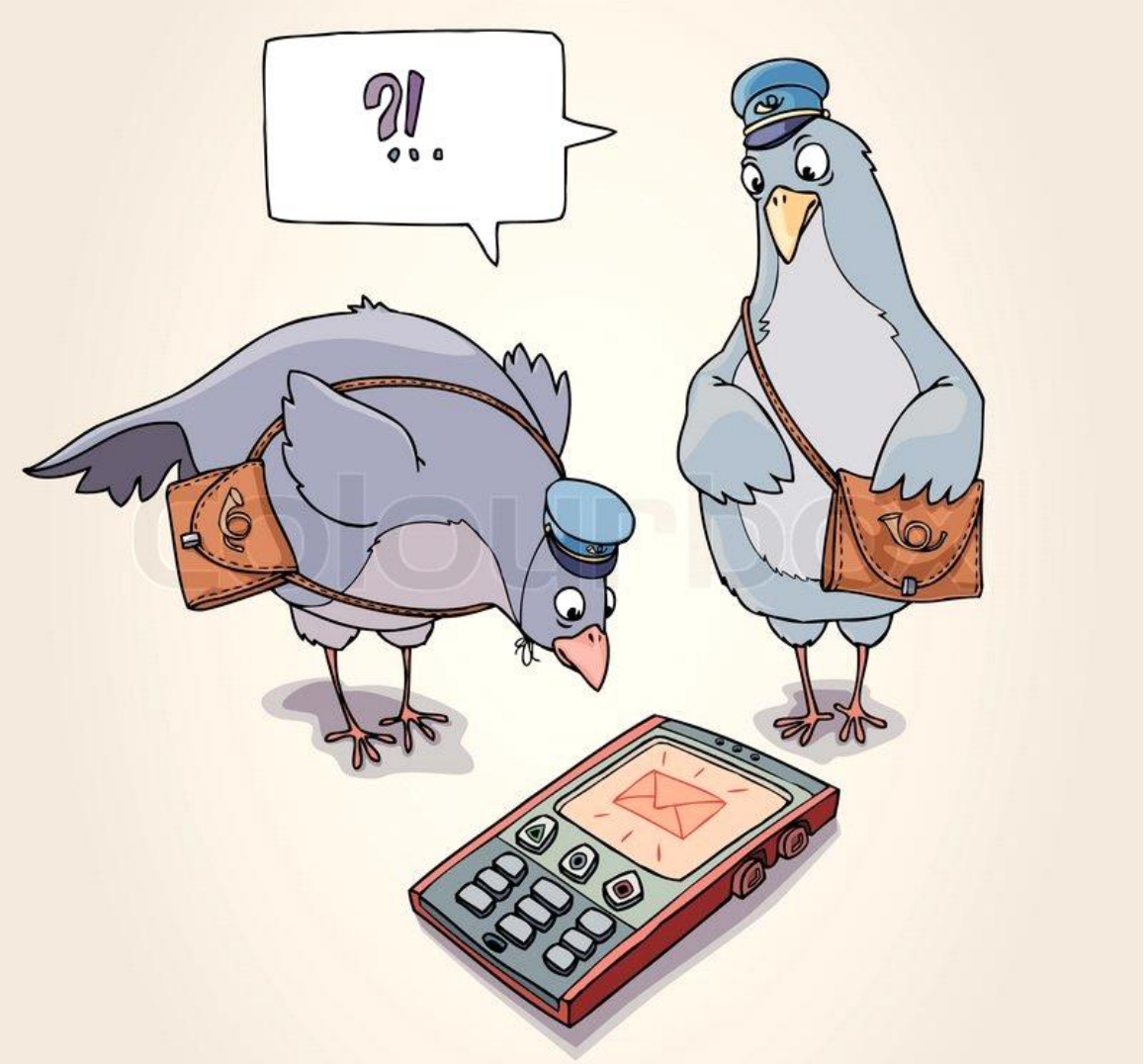
Data Communication Networks - Lecture notes - ECE 487 - 2

Data Communication Networks (University of Alberta)

ECE 487 : Data Communications Networks

Friday, May 29, 2015 8:58 AM



Stefan Martynkiw

Table of Contents

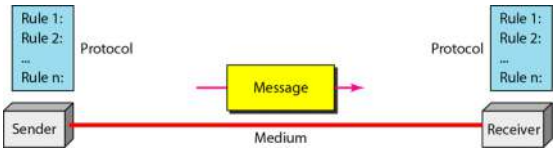
Friday, May 29, 2015 8:58 AM

1.0 Introduction to Data Communications

Monday, March 30, 2015 4:00 PM

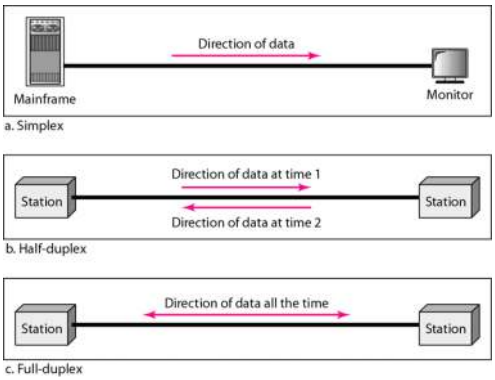
- ***Telecommunication*** = Communication at a distance
- ***Data =*** Information presented in whatever form agreed upon by parties creating and using the data
- ***Data Communications =*** Exchange of data between two devices vis some form of transmission medium.

Components of Data Communication



1. ***Message.*** The information to be communicated
2. ***Sender.*** Devices that sends data messages.
3. ***Receiver.*** Device that receives the data message.
4. ***Transmission Medium.*** Physical path by which a message travels from the sender to receiver.
5. ***Protocol.*** Set of rules that govern data communications. Represents agreement between the communicating devices.

Dataflow



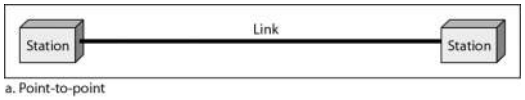
- ***Simplex.*** Communication is unidirectional. Only one of the two devices can transmit; the other must receive.
- ***Half-duplex.*** Each station can transmit or receive, but not at the same time.
  - Like a one-lane road with traffic allowed in both directions.
- ***Full-duplex.*** Both stations can transmit and receive simultaneously.

Networks

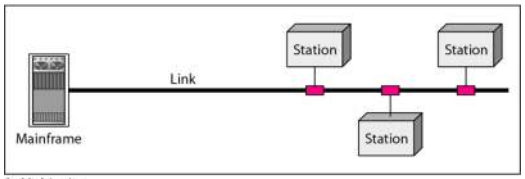
A ***network*** is a set of devices (***nodes)*** connected by communication ***links***.

Types of connections

There are ***point-to-point*** and ***multipoint.***



a. Point-to-point



b. Multipoint

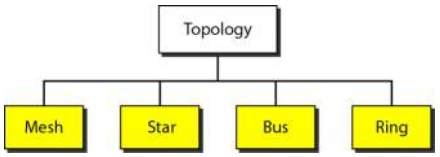
- ***Point-to-point:*** Dedicated link between the two devices. The entire capacity of the link is reserved for the only two devices on it.
- ***Multipoint:*** (multidrop). More than two specific devices share a single link.
    - The capacity of the channel is *shared:*
        - **Spatially Shared:** Several devides can use the link simultaneously.
        - **Time-shared:** Users must take turns

Topology

Physical topology: Which way the network is laid out physically. Two or more devices connect to a link. Two or more links form a topology.

The topology of a network is the geometric representation of the relationship of all links and nodes to one another.

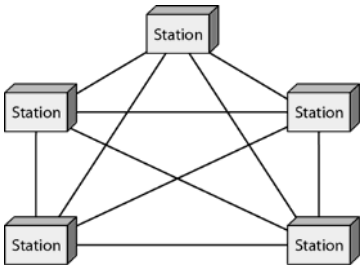
Four main types of topology:



Mesh Topology

Dedicated point-to-point link for every pair.

Need n(n-1)/2 full-duplex links.



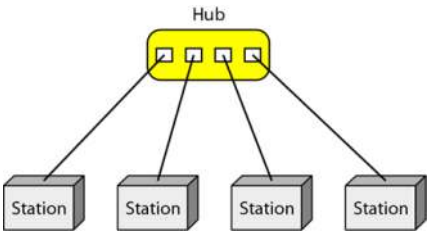
Pros:	• Cons:
- Robust - Good privacy - Easy to identify faults	- Lots of hardware

- Examples:
    - Backbone connecting

Star Topology

Each device has dedicated point-to-point link

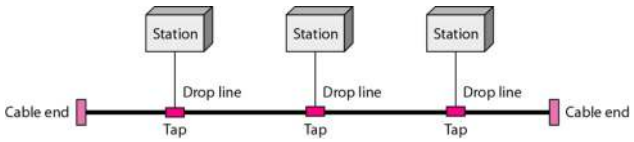
If a hub is used, no privacy, since hubs just dumbly repeat signals to all ports. If a switch/router is used, it only sends the received signal to the target port.



Pros: Cheaper, privacy.
Cons: Single point of failure.
Example: LAN

Bus Topology

Dropline: Connection between device and main cable.
Tap: Connector that splices into the main cable.



As signals travel further along the backbone, they get weaker ==> Need a limit on the number of taps.

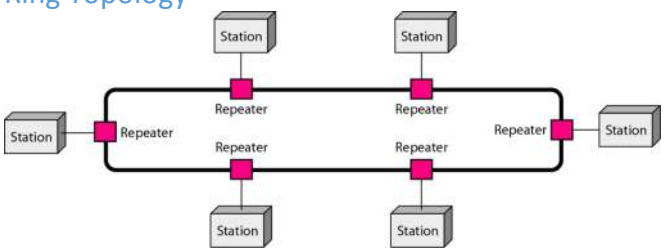
Pros: Easy to install
Cons: Poor fault isolation. Signal reflection causes quality degradation -> Hard to add new nodes.

(3) We need to connect eight stations. How many links are needed in a mesh topology? How many links are needed in a star topology? How many drop lines are needed in a bus topology? How many links are needed in a ring topology (without counting the links that connect stations to repeaters)? Here a link means a wired connection that connects two points. **(4 points)**

28; 8; 8; 8.

Show work here.

Ring Topology

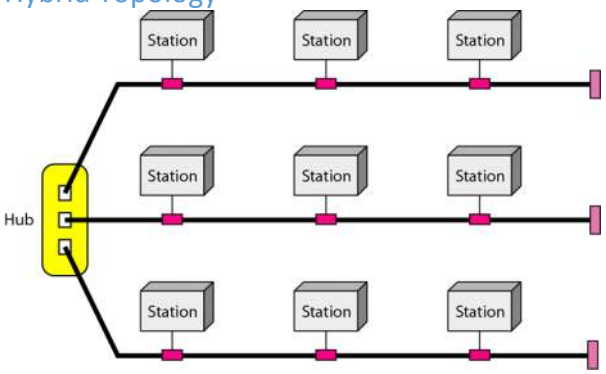


Each device is linked only to its physical neighbours. Only constraints are media/traffic considerations: maximum ring length and number of devices.

Fault isolation is simplified. If one device does not receive a signal within a specified period, it can signal an alarm.

In a unidirectional ring, one break (such as a disabled station) can disable the entire network. We can solve this by using a dual ring.

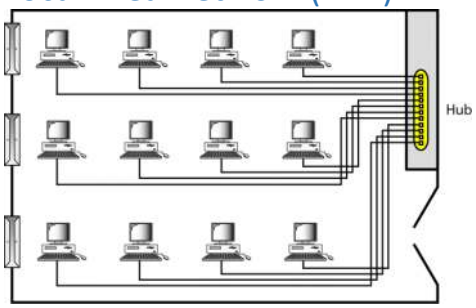
Hybrid Topology



1.2 Categories of networks

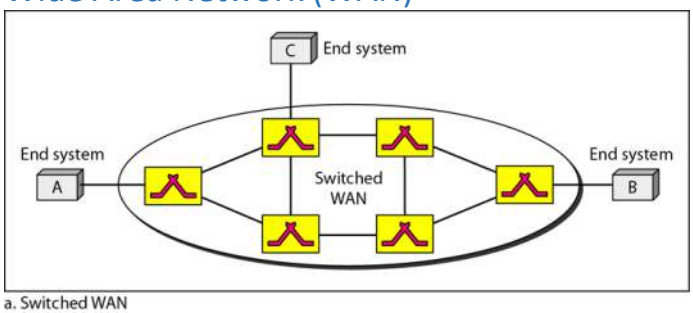
Tuesday, March 31, 2015 9:20 AM

Local Area Network (LAN)

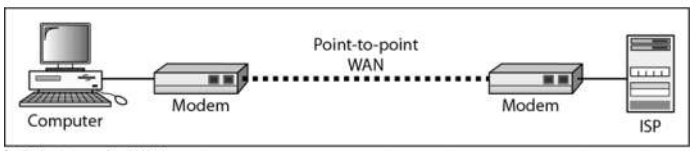


- For resource sharing in a local area (office, campus, etc)
- Max range is a few kilometers.
- One LAN will only use one time of transmission medium.
- Speed:
  - Early LAN: 4Mbps. Later, >1Gbps
- Common topologies:
  - Bus, Ring, Star.

Wide Area Network (WAN)



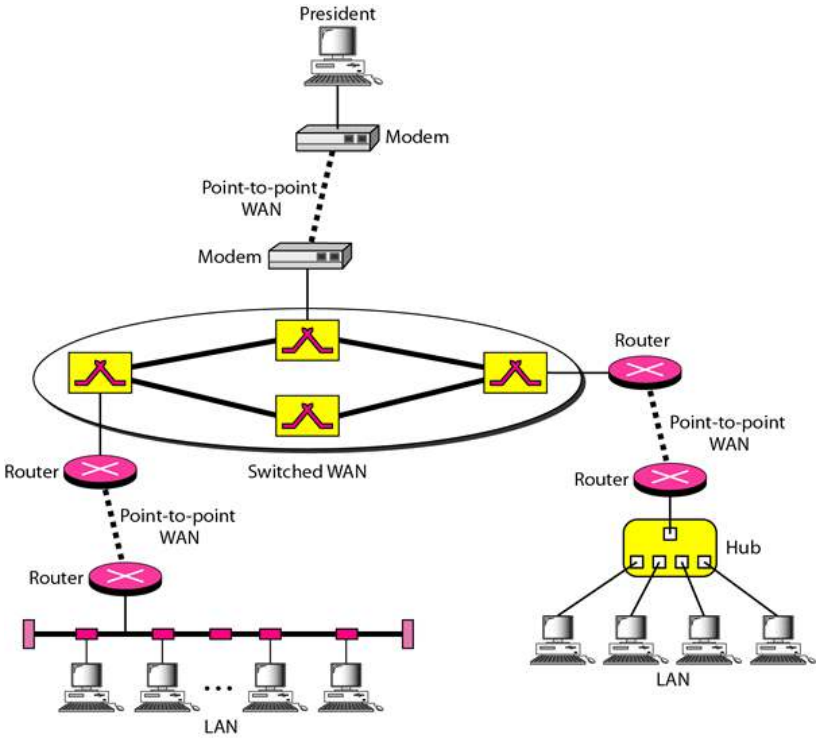
a. Switched WAN



b. Point-to-point WAN

- Long distance transmission of information: a country, continent, or world.
- Complexity:
  - As complex as backbones that connect the internet (**switched WAN)**
  - As simple as a dial-up line that connects a home to the internet. (**point-to-point WAN)**
- ***Switched WAN.***
  - Connects the end systems, which usually have a router that connects another LAN or WAN.
- ***Point-to-point WAN>***
  - Typically a line leased from a telephone or cable TV provider that connects a home computer or small LAN to an ISP.

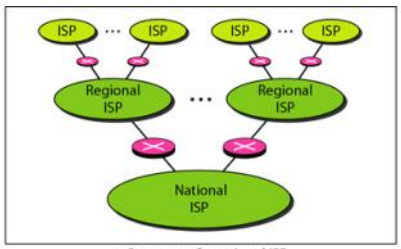
Interconnection of Networks



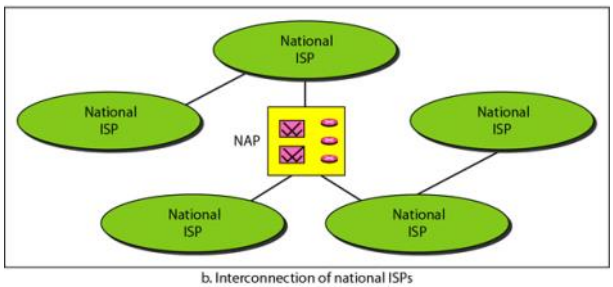
1.3 The Internet

Tuesday, March 31, 2015 9:27 AM

The Internet is made up of many WAN and LANs joined by connecting devices and switching stations.



a. Structure of a national ISP



b. Interconnection of national ISPs

NAP: Network Access Point

1.4 Protocols and Standards

Tuesday, March 31, 2015 9:30 AM

Protocol: Set of rules that govern data communications
Standard: Protocol widely recognized.

