

Lesson 5 - The Data Link Layer

The Data Link Layer - Introduction | Link Layer and Local Area Networks | Link Layer Implementation | Error Detection and Correction Techniques | Parity Checks | Multiple Access Links and Protocols

Lesson Outline

The Data Link Layer - Introduction

Link Layer and Local Area Networks

- Reliable Delivery
- Flow Control
- Error Detection
- Error Correction

Link Layer Implementation

Error Detection and Correction Techniques

Error Detection and Correction Scenario

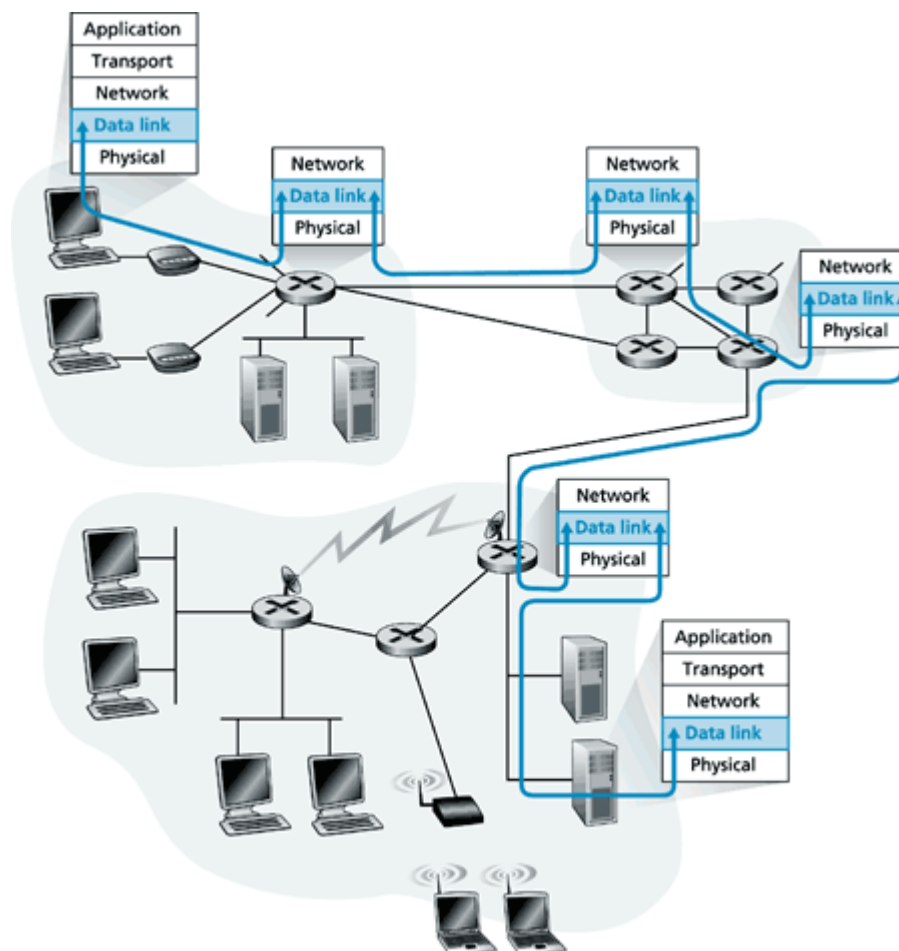
Parity Checks

- One-Bit Even Parity
- Internet Checksum Technique
- Cyclic Redundancy Check (CRC)

Multiple Access Links and Protocols

[GOTO TOP](#)

The Data Link Layer - Introduction



Goals:

Understand principles behind data link layer services:

- error detection, correction
- sharing a broadcast channel: multiple access
- link layer addressing reliable data transfer, flow control

Instantiation and implementation of various link layer technologies

Some Terminology:

Hosts and routers are **nodes** (bridges and switches too)

Communication channels that connect adjacent nodes along communication path are **links**

- wired links

- wireless links
- LANs

2-PDU is a **frame**, encapsulates datagram

The data-link layer has responsibility of transferring datagram from one node to adjacent node over a link

[GOTO TOP](#)

Link Layer and Local Area Networks

In this lesson we examine the data link layer - its services, the principles underlying its operation, and a number of important data link layer protocols. We learn that the basic service of the data link layer is to move a network-layer datagram from one node (host or router) to an adjacent node.

