1 Multiaccess Protocols

In this section, we consider the situation where multiple end host share a single broadcast channel to send and receive data packets. This means that when more than one host trying to send a packet, packets will collide (on the broadcast channel) and are lost. Thus, in order to avoid collisions, and recover from situations where a collision occurred, a (mutliaccess) protocol is needed. What makes this problem challenging is that the individual nodes do not know whether other nodes have packets to be sent.

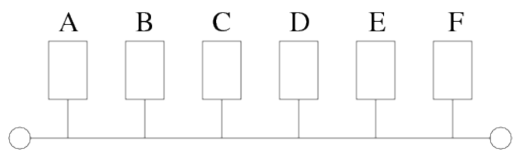


Figure 1: Multiple hosts sharing the same channel

We will focus on contention-based protocols, where (roughly) a node immediately sends a new data packet, hoping that no other host is active and no collision will occur. When a collision occurs, the end hosts will retransmit the packet at a later time. The interesting question here is how, and when, end hosts should schedule their retransmissions.

We will first consider a simple multiaccess model called Aloha. This model allows us to obtain insight into the basic properties of contention-based protocols.

1.1 Slotted Aloha

The Aloha network was developed by the University of Hawaii around 1970. Hawaii consists of several islands and the university has its campuses in each of these islands. So, the university decided to build wireless connections between the computer network of the main campus and those of the remote campuses. Whenever a remote campus has a packet that needs to be sent to the main campus, it just transmits it through the air (cheap!). If another remote campus sends its own packet while the first packet is being transmitted, collision takes place and none of the packets will be received by the main campus. Therefore, a protocol was needed for scheduling the retransmission attempts of backlogged packets. The difficulty here is that the individual nodes do not know whether (a) other nodes have a new packet to be sent, (b) how many nodes were involved in a collision, and (c) when other nodes will attempt to transmit their backlogged packets. Aloha is a protocol that addresses these issues. There are two versions of Aloha: unslotted Aloha and slotted Aloha. The difference between the two versions is that packet transmission can start at any time in an unslotted Aloha system, while packets can only be transmitted during synchronized time slots in a slotted Aloha system. We will focus on slotted Aloha systems.

1.1.1 Assumptions

For the analysis, we make the following assumptions.

Figure 2: Time is divided into unit slot.

- **Slotted system**: Each packet has the same length L. The channel capacity is C. Time is divided into slots and each packet requires one slot for transmission. We re-scale time and set 1 unit time equal to $\frac{L}{C}$ seconds (the transmission delay of one packet).
- **Poisson arrivals**: Each host generates new packets according to a Poisson process. The overall arrival rate of new packets to the system is λ.
- **Collisions or perfect reception**: If more than one host transmit a packet in the same time slot, there is a collision and those packets are lost. If only one packet is transmitted in a time slot, that packet is received correctly.
- **Immediate feedback**: At the end of each slot, each host can determine whether the slot was idle, one packet was successfully transmitted, or a collision occurred.
- **Retransmission of backlogged packets**: Packets that are involved in collision must be retransmitted in some later slot until it is finally received. Nodes that have packets that need to be retransmitted are said to be backlogged. Here, we assume that each backlogged node retransmits with a fixed probability $q_r > 0$ in each successive slot until successful.
- **Infinite number of hosts**: The system has an infinite set of hosts and each host has at most one packet to transmit.

When a collision occurs, each host will discover the collision at the end of the slot, and retransmit packets that collided after waiting for some random time. The average waiting time until the next retransmission attempt is equal to $\frac{1}{q_r}$ time units. One question that we want to answer is how nodes should choose the retransmission probability q_r. Note that when q_r is very small, then the (expected) idle time between retransmission attempts becomes very large. On the other hand, choosing q_r equal to 1 will lead to a system deadlock (why is that?).

1.1.2 Mathematical Model

In this section, we carry out a (simplified) analysis of the slotted Aloha system. We denote with n the number of nodes that have a packet to send. During one time slot, events occur with the following probabilities.

