Notes for MSIT 431 for Fall 2011

Statistical Methods[1] by A. H. Haddad

8. Medium Access Control

8.1 Introduction

In this section we are interested in evaluating the performance of the network layer dealing with medium access control. In order to simplify the problem we shall consider the ALOHA network, since it is easy to analyze and may serve as a model for more realistic networks.

Before we discuss the ALOHA network we have to place it in perspective. The entire subject matter of medium access control is involved with evaluating the performance of networks that have multiple users having access to the network. One way of eliminating the uncertainty when we have multiple users is to allow each user a fixed time slot on a rotating basis. This can be accomplished by a static TDMA system, which has a deterministic performance, but when the number of users is large, it can result in very large delays. The delay is N times larger than the average delay obtained by queuing all users in one large queue. That is why we prefer to use a dynamic channel allocation approach when dealing with multiple users rather than a fixed allocation. Such an allocation is obviously random in general and hence we need to use probabilistic methods to evaluate the resulting performance. The first such system we study is the ALOHA system, which was developed in Hawaii and is the precursor (in principle) to the type of protocol used in a typical Ethernet.

8.2 ALOHA network

We have two types of ALOHA systems. The first is called the pure ALOHA in which any user can transmit at any time at will, and of course when collision occurs, the users have to repeat the message. The second is a slotted ALOHA where each transmission can only start at the beginning of specified periods of time. We shall consider the slotted ALOHA first.

8.2.1 Slotted ALOHA:

We have a base station on a satellite to which ground users send messages, and the base station repeats all messages it receives so that they can be received by the destination station based on the addressing on the message. Users can start sending messages or packets at the beginning of fixed slots of length T. Obviously, since users can send messages at will there will be collisions, which means that such messages are not

[1]Copyright 2011 A. H. Haddad

received by the base station and when no successful broadcast of the message occurs, the sending user will retransmit the message until successful.

The features of the system maybe summarized as follows:

1. Star topology, all users communicate via the base station

2. All users share the same uplink channel

3. Broadcast of all incoming messages by base station on the downlink channel

4. Acknowledgements sent for all correctly received messages

5. Packets not acknowledged within a time-out period are retransmitted at a future randomly selected time

What are some of the parameters of the system?

- N = Number of user is assumed large

- T = Slot size = Message length in time is assumed to be fixed

- S = **Average** number of new messages generated at each slot

- Number of messages per slot is assumed to be Poisson

- p = Probability that a message is transmitted successfully

- G = Average number of messages through the channel

First, each user is restricted to send one size message (packet or frame). Total number of users is assumed to be large and typically is assumed to be N. Each message then is assumed to occupy a slot of time of fixed length T and a fixed starting instant. We assume that the number of messages generated by the many users has an average of S per time slot (the time it takes to send one message or frame), and the actual number of messages per such slot is assumed to satisfy a Poisson model. Due to retransmission, the average number of messages sent through the channel is actually G which is larger than S. Again we are speaking in terms of messages per slot of transmission time. When the load is low the value of G will be approximately equal to S, due to the fewer collisions. However, when more users are attempting to transmit, the number of collisions will be high and the value of G will be much larger than S. A block diagram representation of the system is shown in Figure 8.1.

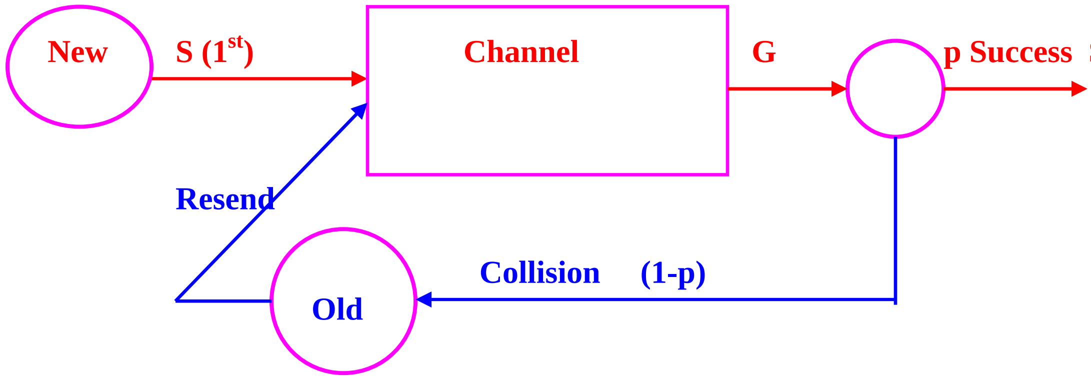

Figure 8.1 Diagram of the operation of ALOHA network

If we assume that the probability of successful transmission is p, then the throughput S is related to the channel transmission rate (one message per slot) by:

$$S = p \times 1 = p \qquad (8.1)$$

What is the value of p?

Obviously, p is the probability of success, namely, the probability of no collision. We therefore have (1 – p) = probability of a collision. If G is low, the probability of a collision is very small and hence p is large, while if G is large the probability of a collision is high and hence p will be small.

How do we find p?

