CPSC 441
COMPUTER COMMUNICATIONS

MIDTERM EXAM SOLUTION

Department of Computer Science
University of Calgary
Professor: Carey Williamson

March 7, 2014

This is a CLOSED BOOK exam. Textbooks, notes, laptops, personal digital assistants, tablets, and cellular phones are NOT allowed. However, **calculators are permitted**.

It is a 50 minute exam, with a total of 50 marks. There are 12 questions, and 7 pages (including this cover page). Please read each question carefully, and write your answers legibly in the space provided. You may do the questions in any order you wish, but please USE YOUR TIME WISELY.

When you are finished, please hand in your exam paper and sign out. Good luck!

Student Name: ____________________

Score: ________ / 50 = _______ %

----------------------------- Privacy Fold Here -----------------------------

Student ID: ____________________

Multiple Choice

Choose the best answer for each of the following 8 questions, for a total of 8 marks.

1 1. Historically, the design philosophy for the network core of the Internet has always been:

(a) simple best-effort datagram service model

(b) connection-oriented virtual circuits to end systems

(c) complex signalling protocol with resource reservations

(d) charge users based on the Quality of Service (QoS) provided

(e) none of the above

1 2. One of the early ancestors of TCP/IP in the 1970's was:

(a) ARPAnet

(b) NSFnet

(c) CERFnet

(d) NCP (Network Control Protocol)

(e) Wimbledon

1 3. Which of these network applications uses TCP as its default transport-layer protocol?

(a) remote login

(b) file transfer

(c) electronic mail

(d) World Wide Web

(e) all of the above

1 4. Which of these network applications uses UDP as its default transport-layer protocol?

(a) BitTorrent

(b) NNTP (Network News Transfer Protocol)

(c) DNS (Domain Name Service)

(d) World Wide Web

(e) none of the above

1 5. What significant event had a major effect on the Internet in the mid-1980's?

(a) the creation of the World Wide Web

(b) the rise of peer-to-peer file sharing

(c) TCP congestion collapse

(d) mid-life crisis

(e) none of the above

1 6. The congestion avoidance and control mechanisms in TCP were initially designed by:

(a) Vinton Cerf

(b) Robert Kahn

(c) Van Jacobson

(d) Tim Berners-Lee

(e) Carey Williamson

1 7. The underlying control principle in TCP's congestion control can be described as:

(a) speed up slowly, and back off quickly

(b) speed up quickly, and back off slowly

(c) speed up quickly, and back off quickly

(d) speed up slowly, and back off slowly

(e) grab everything you can get, and run away

1 8. In general, the throughput achieved by a file transfer using TCP depends on:

(a) the size of the file being transfered

(b) the round trip time of the network path

(c) the packet loss rate of the network

(d) the version of TCP being used

(e) all of the above

Networking Concepts and Definitions

12 9. For each of the following pairs of technical terms, **define** each term, and **clarify** the key difference(s) between the two terms. Be clear and concise.

(a) (3 marks) "positive ACK" and "negative ACK"
Positive ACK: an acknowledgement strategy in RDT protocols that indicates which data segments have arrived SUCCESSFULLY.
Negative ACK: an acknowledgement strategy in RDT protocols that indicates which data segments were UNSUCCESSFUL (e.g., corrupted in transit, detected via checksum).
Positive ACKs are widely used in TL protocols, including TCP. Negative ACKs are rarely used, since this capability can be provided using timers to recover from both corruption and loss of segments.

(b) (3 marks) "cumulative ACK" and "selective ACK"
Cumulative ACK: an acknowledgement strategy in RDT protocols that indicates the sequence number of the highest in-order data received.
Selective ACK: an acknowledgement strategy in RDT protocols that indicates the sequence number of every data segment received (whether in order or not).
The cumulative strategy requires fewer ACKs, but can only ACK in-order data. Selective ACKs provide more precise information, but add complexity to the RDT protocol. Most TCP implementations use cumulative ACKs, while TCP SACK uses selective ACKs.

