SOLUTIONS TO CS 536 MIDTERM, FALL 2019 (PARK)

P1(a) 14 pts

Each TDM frame consists of 100 * 8 = 800 bits.
2 pts

8000 frames need to fit in a 1-second time interval. This is due to supporting voice communication in T1 which assumes 4000 Hz voice bandwidth. By Nyquist's sampling criterion, 4000 * 2 = 8000 samples are needed per second.
3 pts

Put together this results in 6400000 bps, i.e., 6.4 Mbps.
2 pts

Pulse width is 1 / 6400000 sec (about 160 nanoseconds).
2 pts

This is not a viable idea. Since pulse width is 1 / 6400000 and we are using pulse amplitude modulation (square waves), we know that the spectrum of each square wave is a sinc function with width at least 6.4 MHz.
2 pts

Since link bandwidth is 3 MHz, a significant portion of the frequencies (i.e., sinusoids) needed to preserve the shape of the square will be lost. This is likely to result in a signal received that is significantly distorted (not a square wave anymore) which leads to errors decoding 0 or 1.
3 pts

P1(b) 14 pts

3 bits yields 8 values for sequence numbers, say, 0, 1, ..., 7.
2 pts

By the rule SWS < (MaxSeqNum + 1) / 2 = (8 + 1) / 2, we have SWS <= 4.
4 pts

Suppose the sender violates the rule by transmitting 5 packets (or frames) with sequence numbers 0, 1, 2, 3, 4. Suppose all frames are correctly received at the receiver who transmits ACK packets with sequence numbers 1, 2, 3, 4, 5. Suppose all five ACKs are received in time (i.e., before their respective timeouts expire) and the sender transmits the next 5 data packets with sequence numbers 5, 6, 7, 0, 1. Suppose 5, 6, 7 are dropped and only 0 and 1 make it to the receiver.
4 pts

Upon receiving 0 and 1, the receiver cannot determine if (a) 0 and 1 are new data packets (as is the case in this example), or (b) its ACK packets 1, 2, 3, 4, 5 did not reach the sender in time which caused the timers for packets 0 and 1 to expire and be retransmitted. This ambiguity cannot arise if SWS <= 4.
4 pts

[Other example scenarios exist and get full credit if they are correctly explained. Otherwise partial or zero credit.]

P1(c) 14 pts

Assuming two hosts A and B at opposite ends of a 2500 m Ethernet network, the worst-case collision scenario arises when A senses the link to be idle (CS), transmits a frame, and just before the first bit of A's frame arrives at B, B senses the link the to be idle and starts transmitting its frame which causes a collision.
2 pts

B's bits travel to A which altogether (from the moment A starts transmitting to A hearing the bits from B which implies collision) is assumed to take 51.2 microseconds (about the round-trip latency of the 2500 m network). For CD to guarantee that A detects the worst-case collision if it occurs, A must not be done transmitting all the bits of its frame. Only then is A assured of hearing B's bits which implies collision.
2 pts

Since link bandwidth is 10 Mbps, 51.2 microseconds * 10 Mbps = 512 bits (i.e., 64 bytes) must be the minimum size of A's frame so that its transmission time at 10 Mbps is at least 51.2 microseconds. Ignoring Ethernet's preamble/postamble bits (we are being conservative), 6 bytes of source address, 6 bytes of destination address, 2 bytes for type field, and 4 bytes for error detection (CRC field) yields 6 + 6 + 2 + 4 = 18 bytes. Hence 64 - 18 = 46 bytes must be the minimum payload size of an Ethernet frame.
2 pts

Maximum diameter has to decrease by a factor of at least 10 (250 m) to guarantee that A is still transmitting bits at 100 Mbps speed when the worst-case collision signal from the opposite end arrives.
2 pts

Buildings come in all shapes and sizes, and the maximum diameter of its electrical wiring is difficult to predict. Also, electrically wires are typically not terminated by absorbers which prevent signals to be reflected back which can incorrectly be interpreted as collision. Hence CD is difficult to implement.
2 pts

CSMA without CD requires a receiver to send a separate ACK frame to acknowledge receipt of a data frame. That is, perform stop-and-wait operation on the data frame.
2 pts

With thousands of hosts competing to transmit frames over shared electric wiring of a building, CSMA cannot prevent multiple hosts from temporarily sensing the link as idle, transmitting, and causing collision. In high-load environments, CSMA is not an effective protocol for sharing bandwidth.
2 pts

P2(a) 19 pts

To allocate 1000 orthogonal carrier frequencies to 1000 users, divide bandwidth 6 - 5 = 1 GHz by 1000. This yields 1 MHz. Set the carrier frequencies as 5.001 GHz, 5.002 GHz, ..., 5.999 GHz, 6 GHz. This is equivalent to (a) setting the carrier frequencies as 0.001 GHz, 0.002 GHz, ..., 1 GHz, which being integer multiples (i.e., harmonics) of the base frequency 0.001 GHz are guaranteed to be mutually orthogonal over the finite time window (i.e., symbol period) 1 / 0.001 GHz = 1 microsecond; then (b) multiplying by 5 GHz which shifts the 1000 carrier waves to the 5-6 GHz range.
5 pts

