in plaintext?

5. What is the difference between a *digital signature* and a *message digest*? What are the advantages and disadvantages of each?

6. In the lecture notes, it was claimed that no one can generate two messages that have the same message digest. How can the system designer ensure this?

7. The Unix system uses a scheme with some similarities to a message digest for storage of user passwords. In what ways are these similar? NB: if you don't use Unix, or use it infrequently, you may be excused from this question.

8. (Requires basic mathematical skill. This will *probably not* be on the exam!) Using the RSA cryptosystem with $p = 7$ and $q = 11$,
   a) Write down the values of n and X.
   b) Find a suitable value for E. What is the public key K_p corresponding to these values?
   c) Discover a legal value for D. What is the private key K_s corresponding to these values?

9. (Advanced - maths majors only. This will *not* be on the exam!) Again using an RSA cryptosystem, this time with $p = 13$, $q = 31$ and $D = 7$, find E.

10. (Advanced - maths majors only. This will *not* be on the exam!) Using the results of the previous question for p, q, D and E,
    a) Ascertain the largest message size (in bits) which can be encrypted using these values, and
    b) Demonstrate the encryption and decryption of a message.

11. (Very advanced - maths majors only. This will *not* be on the exam!) In the lecture, a demonstration was given of a RSA-like public key encryption system, albeit one using very small primes. The numerical results were given without justification. Attempt to verify the correctness of the calculations presented there.

Tutorial #21

1. Can you think of any other forms of electronic commerce than those mentioned in today's lecture?

2. What is Electronic Data Interchange? How is it different to the Internet and Unix electronic mail facilities you have used?

3. Describe three (or more) advantages of using EDI compared to manual systems. Do any of these seem more important than others?

4. Describe the **three** key components of an EDI system. Are all three of these necessarily going to be required in every EDI implementation?

5. What is the significance of each of the X.12 and EDIFACT standards for EDI?

6. Comment on the EDIFACT message structure. Why is it regarded as somewhat **old fashioned**?

7. (Philosophical question) The emergence of the Internet has changed the way in which many businesses do their computer networking. Discuss the options for EDI in the Internet context. Are there likely to be cost benefits from using the Internet instead of a proprietory EDI service? What factors do you think are limiting the use of the Internet for EDI? What does the future hold?

Tutorial #22

1. The following markup appears in the context of a form on a Web page. How would you expect it to appear on a Web page?

   ```
   <INPUT TYPE="TEXT"  NAME="Name" MAXLENGTH="64" SIZE="20">
   ```

2. What is meant by the following user interface terms:
   - Radio button
   - Checkbox

3. The following string of text has been **urlencoded**. What might the original form have looked like? Artist's impression required!

   ```
   name=Phil+Scott&sex=male&occupation=Ace+Lecturer
   ```

4. The **`GET`** method is recommended for submission of form data if the submission of the form does not have **side effects**. What does this mean? What is the alternative?

5. The following URL is typical of those observed in queries to the AltaVista search engine.

   ```
   http://www.altavista.com/cgi-bin/query?locale=au&q=%22phil+scott%22+bendigo&search=Search
   ```

   a) Does this URL suggest that the **`<FORM METHOD=>`** used to generate it was specified as **`GET`** or **`POST`**?
   b) What is the name of the CGI program which is executed by the Web server?
   c) Give, in plain text, the query string which was used to generate this URL.
   d) Practical exercise: although this URL has been (slightly) edited, it still generates a valid AltaVista search. Copy it from this Web page, and paste it into an "Open Page" dialog in your Web browser. What does it return?

6. How does a CGI program written for a **`GET`** method obtain data from a form?

7. When an HTML <FORM> is specified to use **`METHOD=POST`**, explain how the browser sends the form information (or **`QUERY_STRING`**) to the Web server. How does the CGI program access the information in this case?

Tutorial #23

1. What is a **shopping cart** application?

2. What is meant by **state maintenance** in the context of a shopping cart application? What are the two major technologies which can be used to implement state maintenance?

3. Typically, what information do you think would be contained within a hidden field or cookie? There are a couple of ways you can think about this question, depending on how much of the "state information" is maintained at the server, and how much on the client side (browser) software.

4. What are some of the advantages of cookies over hidden fields? What disadvantages do they have?

5. Under what conditions is a cookie stored on a client system's local disk between "browser sessions"?

6. Discuss the security implications of cookies. In particular, if someone asked you whether it's safe to accept cookies from Web servers, what would you tell them, and why?

7. On many Web Commerce sites (for example, Amazon.com and CDnow.com), cookies are used to authenticate repeated visits to the site. For example, if you have "shopped" at either of the above businesses, they will set a cookie so that you can subsequently "one-click" (or somesuch) to order. It's obviously important that no one else can generate **your** cookie, or they could impersonate you. How could this be implemented?

8. (Hard) What controls do the `domain` and `path` specifiers impose on when your browser sends a cookie to a server? In other words, how are the `domain` and `path` specifiers interpreted in the browser?

9. (Research question) Sites such as Amazon.com maintain a session identifier in URL **Extra Path Information**. Discover how this works, and explain its advantages over other systems. Why would they do this if they can achieve exactly the same effect using cookies?

10. (Discussion question) There's obviously lots of potential for using Java and/or Javascript to build a shopping cart application which runs on the client (browser) instead of using FORMS and CGIs. Is this a good idea? Why, or why not?

Tutorial #24

1. What is a **site certificate**, and why is it considered a desirable (if not essential) tool for conducting Electronic Commerce on the Web?

2. As a Web user, how can you tell if the site you're communicating with is using SSL security?

3. Why is a site certificate encrypted using the private key of the CA which issued it?

4. Why would you want a personal (client) certificate, analagous to a site certificate? Under what circumstances might client cerificates be important?

5. Some time ago (whilst I was busily surfing to the Dilbert site), Netscape presented me with the following message:

   > *The certificate that the site 'www.unitedmedia.com' has presented does not contain the correct site name. It is possible, though unlikely, that someone may be trying to intercept your communication with this site. If you suspect the certificate shown below does not belong to the site you are connecting with, please cancel the connection and notify the site administrator.*
   > *Here is the Certificate that is being presented*:
   >
   > | | |
   > |---|---|
   > | *Certificate for:* | *United Media* |
   > | *Signed by:* | *RSA Data Security, Inc.* |
   > | *Encryption:* | *Export Grade (RC4-Export with 40-bit secret key)* |

   What's going on here? How could it happen?