**CSC 138 Exam 2 Study Guide**

***1. Describe the purpose of the following mechanisms in a reliable data transfer protocol: Checksum, Timer, Sequence Number, Acknowledgment, Negative Acknowledgment, and Window.***

- Checksum: detects bit errors in data, detects if bits have been altered as it moved from source to destination

- Timer: allows sender to wait for a reasonable amount of time for ACK from receiver, safeguards against lost packets

- Sequence Number: used to check if received packet is duplicate, allows sender to know which packet being ACKed by receiver

- Acknowledgment (ACK): tells sender that packet has been received without issue, so that sender can send next packet

- Negative Acknowledgment (NAK): tells sender that there is an issue with received packet, so that sender can re-transmit the packet

- Window: used in pipelined protocols, sender window limits number of sent, but unACK’ed packets allowed, receiver window limits number of packets that can be received

- Sender Window: used in both go-back-n and selective repeat

- Receiver Window: used in selective repeat

***2. Describe the purpose and operation of each field of the UDP segment header.***

- Source Port Number: identifies socket on sender’s host that segment originated from

- If receiving host sends a reply to sender, the destination port number of the reply is extracted from the source port number

- Destination Port Number: identifies socket on receiver’s host that segment should be sent to

- When segment arrives at host, transport layer examines destination port number and directs segment to corresponding socket

- Length: specifies the length (in bytes) of the segment, including header

- Transport layer examines length field to determine the total length of the segment, since size of data field may differ from one segment to the next

- Checksum: detects errors in transmitted segment

- Sender calculates checksum by adding all 16 bit words in segment (with overflow wrapped around) and performing 1s complement

-Receiver adds all 16 bit words (including checksum), result should be all 1s if no bit errors

***3. Describe the purpose and operation of each field of the TCP segment header, ignoring the flags or urgent data field.***

- Source Port Number: used for identifying a TCP socket

- TCP socket identified by four fields: IP address and port number of both source and destination

- When receiving host sends a reply to sender, the destination port number of the reply is extracted from the source port number

- Destination Port Number: used for identifying a TCP socket

- When segment arrives at host, IP address and port number of both source and destination are used to direct segment to correct socket

- Sequence Number: counts by bytes, tells receiver byte offset of application data

- Initial sequence number randomly chosen

- If sequence number of received segment is not byte in data stream expected by receiver, then segment has been received out of order

- Acknowledgment Number: ACKs last byte received

- Receiver puts the next byte expected from sender into the acknowledgment number field of the segment that it sends to sender

- Header Length: 4 bit value, specifies length of header in 32 bit words

- Header can be of variable length due to options field

- Receive Window: used for flow control, tells sender how much free buffer space receiver has

- Maintained by sender

- Buffer created by receiver

- Receive window continuously updated with amount of free buffer space in every segment receiver sends to sender

- Sender keeps amount of unACK’ed data less than the value of receive window

- Sender determines amount of unACK’ed data by subtracting last byte ACKed from last byte sent

- Checksum: detects errors in transmitted segment

- Sender calculates checksum by adding all 16 bit words in segment (with overflow wrapped around) and performing 1s complement

-Receiver adds all 16 bit words (including checksum). Result should be all 1s if no bit errors

- Options: specifies TCP options

- Optional and variable length field

- Maximum segment size (MSS), window scaling factor (for high-speed networks), time- stamping

***4. Describe the routing and forwarding processes as performed by a router (switch) in:***

***a) a traditional router network, and***

***b) an SDN-enabled network.***

- Traditional Router Network

- Routing algorithm (in control-plane) runs in every router

- Routing algorithm determines contents of router’s forwarding table (in data-plane) by communicating (exchanging messages) with routing algorithms in other routers

- Router uses forwarding table to transfer packets from an input link interface to correct output link interface

- SDN-Enabled Network (Software-Defined Networking)

- Physically separate remote controller (in control-plane) computes and distributes forwarding tables (in data-plane) for every router

- Separates control-plane routing from physical router

- Control-plane has top-down view of network

- Routing device performs forwarding only

***5. Describe the following packet scheduling/queue management methods: FIFO, Priority, Round Robin, and WFQ.***

- FIFO (First In First Out): packets sent in order of arrival to queue

- Priority: send highest priority queued packet

- Packets sorted into multiple classes, with different priorities

- Each class typically has its own queue

- Round Robin: cyclically scan class queues, sending a packet from each class (if available)

- WFQ (Weighted Fair Queuing): generalized Round Robin, each class gets weighted amount of service in each cycle

- Each class is assigned a weight

- Most prevalent queuing policy

***6. CIDR and Ipv4 subnetting.***

- Do related exercises

***7. Describe the purpose of each field in the IPv6 datagram header.***

- Version: specifies IP version number

- Priority: specifies priority among datagrams in flow

- Flow Label: identifies datagrams in same flow

- Payload Length: specifies number of bytes in datagram following the header

- Next Header: specifies the protocol to which datagram header will be delivered (e.g. TCP, UDP)

- Hop Limit: determines how many remaining times datagram can be forwarded

- Decremented by 1 by each router that forwards datagram

- Datagram discarded if reaches 0

- Ensures datagrams do not circulate forever in network

- Source Address: specifies IP address of datagram source

- Destination Address: specifies IP address of datagram destination

***8. Describe the purpose of each field in the IPv4 datagram header, ignoring the flags, identifier, and fragmentation offset.***

- Version: specifies IP version number

- Header Length: specifies where in the datagram the payload begins

- IPv4 datagram header length can vary due to variable length options field

- Type of Service: allows different types of datagrams to be distinguished from each other (e.g. real-time from non-real-time)

- Length: specifies total length of the datagram (including header) in bytes

- Time to Live: specifies maximum number of remaining hops

- Decremented by 1 each time datagram processed by router

- Router drops datagram if reaches 0

- Ensures datagrams do not circulate forever in network

- Upper Layer: specifies the protocol to which datagram header will be delivered (e.g. TCP, UDP)

- Header Checksum: aids router in detecting bit errors in received datagram

- Calculated from header

- 32 Bit Source IP Address: specifies IP address of datagram source

- 32 Bit Destination IP Address: specifies IP address of datagram destination

- Options: specifies any options to be used

- Optional field, variable length