15-744 Computer Networks (Fall 2011)
Homework 2

Due: Oct. 12th, 2011, 3:00PM (in class)

Name:
Andrew ID:

October 12, 2011

A Short Questions

1. Which of the following is true about modern high-speed routers?
   - A. A typical crossbar-based router uses input queueing so that it can implement sophisticated Quality-of-Service (QoS) processing.
   - B. The iSlip crossbar scheduling algorithm (a round-robin Parallel Input Matching algorithm) is efficient, but in practice does not fairly allocate time slots.
   - C. High-performance routers can not change the IP packets they forward at all
   - D. High-performance routers use a bus-based achitecture to achieve very high packet forwarding rates.
   - E. In a high-performance router, copies of the forwarding table are kept in the output ports.
   - F. In a high-performance router, copies of the forwarding table are kept in the input ports.

Solution: Only (f) is correct.

2. Chiu and Jain showed in their paper that AIMD was the only choice of increase/decrease algorithms that converged to a fair and efficient utilization. *Briefly* explain why XCP is able to get away with using a multiplicative increse algorithm?

Solution: Because XCP uses an AIMD fairness controller to allocate the increase *between* flows.

B Long Questions

3. The picture above shows the famous TCP saw tooth behavior. We are assuming that fast retransmit and fast recovery always work, i.e. there are no timeouts and there is exactly one packet lost at the end of each "tooth". We are assuming that the flow control window is large and that the sender always has data to send, i.e. throughput will be determined by TCP congestion control.

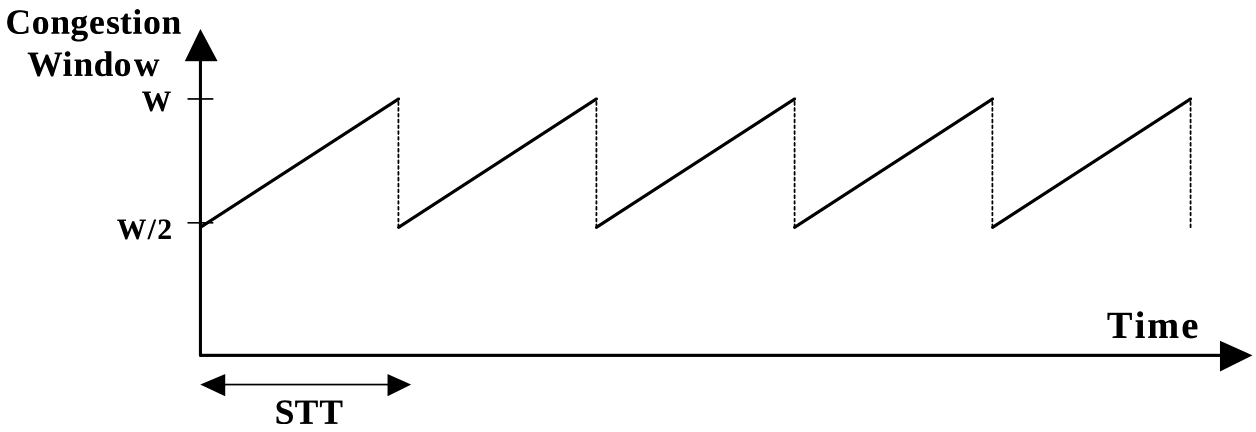


Figure 1: TCP sawtooth diagram

In the picture, W represents the congestion window size at which a congestion packet loss occurs (expressed in maximum transfer units). You can assume that W is large, so feel free to approximate (W-1) or (W+1) by W. STT represents the "saw tooth time" expressed in seconds.

The aim of this exercise is to derive the average throughput of a TCP connection as a function of the roundtrip time (RTT), the maximum transfer unit (MTU), and the packet loss rate (PLR) for the connection. Please use the notation suggested by the figure, i.e. W and STT, as intermediate values if you need them.