- Probability of a idle slot: $(1 - q_r)^n$
- Probability of a successful transmission: $nq_r(1 - q_r)^{n-1}$
- Probability of a collision: $1 - (1 - q_r)^n - nq_r(1 - q_r)^{n-1}$

Let P_{succ} be the probability of a successful transmission during a time slot. We then have

$$\begin{aligned} P_{succ} &= nq_r(1-q_r)^{n-1} \\ &= \frac{nq_r}{1-q_r}(1-q_r)^n \end{aligned}$$

When q_r is small, we can use the following approximations,

$$(1-q_r)^n \approx e^{-nq_r} \qquad and \qquad \frac{nq_r}{1-q_r} \approx nq_r,$$

and we obtain

$$\begin{aligned} P_{succ} &\approx nq_r e^{-nq_r} \\ &= G(n)e^{-G(n)}, \end{aligned} \tag{1}$$

where

$$G(n) = nq_r.$$

As we rescaled time such that the length of each time slot is equal to 1 time unit, when the attempted transmission rate is equal to G then the throughput γ of the system is equal to Ge^{-G} packets per unit time.

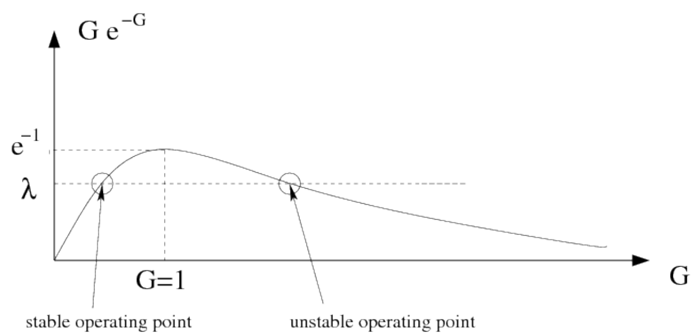


Figure 3: Slotted Aloha: Throughput γ as a function of attempted transmission rate G

Figure 4 shows the relationship between the attempted transmission rate G and the throughput γ. We make the following observations. The highest possible throughput is equal to $e^{-1} \approx 0.368$, and is achieved for the attempted transmission rate $G^* = 1$. When the attempted transmission rate is very small (the system is lightly loaded) then there will be few collision; but the channel will be idle most of the time and the throughput γ will be small. On the other hand, when G is large and the system is heavily loaded, then in almost every time-slot a collision will occur and the throughput will be very small. The stable operating point of the system is when the throughput γ is equal to λ which is the arrival rate of new packets. When $\lambda > \gamma$, then (on average) more packets will arrive than depart, and the number of backlogged nodes, and the attempted transmission rate G, tend to increase. When $\lambda < \gamma$, then (on average) more packets depart than arrive, and the number of backlogged nodes, and the attempted transmission rate, tends to decrease.

In the above figure, we observe that for any $\lambda < \frac{1}{e}$, there are two values of G for which we have $\gamma = \lambda$. Note that the operating point on the left is a stable operating point as for small deviations the throughput will drift back to the operating point. However, the operating point on the right is unstable as for small deviations the throughput will drift away from the operating point. In particular, an additional collision will increase the number of backlogged nodes, and therefore the attempted transmission rate, causing the throughput of the system to drift towards 0, and the number of backlogged nodes will drift towards ∞.

One approach to avoid this instable behavior of the system is to dynamically change q_r as the number of backlogged nodes n changes. Naturally, we would like to choose the retransmission probability to maximize the throughput, and therefore reduce the number of backlogged nodes as fast as possible. As the maximal throughput is obtained for $G = 1$, we should choose q_r such that $nq_r = 1$. Unfortunately, the individual nodes do not know the exact number n of backlogged nodes, but can only be estimated by observing the channel. There are many strategies for estimating n and choosing the retransmission probability q_r. In essence, all of them increase q_r when an idle slot occurs, and decrease q_r when a collision occurs.

1.2 Improving the performance of Aloha

As it turns out, the Aloha protocol is not efficient and can be improved. In particular, we observe that

- **Idle slot**: when a packet arrives during an idle slot, the node needs to wait for the whole slot until the packet can be transmit.
- **Collision**: When a collision occurs, then nodes do not terminated their transmission, but use up the entire slot time.

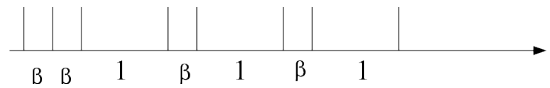
To address this, we can use the following approaches:

- **Carrier Sensing, Multiple Access(CSMA)**: Nodes sense whether the channel is idle or not. As soon as all nodes detected that the channel is idle, new packets can be transmitted, therefore reducing the length of an idle slot.
- **Collisions Detection (CD)**: When a node detects a collision during the transmission, it will stop immediately with the transmission, therefore reducing the length of a collision slot.