Suppose we try to send a packet and suppose there are N users using the system at the same time, with N large. If the rate through the channel is G, then each user is assumed to transmit at a rate of G/N packets per time slot T. The probability of success (no collision) p is equal to the probability that only one user is transmitting at a given slot. We have:

$$P(\text{a user transmits during a slot}) = G/N \qquad (8.2)$$

$$P(\text{a user does not transmit during a slot}) = 1 - G/N \qquad (8.3)$$

$$p = P(\text{no collisions}) =$$
$$P(\text{1 user is transmitting during the slot}) = N\left(\frac{G}{N}\right)\left(1-\frac{G}{N}\right)^{N-1} \qquad (8.4)$$

For N large the approximate expression for p is:

$$p = G\exp(-G) \qquad (8.5)$$

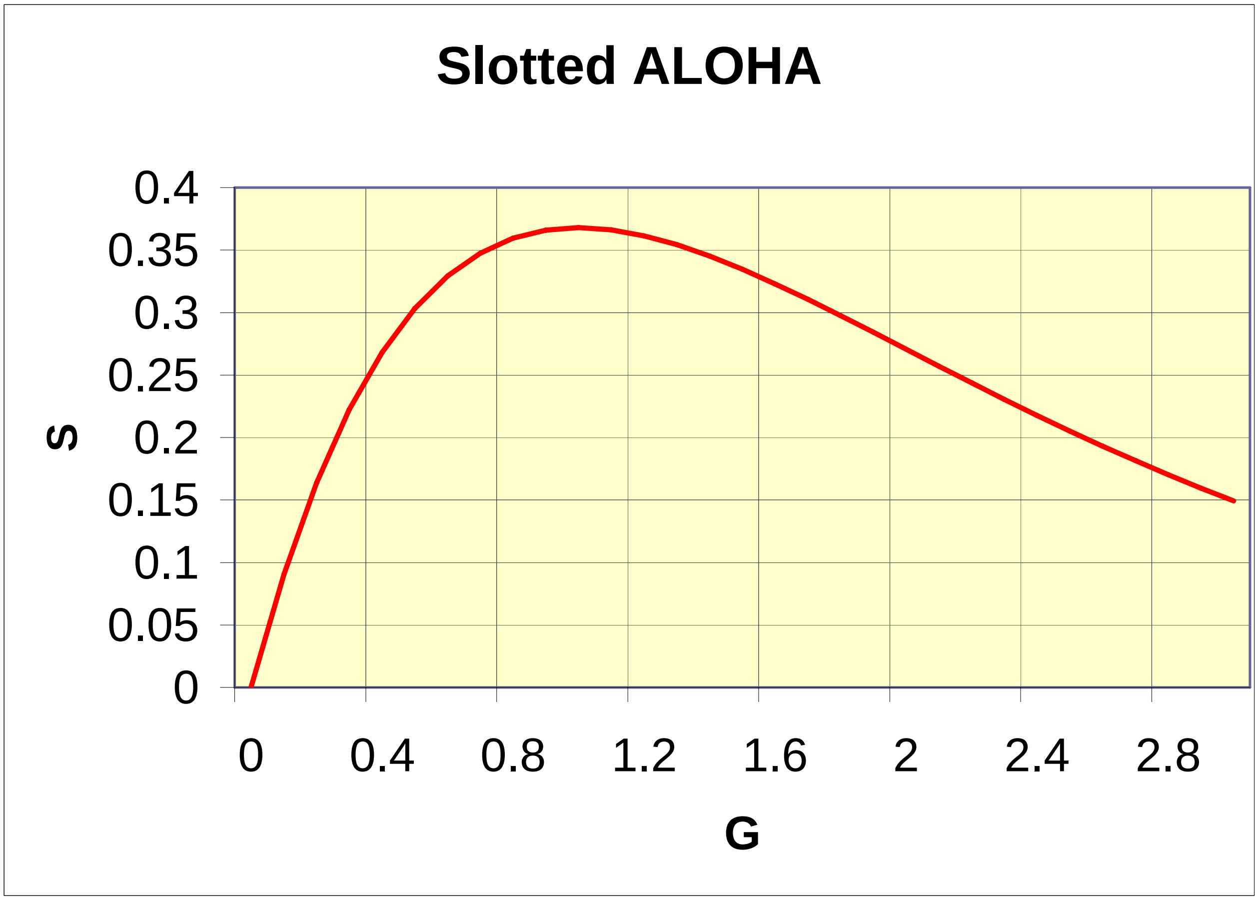

Figure 8.2 Throughput, S, as a function of channel load, G.

Hence, the resulting throughput satisfies the relation:

$$S = p = G \exp(-G) \qquad (8.6)$$

The expression for the throughput as a function of the load G through the channel is shown in Figure 8.2.

From the figure we can show that the maximum throughput occurs at G = 1, and it is equal to

$$S = \exp(-1) = 0.368 \qquad (8.7)$$

It means that the maximum channel utilization (or efficiency) in the slotted ALOHA system is ≈36%. The maximum utilization occurs when exactly one message per slot is transmitted through the channel. We see from the figure that for every value of S we have two values of G, one corresponding to the light load case with almost no collision, and one corresponding to the heavy load case with most of the traffic is made up of retransmitted packets.