Page 56 of the PDF for the textbook.

1.5 Circuit Switching vs. Packet Switching

Tuesday, March 31, 2015 9:32 AM

Circuit Switching

- A circuit is set up between two terminals for the duration of the conversation.
- Resources (bandwidth, time slot, buffer, etc.) reserved among the circuit, used by the two terminals exlusively.
- Circuit is released when the call terminates.
- **Example:** Telephone Network

Packet Switching

- Packets can follow different paths to the destination. Resources in the network can be shared by many connections.
- **Example:** Internet.

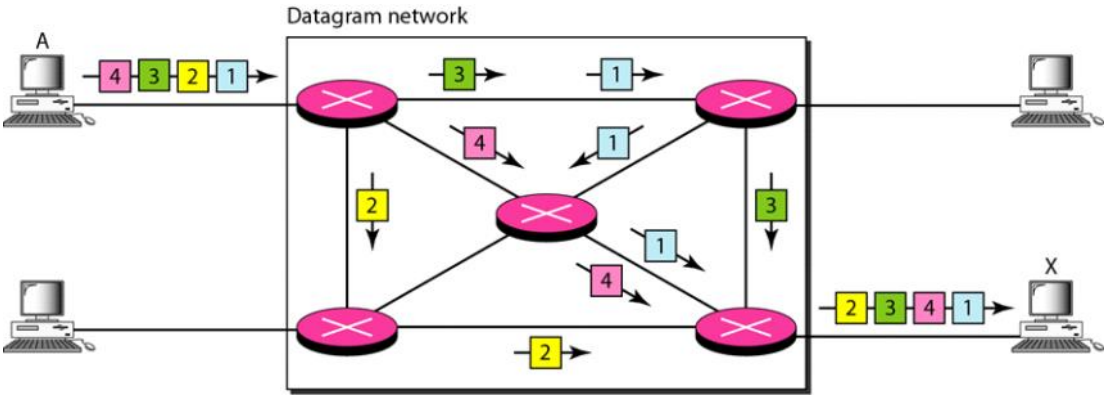
	Circuit switching	Packet switching
Dedicated path	Yes	No
Bandwidth available	Fixed	Dynamic
Potential waste	Yes	No
Store and forward	No	Yes
Call setup	Yes	No need
congestion	At setup time	On every packet
Charge	Per minute	Per packet
Same route/all packets	Yes	No

Telephone Network

- Designed for voice communication (likely with constant rate)
- Less suited to data (eg. Email, ftp, web browsing, etc.) and other non-conversational transmissions.
- Non-voice transmissions tend to be **bursty**: data comes in spurts with idle gaps in between.

Internet

- Suited to data transmission.
  - Provides best-effort services.
  - No guarantee on the delays or loss rate of transmissions.
- Unreliable, offers no security.
- **Self-organizing:** Packets find their way to their destination, even if a link or router breaks down.



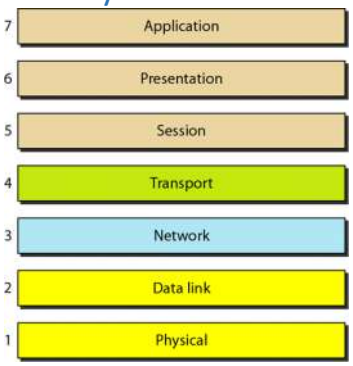
Lecture 2: Network Models

Tuesday, March 31, 2015 9:39 AM

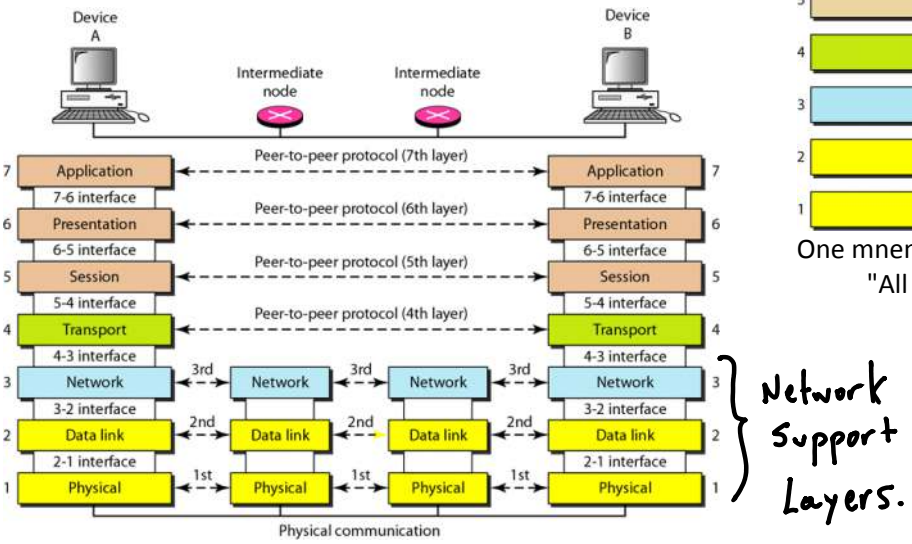
2.2 The OSI Model

Tuesday, March 31, 2015 9:42 AM

The Layers of the OSI Model



One mnemonic to remember the order is:
"All Peters Suck Tits, Not Dicks, Paul."



At the physical layer, communication is direct. At higher layers, communication must move down through the layers on Device A, then back up the layers on Device B. Each later adds its own information to the layer above it, and passes the whole thing along.

Organization of the Layers

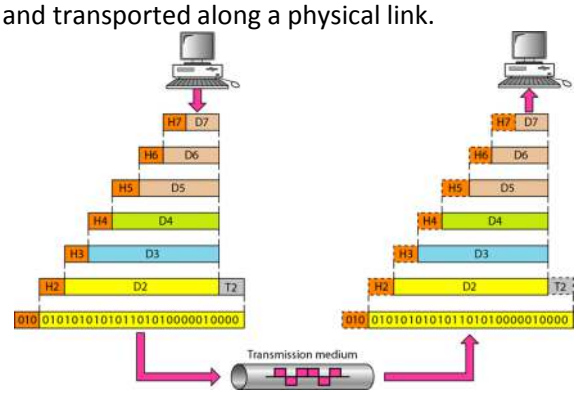
- **Network Support Layers:** Layers 1, 2, 3.
  - Deal with physical aspects of moving data from one device to another.
    - **Ex:** electrical specifications, physical connections, physical addressing, transport timing and reliability.
- **User Support Layers:** Layers 5, 6, 7
  - Allow interopability among unrelated software systems.
- **Layer 4: The Transport Layer**
  - Links the two subgroups and ensures that what the lower layers have transmitted is in a form that upper layers can use.

The upper layers are typically implemented in software. Lower layers are a combination of hardware and software. Physical layer is mostly hardware.

An Exchange using the OSI Model

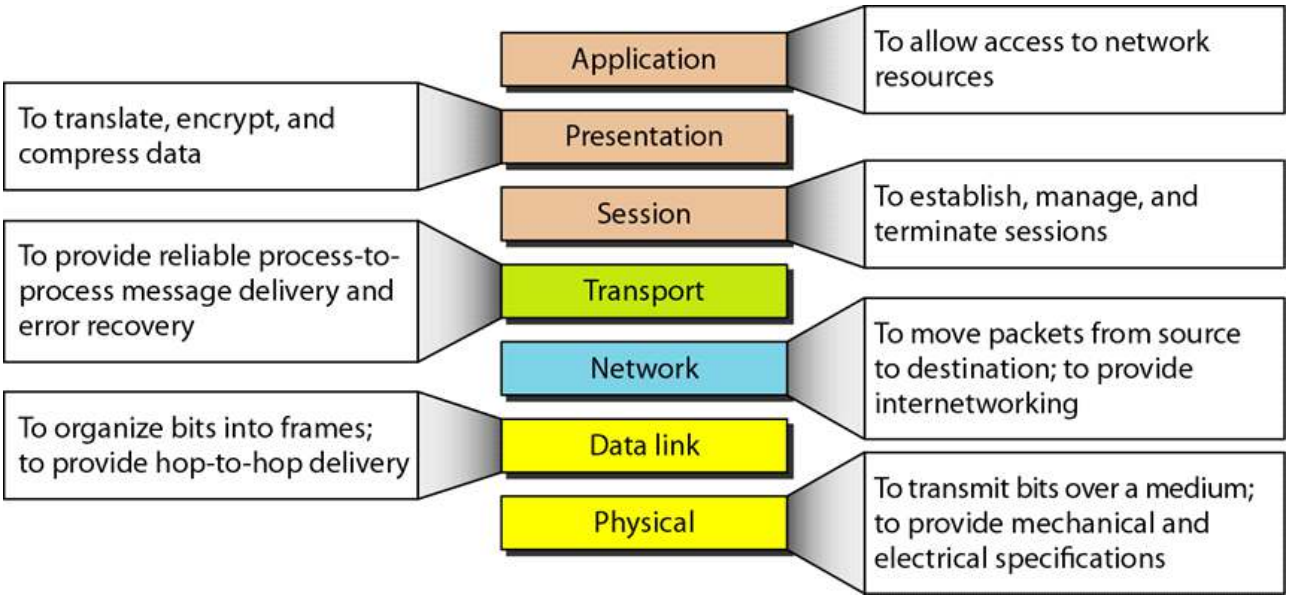
At each layer a **header,** or possibly a **trailer** can be added to the data unit.
Typically the trailer is only added at layer 2.

When the formatted data unit goes through the physical layer, it is changed into an electromagnetic signal and transported along a physical link.



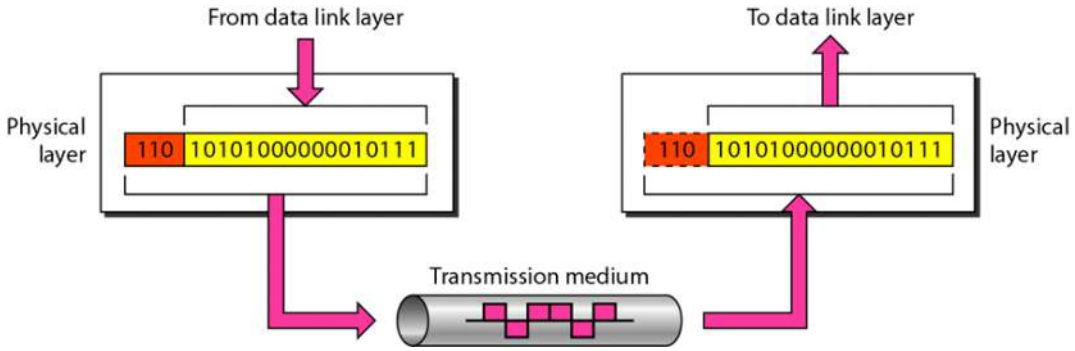
2.3 Layers in the OSI Model

Tuesday, March 31, 2015 10:01 AM



2.3.1 Physical Layer

Tuesday, March 31, 2015 10:03 AM



Coordinates the functions required to carry a bit stream over a physical medium.

- Mechanical and electrical specifications of the interface and transmission medium.
- Defines procedures and functions for physical devices and interfaces

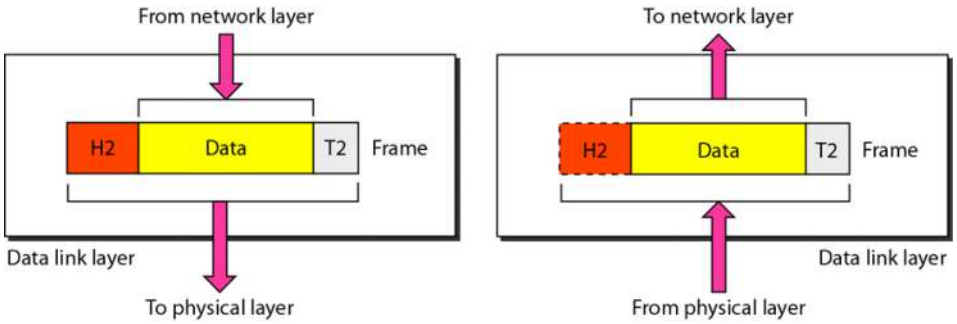
Moves individual bits from one node to the next.

Other Concerns:

- Physical characteristics of interfaces and medium.
- Representation of bits.
  - PHY Layer Data consists of a stream of bits (0's or 1's) with no interpretation.
  - To be transmitted, buts must be encoded into signals -- electrical or optical.
    - PHY defines type of encoding (how 0's/1's are changed to signals)
- Data rate.
  - Transmission rate: Number of bits sent per second.
    - Implies PHY defines the duration of a bit.
- Synchronization of bits.
  - The sender and receiver clocks must be synchronized.
- Line Configuration.
  - The Physical Layer is concerned with the connection of devices to the media. (Point-to-point vs multipoint.)
- Physical topology.
- Transmission mode.
  - Simplex, Half-duplex, full-duplex.

2.3.2 Data Link Layer

Tuesday, March 31, 2015 10:12 AM

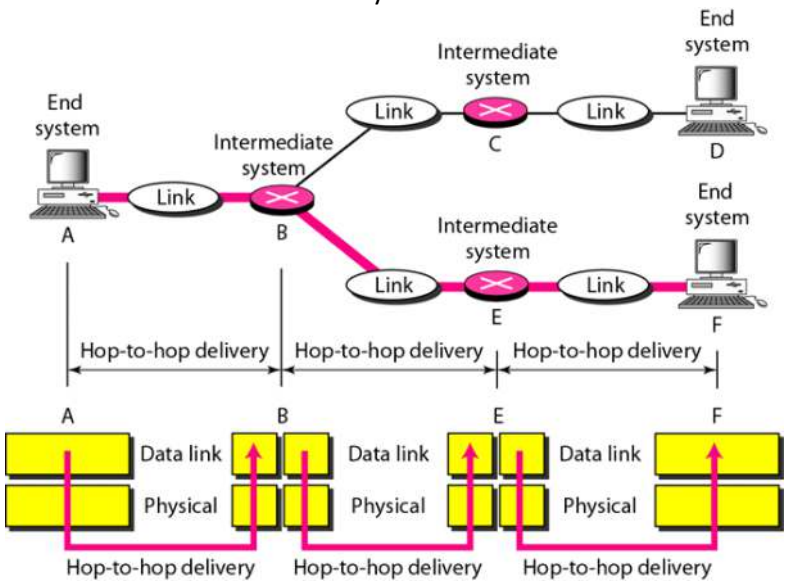


Transforms the PHY layer into a reliable link. Makes PHY appear error-free to the upper layer.

Moves frames from one node to the next.

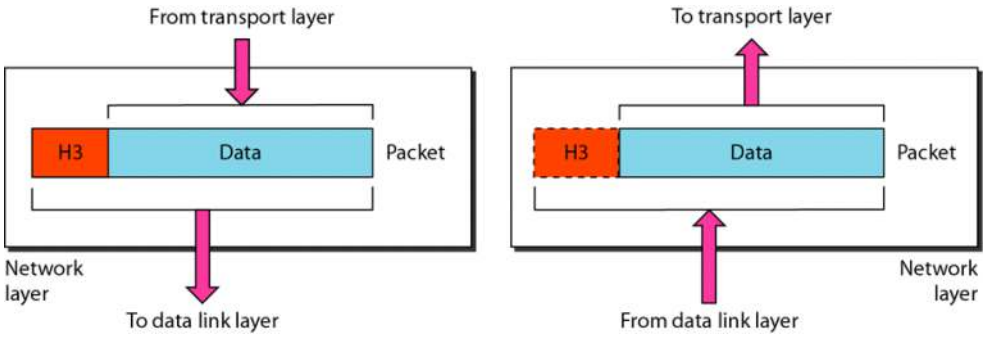
Other responsibilities:

- Framing.
  - Divides the stream of bits received from network layer into manageable data units called frames.
- Physical addressing.
  - If frame going to somewhere in same network:
    - Adds a header to the frame to define sender/receiver of the frame.
  - If frame intended for outside the current network,
    - Receiver address is the one that connects this network to the next one.
- Flow control.
  - Try to avoid overwhelming the receiver
    - If rate(receiver data absorption) < rate(sender data produced)
- Error Control.
  - Add mechanisms to detect and retransmit damaged or lost frames.
  - Recognizes duplicate frames.
  - Typically achieved through trailer added to end of frame.
- Access Control
  - When two or more devices connected to the same link, data link layer determines which device has control over the link at any time.