(a) Calculate the STT as a function of W, and the RTT. (Hint: the congestion window goes from W/2 to W in one STT, and remember the congestion window is increased by 1 MTU every RTT).

> **Solution:** To calculate STT, we need to know the "slope" of the congestion window increase. For TCP, the congestion window is increased 1 MTU every RTT, so
> STT = RTT * W / 2

(b) How much data is sent in one STT? (Hint: how much data is sent each RTT?)

> **Solution:** In one RTT, one congestion window worth of data is sent. There are W / 2 RTTs in STT, and the average size of the congestion window is 3 * W / 4, so
> amount of data = (W / 2) * (3 * W / 4) * MTU

(c) What is the average throughput of the connection?

> **Solution:** The average throughput is the amount of data sent in STT divided by STT, so let us calculate those two entities.
> How much data is sent in that amount of time? This means that the through put is:
> T = (3 * W / 4) * MTU / RTT

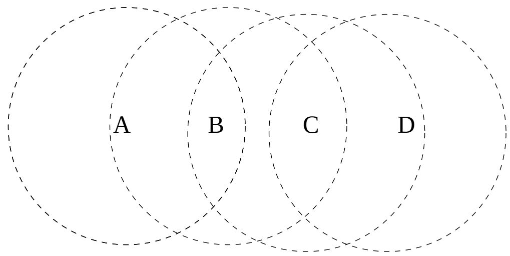
(d) What is the average packet loss rate? (Hint: How many losses occur per STT?)

> **Solution:** Now, how much is W? Well, we lose one packet for every tooth in the picture, so the PLR is 1 divided by the number of packets sent in STT, which we already calculated above.
> PLR = 1 / ((W / 2) * (3 * W / 4))

(e) What is the relationship between the throughput and the packet loss rate?

> **Solution:** which allows us to calculate W as a function of PLR.
> If we substitute W in T by this value, we get:
> T = 1.22 * MTU / (RTT * sqrt(PLR))

4. Peter, excited to setup a new adhoc wireless network on campus, uses the following topology with nodes A, B, C, and D. The dotted lines indicate the range of the wireless transmission from each node. For example, C is within transmission range of nodes B and D. In the following question, you will determine the outcome of transmissions that Peter has scheduled to take place in order to test his new network. Also assume that packets are never rescheduled for transmission, i.e., if a packet did not succeed the first time the node does not bother to retransmit it.



(a) In the first set of scheduled transmissions, carrier sense is enabled, but Peter has disabled RTS/CTS. Each transmission has a source, destination, associated relative start time, and total transmission time. For each transmission, briefly state the result and relate it to other transmission numbers if needed to justify your answer. **In this section, transmissions are never rescheduled.** A result would be "success" or "prevented by ...something..." (8 points)

#	Src	Dst	Start Time	Duration	Type	Result
1	A	B	0	15	DATA	
2	C	B	5	10	DATA	
3	C	D	20	20	DATA	
4	B	C	22	8	DATA	
5	B	A	30	10	DATA	

Solution:

#	Src	Dst	Time	Length	Type	Result
1	A	B	0	15	DATA	Collision with 2 (hidden terminal)
2	C	B	5	10	DATA	Collision with 1 (hidden terminal)
3	C	D	20	20	DATA	Successful
4	B	C	22	8	DATA	Prevented by carrier sense from 3
5	B	A	30	10	DATA	Prevented by carrier sense from 3 (exposed t.)

(b) After the first series of transmissions, Peter enables RTS/CTS on each node and reschedules the same transmissions from *part a* with the same start and transmission times. For each scheduled transmission give the outcome **and** any additional transmissions that may result from them. Make sure to specify the type of packet using DATA, RTS, or CTS. Assign new transmission numbers, but consider the times of RTS/CTS packets to be negligible (do not assign times). You may not need all of the blank lines.

