CS 653 Mobile Computing *Spring 2014*

Problem Set 2 Solutions

Problem 1

Refer to the link-layer throughput calculations for 802.11g that were shown in the class, and to the reference provided on the class website. Using the MAC timing constants as specified in the reference, compute the following.

a. What is the total time taken to transmit a link-layer frame of length 1000 bytes (this length includes the link layer header also) at the bit rate of 54 Mbps? Include the time taken for the DIFS, physical layer preamble, SIFS, and link-layer ACK. Assume that the frame contains UDP data.
b. Suppose the total time to transmit a frame as computed above is *T*, and the time that is spent in actually sending the OFDM symbols corresponding to the link layer frame is *t*. Then, we define the efficiency of the link layer as the fraction *t/T*. What is the efficiency of the link layer in transmitting the 1000 byte frame in part (a) above, for the bit rate of 54 Mbps?
c. Suppose we had the ability to aggregate multiple 1000 byte frames to be sent at once before eliciting the link layer ACK. What is the minimum number of 1000 byte frames that must be aggregated to achieve a link layer efficiency of 90% at a bit rate of 54 Mbps?

Solution

a. 1000 bytes = 8000 bits = 8000/216 ~ 38 symbols. Time spent in overheads = 28 us (DIFS) + 20 us (phy header) + 6 us (tail signal) + 10 us (SIFS) + 30 us (link layer ACK) = 94 us. Time to send data = 38*4 = 152 us. Total time = 246 us.
b. Efficiency = 152/(152+94) = 0.617
c. Suppose we need to aggregate N frames. Then N*152/(N*152+94) >= 0.9. Solving, we get N must be 6.

Problem 2

Consider two clients connected to an access point in a WiFi-like system. One of the clients transmits at a bit rate of 6 Mbps and the other at 54 Mbps. Both clients are continuously generating UDP traffic to the AP. Assume no channel losses at the chosen bit rate, and no losses due to collisions. Below we describe four different MAC protocols that can be used by the system. In each case, calculate the average throughput obtained by each client, and the aggregate throughput of the system (over a long period of time, i.e., in steady state) when clients share the medium using the specified MAC protocol. Ignore all MAC-level timing overheads (i.e., a client sending at the 6 Mbps bit rate all the time can be assumed to get a throughput of 6 Mbps). Also ignore any other implementation overheads of the protocols (e.g., the overhead of synchronizing the clients for TDMA).

a. The CSMA MAC protocol, which gives each client equal transmission opportunities.
b. A TDMA-based solution with a round-robin scheduler, where each client gets equal number of time slots to transmit.
c. A TDMA-based solution with max rate scheduling.
d. A TDMA-based solution with proportionally fair scheduling, where the priority of a node is determined as current_rate/average_rate, and the average is computed over a long period of time.

Solution

Suppose a given MAC protocol gives the 6 Mbps node a fraction x of time to transmit, and the 54 Mbps node gets a fraction (1-x). Then the average throughputs of the clients are 6*x Mbps and 54*(1-x) Mbps, and the aggregate throughput is the sum of the two. Each of the MAC protocols differs in the number x.

a. CSMA gives equal transmission opportunities, so the 6 Mbps node gets 9 times as much time as the 54 Mbps node. So x = 0.9. Plugging it in, we get the client throughputs as 5.4 Mbps each, aggregate throughput of 10.8 Mbps.
b. Round robin scheduler gives equal time slots, so x = 0.5. Clients get 3 and 27 Mbps respectively, aggregate is 30 Mbps.
c. With max rate scheduler, all the time is given to 54 Mbps client. That is, x = 0. So the clients get 0 and 54 Mbps respectively, and aggregate is 54 Mbps.
d. The PF scheduler stabilizes at a point where the priorities of the two clients are equal. That is, 54/54(1-x) = 6/6x, so x = 0.5. That is, the PF scheduler behaves exactly like the round robin scheduler in this case. It schedules the 6 Mbps client once for every 9 times it schedules the 54 Mbps client, so that each client gets equal air time in the end. The benefit of PF scheduler in practice comes from the fact that it can schedule clients when their channel is better (compared to its average) in cases when channel conditions vary. We are not able to see the benefit in this simple example because we are assuming constant channel conditions.

Problem 3

Consider the following variation of the SampleRate algorithm discussed in class, called SampleRate1. SampleRate1 is similar to SampleRate in all aspects except one: while SampleRate sends every 10^{th} packet at a higher rate than the current best rate, SampleRate1 simply sends the 10^{th} packet also at the current best rate only. That is, SampleRate1 has no special clause to deal with every 10^{th} packet, and picks the bit rate with the lowest average transmission time on all packets.

Consider a 802.11g wireless link running SampleRate1, that can operate at 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. Currently, the wireless link between a sender and receiver has the following frame loss rates at each of the bit rates: rates 1 to 18 Mbps are lossless, 24 Mbps has 10% loss, 36 Mbps has 50% loss, and the higher rates do not work at all. Assume that the SampleRate1 rate adaptation algorithm has stabilized to the best rate for this setting. Now, we will provide two different scenarios in which the wireless channel changes from this current state. In each scenario, specify whether SampleRate1 leads to better/worse/same throughput as SampleRate after the channel change, and explain your answer.

a. The channel becomes better so that 24 Mbps is lossless, 36 Mbps has a 10% loss, 48 Mbps has a 50% loss, and higher rates do not work.
b. The channel becomes worse so that 24 Mbps now has a 50% loss rate, 36 Mbps and higher rates do not work. Also, 18 Mbps now has a 10% loss rate, and lower rates are lossless.

Solution

Please note: There has been an inconsistency between how I described SampleRate in this question, and how it is described in the paper. Here is the correct description of what SampleRate does on the 10th packet. It samples a rate that could potentially lead to a higher throughput (it can be a higher or lower rate), i.e., a rate that has a minimum transmission time lower than the current average transmission time. I have mistakenly described that it only samples higher rates in this question, though it can sample higher or lower rates.

a. SampleRate1 still operates at the best rate of 24 Mbps and hence is worse than SampleRate (which adapts the best rate to 36 Mbps). This is because SampleRate1 does not periodically probe for higher rates, and hence cannot adapt when best rate increases.
b. The answer to this part is ambiguous, given the ambiguity in the question I have pointed out above. You will be given credit for any answer, as long as you justify it fully.