2.3.3 Network Layer

Tuesday, March 31, 2015 10:20 AM



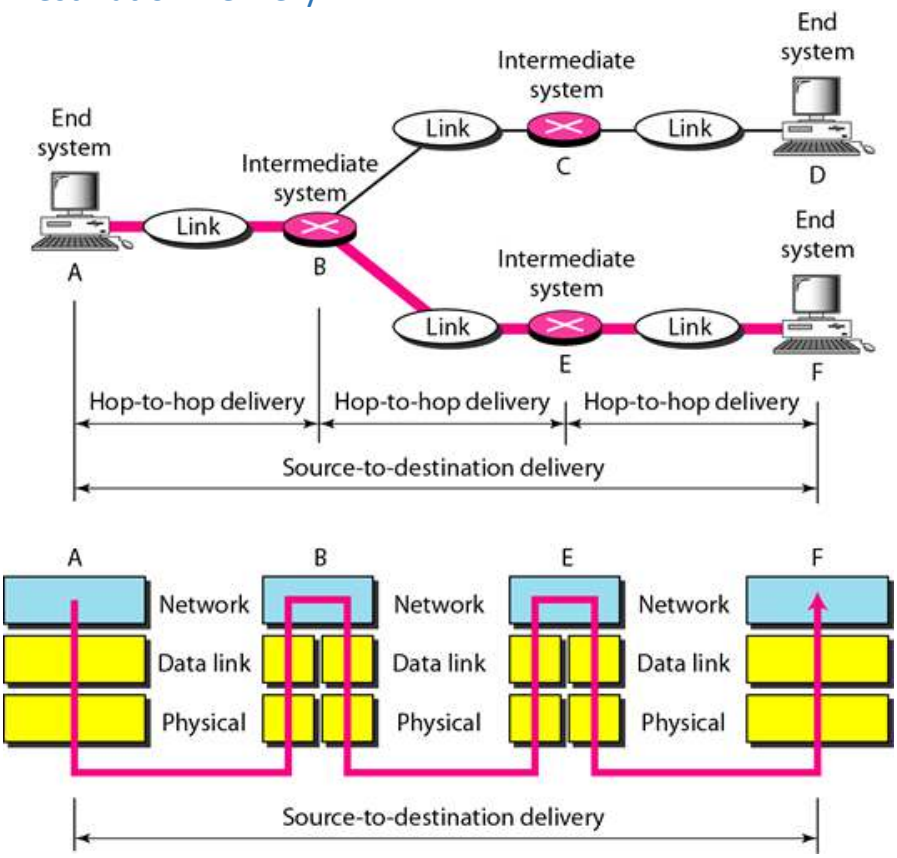
Responsible for source-to-destination delivery of packets, possibly across multiple networks.

Data Link can only deliver the packet within a network.

Other responsibilities:

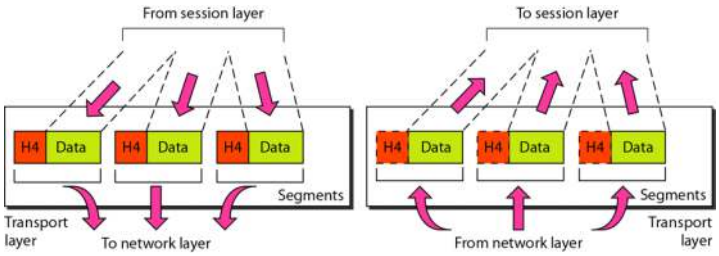
- Logical Addressing.
  - If a packet passes the network boundary, need another addressing system to help distinguish the source and destination systems.
  - Network layer adds a header to the packet coming from the upper layer that contains the logical address of the sender and receiver.
- Routing
  - When independent networks are connected to create internetworks, the routers/switches send packets to their final destination. The network layer does this.

Source to Destination Delivery



2.3.4 Transport Layer

Tuesday, March 31, 2015 10:25 AM



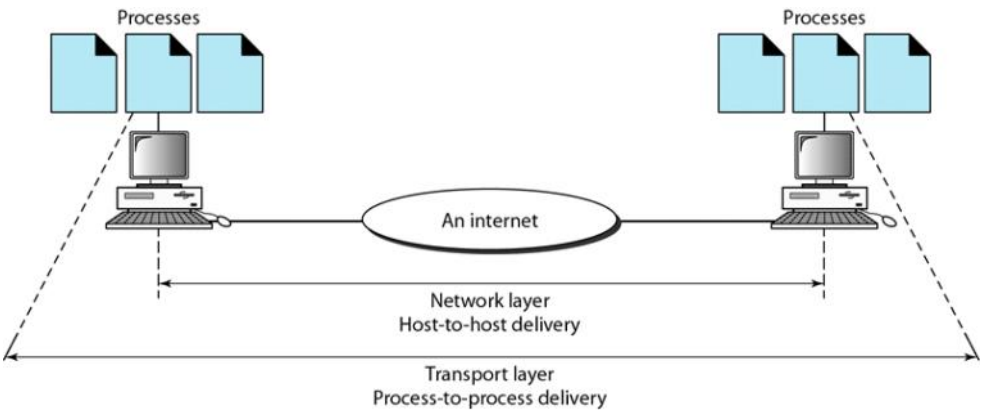
The network layer does not recognize any relationship between packets.

The Transport layer ensures the whole message (group of packets) arrives in tact and in order.

The transport layer also is responsible for delivery of a message **from one process to another.**

Other responsibilities:

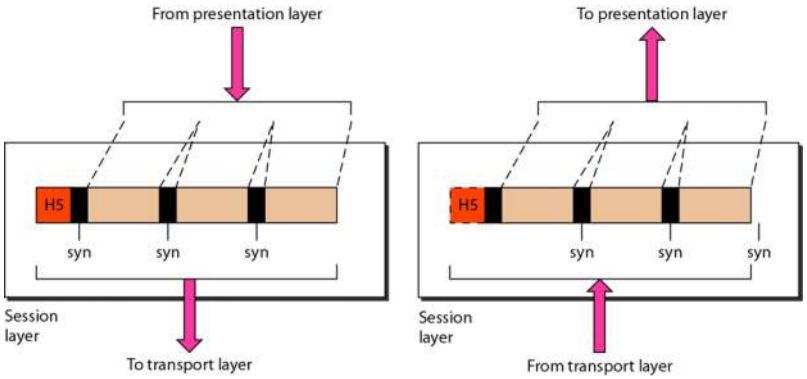
- Service-point addressing.
  - "Computer addresses" have ports (22, 80, etc.).
  - The Transport Layer Header defines a **service-point address (port address)**.
- Segmentation and re-assembly.
  - A message is divided into transmittable segments, each containing a sequence number.
  - On receipt, assemble message in order, replace lost packets.
- Connection control.
  - Can be connectionless or connection-oriented.
    - **Connectionless:** Treats each segment as an independent packet.
    - **Connection-oriented:** Makes a connection with the transport layer at the destination machine first before delivering packets. After all data transferred, connection is terminated.
- Flow control.
  - Like the Data Link Layer, Transport is also responsible for flow control.
  - Flow control at this layer is performed end-to-end rather than link-to-link.
- Error Control.
  - Performed process-to-process rather than across a single link.
  - Sending transport layer makes sure entire message arrives at receiving transport layer without error.
  - Error Correction typically achieved through retransmission.



Reliable process-to-process delivery of a message.

2.3.5 Session Layer

Tuesday, March 31, 2015 10:40 AM



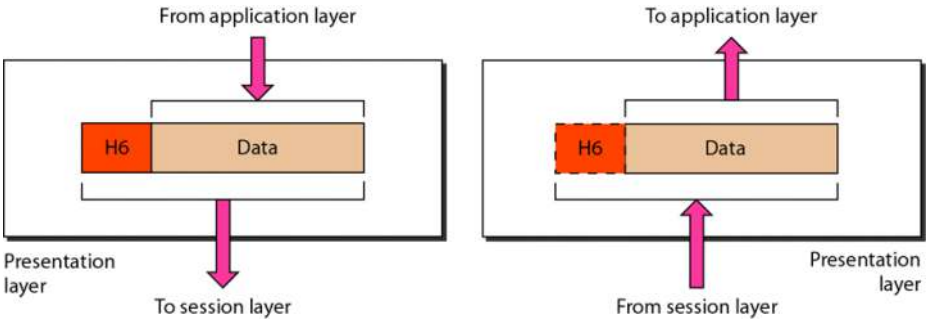
The session layer is the network ***dialog controller.*** It establishes, maintains, and synchronizes the interaction among communicating systems.

The session layer is responsible for dialog control and synchronization.

- **Dialog Control:**
  - Allow two systems to enter into a dialog. Allows communication between two processes to be half-duplex or full-duplex.
- **Synchronization:**
  - Session layer allows a process to add checkpoints, or synchronization points to a stream of data.
  - **Example:** If sending a 2000 page file, add checkpoints every 100 pages to make sure that each 100-page unit is received and acknowledged independently. If a single 100-page section transmission fails, no need to resend the whole file, just that section.

2.3.6 Presentation Layer

Tuesday, March 31, 2015 10:45 AM



The presentation layer is responsible for translation, compression, and encryption.

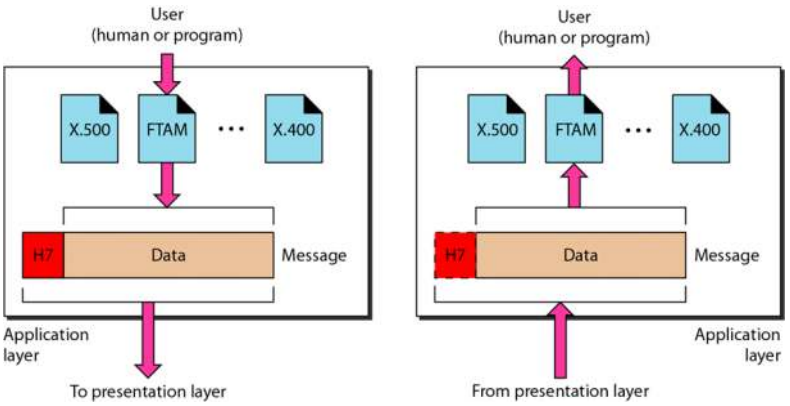
Responsibilities:

- Translation.
  - Changes sender-dependent format into common format. /vise-versa.
    - Example: Character encodings.
- Encryption
- Compression.

2.3.7 Application Layer

Tuesday, March 31, 2015 10:48 AM

The Application Layer is responsible for providing services to the user.

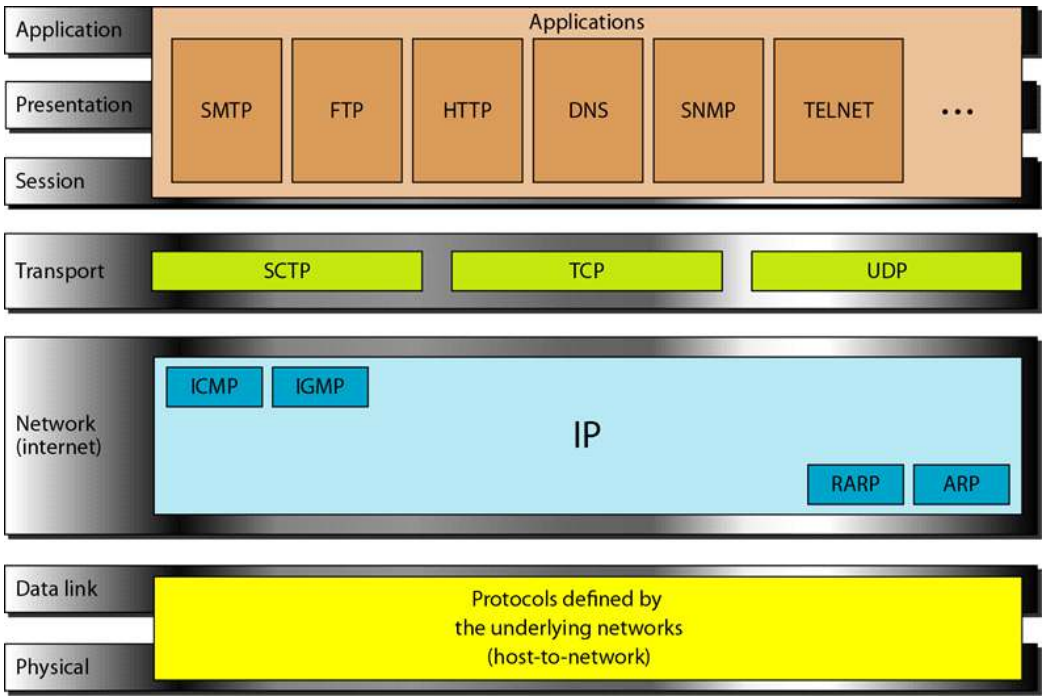


2.4 TCP/IP Protocol Suite

Tuesday, March 31, 2015 10:51 AM

TCP/IP and OSI models were designed at the same time. The layers only sort of match.

TCP/IP Layers		OSI Model Layers
• Application • Transport • Internet • Host-to-network	??	• Application • Transport • Network • Data Link • Physical

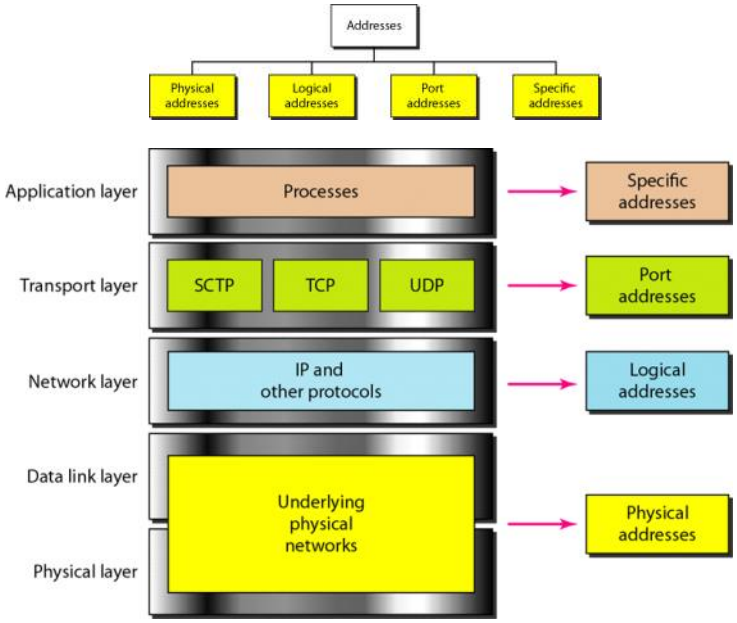


2.5 Addressing

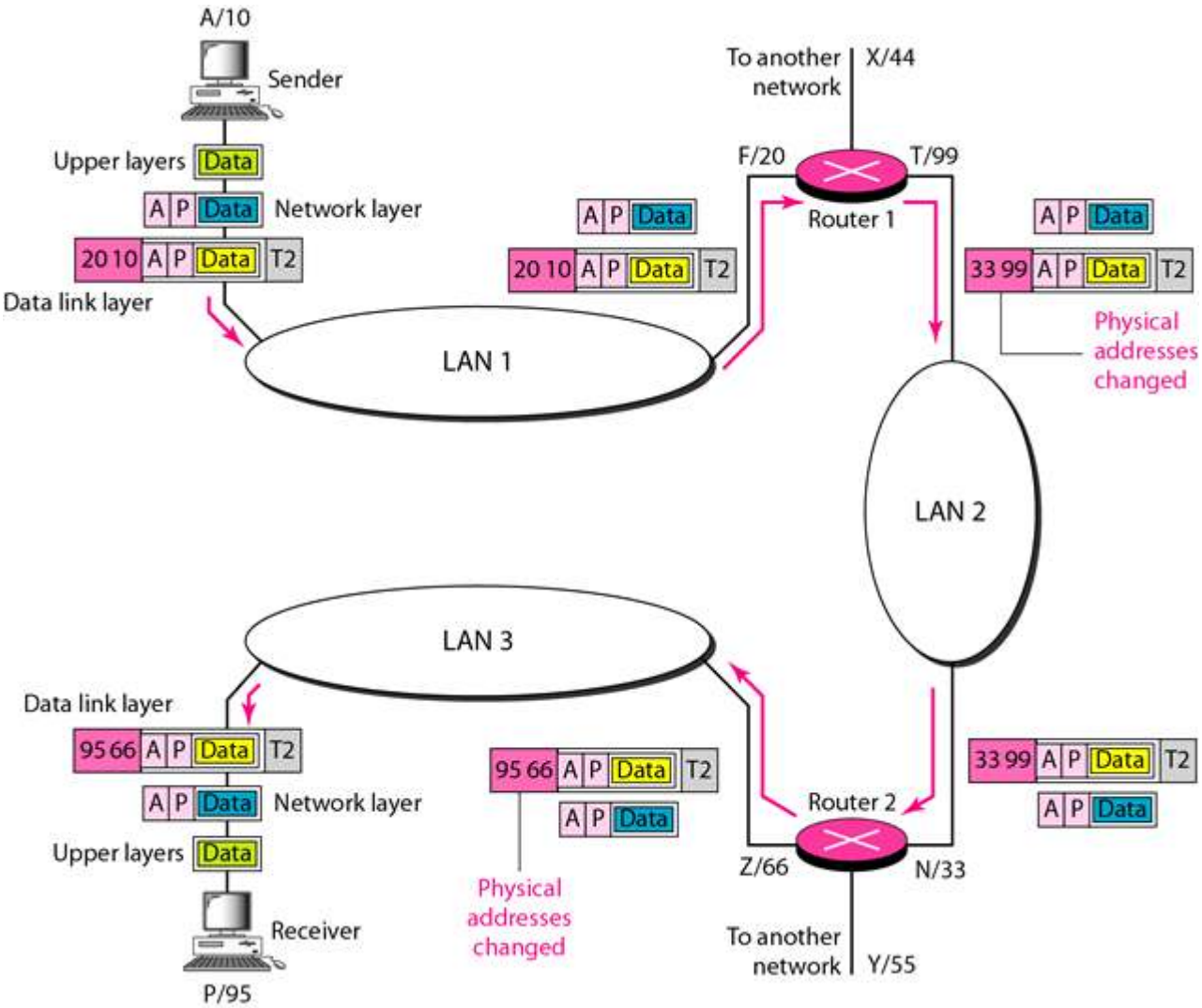
Tuesday, March 31, 2015 11:04 AM

Four levels of addresses are used in an internet employing TCP/IP protocols:

