ECE 333: Introduction to Communication Networks
Fall 2001

Lecture 13: Medium Access Control I

- **Introduction**
- **Static Allocations**
- **Aloha**

In Lecture 1, we classified links as either ***broadcast*** or ***point-to-point.*** With a broadcast link, more than two users share the same transmission media. Often an entire network, in particular a LAN, consists of a single broadcast link connecting a group of users[1]. Recall, in this case we refer to the network itself as a ***broadcast network***. A broadcast link is also called a ***multiaccess channel.*** In such a channel,

1. A transmitter can be heard by multiple receivers
2. A receiver can hear multiple transmitters.

The first point implies that a technique is needed to decide which receiver(s) a transmitted packet was meant for. A direct way to accomplish this is through ***addressing***, i.e., each receiver is assigned an address in the form of a unique bit sequence, and this address is then added to the header field, before sending a packet. Regarding the second point, if two transmitters transmit at the same time, their signals may ***interfere*** or ***collide*** and not be recoverable at the receiver. A method is needed to share the broadcast link among the various transmitters and avoid such collisions; this is called the ***medium access control (MAC)*** problem. We will examine various solutions to this problem in the next several lectures. Before discussing these solutions in detail, we first describe several situations where broadcast links arise. We then look at where the medium access control problem is commonly addressed in layered network architectures. Next we categorize the possible solution approaches, and begin discussing some specific approaches.

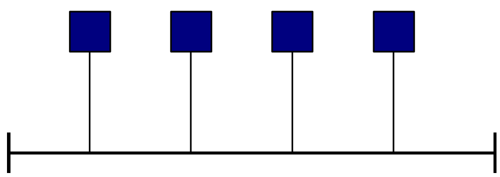
[1] The primary reason LANs have been designed this way is that it is a very cost effective approach for connecting together a group of users in a small geographical area.

Examples of broadcast links:

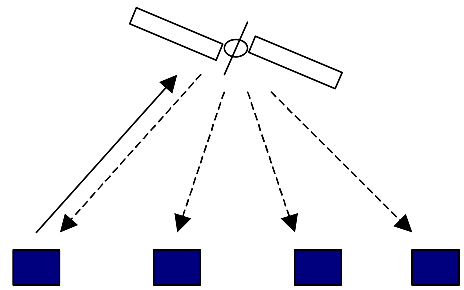
In the following we mention a few representative examples of broadcast links, several other examples will be provided in later lectures.

Shared Bus: A shared bus consists of a single cable (*e.g*. a coaxial cable) to which all users are connected. A signal transmitted by one user will propagate in both directions on the cable and can be received by any other user. Several versions of *Ethernet* use a shared bus, as does the *LocalTalk* LAN developed by *Apple Computer Corp*.

"Bent-pipe" Satellite Link: In a common type of satellite communication, users transmit messages to a satellite in one frequency band and the satellite simply relay's any received signal back down to all users in a second frequency band.



Shared bus



Satellite channel

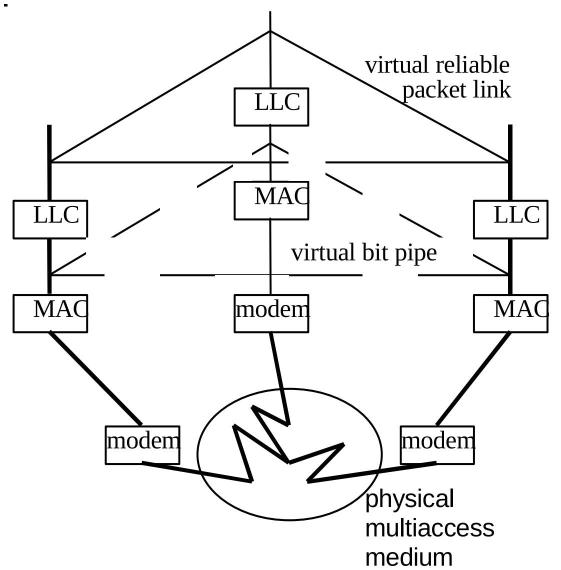
Cable TV Networks: Recall in lecture 4 we described the cable TV distribution network. When this network is used for data communication, upstream communication from users to the head-end is over a shared link, which can be viewed as a broadcast medium.