It should be noted that by using the Poisson assumption, we could have arrived at the probability of no collision as equal to the probability of exactly one transmission when the arrival rate per slot is equal to G, which is indeed G exp(–G) as we obtained by our approximation shown in the analysis above in Equation (8.5). It is of interest to note that even though the efficiency is only 36.8%, the channel is idle (no messages sent per given slot) also 36.8% of the time as in Poisson model P{0 messages} = exp(–G), which for the optimal G=1 yields also 36.8%.

What is the average delay in the slotted ALOHA system? We shall compute the delay relative to the time slot of T for transmitting a single packet. Let T_{prop} define the one way propagation time from a node to the base station, and let B denote the back-off time on average that a node waits to retransmit after a collision, then we can use the methods introduced in Section 4.1 to compute the average delay based on Figure 8.3.

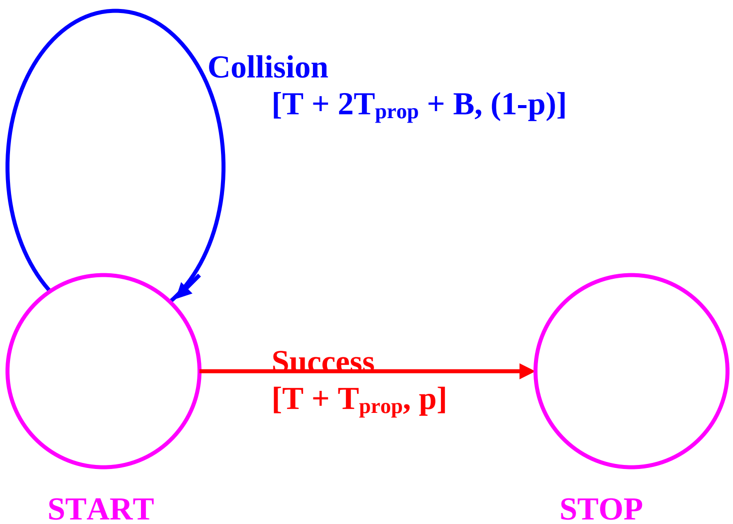

Figure 8.3 Diagram for computing the delay in ALOHA network

We start by writing the iterative equation for the delay from Figure 8.4 at the START node:

$$E\{D\} = p\,[T + T_{prop}] + (1 - p)[T + 2T_{prop} + B + E\{D\}] \quad (8.8)$$

The solution for E{D} is therefore:

$$E\{D\} = T + T_{prop} + \frac{(1-p)}{p}[T + 2T_{prop} + B] \quad (8.9)$$

When we normalize the delay by the time slot of transmitting a single packet, T, and we substitute the value of p from (8.5) and denote the ratio $\alpha = T_{prop}/T$, we obtain for the normalized average delay the expression:

$$E\{D\}/T = 1 + \alpha + [\exp(G) - 1](1 + 2\alpha + B/T) \quad (8.10)$$

8.2.2 Pure ALOHA:

The pure ALOHA preceded the slotted ALOHA, but the reason for our reverse study is that it is easier to handle the slotted ALOHA first. The difference between the two systems is that in the pure ALOHA any user can start transmission anytime. Therefore, if a user starts at some time t_0, then collisions can occur with any message that was started between $(t_0 - T)$ and $(t_0 + T)$. This fact is illustrated in Figure 8.4.

Figure 8.4 A message starting at t_0 may collide with messages starting in $(t_0 - T, t_0 + T)$

This means that a message can collide with messages that start over a two-slots period of 2T. Since the rate of transmission in one slot of size T is G, the average rate of messages in two slots, 2T, is equal to 2G, and hence the probability of successful transmission p is obtained by using the Poisson model and is equal to

$$p = 2G \exp(-2G) \qquad (8.11)$$

Consequently, the throughput as related to the channel load (1 message every 2 slots with probability p) is given by:

$$S = p \times 1/2 = G \exp(-2G) \qquad (8.12)$$

The throughput as a function of G is shown in Figure 8.5.

We see that the maximum throughput now occurs at G = 0.5, and that maximum is equal to:

$$S = 0.5 \exp(-2 \times 0.5) = 0.5 \exp(-1) = 0.184 \qquad (8.13)$$

It is exactly half the throughput for the slotted ALOHA case. It is to be expected since there are no restrictions on when users may try to transmit. The slotted ALOHA was proposed at that time to improve the channel utilization over the pure ALOHA system.