1. Physical
2. Logical
3. Port
4. Specific



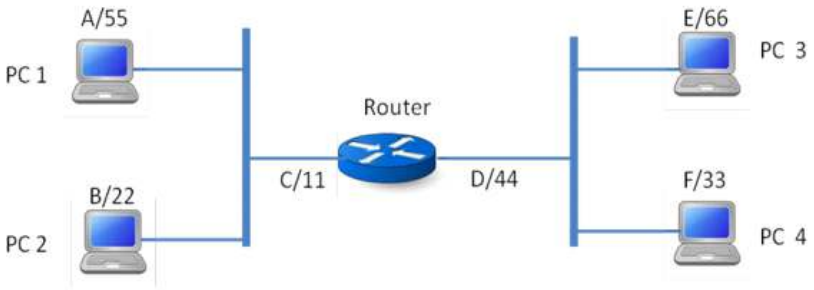
This is taken from Slide 37, Lecture 2.



2.5.a Examples

Tuesday, March 31, 2015 11:12 AM

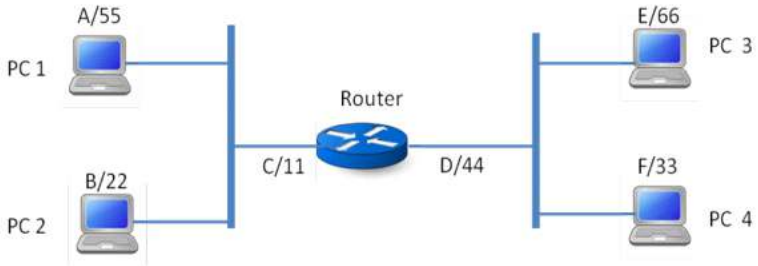
(h) In the following figure, four PCs (with indices 1, 2, 3, and 4) are connected through two bus-topology local area networks (LANs). The address configuration is also shown in the figure, where a capital case letter means an IP address and a number means a physical address. Assume a process with port address 't' on PC 2 sends a message to a process with port address 'q' on PC 4. In the following table, please indicate the addresses used in the header of Layers 2, 3, and 4. **(6 points)**



	Link from PC 2 to the router		Link from the router to PC 4	
	Source address	Destination address	Source address	Destination address
Layer 2 header	22	11	44	33
Layer 3 header	B	F	B	F
Layer 4 header	t	q	t	q

---

5. In the figure on the next page, four PCs (with indices 1, 2, 3, and 4) are connected through two bus-topology LANs. The address configuration is also shown in the figure, where an upper-case letter means an IP address and a number means a physical address. Assume PC 1 sends a message to PC 4. Show the content of the packet and frame at the network and data link layer, respectively, for each of the six connection interfaces (each PC has an interface to its corresponding bus, and the router has two interfaces to the two buses). Example content of a packet or a frame can be found in Slide 37 of Lecture 2. **(5 points)**



Solution:

Packets at the network layer:

At the two connection interfaces of PC1 and the router to the left LAN, and at the two connection interfaces of PC4 and the router to the right LAN, we have

A	F	Layer 4 data

At the connection interface of PC2 to the left LAN, and at the connection interface of PC3 to the right LAN, the corresponding data link frame is dropped, and therefore, we don't have network layer packet.

Frames at the data link layer:

At the three connection interfaces to the left LAN, we have

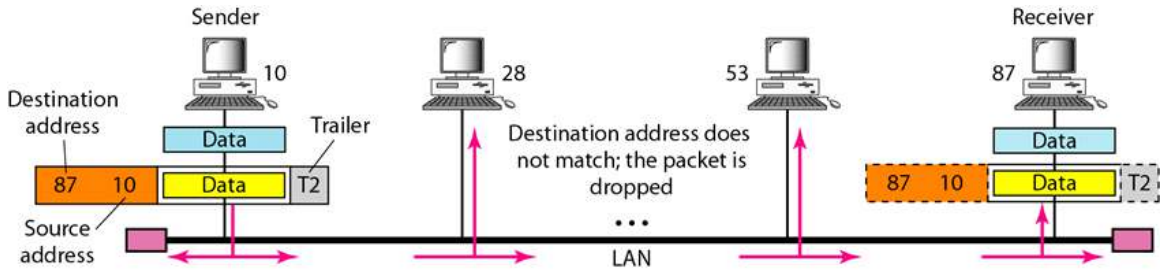
11	55	A	F	Layer 4 data	T2

At the three connection interfaces to the right LAN, we have

33	44	A	F	Layer 4 data	T2

2.5.1 Physical Addressing

Tuesday, March 31, 2015 11:07 AM



Physical address: address of a node as defined by its LAN or WAN standards.

In the above figure a node with physical address 10 sends a frame to a node with physical address 87.

The two nodes are connected by a link (bus topology LAN). As the figure shows, the computer with physical address 10 is the sender, and the computer with physical address 87 is the receiver.

Most local-area networks use a 48-bit (6-byte) physical address written as 12 hex digits. Every byte (2 hex digits) is separated by a colon.

07:01:02:01:2C:4B

A 6-byte (12 hexadecimal digits) physical address.

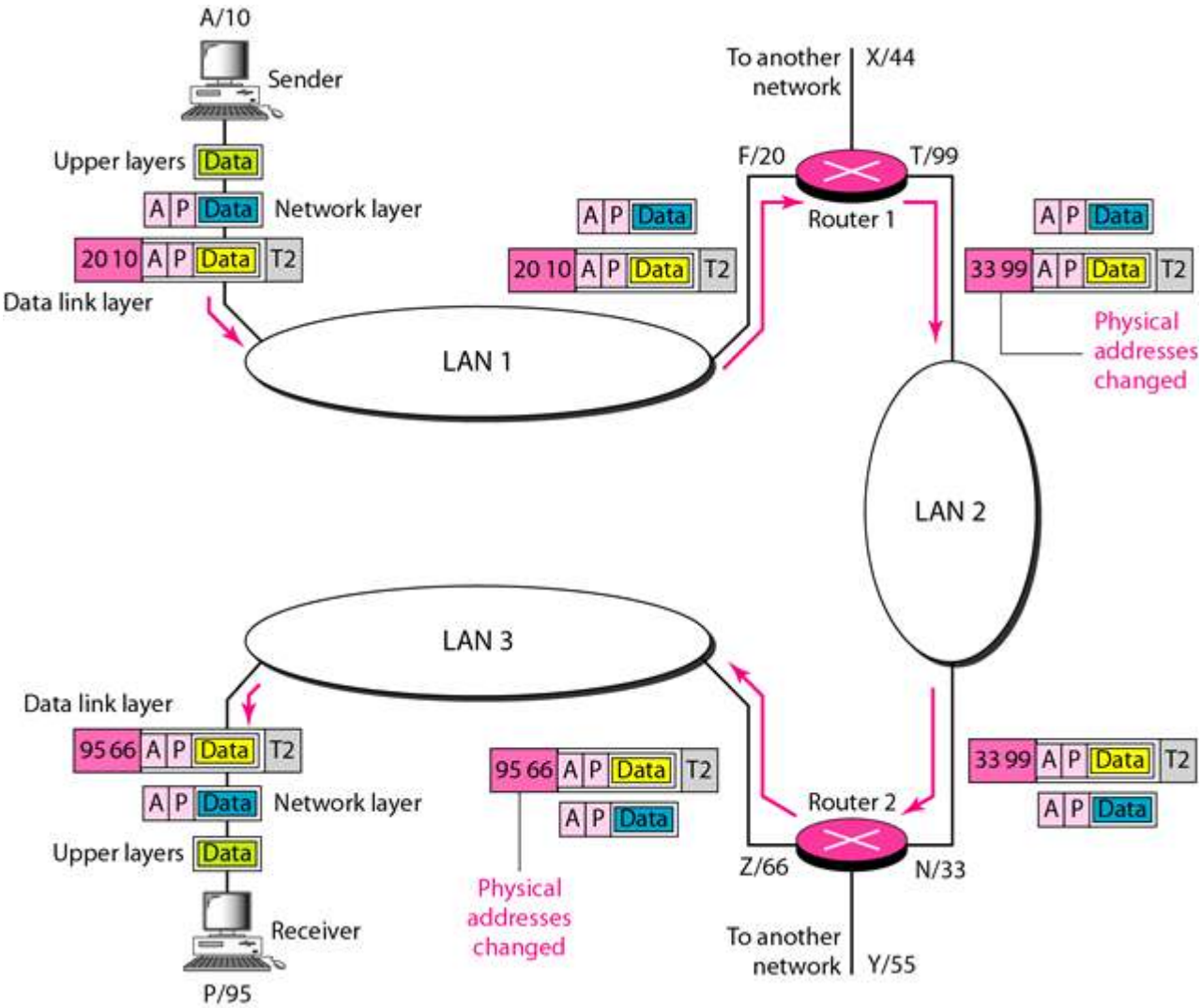
2.5.2 Logical Addressing

Tuesday, March 31, 2015 11:16 AM

Different networks have different physical address formats.→ not adequate for universal communications.

A logic address in the Internet is currently a 32-bit address to uniquely define a host connected to the internet (in IPv4). (128 bits in IPv6)

Next figure shows a part of an internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses. Each router, however, is connected to three networks (only two are shown in the figure). So each router has three pairs of addresses, one for each connection.



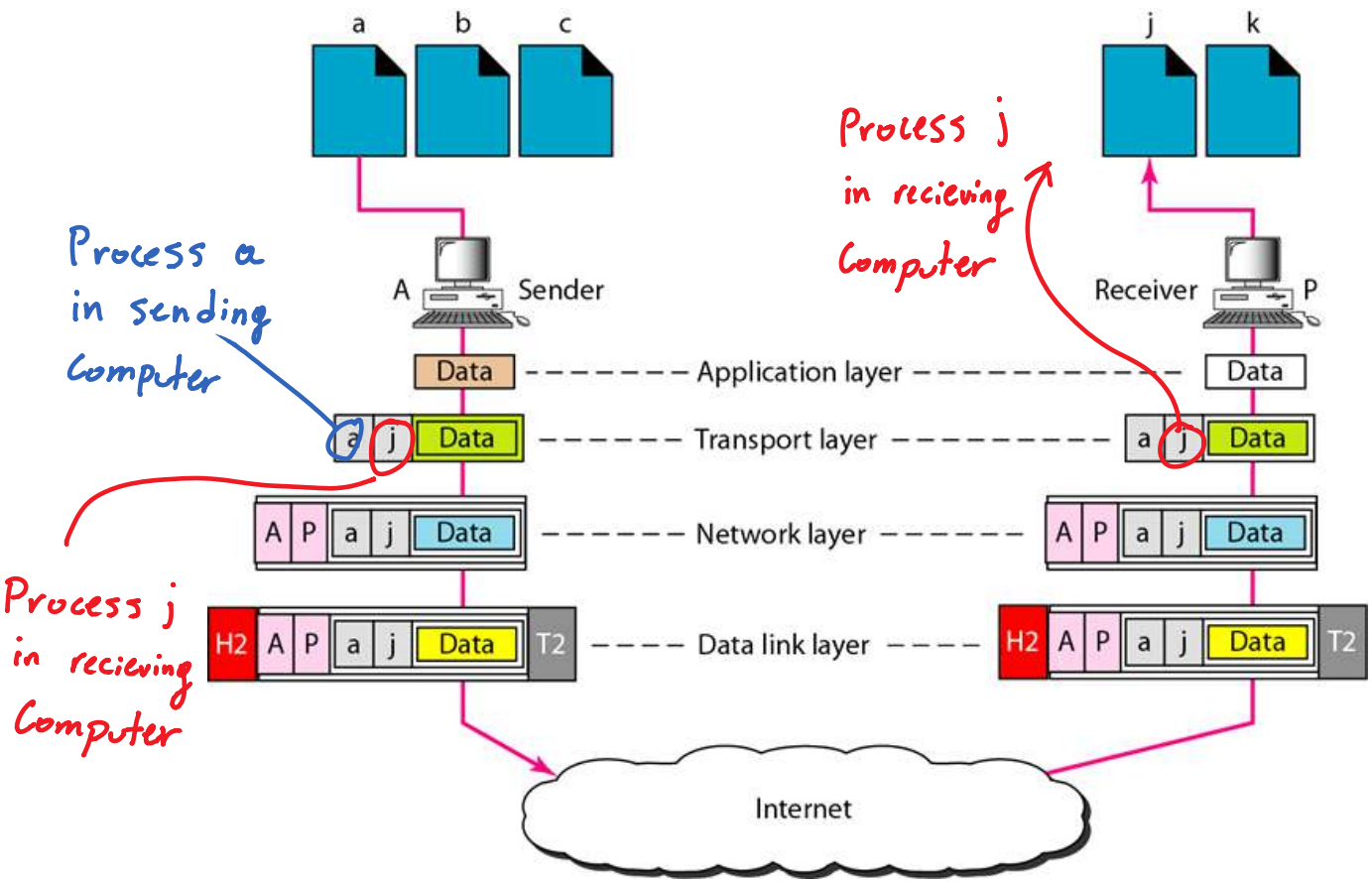
2.5.3 Port Addressing

Tuesday, March 31, 2015 11:26 AM

Port Addresses ate 16-bits, represented by one decimal number.

753

A 16-bit port address represented as one single number.



The physical addresses change from hop to hop, but the logical and port addresses usually remain the same.

If you move and connect your computer to a new network of the same type, your logic address changes, while your physical address keeps the same.

2.5.4 Specific Addresses

Tuesday, March 31, 2015 11:35 AM

User-friendly addresses, such as email address and Universal Resource Locator (URL) (for example, www.ualberta.ca). → get changed to the corresponding port and logic addresses by the sending computer.

Lecture 3: Error Detection and Correction

Tuesday, March 31, 2015 11:36 AM

Chapter 10 in the textbook.

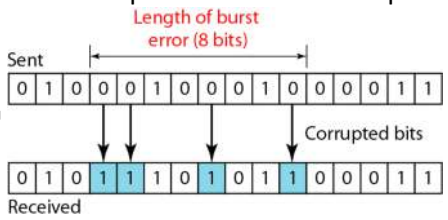
Data can be corrupted during transmission.

Some applications require that errors be detected and corrected.