1.3 Slotted CSMA

In CSMA slotted Aloha system, nodes detect idle periods and can therefore terminate an idle period (slot) quickly. Let β be the propagation and detection delay (in packet transmission units) required for all sources to detect an idle channel after a transmission ends. Before transmitting a packet (which could either be a new packet or a backlogged packet), the node will first need β time units to detect an idle period, and then initiate the packet (re-)transmissions. The model of CSMA slotted Aloha is given as follows (see also figure below).

- Time is divided into slots (unit time = $\frac{L}{C}$ seconds):
- Length of idle slot is equal to β time units.
- Poisson arrivals with aggregated rate of λ packets/time unit



- Immediate Feedback whether a slot was idle or busy.
- Retransmission Probability: q_r

The main difference between CSMA and ordinary Aloha is that idle slots in CSMA have a duration of β time units.

1.3.1 Analysis

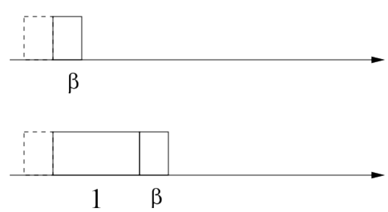
Note that in CSMA, a successful transmission can only take place after an idle slot. Consider a given time slot, and assume that the time previous slot was an idle slot. The probability P_{succ} of a successful transmission after this idle slot is then again given by

$$\begin{aligned} P_{succ} &= nq_r 1 - q_r{}^{n-1} \\ &\approx g(n)e^{-g(n)}, \end{aligned}$$

where

$$g(n) = nq_r.$$

To compute the throughput for CSMA, we have to take into account that idle slots are shorter than busy slots. As a transmission attempt in CSMA only occurs after an idle slot, we can identify the following two "basic" events that can happen after an idle slot (see figure below): (a) another idle slot (of length β) or (b) a busy slot followed by an idle slot (of total length $1+\beta$). Let A be



the even that there is no transmission attempt after the idle slot (and we have another idle slot) and let B be the even that there is at least one transmission attempt (and we have a busy slot). Then the average duration $E[T]$ of an event is given by

$$E[T] = E[T \mid A]P\{A\} + E[T \mid B]P\{B\},$$

and (using the above approximation) we obtain

$$\begin{aligned} E[T] &= \beta \cdot e^{-g(n)} + (1+\beta) \cdot (1 - e^{-g(n)}) \\ &= \beta + 1 - e^{-g(n)} \end{aligned}$$

The throughput of CSMA slotted Aloha (i.e. the average number of successful number of successful transmission per unit time) as a function of the attempted transmission rate g is then equal to

$$\gamma = \frac{P_{succ}}{E[T]} = \frac{ge^{-g}}{E[T]} = \frac{ge^{-g}}{\beta + 1 - e^{-g}}$$

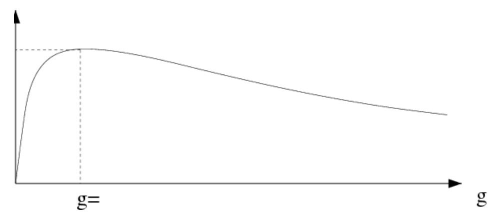


Figure 4: Slotted CSMA Throughput as a function of attempted transmission rate G

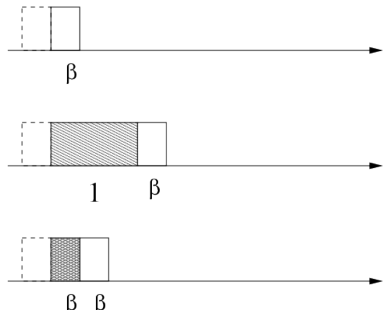
The above figure shows the graph of γ as a function of g. For small β, CSMA has a maximum throughput of approximately $\frac{1}{1+\sqrt{2\beta}}$ at $g = \sqrt{2\beta}$. CSMA slotted Aloha has the same stability problem as ordinary slotted Aloha. For fixed q_r, $g(n)$ grows with the backlog n. When n becomes too large, the departure rate is less than the arrival rate, increasing the number of backlogged packets even more. Note that β/q_r is the expected idle time that a backlogged node must wait to attempt transmission, and for small β and modest λ, q_r can be quite small without causing much delay. This means that the backlog must be very large before instability sets in. So the stability problem is less serious for CSMA than for ordinary Aloha.