We investigate the different services a link layer protocol can provide in addition to this basic service, including link access services, delivery services, flow control services and transmission services. These differences are due in part to a wide variety of link types over which data link protocols must operate.

We examine error detection and correction, services that are often present in link-layer protocols. We investigate multiple access protocols, commonly used in LANs (local area networks).

Link Layer: Setting the Context

Almost all link-layer protocols encapsulate each network-layer datagram within a link-layer frame before transmission onto the link. A frame consists of a data field, in which the network-layer datagram is inserted, and a number of header fields. (A frame may also include trailer fields; however, we will refer to both header and trailer fields as header fields.)

A data-link protocol specifies the structure of the frame, as well as a channel access protocol that specifies the rules by which a frame is transmitted onto the link. For point-to-point links that have a single sender on one end of the link and a single receiver at the other end of the link, the link-access protocol is simple (or non-existent) -the sender can send a frame whenever the link is idle. The more interesting case is when multiple nodes share a single broadcast link -the so-called multiple access problem.

Things to Remember:

Datagram transferred by different link protocols over different links:
e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link

Each link protocol provides different services
e.g., may or may not provide reliable data transfer over link

Transportation Analogy

Trip from Princeton to Lausanne

Limo: Princeton to JFK

Plane: JFK to Geneva

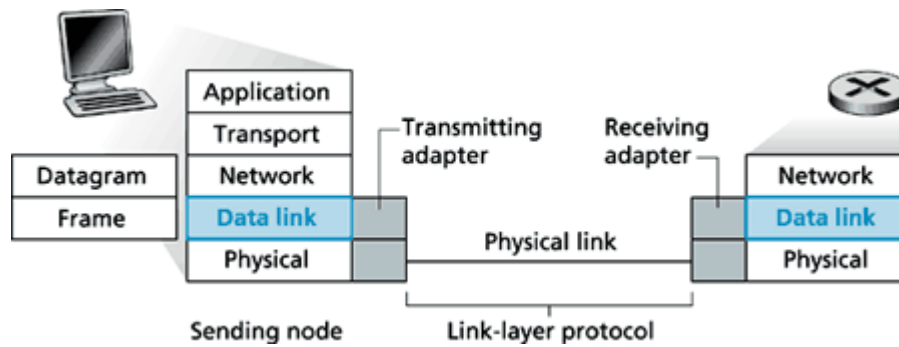
Train: Geneva to Lausanne

Tourist = datagram

Transport segment = communication link

Transportation mode = link layer protocol

Travel agent = routing algorithm



Here, the channel access protocol serves to coordinate the frame transmissions of the many nodes; Frame headers also often include fields for a node's so-called physical address, which is completely distinct from the node's network layer (for example, IP) address.

[GOTO TOP](#)

Link Layer Services

Framing, Link Access

- encapsulate datagram into frame, adding header, trailer channel access if shared medium
- 'physical addresses' used in frame headers to identify source, dest
- different from IP address!

Reliable Delivery

When a link-layer protocol provides reliable-delivery service, it guarantees to move each network-layer datagram across the link without error. Recall that certain transport-layer protocols (such as TCP) also provide a reliable-delivery service. Similar to a transport-layer reliable-delivery service, a link-layer reliable-delivery service is achieved with acknowledgments and retransmissions.

A link-layer reliable-delivery service is often used for links that are prone to high error rates, such as a wireless link. The goal is to correct an error locally, on the link where the error occurs, rather than forcing an end-to-end retransmission of the data by a transport-layer or application-layer protocol. However, link-layer reliable delivery can be considered an unnecessary overhead for low bit-error links, including fibre, coax, and many twisted-pair copper links. For this reason, many of the most popular link-layer protocols do not provide a reliable-delivery service.

Discussion Question: Why do we need both link-level and end-end reliability?

Things to Remember:

- seldom used on low bit error link (fiber, some twisted pair)
- wireless links: high error rates

Flow Control

The nodes on each side of a link have a limited amount of frame buffering capacity. This is a potential problem, as a receiving node may receive frames at a rate faster than it can process the frames over some time interval. Without flow control, the receiver's buffer can overflow and frames can get lost. Similar to the transport layer, a link-layer protocol can provide flow control in order to prevent the sending node on one side of a link from overwhelming the receiving node on the other side of the link.

