Lecture 4

Gidon Rosalki

2025-11-9

Notice: If you find any mistakes, please open an issue at `https://github.com/robomarvin1501/notes_networking`

1 Multiple Access Domains

Consider when there is a single shared broadcast channel, with two or more simultaneous transmissions by nodes. There is a collision if a node receives two or more signals at the same time. Multiple Access Protocols (MAPs) are distributed algorithms that determine how nodes share this channel, or in other words, when a node can transmit. The communication about sharing this channel takes place over the channel itself.

1.1 Broadcast domains

1.1.1 Wired

It used to be that all nodes would be effectively connected to a single wire, potentially indirectly through a *hub*. We do not especially use this today, since this had all nodes in the same collision domain, and massively reduced the amount of data able to be transmitted. Today we mainly use a star topology, with an active *switch* in the centre. Each node runs a separate connection to the switch, massively increasing available bandwidth.

1.1.2 Wireless

Wireless networks suffer from the same problems with collisions, with the additional issues of potentially some nodes not receiving the messages, due to signal attenuation over distance, or potential geographic features blocking the signal.

1.2 MAPs

An idea MAP is as follows:

1. When one node wants to transmit, it can send at rate R
2. When M nodes want to transmit, each can send at an average rate $\frac{R}{M}$
3. It is fully decentralised, there is no node that coordinates transmissions
4. There is no clock / slot synchronisation
5. It is as simple as possible

There are also **Random Access Protocols**, where when a node has a packet to send, it transmits at full channel data rate R, with no a priori coordination between the nodes. Two or more transmitting nodes result in collisions. Therefore, the RAC specifies how to detect collisions, and how to recover from them. Some examples include slotted ALOHA, and ALOHA.

1.2.1 Goodput and throughput

Definition 1.1 (Goodput)**.** *The proportion of successful slots (as in, in how many there was a transmission, without a collision)*

Definition 1.2 (Throughput)**.** *The proportion of slots in which there was a transmission (not necessarily successful)*

1.2.2 Slotted ALOHA

We assume that all frames are the same size, and that time is transmitted into equal size slots (each slot is the time to transmit one frame). Nodes start to transmit only at the beginning of the slot, and they are synchronised. If 2 or more nodes transmit in a slot, all nodes detect the collision.
They operate as follows: When a node needs to transmit a frame, it transmits it in the next slot. If there is no collision, great! If there is a collision, then the node retransmits the frame in the next slot with a probability p, repeated until it succeeds.

This comes with the benefits that a single active node can continuously transmit at the full rate of the channel. It is also highly decentralised, since only the slot times need to be in sync across the nodes. It's also beautifully simple. However, there are inevitably collisions, which result in wasted slots. Additionally, there are often many idle slots, and nodes may be able to detect a collision in less than a slot time. Finally, how are we meant to synchronise the clocks?

Suppose that there are N nodes, each with many frames to send, each transmitting in a given slot with probability p. The probability that a given node succeeds in a given slot is

$$p\left(1-p\right)^{N-1}$$

and the probability that *some* node succeeds is approximately

$$Np\left(1-p\right)^{N-1}$$

To find the maximum efficiency, we want to find the p^* that maximises $Np\left(1-p\right)^{N-1}$. For many nodes, taking the limit of $Np\left(1-p\right)^{N-1}$ as $N \to \infty$ gives a maximum efficiency of $\frac{1}{e} \approx 0.37$, or the channel only sends useful transmissions 37% of the time, which is *not great.*

1.2.3 Pure ALOHA

Pure, or unslotted ALOHA, is simpler, with no synchronisation. Each host behaves as it if it running slotted ALOHA, but the slots are local to itself, rather than shared over the network. It sends a frame with a probability of p, at the beginning of a slot. This increase the probability of a collision, each frame that is sent at time t_0 will collide with frames sent in slots that start anywhere from $(t_0 - 1, t_0 + 1)$.

The goodput is given

$$\begin{aligned}
\mathbb{P}\left[\text{success by given node}\right] &= \mathbb{P}\left[\text{node transmits}\right] \cdot \mathbb{P}\left[\text{no other node transmits in interacting times}\right] \\
&= \mathbb{P}\left[\text{node transmits}\right] \\
&\cdot \mathbb{P}\left[\text{no other node transmits in } \left(t_0 - 1, t_0\right)\right] \\
&\cdot \mathbb{P}\left[\text{no other node transmits in } \left(t_0, t_0 + 1\right)\right] \\
&= p \cdot \left(1-p\right)^{N-1} \cdot \left(1-p\right)^{N-1} \\
&= p \cdot \left(1-p\right)^{2(N-1)}
\end{aligned}$$

Choosing the optimal p, and $n \to \infty$ results in even **worse** than slotted ALOHA, with a result of $\frac{1}{2e} \approx 0.18$.