10.1 Introduction

Tuesday, March 31, 2015 11:38 AM

There are two main types of errors:

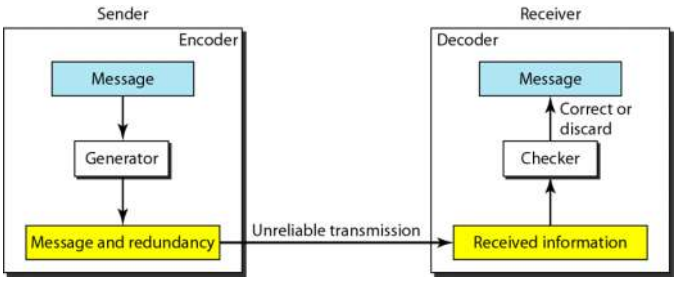
• **Single-bit error** ○ Only 1 bit of a given data unit is flipped (0->1/1->0). ○ Least likely to occur in serial data transmission. ▪ Imagine data sent at 1Mbps. Each bit lasts only 1/1,000,000 [s]. For that, the noise must last only 1 micro second. This is unlikely.	• **Burst Error** ○ 2 or more bits have changed. ▪ These errors do not have to be consecutive. ○ Length of the burst: ▪ First corrupted bit --> Last corrupted bit.  ▪ More likely to occur than a single-bit error, since noise typically occurs and mangles more than 1 bit.

Redundancy

We need to send extra bits with our data.

Detection vs. Correction.

- **Detection.** Only care if an error has occurred. No interest in number of errors.
- **Correction.** Need to know exact number of bits corrupted, and their location.



Coding

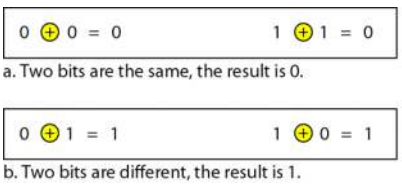
The sender adds redundant bits. The receiver checks the relationships between the two sets of bits to detect or correct errors. The ratio of data bits to redundant bits is very important.
There are two categories of schemes: **Block Coding** and convolution coding.

Modular Arithmetic

We use only a limited range of integers. We define an upper limit, called a ***modulus N***. Then we use only integers **0 to N - 1, inclusive.** This is ***modulo-N arithmetic.***.
Addition and subtraction in modulo arithmetic is simple: no carry between columns at all.

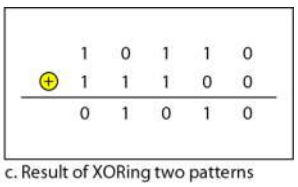
Modulo-2 Arithmetic

In Modulo-2 arithmetic, addition and subtraction are XORs.



a. Two bits are the same, the result is 0.

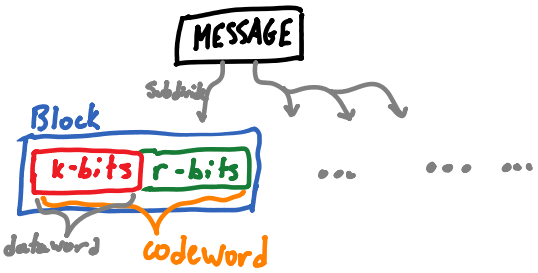
b. Two bits are different, the result is 1.



c. Result of XORing two patterns

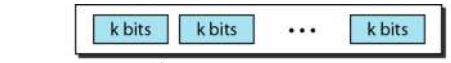
10.2 Block Coding

Tuesday, March 31, 2015 11:57 AM



We divide our message into blocks, each of k-bits into ***datawords.*** We add ***r*** redundant bits to each block to make it length ***n = k + r.***

There's some fun math that makes error checking work.
There are 2^k datawords. There are 2^n codewords. Since $n > k$, then the number of possible codewords is larger than the number of possible datawords. This means that $2^n - 2^k$ Codewords are invalid/illegal.



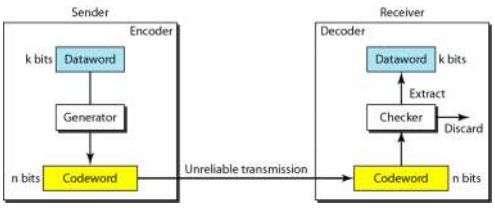
2^k Datawords, each of k bits

2^n Codewords, each of n bits (only 2^k of them are valid)

Error Detection

If the following two conditions are met, the receiver can detect a change in the original codeword:

- The receiver has (or can find) a list of valid codewords.
- The original codeword has changed to an invalid one.



Bogus Example

Let us assume that k = 2 and n = 3. Table 10.1 shows the list of datawords and codewords. Later, we will see how to derive a codeword from a dataword.

Datawords	Codewords
00	000
01	011
10	101
11	110

Assume the sender encodes the dataword 01 as 011 and sends it to the receiver. Consider the following cases:

1. The receiver receives 011. It is a valid codeword. The receiver extracts the dataword 01 from it.
2. The codeword is corrupted during transmission, and 111 is received. This is not a valid codeword and is discarded.
3. The codeword is corrupted during transmission, and 000 is received. This is a valid codeword. The receiver incorrectly extracts the dataword 00. Two corrupted bits have made the error undetectable.

An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected.

Error Correction

We need more redundant bits to perform error correction, compared to detection. The idea is similar to error detection, but the checker functions are more complex.

Bogus Example

Let us add more redundant bits to Example 10.2 to see if the receiver can correct an error without knowing what was actually sent.

We add 3 redundant bits to the 2-bit dataword to make 5-bit codewords. Table 10.2 shows the datawords and codewords.

Table 10.2 *A code for error correction (Example 10.3)*

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

- Assume the dataword is 01.
- The sender creates the codeword 01011.
- The codeword is corrupted during transmission, and 01001 is received.

First, the receiver finds that the received codeword is not in the table. This means an error has occurred. The receiver, assuming that there is only 1 bit corrupted, uses the following strategy to guess the correct dataword.

1. Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits.
2. By the same reasoning, the original codeword cannot be the third or fourth one in the table.
3. The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit. The receiver replaces 01001 with 01011 and consults the table to find the dataword 01.

10.2.1 Hamming Distance

Tuesday, March 31, 2015 12:23 PM

The Hamming Distance between two words (of the same size) is the number of differences between the corresponding bits.

Between word ***x*** and word ***y***, the Hamming Distance is ***d(x,y).***

Procedure to find Hamming Distance

1. XOR the two words.
2. Count the number of 1's in the result.

Minimum Hamming Distance:

- $d_{min}()$
- The smallest Hamming distance between all possible pairs in a set of words.

Parameters for Coding Schemes

All coding schemes are defined by 3 parameters:

- Codeword size ***n***
- Dataword size ***k***
- Minimum Hamming Distance d_{min}

Examples of denoting Coding Schemes:

C(n, k) , d_{min} =?	C(3,2), $d_{min} = 2$	C(5,2), $d_{min} = 3$

To guarantee the detection of up to s errors in all cases, the minimum Hamming distance in a block code must be $d_{min} = s + 1$.

To guarantee correction of up to *t* errors in all cases, the minimum Hamming distance in a block code must be $d_{min} = 2t + 1$.

Guarantees About Performance

- To guarantee ***detection*** of ***s*** errors in all cases, $d_{min} = s + 1$
  - Undetected errors means the errors change a valid codeword to another valid codeword.
  - To change a valid codeword to another valid codeword, we need at least dmin errors.
  - So dmin -1 errors would not change a valid codeword to another valid codeword, and thus, can be detected.

- To guarantee ***correction of upto t errors*** in all cases, $d_{min} = 2t + 1$

Figure 10.8 *Geometric concept for finding d_{min} in error detection*

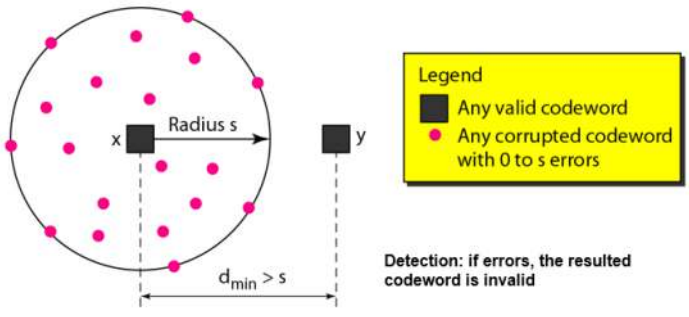
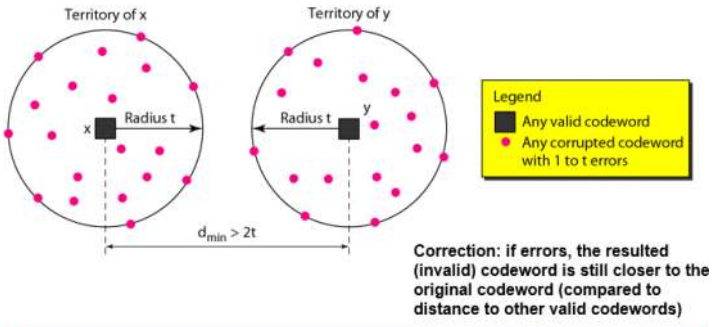


Figure 10.9 *Geometric concept for finding d_{min} in error correction*



10.3 Linear Block Codes

Tuesday, March 31, 2015 12:41 PM

Most block codes used today are ***Linear Block Codes.***

Linear Block Code: The XOR of two valid codewords creates another codeword.

Minimum Hamming Distance for Linear Block Codes

- The number of 1's in the smallest non-zero valid codeword.

Checking if a "code" belongs to Linear Block Codes

1. Check XORing any codeword with another one creates a valid codeword.

10.3.1 Parity Check Codes

Tuesday, March 31, 2015 12:41 PM

The simple parity-check code has an extra **parity bit** added to the dataword.

The parity bit is chosen to make the total number of 1's in the codeword an even number.

C(k+1, k), $d_{min} = 2$

On the sender, the parity bit is generated by:

$r_0 = a_3 + a_2 + a_1 + a_0 \ (modulo - 2)$

If the number of 1's is even, the result is 0. If the number of 1's is odd, the result is 1. In both cases, the number of 1's sent is even.

On the receiver, we generate the **syndrome bit** using the following:

$s_0 = b_3 + b_2 + b_1 + b_0 + q_0 \ (modulo - 2)$

The decision logic checks the syndrome bit.

- Syndrome = 0 --> no error in received codeword
- Syndrome =1 --> received codeword discarded.

A simple parity-check code can detect an odd number of errors.

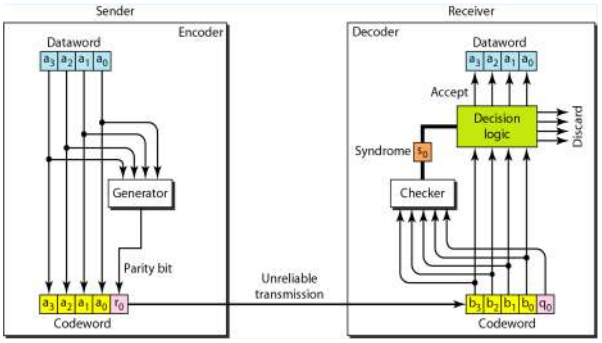
Since $d_{min} = 2$, this code can detect only 2-1 = 1 errors.

A simple parity-check code is a single-bit error-detecting code in which $n = k + 1$ with $d_{min} = 2$.

Table 10.3 *Simple parity-check code C(5, 4)*

Datawords	Codewords	Datawords	Codewords
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

Figure 10.10 *Encoder and decoder for simple parity-check code*

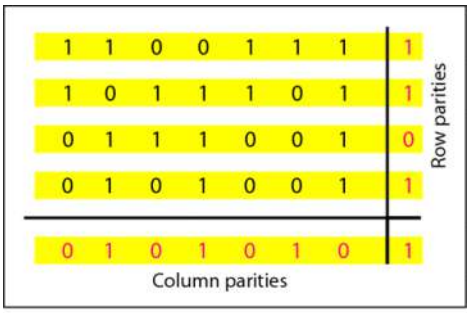


10.3.2 Two-dimensional Parity-check code

Tuesday, March 31, 2015 12:41 PM

In this setup, we take several datawords and arrange them in a 2d array to generate the parity bit check.

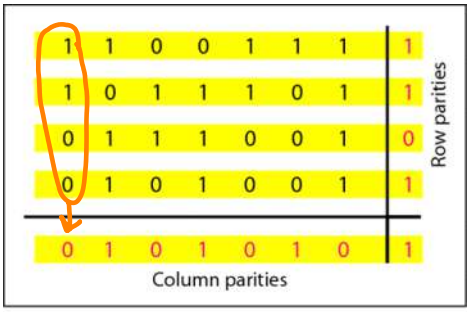
We send the entire "table" over to the receiver. The receiver then checks the rows and columns to calculate a syndrome bit.



a. Design of row and column parities

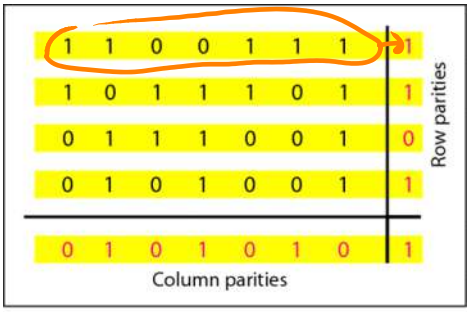
Calculating a Syndrome Bit

1. We need to perform (modulo-2) addition (aka: XORing) along the rows and columns to generate the row and column parities.
2. We check the row and column parities.
   a. If all 13 (in this case) syndromes are zeroes, the receiver thinks there is no error.



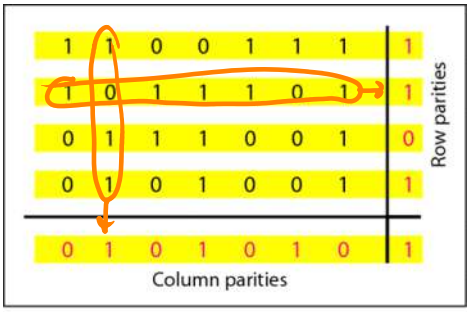
a. Design of row and column parities

1+1+0+0 = 0 (mod 2)



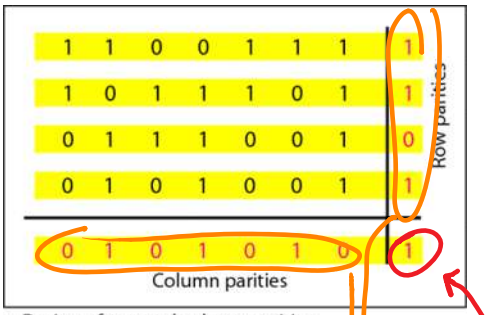
a. Design of row and column parities

Add across mod 2



a. Design of row and column parities

Repeat

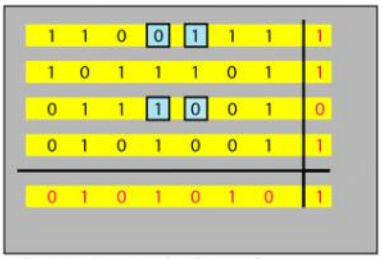


a. Design of row and column parities

If any of these are 1, set syndrome for whole table to 1

Error Detection Pattern

If the errors form a rectangle, then their row/column parities will cancel out.


e. Four errors cannot be detected

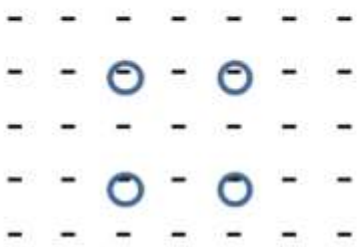
Other patters of four errors will be detected

10.3.2.a Two-dimensional Parity-check code Example (Midterm)

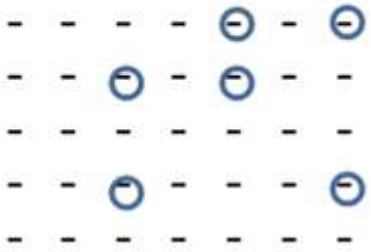
Tuesday, March 31, 2015 12:41 PM

On the Midterm, there was a question about creating an "error pattern" in a 2-d parity check code to generate a certain number of errors.

(8) Please give an example for the case that a two-dimensional parity-check code cannot detect four bit errors; also give an example for the case that a two-dimensional parity-check code cannot detect six bit errors. **(4 points)**



The rationale for the 4 bit-error answer is pretty clear: The 4 bits form a rectangle and in the row/column parities will cancel out.



The rationale in this case is similar. Place 6 circles in the field such that every circle has a "partner" on the same row **AND** column.

10.3.3 Hamming Code C(7,4)

Tuesday, March 31, 2015 12:44 PM

To provide a primer for the material, this the code that is described/ combobulated by the figure at the bottom.

