Lecture 13: Medium Access Control, the ALOHA protocol

Anirudh Sivaraman

2018/12/17

So far we have looked at the application, transport, and routing layers of the Internet stack. These are the layers of the Internet that are traditionally associated with computer scientists and/or computer engineers. Over the next two weeks, we'll cover the two lowest layers of the stack: the medium access control (MAC) layer (also called the link layer) and the physical layer. The lowest two layers have traditionally been the domain of electrical engineers, though there has been much recent research work in these two layers that sits squarely in the middle of electrical engineering and computer science.

These two layers get much closer to several physical realities that constrain communication, e.g., interference between laptops that are close to each other, the loss in intensity of an electromagnetic wave as it propagates from one point in space to another, and how bits are represented by voltages so that they can be transmitted within a cable. The flip side of getting close to reality is that we can no longer paper over problems using abstractions and must confront them head on.

This week, we'll look at the medium-access control (or MAC) layer. In particular, we will look at the problem of arbitrating access to a *shared communication medium* among several different users. Examples of shared communication media are all around us. WiFi networks are one example, where electromagnetic (EM) waves belonging to a certain frequency range (2.4 to 2.48 GHz) are reserved for WiFi communication. Regardless of how many users are in a particular room, the same 80 MHz range (2.4–2.48 GHz) must be shared across all these users in some manner. Another example is the variety of cellular network bands that are available: e.g., there is an LTE band from 2.11 to 2.17 GHz and a GSM band from 390.2 to 399.8 MHz. Again, regardless of the number of users, the same frequencey range must be shared across all users. Decisions regarding which link-layer technology gets to use which frequency range are handled in the U.S. by the Federal Communications Commission (FCC).

The frequency range of a shared communication medium is also called the bandwidth of the medium (e.g., for WiFi this is 80 MHz because it spans the range 2.4–2.48 GHz). In general, the total available capacity in bits/second in a shared communication medium increases with the bandwidth of the medium.[1] This relationship between bandwidth and capacity is partly why the term bandwidth is used interchangeably with capacity in many situations. In reality though, they are different concepts: bandwidth is an analog quantity measured in Hz, while capacity is a digital quantity measured in bits/sec.

Not all media are shared. An Ethernet cable between a desktop and the wall socket is an example of a dedicated medium because each desktop has exclusive access to its own Ethernet cable without sharing it with other desktops. This desktop can use up the full capacity of its own cable. Further, a second desktop can be added with its own cable and can transmit at the full capacity of the second cable. In other words, there isn't a shared medium (such as a particular frequency range in WiFi) that is being divided up among the desktops.[2]

The next two lectures will deal with the problem of arbitrating access to a shared medium such as WiFi or a cellular network like GSM, WiMAX, or LTE. There are a few different ways to do this, and we'll consider each in turn over this and the next lecture.

[1] For the very specific case of a point-to-point communication medium with a single link connecting a single sender and receiver, the relationship between bandwidth in Hz and capacity in bits/sec is captured by an equation known as the Shannon-Hartley theorem. For more complicated communication media, such as multiple senders and a single receiver (e.g., multiple laptops and a WiFi AP), a mathematical characterization of capacity still remains an open problem.

[2] It is possible that both desktops share a common link deep inside the Internet if they are transmitting to the same destination. However, the sharing in that case isn't happening at the link layer. It's happening at the transport layer, when end hosts use AIMD to achieve fair allocation of link capacity.

1 Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA)

A simple mechanism to arbitrate access to a shared medium is Time Division Multiple Access (TDMA), where each user on a shared medium uses the medium for a certain fixed amount of time (called a slot) before relinquishing the medium to the next user in round-robin order. While this is simple, it is not particularly efficient for bursty workloads, where a user transmits a large amount of data, but in bursts separated by large periods of inactivity. For such bursty workloads, TDMA ends up wasting slots. TDMA can be modified to skip slots for users that do not have data waiting to be transmitted, but this ends up complicating TDMA considerably. Another simple mechanism to arbitrate access to a shared medium is Frequency Division Multiple Access (FDMA), where each user is given a slice of the frequency range (or bandwidth). Such a slice is also called a channel.

