Student ID: ..... NetID/UPI: .....

1. **Transmission Schemes** [10 marks]

   Assume that blocks of printable characters are transmitted on a link as *information frames*.

   (a) How does the receiver recognise the start and end of a frame? [2 marks]

   Special chacarters are used to indicate the start and end; they are STX (start of frame) an dX (end of frame).
   This question only talks about characters wthin frames, you didn't need to say anything about how characters *are sent 'on the wire.'*

   (b) What changes to the above are needed if we wish to carry arbitrary (non-printable) bytes within the frames? [4 marks]

   - We need an 'escape' chacater to indicate that the character following it needs special treatment. That's the DLE (data link escape) character.
   - ETX can appear as data wthin the frame, we must use DLE ETX to mark the end of the frame.
   - DLE STX is used to indicate start of a 'transparent' frame (i.e. one that contains DLEs).
   - To get a DLE within the frame we must precede it by another DLE, that's described as 'character stuffing.'

   What difference(s) arise if we use *synchronous* rather than *asynchronous* transmission on the link? [4 marks]

   3 marks for explaining difference between async and sync ...

   - Async uses start, stop and parity bits to frame each character.
   - Per-character bits are reduced from 11 to 8, so frame transmission is faster (*speed* increase of 11/8).
   - Sync keeps the line busy by sending SYN characters while there's no data char to send. *This question is about* character-based *frames, rather than about* bit-synchronous *transmission.*

   One mark for further comments, e.g. ...

   - No change to character framing (when changing from async to sync transmission).
   - For sync transmission we have to synchronise tx and rx clocks, e.g. by using Manchester encoding of data.
   - Async transmission only requires that the rx and tx clocks be running at *approximately* the same rate.

CONTINUED

COMPSCI 314 S1 C 2006 test. Answers to Questions 2, 3, 4

Q2.

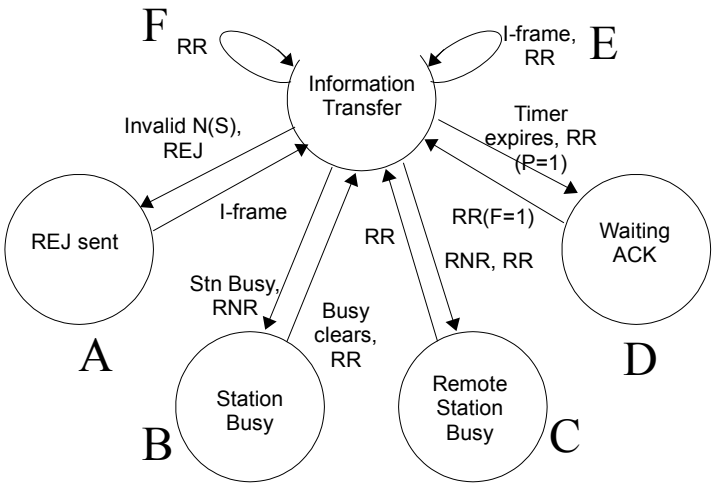
In the following questions, some parts refer to HDLC, and others to aspects of the Internet protocol stack.

Briefly (*no more than 1 or 2 lines*) describe the function of each component

i	supervisory frames	Indicate the **receiving** station status, usually after receiving an information frame. They are Receive Ready (RR), Receive Not Ready (RNR), Reject (REJ) and Selective Reject (SREJ). RR and RNR may be sent regularly as notification of a station status when there is no other traffic.
ii	unnumbered frames	Frames with no sequence number – includes frames to set up and break down a connection, and also unnumbered information
iii	poll/final bit	A bit in the header – set in a command frame (Poll) to require a response, or in a response frame (Final) in response to a Poll. Especially the Final bit may be set on the last of a sequence of Information Frames. Final bit DOES NOT mark end of frame.
iv	command and response frames	A **command** frame is any frame sent by a **primary** station, and a **response** is any frame sent by a **secondary** station.
v	piggyback acknowledgments	A sequence number included in an information frame to acknowledge data in the **other** direction (N(R) in SDLC) Acknowledgment MUST be included in info message to be “piggy-back”.
vi	protocol data unit	A frame as it exists **below** a protocol level, including headers for that level.
vii	physical layer	The lowest level of a protocol stack, used to send bits over a link
viii	network layer	The level of a protocol stack which sends messages between end-nodes, including routing.
ix	transport layer	An upper protocol level that ensures correct end-to-end packet transmission and reassembly, and recovery from broken connections
x	application layer	The topmost protocol layer, dealing with user activities such as file transfer, email, web access, etc

[10 marks]

Q 3.
The diagram shows part of the state diagram for HDLC/SDLC.