The average delay of a transmitted message is similarly obtained as for the slotted ALOHA and is expressed as:

$$E\{D\}/T = 1 + \alpha + [\exp(2G) - 1](1 + 2\alpha + B/T) \qquad (8.14)$$

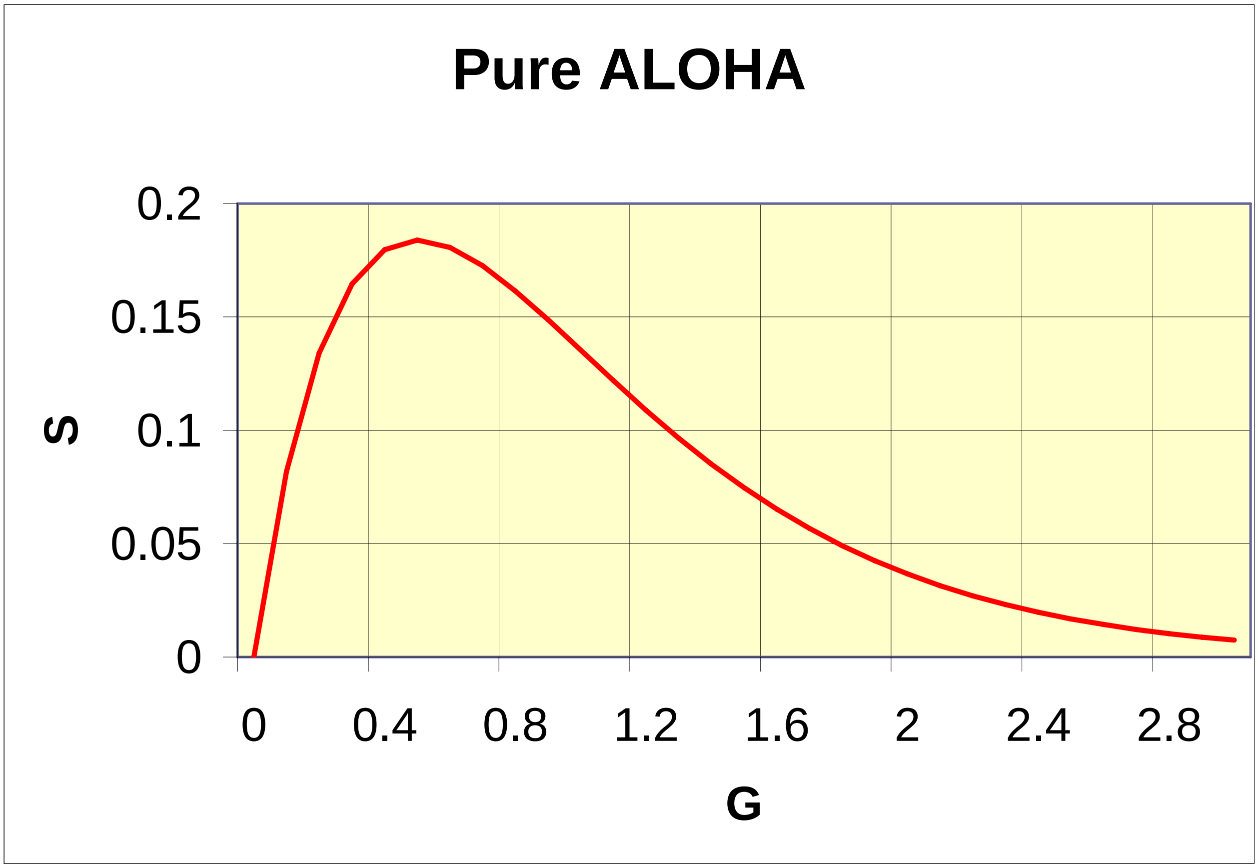

Figure 8.5 Throughput, S, as a function of channel load, G.

8.2.3 ALOHA with reservations:

Another approach to improve the ALOHA system is to include reservations. During a short interval T_{res} contention is allowed and all users may send messages requesting reservations to send a packet at a future slot. During the actual sending of the packet there are no collisions, since only one user is allowed to transmit. The only collisions occur during the reservation slot. We may apply the same rule of the slotted ALOHA that we used for the slot T, to that of the time slot T_{res} used for reservations. The slot length is T_{res} and the channel efficiency is at most 36%, as we obtained for the slotted ALOHA, so that the actual average time required for reservations (including the collisions) is $T_{res}/0.36$. If the transmission time for sending a packet is τ, then the efficiency of the system is:

$$\eta = \tau /[T_{res}/0.36 + \tau] = \tau /[2.8T_{res} + \tau] \quad (8.15)$$

If we assume that the T_{res} is about 5% of the transmission time of a regular message, namely 0.05τ, then the efficiency is approximately 88%.

What is the price of the reservation system? The price is the additional delay in getting the acknowledgement of the reservation message, which is high due to the round trip time of messages to the satellite from the Earth-based nodes.

8.3 Medium Access Control in Ethernet

Medium access control in an Ethernet is similar to ALOHA but with the advantage of shorter distances, which reduce the round-trip time of messages along the entire length of the network. In addition, the time wasted by everyone transmitting regardless if the channel is idle or not, reduces the efficiency of the ALOHA system. Therefore in an Ethernet ***carrier sensing*** is added to prevent a user from sending a message when the channel is not idle. What we have then is what is called a CSMA (Carries-Sensing Multiple Access) protocol. The best way to describe it is to show how it can be evolved from the ALOHA system. As in the ALOHA users may transmit anytime, however, before they do so, they listen to the channel, and only if there are no other messages present (the channel is idle), they start transmitting. Collisions can still occur due to the propagation delay T_{prop} in the system until the message signal reaches the furthest node on the network. There are variations of the CSMA scheme depending on how often nodes sense the channel during the busy period, and with what probability nodes transmit during the idle period. In order to improve on the inefficiency of the CSMA scheme, Collision Detection is added to evolve into the CSMA-CD scheme.