Both TDMA and FDMA typically require a centralized authority such as a base station that decides which time slots belong to which users or which frequency slices/channels belong to which users. Both mechanisms are commonly used in cellular networks, where the centralized base station hands out channels and/or time slots to users connected to it currently. TDMA/FDMA also have the additional benefit of deterministic performance: once a time slot or frequency channel has been allocated to a user, it is not revoked until the number of users changes or the allocated user is done. This allows the user more control over performance because each user knows exactly which time slots or how many frequency channels are available to them.

On the other hand, the requirement of a centralized authority means that TDMA and FDMA are not well-suited to networks like WiFi that are deployed in an ad hoc manner without a formal registration or subscription process for customers of the network. TDMA/FDMA are also not a great fit for the bursty workloads typical of the Internet today because time slots or channels may idle if there is no data to send. For such networks, it is preferable to use a distributed protocol where nodes contend with each other to share the medium whenever they have traffic to send. The difference between TDMA/FDMA and contention-based MAC protocols is similar to the difference between circuit and packet switching.

2 Contention-based protocols and an analogy between MAC and congestion control

An alternative to having a centralized coordinator hand out time slots or channels is for users themselves to determine who should transmit and when in a *distributed* fashion—much like the AIMD algorithm picks the right value of the window size for several window-based senders in a distributed manner. In fact, MAC and congestion control are similar in many respects: both deal with the allocation of a shared resource, either the capacity of a link at the network/routing layer (congestion control) or the capacity/bandwidth of a shared communication medium at the MAC/link layer (MAC).

The difference between them is twofold. First, MAC is a *local* problem at the level of a local WiFi/cellular network, which can be completely isolated from the rest of the network, especially if the bottleneck link is in the local network. By contrast, congestion control is a *global* problem, which requires multiple endpoints scattered across the Internet to coordinate with each other. Second, because MAC is a local problem, MAC can strive to optimize local metrics such as the utilization of the local network or fairness among the local network's users—whatever makes sense locally. Because congestion control is a global problem, it is harder to optimize for a single metric, because it is hard to get global consensus on such a metric. Instead, the goal of most congestion control algorithms is to primarily avoid congestion collapse (something almost everyone agrees is bad), while providing an acceptable level of performance.

3 A simple contention-based protocol: ALOHA

The simplest and earliest example of a contention-based protocol is ALOHA (Additive Links On-line Hawaii Area), which was possibly the first demonstration of a wireless packet-switched network. It dates back to

1971 and was developed at the University of Hawaii. ALOHA was way ahead of its time. The ARPANET, one of the first demonstrations of *wired* packet switching was about two years old at the time ALOHA was demonstrated, and the protocols underlying the Internet (IP and TCP) were yet to be developed! We'll study the MAC mechanism in ALOHA both because it is simple and the ideas continue to permeate WiFi today, some 45 years later. ALOHA also inspired Bob Metcalfe to create Ethernet in 1973, which borrows the idea of collision detection from ALOHA.

In the ALOHA network, several users sent their data over a shared uplink wireless channel to a centralized computer, which would acknowledge receipt of this data using an ACK packet. Because the uplink wireless channel was shared, only one user could transmit at a given instant; transmissions by more than one user would result in a *collision* of their resulting data packets, which would be detected by each of the users by the non-receipt of an ACK.[3] The goal then is to get users to coordinate their transmissions so as to minimize the probability of a collision.

The simplest version of ALOHA is called slotted ALOHA. Here time is derived into discrete time slots, and each packet occupies one time slot.[4] We'll also assume the number of users in the network N is known. Then, at each time slot, a user transmits with a fixed probability p—and hence does not transmit with a fixed probability $1-p$. If you were to guess the right value of p, what would it be?

Let's derive the right value of p. Our goal in picking p is to maximize ALOHA's utilization. In steady state, the utilization of ALOHA is the probability that *some* (i.e., exactly one of the N users) transmits without a collision. This probability is given by:

$$N.p.(1-p)^{N-1} \tag{1}$$

, which captures the fact that one user transmits (p) and the remaining $N-1$ stay silent ($(1-p)^{N-1}$), and there are N distinct possibilities for this one transmitting user. To maximize this probability, we need to set the derivative of the above equation with respect to p to 0, which gives us:

$$\begin{aligned} N((1-p)^{N-1} - p(N-1)(1-p)^{N-2}) &= 0 && (2)\\ (1-p) - p(N-1) &= 0 && (3)\\ 1 - p - Np + p &= 0 && (4)\\ p &= \frac{1}{N} && (5) \end{aligned}$$

So now, what is the probability that some user transmits without a collision on a given time slot assuming we pick the best value of p? This quantity turns out to be:

$$(1-\frac{1}{N})^{N-1} \tag{6}$$

, which tends to $\frac{1}{e}$ (or 37%) as N gets really large (Figure 1 shows the derivation.).