Major parts of the diagram are labelled with large capital letters (A, B, …). Explain which letter or letters are involved with each of the following actions, and briefly explain each action –

NOTE: The circles around the edge are states of this station and must not be confused with transmissions to or from the remote station.
The arrows show state transitions, not data transmissions.

i	Normal information transfer, receiving station stays ready	E	Station receives an I-frame and responds by sending RR (Receive Ready) response (which will include the N(S) sequence number acknowledgment).
ii	Idling with no information to send by either station	F (D)	Station receives an RR and immediately waits for another message, taking no other action. State D might be involved as well, if the other station sends nothing while idle
iii	This station becomes busy (include signalling when it becomes ready again)	B	Station becomes Busy, sends RNR (Receive Not Ready) and enters "Station Busy" state. When Busy clears, send RR and return to "Information Transfer" state. (NOTE RNR & RR reflect the state of **this** station.)
iv	The remote station becomes busy (include signalling when it becomes ready again)	C	This station receives Receive Not Ready (RNR), acknowledges it with RR and enters "Remote Station Busy" state until remote sends RR. (RNR and RR are sent by the remote to signal its state; this is the mirror of (iii) above.)
v	There is no response from the other station	D	Timer expires; send RR to say I am still ready and enter "Waiting ACK" state until remote sends RR to signal it is ready. (P=1 requires response). It is likely that the timer will expire again, so that RR frames will be sent at regular intervals. Note that the sent RR has (P=1) forcing a reply if at all possible; the matching RR (or possibly RNR) **must** have a matching F=1.

Q 4.

A Hamming code is described as a (7,4) code. Explain –

i.	the purpose of a Hamming code	To allow correction (not detection) of errors in the received data
ii	the significance of the "7".	The length of the **codeword** (data+parity)
iii	the significance of the "4".	The number of **data** bits in the codeword. **DO NOT** go on to say "3 parity bits"—that is the next part!
iv	any other important **quantities** following from the (7,4) description	The number of parity bits is 7 - 4=3. **NOTE** how parts (ii) (iii) and (iv) here each ask **one** thing only—don't give all the answers in one part where only one point is needed.
v	the properties of a (15, 11) Hamming code.	It has a codeword length of 15 bits with 11 data bits (and hence 4 parity or check bits)

[5 marks]

A (7,4) Hamming code is known to use even parity and is shown with its bits in the order 7…1, (7 on the left, 1 on the right).

Decode and comment on the two following codewords, recovering the correct information if possible –

i. 1101001 [2 marks]

ii. 0101010 [2 marks]

Bit	7	6	5	4	3	2	1	Bits under parity mask	
Data	**1**	**1**	**0**	**1**	**0**	**0**	**1**		
Parity-1	1	0	1	0	1	0	1	2	OK
Parity-2	1	1	0	0	1	1	0	2	OK
Parity-4	1	1	1	1	0	0	0	3	ERR

Error is in bit 4, data is in bits {7, 6, 5, 3} = 1100

Bit	7	6	5	4	3	2	1	Bits under parity mask	
Data	**0**	**1**	**0**	**1**	**0**	**1**	**0**		
Parity-1	1	0	1	0	1	0	1	0	OK
Parity-2	1	1	0	0	1	1	0	2	OK
Parity-4	1	1	1	1	0	0	0	2	OK

There is no error, data is in bits {7, 6, 5, 3} = 0100

What might cause the recovered information to be wrong? [1 mark]

The code assumes either 0 or 1 error; more errors give a faulty correction.