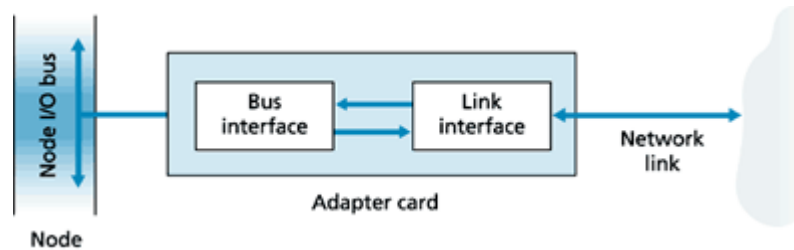
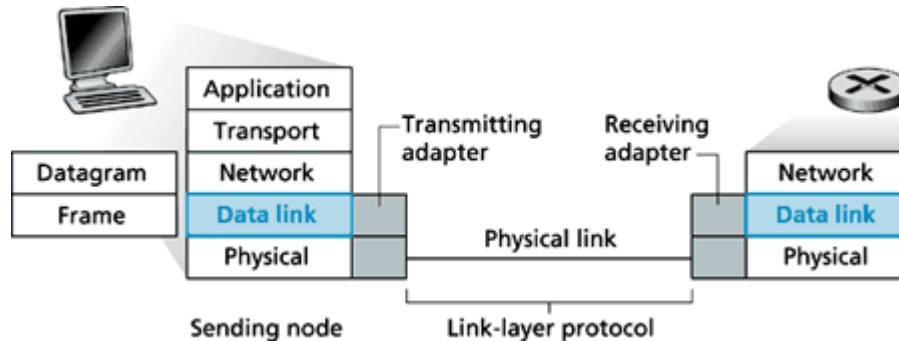
Error Detection

A node's receiver can incorrectly decide that a bit in a frame is zero when it was transmitted as a one, and vice versa. Such bit errors are introduced by signal attenuation and electromagnetic noise. Because there is no need to forward a datagram that has an error, many link-layer protocols provide a mechanism to detect the presence of one or more errors. This is done by having the transmitting node set error-detection bits in the frame, and having the receiving node perform an error check. Error detection is a very common service among link-layer protocols.

Error Correction

Error correction is similar to error detection, except that a receiver cannot only detect whether errors have been introduced in the frame but can also determine exactly where in the frame the errors have occurred (and hence correct these errors).

Link Layer Implementation



For a given communication link, the link-layer protocol is, for the most part, implemented in an adapter. An adapter is a board (or a PCMCIA card) that typically contains RAM, DSP chips, a host bus interface, and a link interface. Adapters are also commonly known as network interface cards or NICs.

The network layer in the transmitting node (that is, a host or router) passes a network-layer datagram to the adapter that handles the sending side of the communication link. The adapter encapsulates the datagram in a frame and then transmits the frame into the communication link. At the other side, the receiving adapter receives the entire frame, extracts the network-layer datagram, and passes it to the network layer.

If the link-layer protocol provides:

- error detection, then it is the sending adapter that sets the error detection bits and it is the receiving adapter that performs error checking;
- reliable delivery, then the mechanisms for reliable delivery (for example, sequence numbers, timers, and acknowledgments) are entirely implemented in the adapters;
- random access, then the random access protocol is entirely implemented in the adapters.