Each of the 1000 carrier waves follows the symbol period of the slowest carrier wave, hence a clock that runs at 1 MHz. Since 8-level AM is used, per symbol period 3 bits can be transmitted per carrier frequency. Thus 1 MHz * 3 bits = 3 Mbps is the bps bandwidth of each carrier wave.
4 pts

Since there are 1000 carrier frequencies, the bps bandwidth of the system as a whole is 3 Mbps * 1000 = 3 Gbps.
2 pts

If the number of users decreases to 100, the base carrier frequency is 1 GHz / 100 = 10 MHz. Hence each user get bps bandwidth 10 MHz * 3 bits = 30 Mbps. System bps bandwidth is 30 Mbps * 100 = 3 Gbps. Thus system bandwidth remains unchanged but each carrier wave is able to transmit 10 times faster.
4 pts

To operate the above OFDMA system in the 3-4 GHz range, all that has to change is step (b): multiplify the carrier waves by 3 GHz sinusoid instead of 5 GHz sinusoid at the sender.
2 pts

Multiplying the carrier waves by 5 GHz or 3 GHz sinuoid is a shift operation and has no effect on bps bandwidth.
2 pts

P2(b) 19 pts

Since there are no losses and assuming timeout greater than d, there will be no retransmissions and a total of S / P data packets will need to be sent.
2 pts

For each data packet, its completion time (i.e., time to reach the receiver) is given by (F / B) + d which is transmission time plus link latency.
2 pts

After the last bit of the data packet has reached the receiver, the receiver transmits an ACK packet to the sender. The completion time of the ACK packet (i.e., time to reach the sender) is given by (D / B) + d.
2 pts

Hence the completion time of a single data packet using stop-and-wait is (F / B) + d + (D / B) + d = 2 * d + (F + D) / B.
2 pts

Thus total completion time to transmit S / P data packets back-to-back using stop-and-wait (S / P) * (2 * d + (F + D) / B).
2 pts

Throughput is S / total completion time which is P / (2 * d + (F + D) / B).
2 pts

If every other packet is dropped and timeout is 2 * d, then for half of the data packets, i.e., (S / P) / 2 the above completion time (per data packet) will apply.
2 pts

For the other half, additional delay will be incurred due to waiting for timeout. Since 2 * d < (2 * d + (F + D)) / B = RTT, a timeout will occur for every data packet irrespective

of whether it is dropped or not. Hence in the loss case, the completion time for
a dropped packet is 2 * d + (2 * d + (F + D) / B).
Thus total completion time is ((S / P) / 2) * (K + L) where K = (2 * d + (F + D) / B) and
L = (2 * d) + (2 * d + (F + D) / B).

Caveat: The above calculation holds if latency dominates transmission time. Otherwise, since for every data packet there will be a timeout that causes retransmission, in some situations this may delay transmission of the next data packet. Therefore the loss component of the problem has side effects that can affect total completion time calculation across multiple packets. As inter-packet dependencies were not part of our discussion of stop-and-wait performance, full credit (4 pts) are given to the loss component of the problem.
2 pts

The delay-bandwidth product is d * B which is in unit of bits. If F is significantly smaller than d * B, i.e., F << d * B, then the link is being underutilized by stop-and-wait and using sliding window is likely to result in significant throughput gain.
3 pts

P3 20 pts

A host (i.e., its NIC) that detects collision will wait a uniformly random time chosen from the interval [0, 51.2] (unit in microseconds).
2 pts

An application of the Bakery Problem, each host choosing uniformly randomly over [0, 51.2] results in wait times that lead to least number of (expected) collisions.
3 pts

If retransmission results in collision, the interval is doubled to [0, 102.4] and a uniformly random wait time chosen.
2 pts

Upon k consecutive collisions, the interval becomes [0, 51.2 * 2^(k - 1)]. That is, at each consecutive collision, the previous interval is doubled leading to exponential backoff of wait time.
2 pts

Ethernet's exponetial backoff aims to aggressively increase the interval over which wait time is uniformly randomly chosen to significantly reduce the likelihood of future collision. Exponentially increasing the interval accomplishes that.
3 pts

A less aggressive method for increasing the wait time interval upon consecutive collision is linear increase, i.e., [0, 51.2 * k]. It is a reasonable method when load is not high.
2 pts

Ethernet is not reliable since if 16 consecutive collisions occur, the sender gives up.
2 pts

Under the assumption that typical wired LANs and wireless LANs are not heavily loaded, it makes sense use CSMA/CD or CSMA since the likelihood of collision is not high and there is no coordination overhead as is the case for OFDMA, TDMA, and CDMA.
2 pts

OFDMA, TDMA, or CDMA are preferable if load is high.
2 pts

Bonus 10 pts

If the ACK packet is dropped, the server is left waiting indefinitely which it cannot afford to do.
3 pts

In our file transfer protocol, the server retransmits the last data packet up to 3 times. If it still does not receive an ACK, it will assume that the data packet has been received and terminate.
3 pts

This is not a correct solution since it is possible that all 4 transmissions (original transmission plus 3 retransmission) of the last data packet were lost and that is why an ACK has not been received (i.e., the client has not sent an ACK). If the server gives up after 3 tries, the client is left waiting which our protocol has not addressed.
4 pts