- *There were many problems with bit numbering—read the question!!*
- *The question asks for the information bits, so you must give them.*
- *The Hamming code has nothing to do with encryption.*

More comments on Q 3, HDLC states.

There is considerable doubt about the interpretation of the state diagram.

The general process is –

1) *some situation arises, to which the station usually (but **not** always) reacts by*
2) *sending some notification message to its "partner" station and*
3) *moving itself to some other state.*

*Assume that the "home" station is transferring data and then becomes busy; this corresponds to **B** in the diagram, and to **C** for its partner.*

1. *The **home** station recognises the new "station busy" condition, immediately sends a RNR frame and moves itself to the "Station busy" status.*
2. *The **partner** station receives the RNR frame, acknowledges it with a RR (signaling both "RNR received" and "I am still ready") and moves itself to "Remote Station busy" status (the partner sees the home station as its remote.).*
3. *Eventually the busy condition clears; the **home** station sends a RR and moves to "Information Transfer" state.*
4. *The **partner** receives the RR and immediately moves from "Remote Station Busy" state to "Information Transfer".*

Both stations are now in "Information Transfer" state and normal operation resumes.

More comments on Q 4, Hamming Codes.

*Too many students know **nothing** much about Hamming codes, or so it seems.*

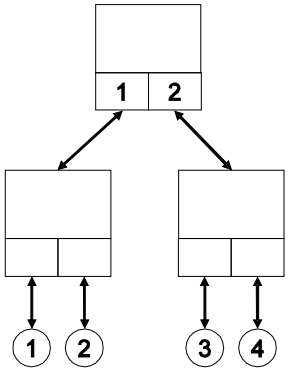
- *The Hamming code, in this example, uses a 7-bit code with parity bits in positions 1, 2, 4 (the powers of 2). NOTE that bits are 7 on the left, 1 on the right (as stated in the question!).*
- *The remaining bits {3, 5, 6, 7} are available for user data*
- *Bit 1 checks bits numbered 1, 3, 5, 7 (those with the $2^0 = 1$ bit in their binary number)*
- *Bit 2 checks bits numbered 2, 3, 6, 7 (those with the $2^1 = 2$ bit in their binary number)*
- *Bit 4 checks bits numbered 4, 5, 6, 7 (those with the $2^2 = 4$ bit in their binary number)*
- *Add together the parity bit numbers of the "failing" parity groups to get the number of the bit to correct. So if the groups for bits 1 and 2 fail but bit 4 succeeds, the error is in bit 1+2=3. If groups 1 and 4 fail, then correct bit 5.*

- *In the first example the parity groups are – (Bit 4 = ERR, bit 2 = OK, Bit 1 = OK); the error pattern or **syndrome** is then 100 showing the error is in bit 4. (This is a parity bit, but that doesn't matter). The question asks for the **information** bits (not the codeword); extracting bits {7, 6, 5, 3} gives 1100 as the information (note the bit ordering).*
- *In the second case all parity groups "succeed" (no errors) and no correction is needed. Extracting bits {7, 6, 5, 3} gives 0100 as the information*

COMPSCI 314 S1C 06 Test Questions

- No marks will be awarded if you merely state a correct answer. To obtain full credit, your script must clearly explain *why* your answer is correct.
- If you are not given enough information to answer a question, you should make a reasonable assumption as required to answer the question, and you should explain your assumption on your script.

Q1. The following questions refer to the network shown below. This network has one bridging hub, two repeating hubs, and four stations. You should assume the bridging hub uses the "transparent hub" routing procedure which was described in the lecture slides and explored in Assignment 3.



a. Assume the forwarding database of the Bridging Hub is empty, immediately *before* station 1 sends a frame to station 2. What is the state of the forwarding database, immediately *after* this frame has been processed by the bridging hub? **[5 marks]**

The BH's forwarding database contains the record (1,1), indicating that station 1 is located on port 1. The BH has learned this information by extracting the source address field (containing MAC address = 1) from the frame it has just received on port 1. The BH does not know the location of station 2, so it floods this frame onto its other port.

b. Explain how station 3 might eavesdrop on frames sent from station 1 to station 2. You should assume that station 2 is completely passive, that is, it never transmits frames. **[5 marks]**