From the above discussion, we can see that CSMA slotted Aloha significantly increases the throughput of multiaccess channels by shortening the idle slots in systems. But another problem comes up. If the packets of data being transmitted are long, why waste these long slot times on sending colliding packets? It would be fore more efficient if the slots wasted by idles or collisions are all shorts.

1.4 Slotted CSMA/CD

Assume that nodes also detect collisions and abort their transmission after β time units when a collision occurs, where β is the (propagation) time needed to sense a signal on the channel. As in CSMA systems, before transmitting a packet (which could either be a new packet or a backlogged packet), the node will first need β time units to detect an idle period and then initiate the packet (re-)transmissions.

To compute the throughput for CSMA/CD, we have to take into account that idle and collision slots are shorter than slots with a successful transmission. As a transmission attempt in CSMA/CD only occurs after an idle slot, we can identify the following to "basic" events that can happen after an idle slot (see figure below): (a) another idle slot (f length β), (b) a busy slot followed by an idle slot (of total length $1 + \beta$), or a collision slot followed by an idle slot (of total length 2β) Let A be the even that there is no transmission attempt after the idle slot (and we have another idle slot), let B be the even that there is exactly one transmission attempt (and we have a successful slot),



let C be the even that there is more than one transmission attempt (and we have a collision slot). The average length of an event is then given by,

$$\begin{aligned} E[T] &= E[T \mid \text{A}]P\{A\} + \\ &\quad E[T \mid B]P\{B\} + \\ &\quad E[T \mid C]P\{C\}, \end{aligned}$$

and we obtain

$$\begin{aligned} E[T] &= \beta \cdot e^{-g(n)} + \\ &\quad (1+\beta) \cdot (g(n)e^{-g(n)}) + \\ &\quad 2\beta \cdot (1 - e^{-g(n)} - g(n)e^{-g(n)}) \\ &= \beta + g(n)e^{-g(n)} + \beta[1 - (1+g(n))e^{-g(n)}]. \end{aligned}$$

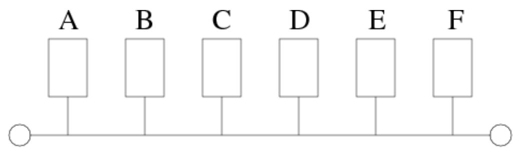
The throughput γ as a function of the attempted transmission rate g is then given by

$$\gamma = \frac{P_{succ}}{E[T]} = \frac{ge^{-g}}{\beta + ge^{-g} + \beta[1-(1+g)e^{-g}]}$$

For CSMA/CD system is stabilized, the maximal throughput is equal to $\frac{1}{1+3.31\beta}$ at $g = 0.77$. As in the other system, CSMA/CD can become unstable.

Note that CSMA/CD and CSMA become increasingly inefficient with increasing bus length(that is increasing propagation delay), with increasing data rate, and with decreasing data packet size.

1.5 Ethernet



- Uses CSMA/CD
- Uses Binary exponential Backoff
- Does not use time slots
- Ethernet provides connectionless service

The frame of Ethernet is briefly showed as following:

- **Preamble (8 bytes)**: Synchronization
- **Destination Address (6 bytes) - Source Address (6 bytes)**
- **Type (2 bytes)**: Multiplexing (of Network protocols)
- **Data (46-1500 bytes)**
- **Cyclic-Redundancy Check (4 bytes)**: Error detection

1.5.1 Ethernet Protocol

- If the adapter senses that the channel is idle and has a frame to transmit, it starts to transmit the frame. If the adapter senses that the channel is busy, it waits until it senses no signal (plus 96 bit times) and then starts to transmit.
- If the adapter detects a signal from other adapters while transmitting, it stops transmitting its frames and instead transmits a 48-bit jam signal.
- After aborting, the adapter enters an **exponential backoff** phase.
- After experiencing the nth collision in a row for this frame, the adapter chooses at random a value K from $\{0, 1, ..., 2^{m-1}\}$ where $m := \min(n, 10)$. The adapter then waits $K \cdot 512$ bit frames and then tries to retransmit the frame.