Assignment 2 : Broadcast Local Area Networks

Due the week of September 24/25, 2015

This week, we will explore the operation of the most widely used local area network technology, Ethernet, and begin to examine wireless networking technology.

I would like you to read §2.6 and §2.7 of Peterson and Davie this week. The first section covers Ethernet. The second section discusses wireless networks. These two topics are closely related since they are both examples of broadcast networks that depend on randomized collision resolution processes to share a single medium connecting many devices.

The treatment of these topics in the text is relatively brief because it presents the algorithms that are used without exploring the performance of these algorithms. The additional readings described below are designed to help you explore these performance issues in the context of Ethernet. Later in the semester, we will return to explore wireless networks more thoroughly.

The first additional reading is a "Classic" paper, "Ethernet: Distributed Packet Switching for Local Computer Networks" by R.M. Metcalfe and D.R. Boggs. This is the paper describing the original experimental system out of which "Ethernet" emerged.[1] The second paper has the distinction of having one of Ethernet's inventors as a co-author. The paper is "Measured Capacity of an Ethernet: Myths and Reality" by D.R. Boggs, J.C. Mogul and C.A. Kent.

The performance of an Ethernet depends on when the computers connected attempt to transmit data and on how much data they attempt to transmit. These details cannot be predicted exactly. Instead, analyses of Ethernet performance attempt to predict the network's expected behavior based on probabilistic models of the behavior of the attached computers. As a result, I will provide a link to a copy of some material about probability theory from a text by Walrand that may be helpful. This reading is optional.

Exercises

1. This question is inspired by the following question from a text by Tanenbaum.

    > Two CSMA/CD stations are trying to transmit long (multiframe) files. After each frame is sent, they contend for the channel, using the binary exponential backoff algorithm. What is the probability that the contention ends on round k, and what is the mean number of rounds per contention period?

    Tannenbaum describes a scenario involving multiple frames, but I would like you to limit your attention to the first pair of frames sent and to assume that where Tanenbaum says "mean" he meant to say "expected."

    The goal of this question is to get you to see how complex the behavior of the exponential backoff algorithm really is. My hope is that this will make you appreciate the need for some of the approximations in the analysis of Ethernet presented in the original paper. As a result, the

[1] Those who want a bit more background on the invention of the Ethernet might enjoy "Dealers of Lightning: Xerox PARC and the Dawn of the Computer Age" by Michael Hiltzik, a history of the research center at which Ethernet and many other computer technologies we take for granted today were first developed. This book is available in Schow. (It is not a required reading.)

answer to this question is not nice and simple. Expect nasty products or summations. Don't spend a lot of time trying to simplify terms, just try to mathematically capture the behavior of a real Ethernet as closely as you can.

In addition, please provide a formula for the mean number of "slots" per contention period. (Note that this quantity is what Metcalfe and Boggs associate with the variable W in their analysis.)

2. There is an interesting behavior exhibited by the binary exponential backoff algorithm known as the Ethernet capture effect.

   (a) To learn about it, complete parts a and b of question 2.43 from Peterson and Davie. (I don't like part c of their question. It is just a bit too vague and/or difficult mathematically.)

   (b) In their discussion of Ethernet performance, Boggs, Mogul and Kent make a statement that suggests the scenario explored by the Peterson and Davie question could not occur in their environment. Explain why. Could a similar situation occur in their environment? Explain.

3. Boggs, Mogul, and Kent clearly state that the failure to understand the correct interpretation of theoretical analyses of Ethernet behavior has led to "myths", but they never quite clearly state whether their results directly conflict with any published analytical results.

   One of them does.

   Consider the data shown in the very first graph presented in their paper, Figure 3.3. It shows that the efficiency achieved with 24 stations transmitting 64 byte packets is 85%. The analysis in the Metcalfe and Boggs paper provides the means to predict this efficiency. If, as they assume, binary exponential backoff effectively estimates Q, then the analysis given in §6 of Metcalfe and Boggs should enable you to estimate the efficiency when 24 stations are sending 64 byte packets.

   (a) Use the results in §6 of Metcalfe and Boggs to calculate the estimated efficiency when 24 stations are sending 64 byte packets. Your estimate should fall far below 85%.