4 Adaptively determining probabilities in ALOHA

The problem with ALOHA as described so far is that the right value of the probability p depends on N, analogous to how the right value of the window size for each sender in the sliding window protocol depends on the number of senders in the network (i.e., $\frac{BandwidthDelayProdoct}{Numberofsenders}$). We use a mechanism similar to AIMD to determine the right value of the probability p. Each user independently starts from some initial probability p_{min} and then increases/decreases its probability depending on whether a packet was successfully ACKed or not (which indicates a collision). On an ACK, the user increases the probability (up to a max probability p_{max}), while on a collision, it decreases the probability. The decrease rule is typically multiplicative for the same reasons as TCP: you want to backoff from a bad situation quickly. In fact, the exponential backoff algorithm in TCP was adopted from ALOHA. The increase rule can be additive such as TCP's congestion avoidance phase or multiplicative such as TCP's slow start phase.

[3] Typically this non-receipt is detected using a timeout similar to how TCP detects lost packets.

[4] You can think of a time slot as the duration of time required for a user to transmit a packet and receive its ACK from the centralized computer.

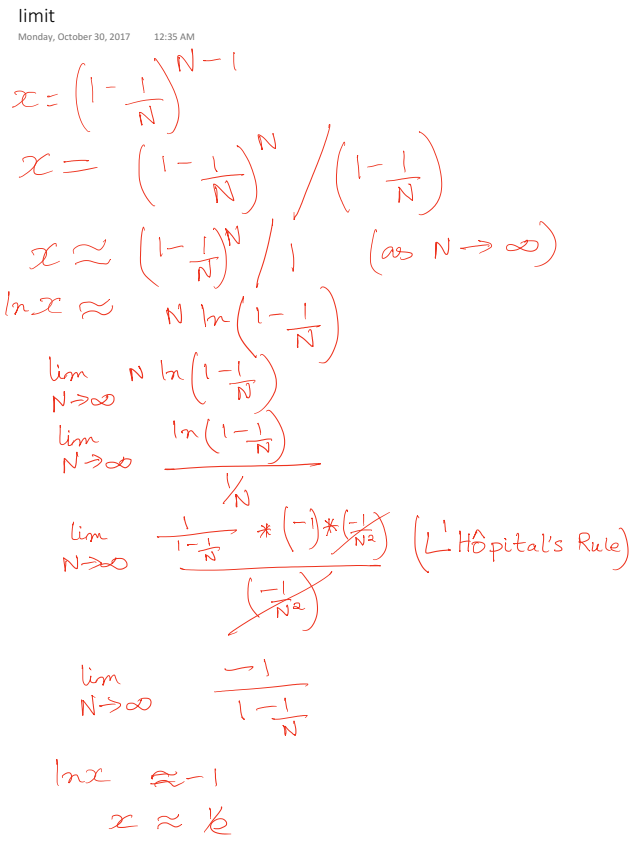

Figure 1: Derivation of transmission probability in ALOHA

5 Limitations with ALOHA and the development of carrier sense

ALOHA operated in a very challenged environment. Each of the users was potentially on a different Hawaiian island, and the centralized computer was on the main University of Hawai campus in Oahu. Because the users could be on different islands, the users could not hear each other and simply cease transmissions when another user had started transmitting. The ability to cease transmissions while another user is in the process of transmitting is called carrier sense and is analogous to day-to-day human conversation, where we ideally "listen before we speak."

Link-layer technologies that followed ALOHA, such as Ethernet and WiFi, adopted many of its ideas such as the ability to handle collisions at the link layer and the use of randomization, but adapted it to slightly less challenging situations like WiFi and Ethernet where carrier sense was an option. We'll talk about carrier-sense-based protocols in the next class.