(c) (3 marks) "SOCK_DGM" and "SOCK_STREAM"
SOCK_DGM: the parameter used when creating a socket to invoke UDP.
SOCK_STREAM: the parameter used when creating a socket to invoke TCP.
These parameters specify either unreliable (connection-less) or reliable (connection-oriented) transport-layer service in TCP/IP socket programming.

(c) (3 marks) "Round Trip Time (RTT)" and "Retransmit Timeout (RTO)"
RTT: elapsed time between sending a data segment and getting an ACK for it.
RTO: timer value used to detect loss (or delay) of a transmitted segment.
RTT is dynamically estimated in TCP. The RTO value is usually set to a multiple of the expected RTT value (e.g., 2-4x larger).

Reliable Data Transfer Protocols

10 10. In class, we discussed a simple One-Bit Sliding Window Protocol (OBSWP).

(a) (4 marks) What were the key elements of **state information** maintained by OBSWP? Where did the state information reside, and why?

Finite State Machine (FSM) to keep track of the sender's state.
Another FSM to keep track of the receiver's state.
Both endpoints (S and R) are identical to support full-duplex data exchange.
Sequence number (0 or 1) for S and R to use on outbound segments.
Expected sequence number (0 or 1) for S and R to see on incoming segments.
Timers at S and R to detect loss or delay of segments.

(b) (2 marks) Would OBSWP be suitable for general-purpose data transfer on the wide-area Internet? Why or why not?

No. OBSWP can have AT MOST ONE data segment in flight from sender to receiver at a time. This will be very inefficient if the RTT is large.

(c) (4 marks) What additional features are needed to generalize OBSWP effectively to a WAN environment? What additional state information is required to support these features, and where would this state information reside?

Pipelining: a sliding window protocol that allows multiple segments (data and ACKs) to be in transit at a time between the sender and the receiver.
State requirements: larger sequence number space; sender window to determine segments eligible for transmission; receiver window to determine eligible incoming segments; buffer space at sender to facilitate retransmissions if needed; buffer space at receiver to handle out-of-order data; timers on each outstanding segment (or at least oldest outstanding one); error recovery mechanism (e.g., Go-Back-N or Selective Repeat).

Transmission Control Protocol (TCP)

10 11. The diagram below illustrates the generic segment structure used by the Transmission Control Protocol (TCP). Use your knowledge of TCP to answer the following questions.

<table>
<tr><td colspan="2">Source Port</td><td>Destination Port</td></tr>
<tr><td colspan="3">Sequence Number</td></tr>
<tr><td colspan="3">Acknowledgement Number</td></tr>
<tr><td>Length</td><td>Flags</td><td>Window</td></tr>
<tr><td colspan="2">Checksum</td><td>Urgent Pointer</td></tr>
<tr><td colspan="3">Options</td></tr>
<tr><td colspan="3">DATA</td></tr>
</table>

(a) (1 mark) Does TCP support simplex, half-duplex, or full-duplex data transfers?
`Full-duplex`

(b) (2 marks) Describe precisely what information is carried in the *Sequence Number* field.
`32-bit field specifying the sequence number of the first byte of outgoing data carried in the segment.  Note that TCP sequence numbers are expressed in bytes, not in segments.`

(c) (2 marks) Describe precisely the information carried by the *Acknowledgement Number*.
`32-bit field specifying the next byte of data expected by the receiver. TCP uses ‘‘lookahead’’ ACKs, and expresses these in bytes, not segments.`