Table 10.4 *Hamming code C(7, 4)*

Datawords	Codewords	Datawords	Codewords
0000	0000000	1000	1000110
0001	0001101	1001	1001011
0010	0010111	1010	1010001
0011	0011010	1011	1011100
0100	0100011	1100	1100101
0101	0101110	1101	1101000
0110	0110100	1110	1110010
0111	0111001	1111	1111111

Hamming Code Constraints

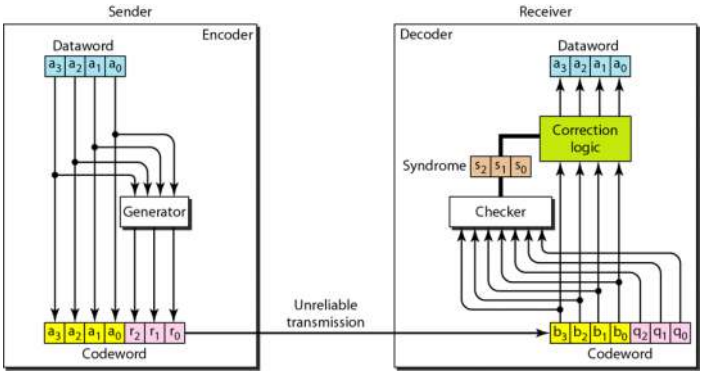
In this course, we have

- $d_{min} = 3$
- $n = 2^m - 1$
  - **m:** number of check bits
  - **n:** number of bits in a codeword

This implies that we can:

- Detect upto 2 errors
- Correct upto 1 error

The encoder and decoder follow the notation shown below-right.



Now for the magic figure that shows all the things.

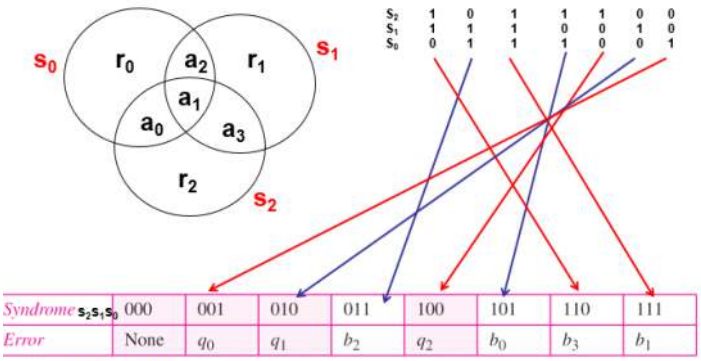
The parity bit generator is typically given in the question. Write it out nicely as shown.

These lines show the correlation between the syndrome bits and the error'd received bit.

For example:

- 011 is in the b_2 column. This means the syndrome code 011 corresponds to an error in the b_2 bit.

$$r_2 = a_3 + a_1 + a_0 \text{ modulo-2} \qquad s_2 = b_3 + b_1 + b_0 + q_2$$
$$r_1 = a_3 + a_2 + a_1 \text{ modulo-2} \qquad s_1 = b_3 + b_2 + b_1 + q_1$$
$$r_0 = a_2 + a_1 + a_0 \text{ modulo-2} \qquad s_0 = b_2 + b_1 + b_0 + q_0$$



The Venn Diagram is mainly for style points. Notice how it expresses the same facts as the table below.

Syndrome $s_2s_1s_0$	000	001	010	011	100	101	110	111
Error	None	q_0	q_1	b_2	q_2	b_0	b_3	b_1

Table 10.5 *Logical decision made by the correction logic analyzer*

The error field shows which bit is incorrect.

Upon receipt:

- Do calculation
- See which bit is an error
- Flip that bit
- Do the calculation again

What does the step "do calculation" mean? Notice how the syndrome bits refer to b_x. Simply plug and chug and see what the actual syndrome bitfield calculates to. Then cross-reference the table and see which bit was incorrect on receipt.

10.3.3.1 Midterm Example

Thursday, April 2, 2015 9:39 AM

This question was a large part of the midterm exam.

2. Consider the encoder and decoder for a Hamming code. Denote the 4-bit dataword at the sender as $a_3a_2a_1a_0$, and the 7-bit codeword at the sender as $a_3a_2a_1a_0r_2r_1r_0$. The three parity check bits are given as follows:

$r_2 = a_2 + a_1 + a_0$ modulo-2 (so r_2 is parity check for a_2, a_1, and a_0)

$r_1 = a_3 + a_1 + a_0$ modulo-2 (so r_1 is parity check for a_3, a_1, and a_0)

$r_0 = a_3 + a_2 + a_1$ modulo-2 (so r_0 is parity check for a_3, a_2, and a_1)

The received codeword at the receiver is denoted as $b_3b_2b_1b_0q_2q_1q_0$.

(a) How does the receiver calculate the three syndrome bits? **(3 points)**

$S_2 = b_2 + b_1 + b_0 + q_2$ modulo-2

$S_1 = b_3 + b_1 + b_0 + q_1$ modulo-2

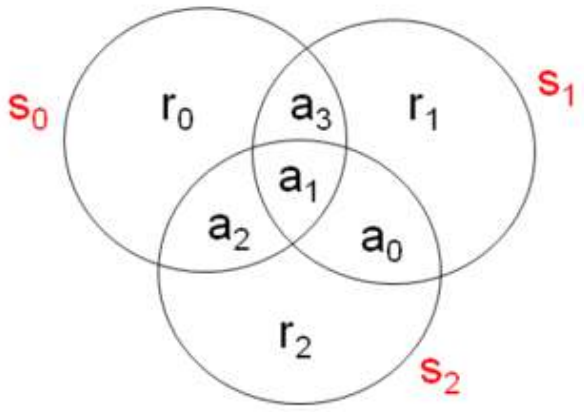
$S_0 = b_3 + b_2 + b_1 + q_0$ modulo-2

(b) The receiver assumes there is at most one bit error in the received codeword. The three-bit syndrome creates eight different bit patterns ("000" to "111"). For each bit pattern, please indicate which bit (among the seven bits in the received codeword) the receiver considers corrupted. **(7 points)**

(c)

Syndrome $S_2S_1S_0$	000	001	010	011	100	101	110	111
corrupted bit	none	q_0	q_1	b_3	q_2	b_2	b_0	b_1

(d)



(e)

10.4 Cyclic Codes

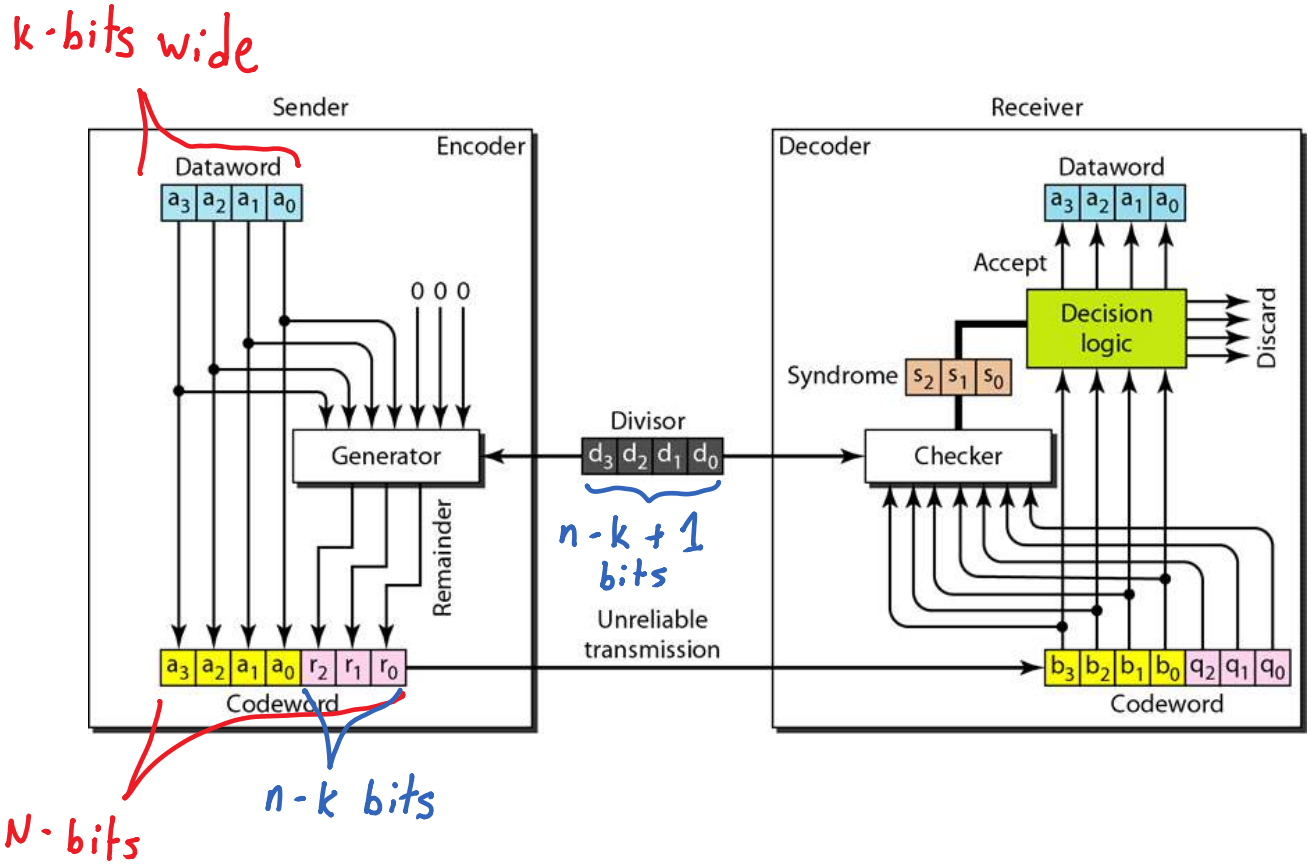
Tuesday, March 31, 2015 12:44 PM

Cyclic Code: If a codeword is cyclically shifted (rotated), the result is another codeword.

In this course, we only cover the ***Cyclic Redundancy Check (CRC).***

Cyclic rotation is typically very fast on CPUs. They usually have a rotate word instruction that moves/rotates the words around with wrap-around.

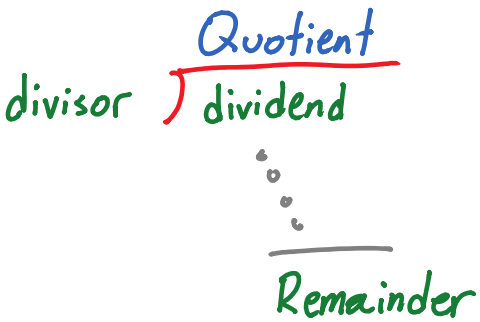
The divisor is agreed upon before transmission.



Creating/Parsing the Codeword.

This is shown on the next page. It involves long-hand binary division. (In this division, addition and subtraction are simply XOR operations with no carry).

Question: What are the things called in long-hand division?
Answer: Grade 3 FTW!

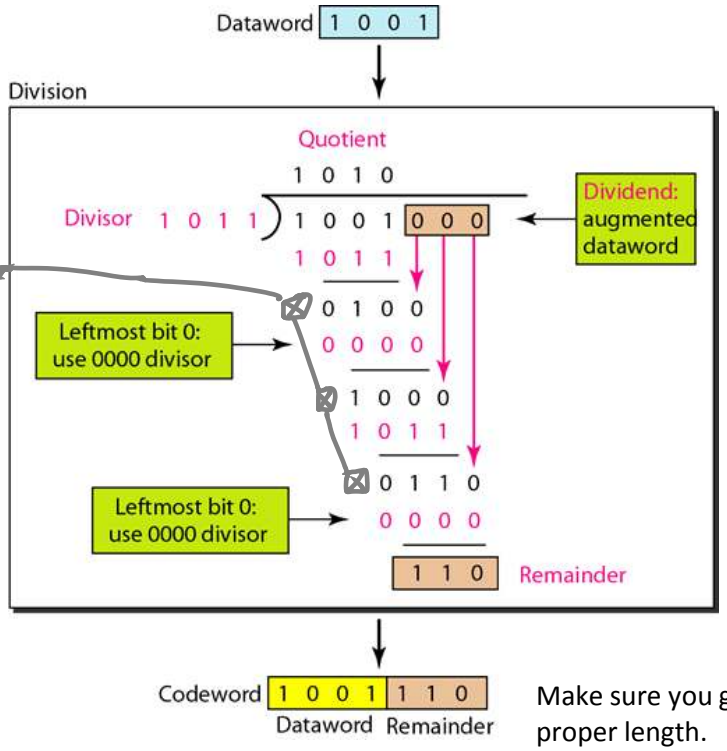


10.4.1 Encoding/Decoding CRC Codewords

Thursday, April 2, 2015 9:53 AM

Encoding

Done as follows:



Strategy:

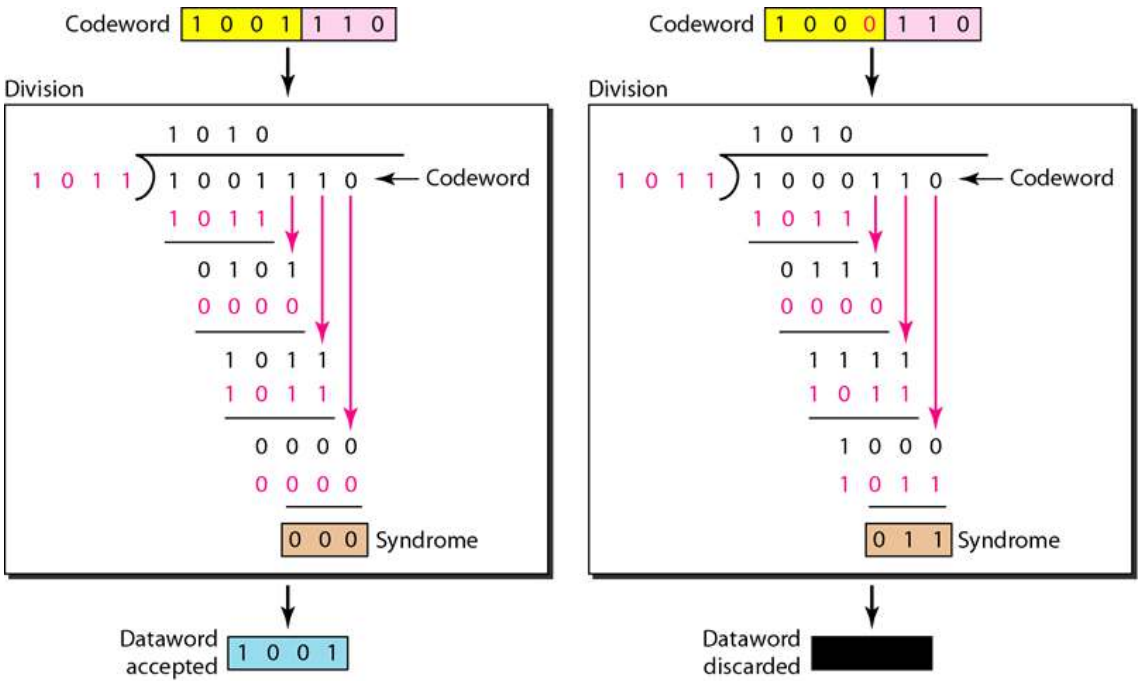
Choose the digit in the quotient that makes this XOR subtraction = 0 so we can move down and right.

n-k bits augmented

Make sure you get a remainder of the proper length.

Decoding:

Perform the same thing, but instead of appending the remainder, check and see that it makes sense (= 000…)



10.4.1.1 Hardware Division for the CRC

Thursday, April 2, 2015 10:00 AM

There is another method for performing the addition quickly, as discussed in ECE 315.