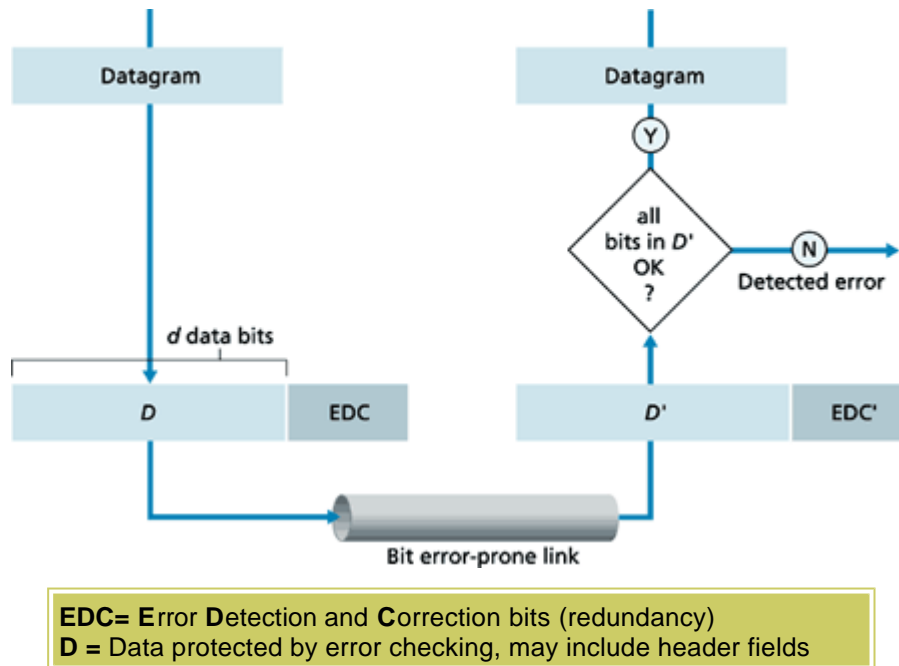
Things to Remember:

Implemented in 'adapter'

- e.g., PCMCIA card, Ethernet Card
- typically includes: RAM, DSP chips, host bus interface and link interface

Error Detection and Correction Techniques

Error Detection and Correction Scenario



The receiver's challenge is to determine whether or not D' is the same as the original D , given that it has only received D' and EDC' .

Error-detection and correction techniques allow the receiver to sometimes, but not always, detect that bit errors have occurred. Even with the use of error-detection bits there will still be a possibility that undetected bit errors will occur, that is, that the receiver will be unaware that the received information contains bit errors.

As a consequence, the receiver might deliver a corrupted datagram to the network layer, or be unaware that the contents of some other field in the frame's header have been corrupted. We thus want to choose an error-detection scheme so that the probability of such occurrences is small.

Things to Remember:

- Error detection is not 100% reliable!
- Protocol may miss some errors, but rarely
- Larger EDC field yields better detection and correction

Generally, more sophisticated error-detection and correction techniques (that is, those that have a smaller probability of allowing undetected bit errors) incur a larger overhead - more computation is needed to compute and transmit a larger number of error-detection and correction bits.

Note: Error detection and correction services are also often offered at the transport layer as well.

Parity Checks

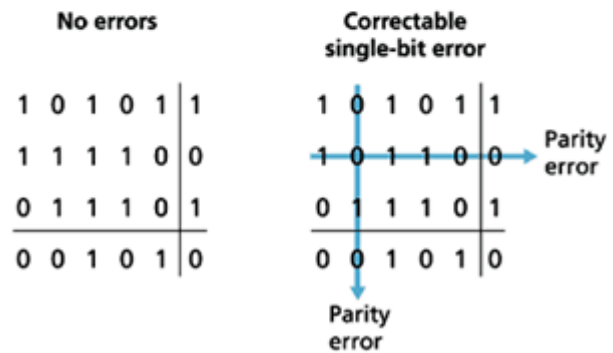
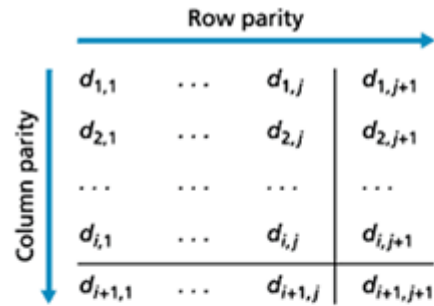
Single Bit Parity:

Detects single bit errors



Two Dimensional Bit Parity:

Detects and corrects single bit errors



Perhaps the simplest form of error detection is the use of a single parity bit. Suppose that the information to be sent, D has d bits. In an even parity scheme, the sender simply includes one additional bit and chooses its value such that the total number of 1s in the $d + 1$ bits (the original information plus a parity bit) is even. For odd parity schemes, the parity bit value is chosen such that there are an odd number of 1s.

One-Bit Even Parity

Receiver operation is also simple with a single parity bit. The receiver need only count the number of 1s in the received $d + 1$ bits. If an odd number of 1-valued bits are found with an even parity scheme, the receiver knows that at least one bit error has occurred. More precisely, it knows that some odd number of bit errors have occurred.