With the Collision Detection feature, the node stops transmitting as soon as a collision is detected. Since the maximum time needed to detect a collision will be the one that occurs between the furthest nodes on the network, the maximum time to detect a collision is therefore $2T_{prop}$, where T_{prop} is the one way propagation time from one end to the other end of the network. If a collision is detected, the node waits a random time-interval, and then repeats the process. The random time is determined based on the number of collisions and retransmission and is a multiple of the time-slot it takes to transmit 512-bits packet. After 1^{st} collision the node waits either 0 or 1 slot with equal probability. After the 2^{nd} collision, it waits from 0 to 3 slots delay, again selected randomly. This algorithm continues until the 9^{th} collision, namely after the m-th collision the delay time is from (0 to $2^m - 1$) slots. Between the 10 and 15 collisions the algorithm is the same as after the 10^{th} collision. If collisions occur during the 16^{th} retransmission, then no additional attempts are made. The exponential back-off algorithm helps reduce congestion in the network and as a result reduces the probability of additional collisions. After the first collision, since the choice is between two delays, the probability of a collision becomes $(0.5)^2 = 0.25$. After the second collision, the probability of a collision becomes $(0.25)^2 = 0.0625$, and so on. Between the 10^{th} and 15^{th} tries at retransmission the probability of a collision becomes $(1/2^{10})^2 = 2^{-20} \approx 10^{-8}$.

8.3.1 Performance of CSMA-CD:

Let us denote the maximum propagation time by $\rho = T_{prop}$. Since it takes 2ρ to determine a collision, let us assume that nodes listen to find whether a channel is idle every time-slot of length 2ρ. Let the transmission time of a packet be denoted by τ, and let M denote the average number of slots of size (2ρ) that are wasted due to collisions. The channel efficiency is then equal to:

$$\eta = \text{Useful time}/(\text{Total time}) = \tau /[M(2\rho) + \tau] \quad (8.16)$$

We now need to determine the average number M of slots wasted in collisions. Suppose the probability of success (no collision) is p, then the average number of slots needed until success can be derived from Figure 8.6.

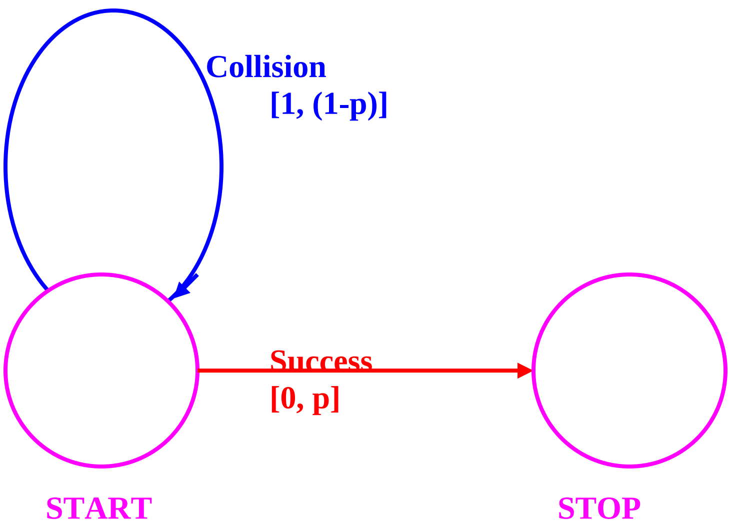

Figure 8.6 Diagram for computing the collisions in Ethernet

If a transmission is successful there are no wasted slots as the message will go through. If a collision is detected, then we are back to the start after one slot is wasted. The average M is therefore expressed as:

$$M = 0\times p + (1 - p)(1 + M) \quad (8.17)$$

So that the solution for M is:

$$M = \frac{1-p}{p} \quad (8.18)$$

Now we have to determine the value of p. Suppose that the network has N nodes, and assume that the probability that a node attempts to transmit is p_1. There is no collision if exactly one node only attempts to transmit. We have a binomial distribution with probability of p_1 and we want to find the probability exactly only 1 occurs:

$$p = P\,\{1 \text{ attempt out of } N\} = N(p_1)(1 - p_1)^{N-1} \quad (8.19)$$

Since we would like to have as large a p as possible, we maximize the value of p in (8.19) by selecting the probability p_1 of an attempt by any single node. The maximum occurs when

$$p_1 = 1/N \tag{8.20}$$

The resulting value for p becomes:

$$p = (1 - 1/N)^{N-1} \tag{8.21}$$

For large values of N the value of p approaches exp(–1) = 0.368, hence for a wide range of values of N we can use $p \approx 0.4$.