The CRC-32 Calculation

- Polynomial division over the finite field GF(2) is easy to implement in hardware. The hardware just requires shift operations and binary XORs.
- For an *n*-bit binary CRC, the *divisor polynomial* contains *n*+1 bits. The most significant bit (MSB) of the divisor is always a 1.
- The divisor is shifted past the input bit stream, starting out left justified with the most significant bit (e.g., the first arriving bit).

```
10110110110110111101  <-- Input bitstream
1101                  <-- Divisor for 3-bit CRC
```

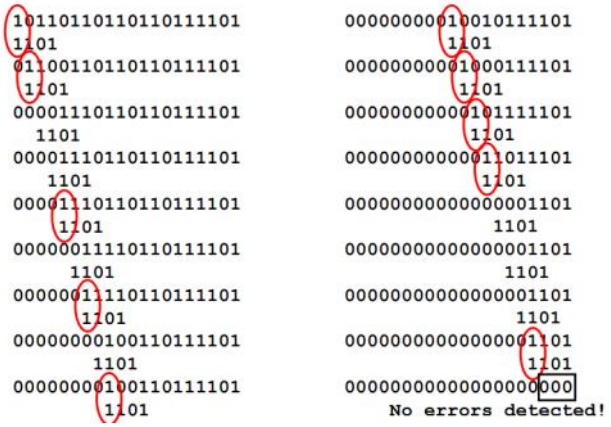
- If the divisor MSB is alongside a 1 in the input stream, then the two vectors are XOR-ed together bitwise. Shift the divisor right, and repeat.

```
01100110110110110101
1101
```

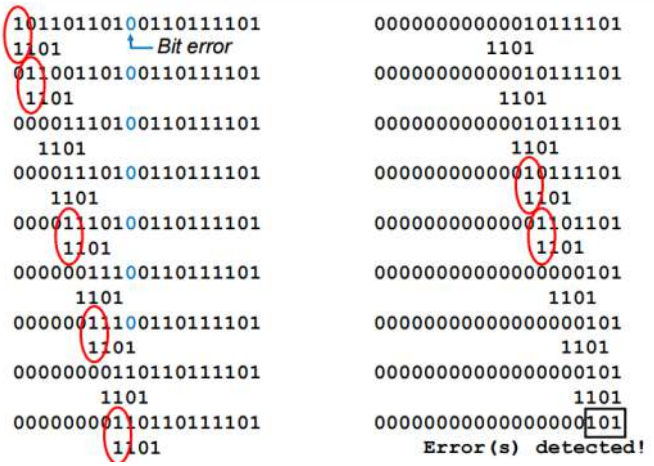
- Once the process is finished, all of the input bits will have been zeroed, except maybe the remainder at the right end. This is the computed CRC.

```
00000000000000000000  <-- CRC (zero, so no errors detected)
      1101
```

Example CRC-3 Calculation (no error)



Example CRC-3 Calculation (with detected error)



10.4.2 CRC Examples (Midterm)

Thursday, April 2, 2015 10:09 AM

3. Consider the encoder and decoder for a Cyclic Redundancy Check (CRC) code. Denote the 4-bit dataword at the sender as $a_3a_2a_1a_0$, and the 7-bit codeword at the sender as $a_3a_2a_1a_0r_2r_1r_0$. The received codeword at the receiver is denoted as $b_3b_2b_1b_0q_2q_1q_0$. The divisor at the sender and receiver is $d_3d_2d_1d_0$=1011.

   (a) Please give the codeword for dataword '1001'. Show your steps. **(4 points)**
   (b) We know that codeword '1110100' is a valid codeword. Assume that the codeword is sent from the sender side. Also assume that the channel from the sender to the receiver may have a burst error. Give example of a burst error that cannot be detected at the receiver side. Show your steps. **(6 points)**

(a)

```
      1010
1011 ) 1001000
       1011
        0100
        0000
         1000
         1011
          0110
          0000
           110
```

Codeword is 1001110

The bottom one is from the ECE 315 Midterm.

(b) The Cyclic Redundancy Check (CRC) bits are appended at the end of the Ethernet frame for the purpose of error detection. For a CRC generator polynomial given by $x^3 + x + 1$, compute the CRC bits to be appended to the following stream of bits:
1 1 0 0 1
[2 marks] The coefficients of the generator polynomial are given by 1011 $(1.x^3 + 0.x^2 + 1.x + 1)$.
[3 marks] Since the polynomial contains 4 bits, the CRC will be 3 bits. Before computing the CRC, 3 zeros should be padded at the end of the stream.
[5 marks] The computed CRC is "1 1 1" as obtained below:

```
1 1 0 0 1 0 0 0
1 0 1 1
0 1 1 1 1 0 0 0
  1 0 1 1
0 0 1 0 0 0 0 0
    1 0 1 1
0 0 0 0 1 1 0 0
      1 0 1 1
0 0 0 0 0 1 1 1
```

10.5 Checksum

Tuesday, March 31, 2015 12:44 PM

We don't really touch on Checksums in this course too much, but it's included in the lecture slides.

Example 10.18

- ***Suppose our data is a list of five 8-bit numbers that we want to send to a destination.***
- ***In addition to sending these numbers, we send the sum of the numbers.***
    - ***For example, if the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12, 0, 6, 36), where 36 is the sum of the original numbers.***
        - ***The receiver adds the five numbers and compares the result with the sum.***
            - ***If the two are the same, the receiver assumes no error, accepts the five numbers, and discards the sum.***
            - ***Otherwise, there is an error somewhere and the data are not accepted.***
- ***We can make the job of the receiver easier if we send the negative (complement) of the sum, called the checksum. In this case, we send (7, 11, 12, 0, 6, −36). The receiver can add all the numbers received (including the checksum). If the result is 0, it assumes no error; otherwise, there is an error.***

Lecture 5: Multiple Access

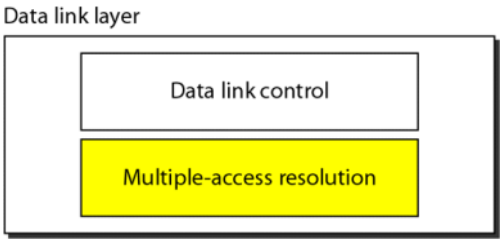
Thursday, April 2, 2015 10:16 AM

Chapter 12 in the textbook.

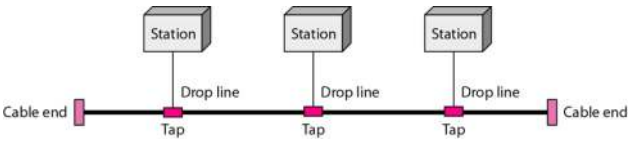
The Data Link Layer is actually made of two sublayers. So far in the course, we have assumed that between every sender and receiver, there is a dedicated link (channel). This is typically seen in PPP (Point-to-Point Protocol).

If we need to share the channel, then we need multiple access. Consider a person using a cell phone only a few feet away from you. Both of you are probably using the same frequency band / channel. How do we split this up so both of you get service?

To do this, we need **Multiple-Access Resolution.** Thus, we can view the Data-Link-Layer as having two separate sub-layers.



To further motivate the discussion, consider the bus network topology.



How do we decide which station has access to the bus? Is it divided by time?
By Frequency (on a spectrum)?

12.1 Random Access

Thursday, April 2, 2015 10:22 AM

Random Access (Contention) *methods:*

- No station is superior.
  - No station assigned control over another.
  - No station does not permit/not permit another station to send.
    - The decision is based on the state of the medium
      - (Aka: is the bus busy with things on it?).
- There is no scheduled time for a station to transmit.
  - Every station transmits whenever it feels like it.
- There is no agreement on who transmits *next*.
  - Stations compete (**contention)** on who takes the medium next.

If more than one station attempts to send at the same time there is an **access conflict/collision** and frames are either destroyed/lost/or modified.

- To avoid this, we need to follow a procedure.

Any procedure for multiple access must answer the following questions:

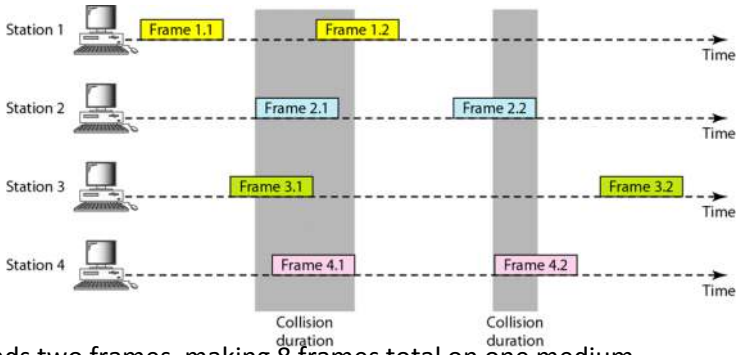
- When can the station access the medium?
- What can the station do if the medium is busy?
- How can the station determine the success for failure of the transmission?
- What can the station do if there is an access conflict?

The following protocols for **random-access** are discussed:

- ALOHA
- Carrier Sense Multiple Access (CSMA)
- Carrier Sense Multiple Access with Collision Detection
- Carrier Sense Multiple Access with Collision Avoidance.

12.1.1 ALOHA

Thursday, April 2, 2015 10:28 AM



Main Idea

- Each station sends a frame whenever it has a frame to send.
- Since there is only one channel, there is a possibility of collisions.

In the figure above, there are 4 stations. Each station sends two frames, making 8 frames total on one medium. Some of these frames collide. The above figure shows that only two frames survive.

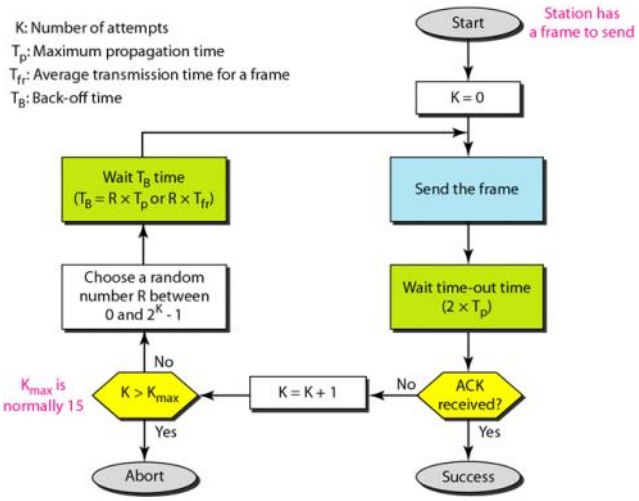
Pure ALOHA relies on ACKnowledgements from the server. If the ACK does not arrive in a specific time period, then the sender just re-sends the frame.

- **Problem:** If two stations re-send frames at the exact same times on the same medium, both frame will be lost. (Hello Race Condition!)
- **Solution:** Pure ALOHA dictates that a sender, upon failing to get an ACK (ACK timed out), will wait a ***random amount of time*** before re-sending the frame. This is the ***back-off time,*** T_B

What if the channel is really crappy and it gets filled with re-sends? This creates congestion.

- Pure ALOHA prevents congesting the channel with re-transmitted frames.
  - After a maximum number of re-transmission attempts (K_{max}), a station must give up and try again later.

The above discussion is shown in the figure below.



Constraints/Figures:

- **Time-out Period:**
  - = maximum possible round-tip delay.
- **Back-off Time T_B**
  - Typically related to K (maximum re-transmission attempts.
  - Implementation Specific.

Back-off Time Calculation.

In this course, we have a formula for calculating T_B

Procedure:

1. Randomly pick a number between $0 \dots 2^K - 1$. Call this number Z. (Where Z is a discrete range of integers).
2. $T_B = Z * T_P$, where T_P is the maximum propigation time.
3. ***OR*** $T_B = Z * T_{rr}$, where T_{rr} is the average time required to send out a frame.

The value of K_{MAX} is typically chosen as 15.

Example

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3×10^8 m/s, we find

$$T_p = (600 \times 10^5) / (3 \times 10^8) = 2 \text{ ms.}$$

Now we can find the value of T_B for different values of K .

a. For K = 1, the range is {0, 1}. The station needs to generate a random number with a value of 0 or 1. This means that T_B is either 0 ms (0 × 2) or 2 ms (1 × 2), based on the outcome of the random variable.

b. For K = 2, the range is {0, 1, 2, 3}. This means that T_B can be 0, 2, 4, or 6 ms, based on the outcome of the random variable.

c. For K = 3, the range is {0, 1, 2, 3, 4, 5, 6, 7}. This means that T_B can be 0, 2, 4, . . . , 14 ms, based on the outcome of the random variable.

d. We need to mention that if K > 10, it is normally set to 10. (in other words, the range is {0,1,2,…,1023})

12.1.1.a ALOHA Vulnerable Time

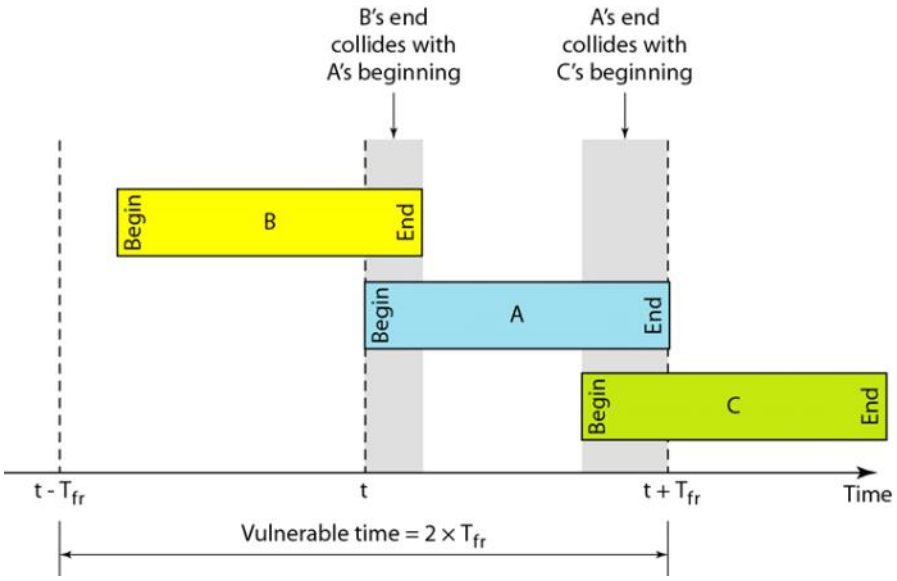
Thursday, April 2, 2015 10:48 AM

Vulnerable Time: The length of time in which there is a possibility of collision.

Station A sends a frame at time t. Imagine Station B has sent a frame between $t - T_{fr}$ and t.

This leads to a collision between frames from Stations A and B. The end of B's frame collides with the start of A's frame.

Suppose C sends a frame between $t + T_{fr}$ and t. Here a collision occurs between the end of A's frame and the start of C's frame.



Example 12.2

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} is 200 bits/200 kbps or 1 ms. The vulnerable time is 2 × 1 ms = 2 ms. This means no station should send later than 1 ms before this station starts transmission and no station should start sending during the one 1-ms period that this station is sending.

12.1.1.b ALOHA Throughput

Thursday, April 2, 2015 10:54 AM

The throughput for pure ALOHA is
$S = G \times e^{-2G}$.
The maximum throughput
$S_{max} = 0.184$ when G= (1/2).

G: average # of frames generated by the system during one frame transmission time

S: average # of successful frames during one frame transmission time.

Example

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces
a. 1000 frames per second b. 500 frames per second c. 250 frames per second.

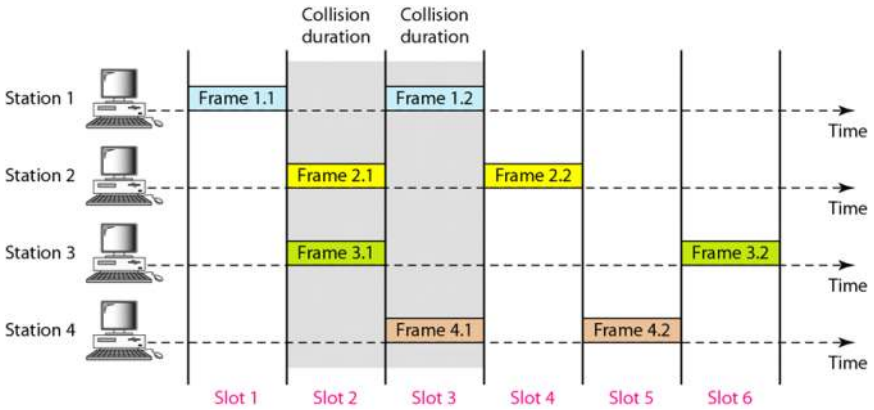
Solution
The frame transmission time is 200/200 kbps or 1 ms.
a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case S = G× e ^{2G} or S = 0.135. In a second, there are 135 successful frames. This means that 13.5% of the generated frames will probably survive.
b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case S = G × e ^{2G} or S = 0.184. In a second, there are 184 successful frames. This means that 36.8% of the generated frames will probably survive.
c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case S = G × e $^{-2G}$ or S = 0.152. In a second, there are 152 successful frames. This means that 60.8% of the generated frames will probably survive.

# of frames generated/second	G (# of frames generated per T_t)	S (# of successful frames per T_t)	# of successful frames/second	Successful prob.
1000	1	0.135	135	13.5%
500	0.5	0.184	184	36.8%
250	0.25	0.152	152	60.8%

12.1.1.c Slotted ALOHA Vulnerability Time

Thursday, April 2, 2015 10:58 AM

Pure ALOHA did not have a restriction on when stations can send. This means that stations could start spamming, sending just after/before another station has finished sending stuff.