   (b) One possible explanation for the mismatch between your calculations and the measured efficiency is the Ethernet capture effect explored in question 2. In their paper, Metcalfe and Boggs assume that the "Ether's time is divided between transmission intervals and contention intervals." The formula they give for efficiency assumes the network alternates between single successful packet transmissions and periods of contention. If the capture effect is significant, this assumption would be invalid. Instead, the network would alternate between periods where a station that had captured the network transmitted several packets, winning each contention round between packets in a single slot, and periods of true contention.

   Propose a formula for the efficiency of an Ethernet in this case. You will have to include an extra variable, S, denoting the expected number of consecutive transmissions made by a station while it has captured the network. Remember, that in their analysis, Metcalfe and Boggs made significant assumptions/approximations. You should not be surprised if you need to do the same. You should, however, state such assumptions clearly.

   (c) Using your formula and the data for the situation where 24 stations are sending 64 byte packets, can you find a value of S that is consistent with the 85% efficiency figure from the paper?

4. Warning. This is a very open-ended question.

Boggs, Mogul and Kent question the validity of the results published by many others. Anyone who does this deserves to have their own work subjected to similar scrutiny. I'd like you to take on the job.

I'd like you to imagine what you might do if you were one of them getting ready to submit the paper for publication. The paper depends on a great deal of data collected by programs that were probably complex enough to contain a bug or two. What if the bugs affected the data? This is a risk one must address when using software to collect data in an experiment.

There are many ways to validate such data. One approach is to look for properties of the data that can be predicted from "theory" and see if the data collected agrees with these prediction. Normally, it is not possible to predict all the data from theory. One can, however, often find limited cases or relationships between data collected in different scenarios that can be predicted.

For example, while the general behavior of an Ethernet may be too difficult to precisely analyze mathematically, one might be able to say something precise about situation where only two stations are active. Alternately, while one might not be able to predict the exact utilization with 25 stations, one might be able to predict the ratio of the utilization with 25 stations sending 1k packets to that with 25 stations sending 2k packets. If one makes such a prediction and the data contradicts it, then it is time to question the accuracy of the experiment.

For this problem I am asking that you try to find a few ways in which you can either argue mathematically that their data meets expectations or that there appear to be surprises. (Of course, surprises will be most interesting) To some extent, problem 3 is an example of what I have in mind (but don't use that example for this question). If you find any surprises, did the authors notice them as well? Do they explain them in the paper?

5. Consider the descriptions of the hidden node and exposed node problems given in §2.7.1 of Peterson and Davie.[2]

   (a) At the top of the text on page 138, the authors state that 802.11 **addresses** these problems (note the plural) using CSMA/CD. In the following paragraphs, they discuss how various mechanisms in 802.11 address the hidden node problem, but they never mention the exposed node problem again.

   Do either the introduction of ACKs alone or the combination of ACKs and RTS-CTS in 802.11 address the exposed node problem? Justify your answer.

   (b) Even the discussion of the hidden node/terminal problem on pages 138 and 139 is a bit confusing. The authors first indicate that 802.11 introduces ACKs "because of the hidden terminal problem." Then, however, in the next paragraph they say that the introduction of another (optional) mechanism, RTS-CTS, "goes some way to addressing the hidden terminal problem."

   So, it is actually unclear whether ACKs alone address the hidden terminal problem, whether ACKs in conjunction with RTS-CTS address the problem or whether even together, these two mechanisms fail to eliminate the hidden terminal problem in all cases.

   So, please argue either that...

      i. ACKs alone are sufficient to eliminate undetected collisions due to the hidden terminal problem. In this case, please explain what, if any, benefit RTS-CTS provides.

[2] Actually, the authors sometimes refer to these issues as the hidden and exposed node problems and at other points call them the hidden and exposed terminal problems. In fact, these are the same thing. Assume that node = terminal.

ii. ACKs alone are insufficient to eliminate undetected collisions due to the hidden terminal problem, but the combination of ACKs and RTS-CTS solves the problem. In this case, be prepared to demonstrate that either of the two mechanisms alone are not sufficient.

iii. Even ACKs and RTS-CTS together do not solve the hidden terminal problem. In this case, you must give specific scenarios showing how failure would occur.

iv. Did I miss any other options? If so, go for it.