However, measurements have shown that rather than occurring independently, errors are often clustered together in 'bursts'. Under burst error conditions, the probability of undetected errors in a frame protected by single-bit-parity can approach 50 percent. Clearly, a more robust error-detection scheme is needed.

Internet Checksum Technique

Goal: detect "errors" (e.g., flipped bits) in transmitted segment (note: used at transport layer only)

In checksumming techniques, the d bits of data are treated as a sequence of k -bit integers. One simple checksumming method is to simply sum these k -bit integers and use the resulting sum as the error detection bits. The so-called Internet checksum is based on this approach -bytes of data are treated as 16-bit integers and their ones-complement sum forms the Internet checksum.

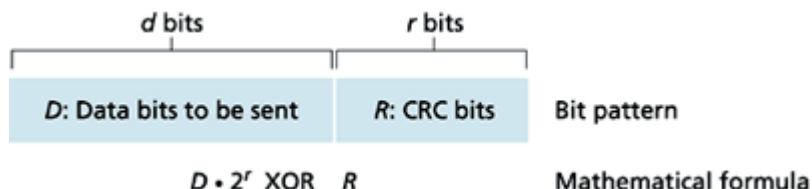
The receiver calculates the checksum over the received data and checks whether it matches the checksum carried in the received packet. RFC 1071 discusses the Internet checksum algorithm and its implementation in detail. In the TCP/IP protocols, the Internet checksum is computed over all fields (header and data fields included). In other protocols, for example, XTP, one checksum is computed over the header, with another checksum computed over the entire packet.

| Sender: | Receiver: |
|--|---|
| <ul style="list-style-type: none"> • Treat segment contents as sequence of 16-bit integers • Checksum: addition (1's complement sum) of segment contents | <ul style="list-style-type: none"> • Compute checksum of received segment • Check if computed checksum equals checksum field value: |

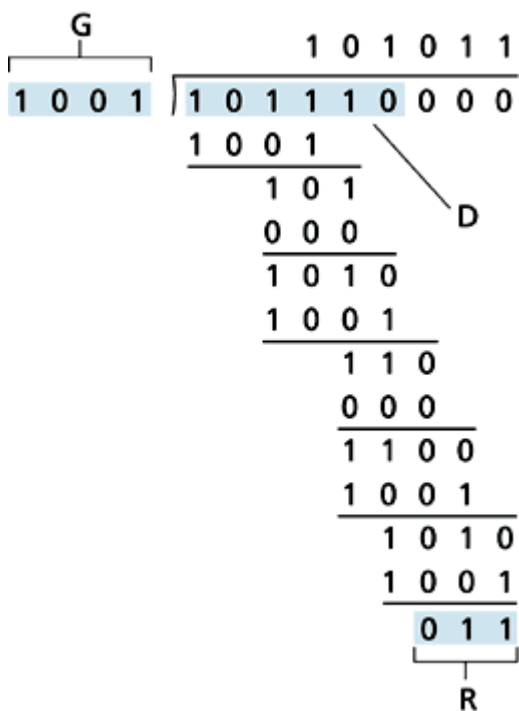
- | | |
|--|---|
| <ul style="list-style-type: none"> • Sender puts checksum value into UDP checksum field | <ul style="list-style-type: none"> ○ NO - error detected ○ YES - no error detected. But maybe errors nonetheless? More later |
|--|---|

Cyclic Redundancy Check (CRC)

An error-detection technique used widely in today's computer networks is based on cyclic redundancy check (CRC) codes. CRC codes are also known as polynomial codes, since it is possible to view the bit string to be sent as a polynomial whose coefficients are the 0 and 1 values in the bit string, with operations on the bit string interpreted as polynomial arithmetic.



CRC codes operate as follows. Consider the d-bit piece of data, D, that the sending node wants to send to the receiving node. The sender and receiver must first agree on an r + 1 bit pattern, known as a generator, that we will denote as G. We will require that the most significant (leftmost) bit of G be a 1. The key idea behind CRC codes is shown in the above diagram. For a given piece of data, D, the sender will choose r additional bits, R, and append them to D such that the resulting d + r bit pattern (interpreted as a binary number) is exactly divisible by G using modulo 2 arithmetic. The process of error checking with CRCs is thus simple: The receiver divides the d + r received bits by G. If the remainder is non-zero, the receiver knows that an error has occurred; otherwise the data is accepted as being correct.