(Fill in the table on the next page) (7 points)

#	Src	Dst	Time	Length	Type	Result
1	A	B			RTS	Success
2	B	A			CTS	Success
3	A	B	0	15	DATA	Success
4	C	B	5	10	DATA	Prevented: heard CTS#2

Solution:

#	Src	Dst	Time	Length	Type	Result
1	A	B			RTS	Success
2	B	A			CTS	Success
3	A	B	0	15	DATA	Success
4	C	B	5	10	DATA	Prevented: heard CTS (#2)
5	C	D			RTS	Success
6	C	D			CTS	Success
7	C	D	20	20	DATA	Success
8	B	C	22	8	DATA	Prevented: heard RTS (#5)
9	B	A	30	10	DATA	Prevented

5. Bin sets up a wireless AP (access point) at home, and finds the performance to be much poorer than he expected. In this question, you are going to help him figure out possible reasons.

   (a) First, Bin finds that since the access point is open, some neighbor has also associated with the AP. While Bin is close to the AP and can receive data at 11Mbps, the neighbor has a much lower signal strength and thus can only receive data at 1Mbps. Assume the AP always has full-size (1500 bytes) packets to send to both Bin and his neighbor, and it uses round-robin to schedule between the two transmissions. If we ignore all overheads (no headers or preambles, no packet loss, no acknowledgements), what is the throughput Bin will get? What if Bin can receive data at 54Mbps?

   Solution:
   At 11Mbps, the throughput is $\frac{1}{1+\frac{1}{11}} = 0.917$Mbps.
   At 54Mbps, the throughput is $\frac{1}{1+\frac{1}{54}} = 0.982$Mbps.

   (b) After finding this out, Bin has used several techniques (enable security, enable MAC address filtering) to prevent his neighbor from using his access point. In the next two questions, we will look at the difference between physical data rate and the TCP throughput.

   Here is how the 802.11 MAC protocol works with only one sender and receiver in the network: the sender would sense the channel before transmitting, and if the channel is clear and has been clear for DIFS (Distributed Coordinate Function Interframe Space) time, the sender would send out the frame. After receiving the frame, the receiver would wait for SIFS (Short Interframe Space) time and send out a link-layer acknowledgement. 802.11 protocol also requires a preamble to let the receiver to synchronize with the sender and to be notified that the data is on its way, and the preamble is transmitted at the lowest data rate.

Assuming Bin is using 802.11b wireless devices, here are the parameters for 802.11b: preamble take $192\mu s$, DIFS = $50\mu s$, SIFS = $10\mu s$, payload size = 1500 bytes, header size = 30 bytes, TCP ACK frame size (including all headers) = 70 bytes, link layer ACK size = 14 bytes. So what is the TCP throughput if Bin can receive at 11Mbps physical data rate (i.e., the payload, headers, and link layer ACKs are transmitted at 11Mbps)? In this problem, you can assume the wireless link is the bottleneck, and ignore TCP slow start, congestion control, round-trip time, delayed ACK.

Solution: The total time for a TCP exchange is: $(50 + 192 + 30 * 8/11 + 1500 * 8/11) + (10 + 192 + 14 * 8/11) + (50 + 192 + 70 * 8/11) + (10 + 192 + 14 * 8/11) = 2072\mu s$. Useful data size is 1500 bytes. Thus the effective data rate is $1500 * 8/2072 = 5.8$Mbps.

(c) Bin is now using 802.11g devices, with the following parameters: preamble take $20\mu s$, DIFS = $34\mu s$, SIFS = $16\mu s$, payload size = 1500 bytes, header size = 30 bytes, TCP ACK frame size (including all headers) = 70 bytes, link layer ACK size = 14 bytes. So what is the TCP throughput if Bin can receive at 54Mbps?

Solution: The total time for a TCP exchange is: $(34 + 20 + 30 * 8/54 + 1500 * 8/54) + (16 + 20 + 14 * 8/54) + (34 + 20 + 70 * 8/54) + (16 + 20 + 14 * 8/54) = 421.2\mu s$. Useful data size is 1500 bytes. Thus the effective data rate is $1500 * 8/421.2 = 28.5$Mbps.