The value of M then becomes:

$$M = (1 - p)/p = 0.6/0.4 = 1.5 \tag{8.22}$$

The approximate efficiency of an Ethernet with this scheme is

$$\eta = \tau /[M(2\rho) + \tau] = \tau /[3\rho + \tau] = 1/(1 + 3a) \tag{8.23}$$

Here we defined $a = \rho /\tau$ as the ratio of the maximum propagation time to the time of transmission of a single packet. In practice the efficiency is more approximately given by:

$$\eta = \tau /[5\rho + \tau] = 1/(1 + 5a) \tag{8.24}$$

The difference between the approximate expression and the practical one stems from the fact that in the approximation we did not include the back-off algorithm when a collision was detected at the first or subsequent times.

Clearly, the performance depends on distance, which determines the one way propagation time, and on the parameter a, which depends on minimum packet length. In addition, the performance of course depends on the maximum transmission rate.

Notes for MSIT 431 for Fall 2011

Statistical Methods[2] by A. H. Haddad

9. Retransmission Protocols

9.1 Introduction

In this section we discuss additional network applications of probabilistic methods. We address the data link layer that in many cases handles the retransmission of incorrectly received packets. Sometimes this task is handled by a higher-level layer (such as the transport layer), which provides an end-to-end control rather than link-by-link control for correctly transmitting packets. Whatever the location of such control function our objective is to discuss several protocols used for retransmission of lost or incorrectly received packets and their analysis. Wherever the task resides the idea is that if a packet contains errors (based on some error control method) or a packet is missing from a sequence of packets in a message, the receiver asks for retransmission. However, the retransmission may also be performed automatically, if an acknowledgement that the packet has been received correctly is not received in time by the sender. Our objective is to analyze such protocols in terms of efficiency, which usually requires statistical methods. Protocols also have to be correct, in that they must ensure that exactly one correct copy of every packet is received. We shall not address that issue in detail here and leave such analysis to Networks I.

There are four commonly used protocols for retransmission (there are many more, but we shall discuss only these four basic types, as they provide a good illustration of how to use statistical methods):

1. Stop and Wait Protocol:

The transmitter waits after each packet to receive an acknowledgement that the packet was received correctly. If after a time-out period no acknowledgement is received, the transmitter sends the packet again.

2. Alternating Bit Protocol:

The transmitter numbers the packets with 0 or 1 alternately, and the acknowledgement are similarly numbered so the waiting time can be made shorter before the sender sends the next packet, rather than wait the full time-out of a round-trip transmission time.

[2]Copyright 2011 A. H. Haddad

3. Go Back N Protocol:

This protocol is the same as the alternating bit protocol but with a larger numbering sequence. A window of size W is used and a number of packets, which is W or less, is sent without waiting for the acknowledgement to arrive and are given consecutive number from 0 to W–1. Acknowledgements are similarly numbered, so that at the receiver packets losses can be identified. When a packet is lost it is retransmitted together with all packets that followed it in sequence.

4. Selective Repeat Protocol:

This protocol is the same as the Go Back N protocol except that only the packets that are not received are retransmitted.

The analysis of such protocols is based on assuming some probability of error (either a packet is dropped or received incorrectly, or the acknowledgement is lost). Then we can use the finite state transition model we studied in the past to analyze such protocols.

9.2 Stop & Wait Protocol (SWP)

This is the simplest of retransmission protocols as the sender sends one packet at a time and waits a time T which is larger than the round trip transmission time S for the packet and the acknowledgement to reach back to the sender. Figure 9.1 illustrates how this protocol works. The sender is shown in the top line and the receiver at the bottom lines. The arrows indicate a packet or an acknowledgement is sent. The arrows that end in nowhere indicate that the packet or the acknowledgement was not received. We see here that packet 1 had to be retransmitted and it appears that packet 2 will have to be retransmitted. The packets are identified with a number (e.g. 0 or 1) so that the receiver will know that a packet is a new one or a retransmission of a lost or late acknowledgement.

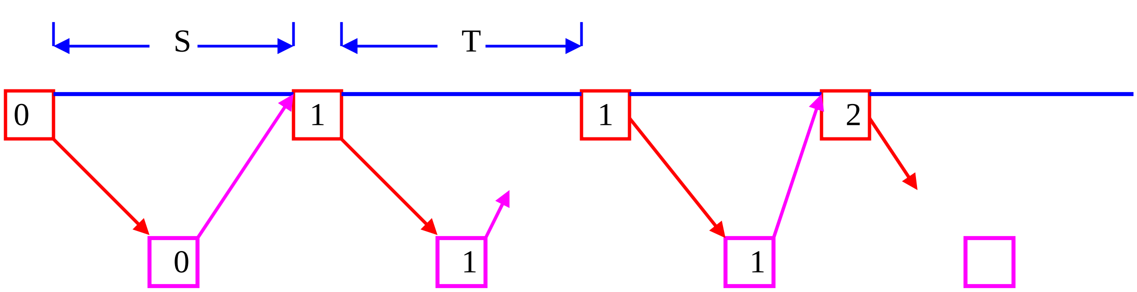

Figure 9.1An illustration of the stop and wait protocol

We shall now evaluate the efficiency η of the protocol. The efficiency is the ratio of the time it takes to transmit one packet to the average of total time that is consumed in sending and receiving correctly one packet. Another way to define it is by the throughput, which is equal to the number of packets per second that the channel is actually transmitting correctly, to the channel capacity in packets per second.