Want:
 $D \cdot 2^r \text{ XOR } R = nG$
equivalently:
 $D \cdot 2^r = nG \text{ XOR } R$
equivalently:
 if we divide $D \cdot 2^r$ by G, want remainder R

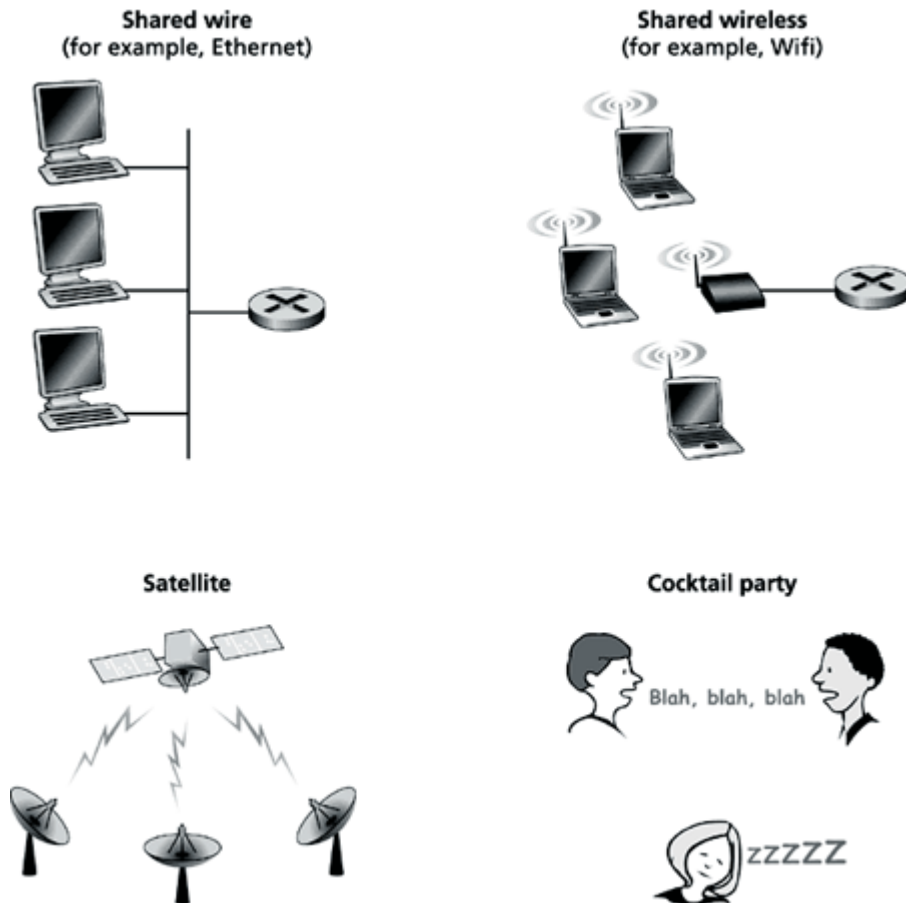
$R = \text{remainder} [D \cdot 2^r / G]$

Things to Remember:

- View data bits, D, as a binary number
- Choose r+1 bit pattern (generator), G
- Goal: choose r CRC bits, R, such that
 - $\langle D, R \rangle$ exactly divisible by G (modulo 2)
 - receiver knows G, divides $\langle D, R \rangle$ by G. If non-zero remainder: error detected!
 - can detect all burst errors less than r+1 bits

Widely used in practice (ATM, HDCL)

Multiple Access links and Protocols



There are two types of network links -point-to-point links, and broadcast links. A point-to-point link consists of a single sender on one end of the link, and a single receiver at the other end of the link. Many link-layer protocols have been designed for point-to-point links;PPP (the point-to-point protocol) and HDLC are two such protocols.

The second type of link, a broadcast link, can have multiple sending and receiving nodes all connected to the same, single, shared broadcast channel. The term broadcast is used here because when anyone node transmits a frame, the channel broadcasts the frame and each of the other nodes receives a copy.