(Terrestrial) Wireless Networks: In wireless networks users communicate over a shared wireless channel, for example a given frequency band specified by the FCC. By nature a wireless channel is a broadcast medium. One of the most widespread types of wireless networks is a cellular telephone network. Another type of wireless network that is increasingly common is a wireless LAN. A number of different wireless LAN standards have been defined including the *IEEE 802.11* wireless LAN standard and the *Bluetooth* standard.

MAC Sublayer

In the OSI protocol stack, channel allocation is addressed in the ***Medium access control (MAC)*** sublayer. This is a sublayer of the Data Link Layer - considered to be below the Logical Link Control (LLC) sub-layer. Many LAN technologies, such as Ethernet are based on this type of architecture. The MAC layer provides an unreliable connectionless service; if required, the LLC layer can convert this into a reliable service.

Network Layer
Logical Link Control Sublayer
Medium Access Control Sublayer
Physical layer



Channel Allocation

Basic problem: How to allocate a multiaccess channel among competing users.

In other words, we need a set of rules (i.e. a protocol) to allow each user to communicate and avoid interference. There are a variety of solutions to this problem that are used in practice. These solutions can be classified as either ***static*** or ***dynamic***. With a static approach, the channel's capacity is essentially divided into fixed portions; each user is then allocated a portion for all time. If the user has no traffic to use in its portion, then it goes unused. With a dynamic approach the allocation of the channel changes based on the traffic generated by the users. Generally, a static allocation performs better when the traffic is predictable. A dynamic channel allocation tries to get **better utilization and lower delay** on a channel **when the traffic is unpredictable**.

Static Channel Allocation Techniques

Two common static channel allocation techniques are TDMA and FDMA.

Time Division Multiple Access (TDMA) – With TDMA the time axis is divided into time slots of a fixed length. Each user is allocated a fixed set of time slots at which it can transmit. TDMA requires that users be synchronized to a common clock. Typically extra overhead bits are required for synchronization.

Frequency Division Multiple Access (FDMA) – With FDMA the available frequency bandwidth is divided into disjoint frequency bands. A fixed band is allocated to each user. FDMA requires a ***guard band*** between user frequency bands to avoid ***cross-talk.***

Another static allocation technique is ***Code Division Multiple Access (CDMA),*** this technique is used in many wireless networks (you can learn more about CDMA in ECE 380).

To a first approximation, if a channel has a capacity of R bps, and either FDMA or TDMA is used, then each user will get an effective rate of R/N bps, where N is the number of users. (This is an approximation, because we have neglected the overhead required for timing or guard bands as discussed above.)

The performance of static channel allocation depends on:

- The variation in the number of users over time
- The nature of the traffic sent by the user

If the traffic on a shared medium is from a fixed number of sources each transmitting at a fixed rate, static channel allocation can be very efficient.

Voice and Video (in their fixed rate forms) have this property and commonly are placed in a shared channel using a static channel allocation.

The variation in the number of users over time impacts the performance of a static allocation because some method is needed to allocate the slot to users as they come and go.

When the traffic sent by a user is bursty, then, under a static allocation, a user's portion of the channel may be empty when another user could use it. This leads one to think that a dynamic allocation will perform better in such cases. This idea is made precise in the following example (which is related to the last problem from Problem Set 4).

Example:

Consider ***K*** users transmitting over a common shared link with capacity ***C***.

Assume by TDMA, or FDMA each user is statically allocated a portion of the link with capacity ***C/K***.

Packets for each user arrive according to a Poisson process with rate λ and are placed into a buffer until they are transmitted.

Assume that packets for each user have lengths that are exponentially distributed with mean **1/μ,** and thus have an average transmission time of ***K/μC***. In this case each user's packets wait in a separate M/M/1 queue.

The average delay in such a system is

$$T_{TDMA} = \frac{1}{\mu C / K - \lambda}$$

Suppose that instead the packets from all ***K*** users could be placed into a single buffer and served FCFS.