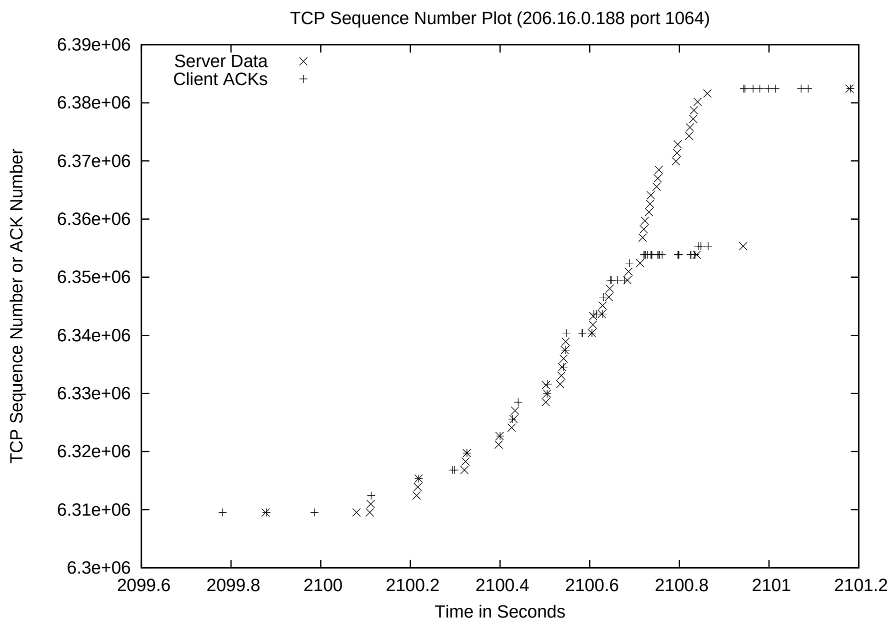
(d) (3 marks) Give 3 examples of *Flags* field bits used to convey TCP state information.
`S: Synchronize flag in opening 3-way handshake to start a new TCP connection.`
`F: Finish flag in closing handshake to terminate a TCP connection.`
`A: Acknowledgement flag is on if the Acknowledgement Number field is valid.`
`P: Push flag if the data should be delivered to the application right away.`
`R: Reset flag indicates that an unrecoverable TCP error has occurred.`
`U: Urgent flag indicates that this segment carries some urgent data, and Urgent Pointer is valid.`

(e) (1 mark) Does the *Window* field represent the Sender Window (W_S), the Receiver Window (W_R), or the Congestion Window (`cwnd`)?
`Receiver Window (indicates available buffer space, expressed in bytes)`

(f) (1 mark) Give one example of a *TCP Option* that could be negotiated during an opening 3-way handshake.
`Timestamp Echo for fine-grain RTT estimation`
`Window Scaling for high delay-bandwidth product`
`SACK for Selective ACKnowledgement TCP`

TCP Performance

10 12. The graph below shows one of the TCP sequence number plots that we discussed in class. Using lines, arrows, circles, and handwritten text as appropriate, **annotate the graph** to show **at least 5 features** of the TCP transport-layer protocol, and/or any unusual performance anomalies that occur during this particular TCP connection. You may write directly upon the graph, or in the white space around it. Up to 2 marks are available for each **distinct and relevant feature** that is correctly identified and explained.




```
Opening 3-way handshake starting near time 2099.8
Non-negligible RTT between client and server
Evidence of TCP slow start for initial cwnd evolution
Delayed ACKs that limit progression of cwnd to Fibonacci-style pattern
A few Duplicate ACKs (and possible Fast Retransmit) around time 2100.7
About twenty Duplicate ACKs (and possible Fast Retransmit) around time 2100.8
Distinct pattern of TCP NewReno Fast Recovery (3 pkts at a time) around time 2100.7
A lost packet is retransmitted around time 2100.8
Another lost packet (consecutive lost packets!)  retransmitted around time 2100.8
Large jump (about 25 KB) in cumulative ACKs once loss recovery is completed
Multiple Duplicate ACKs around time 2101.0 as socket buffer empties
Closing 3-way handshake just before time 2101.2
Total (vertical) data transfer is approximately 75 KB
Total (horizontal) connection duration is approximately 1.4 seconds
Average connection throughput is approximately 440 kbps
```

*** THE END ***