Ethernet is probably the most widely deployed broadcast link technology. In this section we will take a step back from specific link-layer protocols and first examine a problem of central importance to the data-link layer: how to coordinate the access of multiple sending and receiving nodes to a shared broadcast channel- the so-called multiple access problem. Broadcast channels are often used in local area networks (LANs), networks that are geographically concentrated in a single building (or on a corporate or university campus). Thus, we will also look at how multiple access channels are used in LANs.

Computer networks similarly have protocols -so-called multiple access protocols -t which nodes regulate their transmission onto the shared broadcast channel. Multiple access protocols are needed in a wide variety of network settings, including both wire and wireless local area networks, and satellite networks. In practice, hundreds or even thousands of nodes can directly communicate over a broadcast channel.

Multiple Access Protocol

- Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- Communication about channel sharing must use channel itself!
- What to look for in multiple access protocols:
 - synchronous or asynchronous
 - information needed about other stations

Things to Remember:

Three types of "links":

- Point-to-Point (single wire, e.g. PPP, SLIP)
- Broadcast (shared wire or medium;e.g. Ethernet, Wavelan, etc.
- Switched (e.g. switched ethernet, ATM, etc)

Things to Remember:

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
- only one node can send successfully at a time

- robustness (e.g., to channel errors)
- performance

MAC protocols: a taxonomy

Three broad classes:

- Channel Partitioning
 - divide channel into smaller “pieces” (time slots, frequency, code)
 - allocate piece to node for exclusive use
- Random Access
 - channel not divided, allow collisions
 - “recover” from collisions
- “Taking turns”
 - tightly coordinate shared access to avoid collisions

We are all familiar with the notion of broadcasting, as television has been using it since its invention. But traditional television is a one-way broadcast (that is, one fixed node transmitting to many receiving nodes), while nodes on a computer network broadcast channel can both send and receive. Perhaps a more apt human analogy for a broadcast channel is a cocktail party, where many people gather together in a large room (the a providing the broadcast medium) to talk and listen. A second good analogy is somethin many readers will be familiar with -a classroom, where teacher(s) and student(s) similarly share the same, single, broadcast medium. A central problem in both scenarios is that (determining who gets to talk (that is, transmit into the channel), and when.

As humans, we have evolved an elaborate set of protocols for sharing the broadcast channel:

- "Give everyone a chance to speak."
- "Don't speak until you are spoken to."
- "Don't monopolize the conversation."
- "Raise your hand if you have a question."
- "Don't interrupt when someone is speaking."
- "Don't fall asleep when someone else is talking."

Read Chapter 5 of **Computer Networking: A Top Down Approach Featuring the Internet, 2nd edition.** James Kurose, Keith Ross Addison-Wesley, July 2002.

Recommended Links

Charles Spurgeon's Ethernet Web Site <http://www.ethermanage.com/ethernet>

This site provides extensive information about Ethernet (IEEE 802.3) LAN technology. This includes the original 10 Mbps system, 100 Mbps Fast Ethernet (802.3u), 1000 Mbps Gigabit Ethernet (802.3z/802.3ab), and 10 Gigabit Ethernet (802.3ae).

Godred Fairhurst's Pages on LANs <http://www.erg.abdn.ac.uk/users/gorry/eg3561/lan-pages/enet.html>

Godred Fairhurst of the University of Aberdeen maintains a comprehensive set of Web pages on Ethernet, CSMA/CD, bridges, ARP and many other networking topics.

802.11 Planet.com <http://www.80211-planet.com>

802.11 Planet.com is a Web site devoted to products based on the 802.11 wireless networking protocol. With daily news, features, reviews, and tutorials, this site covers all areas of the rapidly changing 802.11 universe.

Working Group for Wireless LAN Standards <http://grouper.ieee.org/groups/802/11>

This is the Web page for the working group that creates the wireless LAN standards, including the standards for IEEE 802.11

Bluetooth Sites <http://www.bluetooth.com>, <http://www.bluetooth.org>

Visit these sites to learn about what's currently happening in the Bluetooth arena.

Tutorials by Nitin Vaidya <http://www.crhc.uiuc.edu/~nhv/presentations.html>

This page provides access to several nice tutorials on wireless LANs and ad hoc networking.