ECS 152A Project

Winter 2015

Simulation Analysis of an IEEE 802.11-Based Wireless Local Area Network (WLAN)

Phase II

ECS152A: Computer Networks
Department of Computer Science
University of California, Davis

Introduction

In this project, you will build a discrete-event simulation model to study the behavior of an 802.11-based Wireless Local Area Network (WLAN). The first Phase of the project included the implementation of a single server queue model. Now, in Phase II, we focus on modeling and analysis of 802.11-based WLAN. This modeling will require the single server model implementation of Phase I. The following references and resources will be used in this project.

- Handouts and lecture/discussion materials.
- Textbook: J. F. Kurose and K. W. Ross, "Computer Networking, A Top Down Approach," Chapter 6, Sections 6.3, 6.3.1, and 6.3.2.

Schedule, milestones, and other administrative issues related to this project are the following:

- You are required to submit a final write-up for both phases of the project. The write-up contains a) a brief description of discrete-event simulation, b) description of Phase I, c) brief discussion of the overall logic of Phase I, d) description of Phase II, e) brief description of Phase II implementation logic, f) hard copy of the code for Phase II, g) Phase II results, and h) analysis of Phase II results.
- In phase II, you will be asked to explain your code and demonstrate your results to the TA during office hours. The schedule for project demonstration will be announced later.
- Phase II is due on **Tuesday, March 10, 2015, at 4:15 pm.**
- There may be questions related to this project in the final exam.

Phase II: Simulation Analysis of an 802.11-Based WLAN Event

In this phase, we will extend the single-server and single-queue model to multiple independent hosts that communicate using the 802.11 WLAN protocol. The goal is to analyze the behavior of the 802.11 CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol. The model of the system is shown in Figure 6.8 of our textbook (Kurose).

In order to develop this simulation model, we will make the following assumptions:

- We consider an ad hoc WLAN with N hosts. All the hosts are within each other's transmission range. Therefore, when a host transmits, every other host can hear the transmission. In such simple scenario, there will be no problem with hidden terminals (therefore, no RTS-CTS mechanism is required to implement).
- Each wireless host communicates with its random neighbor during simulation. Each wireless host has infinite FIFO buffer to hold outstanding frames.
- We assume that wireless hosts communicate on a shared error-free channel. That means, if there is no collision, the transmission between hosts will be successful.
- The wireless hosts use link layer acknowledgement mechanism as described in Section 6.3.2 of the textbook (Kurose). In this mechanism, a transmitting host will send data frame and the receiving host will send acknowledge frame in response.
- As in Phase I, λ will denote the mean arrival rate of frames. We will assume that the hosts are identical, each generating frames with a mean arrival rate of λ frames/second. The arrival process follows a negative exponential distribution (same as in Phase I).
- CSMA/CA with Distributed Coordination Function (DCF) will be used as underlying WLAN protocol. Therefore, the main focus of the transmission mechanism will be collision avoidance. Section 6.3.2 of our textbook (Kurose) provides the details of CSMA/CA protocol with DCF. In CSMA/CA, suppose that a wireless host has a frame to transmit:
    1. If initially the host senses the channel idle, it transmits its frame after a short period of time known as Distributed Inter-frame Space (DIFS) (see Figure 6.10 of the text book (Kurose)).
    2. Otherwise, the host chooses a random backoff value and counts down this value when the channel is sensed idle. While the channel is sensed busy, the counter value remains frozen[1].
    3. When the counter reaches zero (note that this can only occur while the channel is sensed idle), the host transmits the entire frame and then waits for an acknowledgement.
    4. If the receiving host receives the frame correctly, it waits for a shorter period of time known as the Short Inter-frame Spacing (SIFS) and then sends back an acknowledgment frame.
    5. If an acknowledgement is received, the transmitting host comes to know that its frame has been correctly received at the receiving host. If the host has another frame to send, it begins CSMA/CA protocol at step 2. If the acknowledgement is not received, the transmitting host can realize that collision occurred. Then, the transmitting host reenters the backoff phase in step 2, with the random value chosen from a larger interval.[2]

[1] If you have to generate a random backoff value which is uniformly distributed in the range (0, T), then you can do the following: 1) Generate a uniform random variable between (0, 1); 2) multiply that by T; and 3) round it off to the nearest integer.

[2] In this case, random backoff value is chosen from a larger interval. If number of unsuccessful attempts for transmission is n, then the interval can be (0; n ×T):

- The data frame length (r) (identically distributed for all nodes) is a negative exponentially distributed random variable in the range $0 < r \leq 1544$ bytes. The acknowledgement frame is constant in size (64 bytes). We also assume that the wireless channel capacity is 11 Mbps. Now, for example, if we ignore propagation delay, a wireless host will take $(1544 \times 8) / (11 \times 10^6) =$ 1.12 *msec* to transmit a 1544-byte long frame. This is similar to $1/\mu$ (service/ transmission time) in Phase I.

Enhance the code that you have developed in Phase I to model the above system. In this phase, we are interested in two parameters for investigating the 802.11-based WLAN's performance. We want to measure them by the influence of different number of hosts and the arrival rate:

1. Throughput: Throughput is defined as the number of bytes transmitted per unit time. In the simulation, you can count the number of bytes that are successfully transmitted and divide that by the whole simulation time to find the throughput.
2. Average network delay: We intend to find average network delay for the overall network. In the simulation, this delay includes the queuing delay and transmission delay for each host; ignore the propagation delay between the hosts. For our purposes here, assume channel transmission capacity to be 11 Mbps. If we divide the total delay for all the hosts with the achieved throughput, we will get the average network delay.

You need to obtain the following results:

a. Plot the throughput and average network delay as a function of λ. Let number of wireless hosts, N = 10. Obtain the throughput for the following values of λ = 0.01, 0.05, 0.1, 0.3, 0.6, 0.8, and 0.9 packets/second. Let us assume SIFS = 0.05 *msec* and DIFS = 0.1 *msec*. Your CSMA/CA implementation can sense the channel every 0.01 *msec*. Vary *T* in the random backoff algorithm to find out the effect of *T* on throughput and average network delay by keeping λ constant.

b. Repeat part [a] with N = 20.

Note that, to present your result clearly, it is better for you to plot two figures: one for throughput, and the other for average network delay.

Interactive grading will also be required. Please sign up for a time slot with the TA.