1.3 CSMA

Carrier Sense Multiple Access type protocols *listen* before they transmit. If a channel is sensed as idle, transmit the entire frame, and if it is busy, defer the transmission. This is like hearing someone else is talking, and waiting for them to finish before you speak.

1.3.1 CSMA/CD

Collisions can still occur, the propagation delay means that two nodes may not hear each others transmission. If there is a collision, then the entire packet transmission time is wasted. It should be noted that distance and propagation delay both play a significant role in determining collision probability.

So how do we add collision detection? In a wired LAN we measure the signal strength, and compare the strengths of transmitted vs received signals. This is harder in wireless, since the received signal strength is overwhelmed by local transmission strength. Should we detect a collision, then the colliding transmissions are aborted, reducing the channel wastage.

Adding this in, when a host has a packet to transmit, it checks that the line is quiet before transmitting. If it detects a collision, then it should stop transmitting, wait a random time, and return to the beginning. This together is called **CSMA/CD**.

To ensure that a packet is transmitted without a collision, a host must be able to detect a collision *before* it finishes transmitting a packet. As a result, we can see that for a host to detect a collision before it finishes transmitting a packet, we require the transport time to be

$$\text{Transport time} > 2 \cdot \text{ propagation delay}$$

In other words, there is a minimum length packet for CSMA/CD networks, and that is at least 2 times the propagation delay.

Let us consider the goodput. The time is slotted, and all packets are the same length. A time slot is equal to 2· propagation time. In any given time slot, a host will decide to transmit (or not) with the probability p. So:

$$\eta = \frac{\text{Time taken to send data}}{\text{Time taken to send data + overhead}}$$

Therefore, to find the goodput $\alpha(p)$, we take the probability that exactly one node transmits in a given slot

$$\begin{aligned}\alpha(p) &= \binom{N}{1} p(1-p)^{N-1} \\ \frac{d\alpha}{dp} &= N(1-p)^{N-1} - pN(N-1)(1-p)^{N-2} \\ \implies p = \frac{1}{N} &\implies \alpha_{max} \approx 37\%\end{aligned}$$

We will define A to be the expected number of time slots **wasted** before a packet is successfully transmitted. Consider for example a coin X, such that

$$\mathbb{P}[X = H] = 0.4 \overset{def}{=} a$$

Therefore, the expected number of coin tosses before the first head is $\frac{1}{0.4} = 2.5$, and therefore $A = 1.5$ unsuccessful attempts, which would then be followed by one successful one.
Let us call the minimum length packet TRANSP. So

$$\begin{aligned}\eta_{CSMA/CD} &= \frac{TRANSP}{TRANSP + \mathbb{E}[\text{Number wasted slots per packet}]} \\ &= \frac{TRANSP}{TRANSP + A(2 \cdot PROP)} \\ &= \frac{TRANSP}{TRANSP + (3 \cdot PROP)}\end{aligned}$$

2 Single (Logical) Link in practice

2.1 MAC Addresses

Devices connected to a LAN have a MAC address, whose function is to get frames from one interface, to another physically connected interface on the same network. Most LANs use a 48 bit MAC address, burned into the NIC ROM. Despite this, it is sometimes settable in software. Each adaptor on the LAN has a unique MAC address. Their allocations are administered by IEEE. Each manufacturer buys a portion of the MAC address space (which ensures uniqueness). Consider a MAC address like a personal ID number. This uniqueness ensures that we can move network cards between LANs without issue (not necessarily true for every address type, consider IP which are hierarchical, and thus not portable).

2.2 Ethernet

Ethernet is the dominant wired LAN technology. It has nice cheap NICs, and is the first widely used LAN technology. Over the years, new versions have been released to keep up with speed improvements.

Ethernet is a CSMA/CD algorithm:

1. NIC receives a datagram from the network layer, and creates a frame
2. If the NIC senses that the channel is idle, it starts frame transmission, and if it senses that it is busy, then it waits until the channel is idle, and then transmits.
3. If the NIC transmits an entire frame without detecting another transmission, then it is done with the frame.
4. If the NIC detects another transmission *while* transmitting, then it aborts, and sends the **jam** signal
5. After aborting, the NIC enters **exponential backoff**. After the mth collision, it chooses k at random from $\{0, 1, \ldots, 2^m - 1\}$, and waits $k \cdot 512$ bit times, before returning to step 2