We use the state diagrams we used before in evaluating the efficiency of the ALOHA and other channels. We may number the states by a consecutive packet number (as in Figure 9.1). However, since we are looking at some general packet we number the packets from k to k+2 (instead of 0 to 2 shown in Figure 9.1). We assume that there is probability p of no error or no losses in the packet or the acknowledgement, and we define the time X as the random time it takes to successfully finish the transmission process of one packet, namely moving from state k to state (k+1). The transitions are shown in the Figure 9.2 together with the times and probabilities involved. The times shown in the figure are the round-trip transmission time of a packet, S, and the time-out period, T.

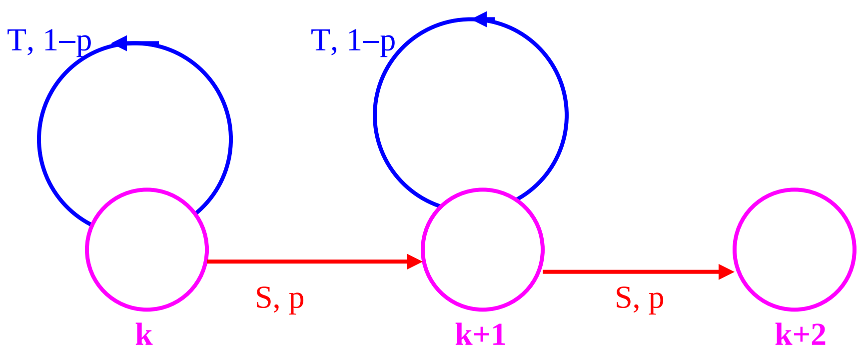

Figure 9.2 Diagram for evaluating performance of the SWP

We have from node k the following expression for E{X} = the average time to finish the successful transmission of one packet:

$$E\{X\} = p\,S + (1-p)[T + E\{X\}] \qquad (9.1)$$

The solution for E{X} becomes:

$$E\{X\} = S + \frac{1-p}{p}T \qquad (9.2)$$

If the one way transmission time of each packet is τ, then the efficiency of the protocol is:

$$\eta = \frac{\tau}{E\{X\}} = \frac{\tau}{S + T(1-p)/p} \qquad (9.3)$$

Note that S is the round trip time to send the packet and receive an acknowledgement without errors. Indeed if p = 1, all packets are received correctly, then the efficiency is just

$$\eta = \frac{\tau}{S} \qquad (9.4)$$

This protocol is not used, as it is very inefficient. It is discussed here because it is easy to follow the analysis, and thus it serves as a basis for the analysis of more complex protocols.

9.3 Alternating Bit Protocol (ABP)

This protocol is the same as the SWP except that the packets are provided an alternating number 0 or 1. The acknowledgements are also so identified, so that the sender waits exactly S for an acknowledgement for a packet labeled 0 to arrive before sending the next packet, which will be labeled 1. In a way, this protocol is still wasteful, with the only savings is the waiting time T when the packet is lost, which in this case can be as small as the time to transmit a single packet. The efficiency has exactly the same expression as the SWP except for the smaller value of T, since it needs no longer be larger than S.

9.4 Go Back N Protocol (GBN)

This protocol uses the network resources much more efficiently that either of the other two described earlier. It also numbers the packets but it has a much larger window than the binary case above. It uses a window size W (integer) so that W packets are transmitted before working on retransmission of lost packets. Each packet is numbered from 0 to $W-1$, and the acknowledgements are also numbered so that the identity of the packet that is in error or that its acknowledgment is in error can be specified. What this protocol does when such an error occurs is to retransmit that packet and all the ones that follow it in the window. If it does not receive acknowledgements after a time-out of length T, then all packets in the window are retransmitted. This last case is the same as when the acknowledgment to the packet labeled 0 is lost. The efficiency of the protocol when there are no errors is very good and is equal to 1 when the time to send the W packets in the window is larger than S (the round trip time for the acknowledgement to arrive at the sender). If on the other hand the time to transmit W packets is less than S, the efficiency will be equal to the ratio $W\tau/S$, assuming the time-out is equal to S (there is no need for it to be larger than S). We next discuss what happens to the efficiency in the presence of errors or losses.

Again we use the same figure we used before and it is shown in Figure 9.3 with the parameters relevant to GBN protocol. In this case we have the same type of states in sending packet k out of a window of size W. The difference is that it takes a time equal to, τ, the transmission time of a packet to be successful, while if unsuccessful it takes time S (the time-out period), which in this case is equal to $W\tau$. The latter assumption provides for the best efficiency when there are no errors, as we mentioned earlier.