Because station 2 is completely passive, the BH never learns its location. All frames sent from station 1 to station 2 are repeated through the left-hand RH, then flooded through the BH onto the right-hand RH, which then repeats these frames to the links connecting to stations 3 and 4. If station 3 is set to receive promiscuously then it will be able to read these frames.

c. If you were administering this network, how could you make it impossible for station 3 to eavesdrop on frames sent from station 1 to station 2? To receive full credit, your changes should not require the purchase of unnecessary hardware, and each of the four stations must still be able to communicate with each of the other stations. You should assume that nobody (other than you) is able to change

the way in which stations are connected to hubs, and how hubs are connected to each other. **[10 marks]**

I would use my administrative privileges to write a static entry (2,1) into the BH's forwarding database, so that the BH will "know" that station 2 is located on its port 1. With this entry in the database, the BH will discard any frame addressed to station 2 if this frame is received on port 1; and it will forward any frame addressed to station 2 if this frame is received on port 2. Thus all stations will still be able to send frames to station 2, however station 3 will not be able to eavesdrop on frames sent from station 1 to station 2. There is no necessity to install a VLAN switch or VLAN NICs; and there is no necessity to use encryption.

Q2. Recall the portions of the IEEE 802.3-2002 standard which you read while answering questions in Assignment 3. Briefly describe one way in which Gigabit Ethernet could be used to connect two stations which are 1 kilometer apart. **[5 marks]**

A single collision domain in Gigabit Ethernet can be at most a few hundred meters in diameter, even if low-latency fibre optic cabling is used. (Coax and twisted-pair cabling has somewhat higher latency, so the maximum size of collision domain is smaller on such links.) One way to connect two stations would be to set up four Gigabit Ethernet fibre-optic links, each 250 metres in length; with bridges or switches between each link to resolve the collisions on each link separately.

Alternatively, if there are only two stations on this network, then it'd probably be cheapest to set up a full-duplex connection (that is, two half-duplex links with no collision resolution) between these two stations. According to the IEEE 802.3-2002 standard, full-duplex fibre optic links can be more than 1 km in length.

Q3. Consider the SSL protocol, as described in your textbook, lecture and Assignment 3. Briefly describe two ways that an SSL client can check a server's certificate for validity. **[10 marks]**

The validity of the cryptographic signature (typically an MD5 or SHA1 hash digest) on any digital certificate provided by a server can be verified by an SSL client – if the client has the public key of the issuing authority in its certificate store. If the client can not verify the issuer's signature on the certificate, this means that the issuer is "not trusted". In this case the SSL client should issue a prominent warning message to the end user.

A SSH client might also check that the certificate does not appear on the "certificate revocation list" (CRL) of the issuing authority. Reputable issuing authorities publish such revocation lists regularly, as this is the only way to address the security risks arising when the confidentiality of a server's private key has been compromised. Note that anyone who knows a server's private key can falsely authenticate themselves as the server.

The SSL client software should also check whether the server URL named on the certificate is the same as the URL used to connect to the server. It is also important

for the client to examine the "valid from" and the "valid to" dates on the certificate, to confirm that today's date is within this range.

The SSL client software might also check whether or not the user has ever accepted this certificate in the past. It is reasonable to presume that, if the certificate was accepted previously by the user, and if the user hasn't subsequently removed it from their computer's trusted-certificate store, then the user would be willing to accept it again.

The user (as opposed to the SSL client software) might also check whether the certificate is issued to an appropriate party. For example, if I'm trying to access my account at the "Bank of New Zealand", then I will not accept a certificate (even if it is issued by Verisign) from a server named "Bank of Nigeria". In practice such name checks are not very important, for users who are smart enough not to follow links in email, and who also pay attention to the URL field in their browser. If I'm trying to connect to the Bank of New Zealand, and my brower redirects to https://www.bankofnigeria.com, I hope I would notice this, because such clever fraudsters might have obtained a valid SSL certificate for this URL – so my SSL client won't be able to warn me of the fraud. So the most important thing for a user to do, when validating an SSL certificate, is to verify that the URL displayed in the browser window is a valid address for the server they're trying to use.