Then the total arriving traffic stream would be a Poisson process with rate ***Kλ***. The average transmission time of a packet would now be **1/μC.**

Thus the average delay in the systems is now

$$T = \frac{1}{\mu C - K\lambda} = \left(\frac{1}{K}\right) \cdot \frac{1}{\mu C / K - \lambda} = (1/K) T_{TDMA}$$

⇒ By serving the packets FCFS, the average delay is reduced by a factor of 1/*K*.

(When used to combine packets from various sessions on a point-to-point link serving packets FCFS in the this way is called *statistical multiplexing*.)

The above suggests that to minimize delay with bursty traffic, the channel should be allocated FCFS to the various users. However there is a problem in doing this in a multiaccess channel. Specifically, the packets of the various users are not placed into a single buffer, but are buffered at each source. The various users are unaware of the arrivals at the other users. The fact that this information is distributed among the users is the main challenge to be overcome by a dynamic channel allocation strategy.

Dynamic Channel Allocation:

Many different dynamic allocation strategies have been developed. They can be broadly classified as:

- **Contention resolution approaches** - users transmit a packet when they have data to send- if multiple users transmit at the same time a **collision** occurs and the packets must be retransmitted according to some rule.
- **Perfectly scheduled approaches -** Users transmit contention free according to a schedule that is formed based on which users have data to send, e.g. polling, reservation.

Various combinations of these approaches also exist. We will look at several specific examples of these approaches in the next few lectures. We begin with a basic contention resolution approach below.

As we look at different approaches keep in mind the following two performance criteria:

1. The delay at low load.
2. The throughput (channel efficiency) at high load.

Multiaccess model

We will consider a (simplified) multiaccess model with the following key assumptions:

1) **User Model**: There are ***N*** users or stations. Frames randomly arrive at each station to be transmitted. The arrivals at each station are independent of each other.

2) **Channel model:** All communication is over a single channel and there is no other means of communicating between users.

3) **Transmission model:** Whenever frames overlap in time the frame is garbled and can't be received. If no other frames overlap with a frame it is successfully received, i.e. the only errors are due to collisions.

4) **Feedback model:** All stations are able to detect collisions (or successes) after sending a full frame. There are several ways collisions can be detected. For example, in a system with a central repeater (e.g., a satellite) a station can listen for the return of its message. (In many cases this feedback may be delayed, but we ignore this for now.)

(We will look at variations of these assumptions later.)

Aloha Protocols

Suppose when a user has data to send, the user simply transmits the frame. In our model, if no other user happened to have a frame to transmit then this frame would be successfully received. What if a collision occurs? The user could simply retransmit, but this would not help, because the other user involved in the collision would also retransmit, resulting in another collision. One way to avoid the problem is to wait a random amount of time before retransmitting.

The above idea is the basis for the ***Aloha*** MAC protocol. The Aloha protocol was developed at the University of Hawaii in the 1970's by Norman Abramson. This protocol has served as the basis for many other contention-based protocols.

(Pure) Aloha:

- Users transmit whenever they have data.
- Senders wait to see if a collision occurred (after whole message has been sent).
- If collision occurs, station waits a **random amount of time** then tries again.

Questions:

- Is this a good protocol (i.e. what is the performance)?
- How should the random time for waiting be chosen?

Aloha Performance

Intuitively, one would expect performance to depend on the system load. As more and more users try to send information, more collisions occur and more of the bandwidth of the channel will be wasted on collisions. A common way to describe the performance of a dynamic allocation scheme is to look at how much **average throughput** is achieved as a function of the **average offered load** on the system.

At low offered load, the probability of a collision will be small; so, most messages get through without a collision. However, the throughput is also low, since there isn't much traffic. In this case, the throughput will be approximately the same as in a static allocation (e.g. TDMA), but the delay will be much smaller, since users can transmit immediately when a frame arrives.

At very high offered load, almost every message will experience a collision, and once again, the throughput will be low. In this case the throughput will be lower than in a static allocation, due to the collision.

What happens in between these points?