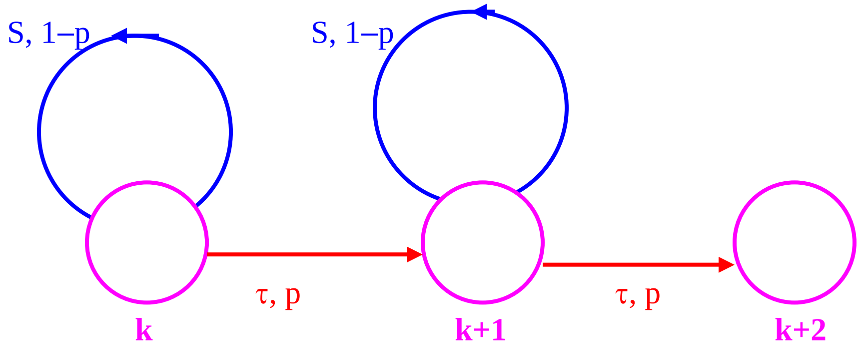

Figure 9.2 Diagram for evaluating performance of GBN

Here again we shall denote the time to transmit one packet successfully as X, so that the average is equal to:

$$E\{X\} = p\,\tau + (1 - p)\,[S + E\{X\}] \qquad (9.5)$$

The solution for the average time is:

$$E\{X\} = \tau + \frac{1-p}{p} S \qquad (9.6)$$

The efficiency for the protocol will be the ratio of τ to E {X}:

$$\eta = \frac{\tau}{\tau + S(1-p)/p} = \frac{1}{1+(S/\tau)(1-p)/p} = \frac{1}{1+W(1-p)/p} \qquad (9.7)$$

The last expression is obtained if indeed we select the window size such that it satisfies:

$$S = W\,\tau \qquad (9.8)$$

Note that while such a choice provides us with 100% efficiency in the error-free case, it does worsen depending on the loss or error probability (1–p), but it does not get as bad as the previous two protocols we considered.

For W = 16, and p = 0.9 we obtain 36% efficiency, and for p = 0.99 we obtain efficiency of 86%. On the other hand if we select T = τ, while S remains as 16τ, in the ABP we obtain an efficiency of 6.2% and 6.25% for the two values of the error probability.

9.5 Selective Repeat Protocol (SRP)

As we found out in the GBN case, while the efficiency is excellent when there are no errors, it becomes worse as the probability of error increases. The main problem is the fact that when a packet in the window is lost all succeeding packets have to be retransmitted as well. This deficiency can be corrected if we allow for packet storage at the receiving end, so that only lost packets will have to be repeated. The receiver needs to store up to (W−1) packets, so that it can deliver W intact packets even if one of the packets arrives out of order, if it is lost the first time it was transmitted. We call this protocol Selective Repeat since it only retransmits packets with no acknowledgment received by the sender. If the packet was indeed lost, the receiver waits for the missing packet before delivering the W packets together. If it was retransmitted because the acknowledgement was lost, then the receiver still keeps the packets so that it can determine that this is a duplicate packet and can discard it while delivering the rest of the packets. To avoid ambiguity, even though the window size is W the packets need to be numbered using 0, 1, 2, …, 2W−1.

The efficiency of this protocol is the same as the GBN when there are no errors in the system. If $S = W\tau$, then we obtain an efficiency of 100% when there are no errors. The efficiency for the case with errors follows the same model as the GBN except that the actual time wasted in the retransmission of a packet is only τ and not S. The delay (or waiting time) maybe S but not the time wasted on retransmission. Hence an approximate value for the efficiency is:

$$\eta = \frac{\tau}{\tau + \tau(1-p)/p} = \frac{1}{1+(1-p)/p} = p \tag{9.9}$$

This is a very crude approximation as it assumes that the only wasted slot is the time to retransmit a single packet and hence the efficiency is equal to p = P{no error or loss}.

9.6 Congestion Control

In this section we consider another application of probability and that is one approach to the analysis of congestion control methods in a TCP network. The network establishes a window W that determines the number of messages that a source can transmit. One approach to the determination of the window size is that a source starts with some window size (say 1) and adjusts the window size based on congestion. Congestion is determined by the fact that a packet is lost (it means that there is no acknowledgement, for example). When congestion is observed the window size for the next transmission is halved. If a packet is not lost then the window size is incremented by 1/W. Suppose the probability that a packet is lost is p (we used p in Sections 9.2 to 9.5 to denote the opposite). We can therefore describe the model for the window size as follows:

$$W_{k+1} = 0.5\, W_k, \qquad \text{with probability } p \tag{9.10}$$

$$W_{k+1} = W_k + 1/W_k, \qquad \text{with probability } (1-p) \tag{9.11}$$

We need therefore to find the average window size in a steady-state operation, which means that the average of W_{k+1} is equal to the average of W_k:

$$E\{W_{k+1}\} = p\, 0.5\, E\{W_k\} + (1-p)\, E\{W_k + 1/W_k\} \tag{9.12}$$

If we assume that all the averages are the same in steady state and if we define W as the steady-state average window size, we obtain:

$$0 = -0.5p\,W + \frac{1-p}{W} \qquad (9.13)$$

Te window size becomes therefore:

$$W = \sqrt{\frac{2(1-p)}{p}} \approx \sqrt{\frac{2}{p}} \qquad (9.14)$$

The above result for small p becomes the well-known square-root law. The window size is inversely proportional to the square root of the loss probability.

There are many other approaches to congestion control, but will not be explored further in these notes.