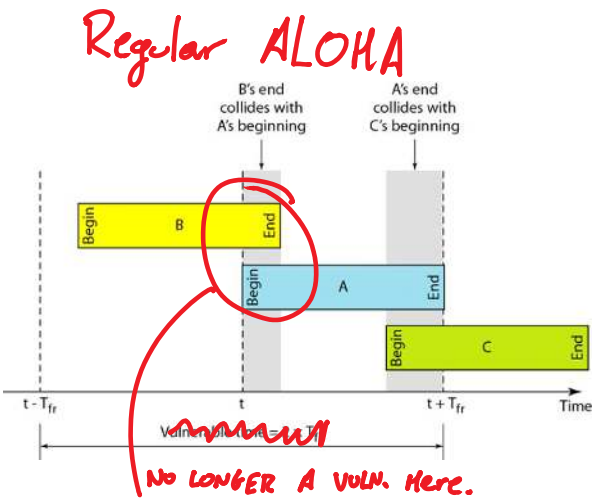


Slotted ALOHA: We divide time into **time slots** of size T_{FR}.We force the station to send only at the beginning of the timeslot.

These timeslots are agreed upon by all stations.

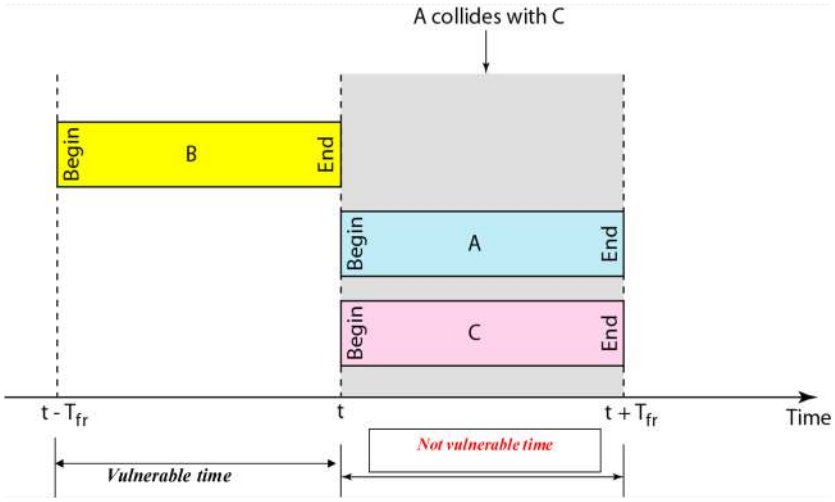
A station is only allowed to send at the beginning of a time slot. If it misses the start of the slot, it must wait until the beginning of the next time slot.

- This implies the station which started at the beginning of the time slot has already finished sending its frame.
    - This means that we no longer have a vulnerability as shown.



Slotted ALOHA Vulnerable Time = T_{FR}

Figure 12.7 *Vulnerable time for slotted ALOHA protocol*



12.1.1.d Slotted ALOHA Throughput

Thursday, April 2, 2015 11:10 AM

The throughput for slotted ALOHA is $S = G \times e^{-G}$.
The maximum throughput $S_{max} = 0.368$ when G = 1.

Example 12.4

A slotted ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces
a. 1000 frames per second b. 500 frames per second c. 250 frames per second.

Solution
The frame transmission time is 200/200 kbps or 1 ms.
a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{G}$ or S = 0.368. In a second, there are 368 successful frames. This means that 36.8% of the generated frames will probably survive.
b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{G}$ or S = 0.303. In a second, there are 303 successful frames. This means that 60.6% of the generated frames will probably survive.
c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case $S = G \times e^{G}$ or S = 0.195. In a second, there are 195 successful frames. This means that 78% of the generated frames will probably survive.

12.1.2 Carrier Sense Multiple Access (CSMA)

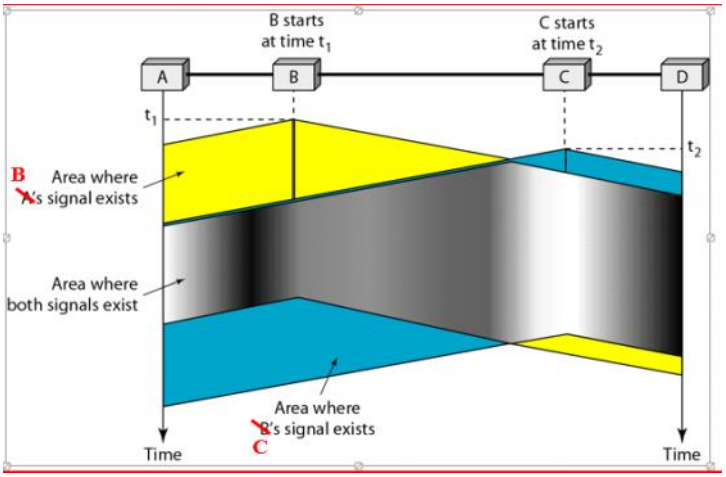
Thursday, April 2, 2015 11:12 AM

The chance of collision can be reduced if a station senses the medium before trying to use it.

CSMA requires:

- Each station first listen to the medium (aka: check the state) before sending
- "Sense before transmit" / "Listen before talk"

CSMA can reduce possibility of collision, but not eliminate it.



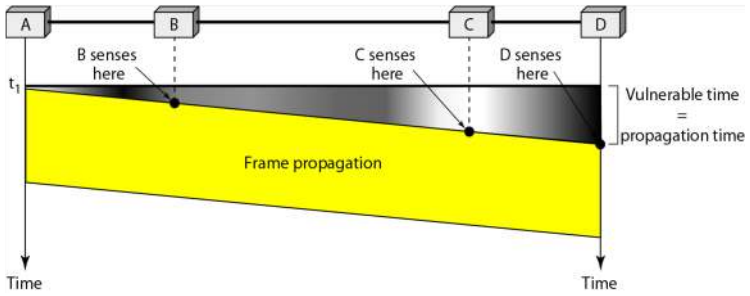
The possibility of collision still exists because of propagation delay. When a station sends a frame, it still takes time for the first bit to reach every station and for every station to sense it.

- A station may sense the medium and find it idle, only because the first bit sent by another station had not been received (due to propagation delay).

In the above-right diagram:

1. At time T1, station B senses the medium and finds it idle, so it sends a frame.
2. At time T2 > T1, station C senses the medium and finds it idle, because at time T2, the first bits from station B haven't made it to station C.
3. Both stations have sent a frame at the "same-enough" time. Both frames are destroyed.

Figure 12.9 *Vulnerable time in CSMA*



Vulnerable Time in CSMA

(pg 408 in the textbook PDF).

12.1.2.1 CSMA Persistence Methods.

Thursday, April 2, 2015 11:23 AM

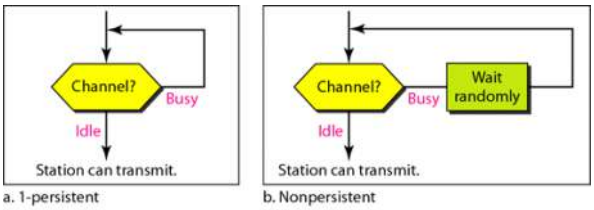
There are several questions to answer:

- What should a station do if the channel is busy?
- What should it do if the channel is idle?

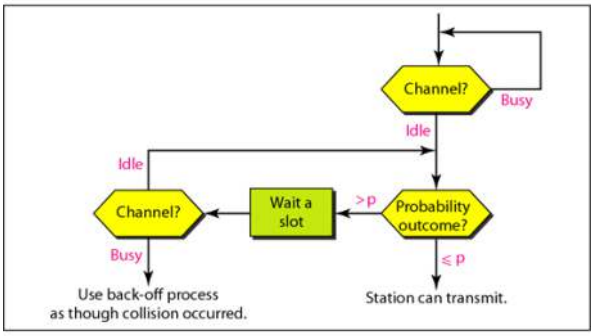
There are 3 persistence methods:

- **1-Persistent:**
  - After the station finds the line idle, it sends the frame immediately (with probability 1).
    - Highest chance of collisions --> since 2 or more stations may find the line idle and send their frames immediately.
    - Ethernet does this.
- **Non-persistent**
  - If the line is idle, sends frame immediately. If the line is not idle, wait a random amount of time before sensing the line again.
    - Reduces chances of collisions. Reduces network efficiency.
- **P-persistent**
  - Used if the channel has time slots with a ***slot duraction >= propagation time***. Reduces chance of collision, improves efficiency.
    - Procedure:

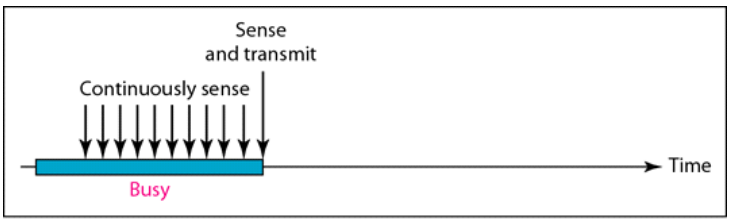
a. With probability p, the station sends its frame.
b. With probability q = 1 - p, the station waits for the beginning of the next timeslot and checks the line again
   i. If line is idle, go to step 1.
   ii. If line is busy, behave as if collision occurred and used *back-off procedure.*



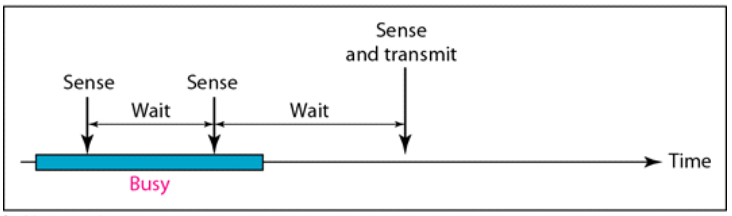
a. 1-persistent
b. Nonpersistent



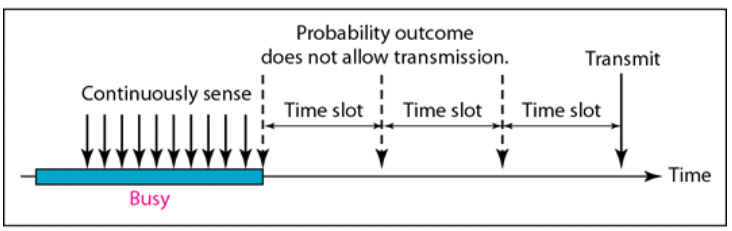
c. p-persistent



a. 1-persistent



b. Nonpersistent



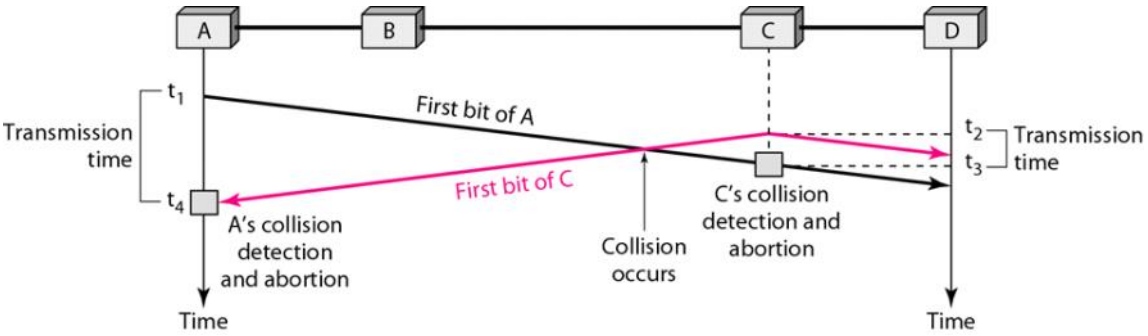
c. p-persistent

12.1.3 (CSMA/CD) Carrier Sense Multiple Access with Collision Detection

Monday, April 13, 2015 8:06 AM

CSMA does not specify what to do if there is a collision. CSMA/CD augments the algorithm the handle the collision.

A station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. Otherwise, the frame is sent again.



1. At time t1, A starts sending bits of the frame.
2. At time t2, C hasn't sensed the first bits from A.
3. C starts sending bits, which go left and right.
4. Sometime after t2, a collision occurs.
5. At time t3, C detects a collision when it received part of A's frame.
    a. (Sending things…. Oh shit! We have some of A's frame before mine should get anywhere!)
6. C immediately stops transmission.
7. A detects collision at time t4 when it gets part of C's frame. A immediately aborts transmission.
8. We see that
    a. A was spamming for time t4 - t1
    b. C was spamming for time t3 - t2.

For the protocol to work, $\frac{length\ of\ any\ frame}{bit\ rate\ in\ protocol} > (t_4 - t_1) and\ (t_2 - t_3)$.

Minimum Frame Size

The ***frame transmission time*** (T_{FR})must be > 2* propagation time.

$T_{FR} > 2T_p$

To reason why, imagine worst case scenario:

- If two stations involved in a collision are max distance apart, the signal takes T_p to go from one end to the other. The other station's message also takes time T_p. The first station must still be transmitting after time $2T_p$.

12.1.3.a Minimum Frame Size

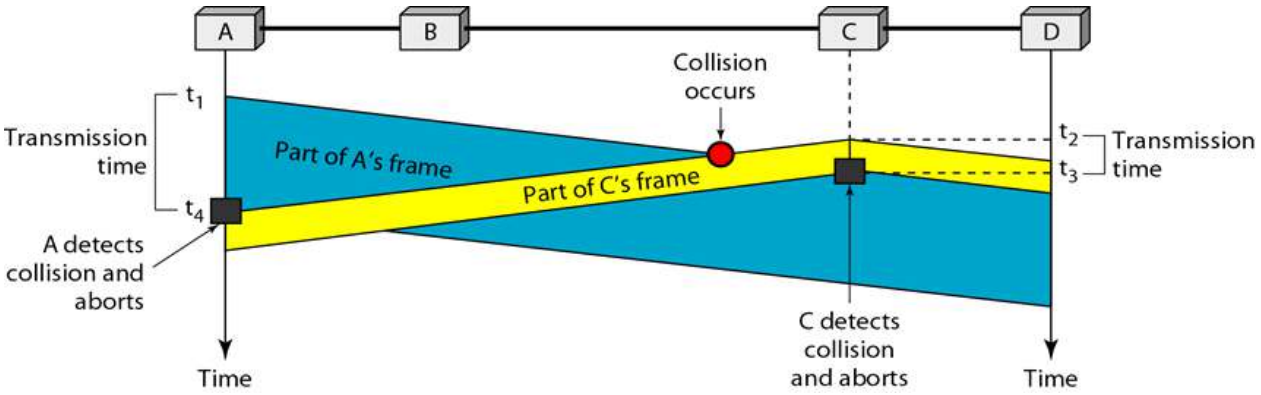
Monday, April 13, 2015 8:20 AM

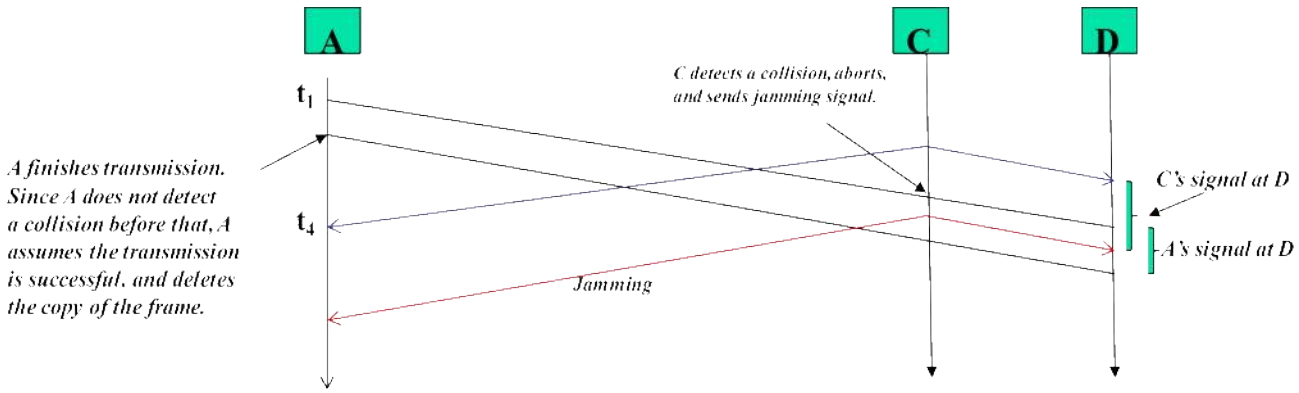
Example 12.5

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6 µs, what is the minimum size of the frame?

Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2$ µs. This means, in the worst case, a station needs to transmit for a period of 51.2 µs to detect the collision. The minimum size of the frame is 10 Mbps × 51.2 µs = 512 bits or 64 bytes. This is actually the minimum size of the frame for Standard Ethernet.

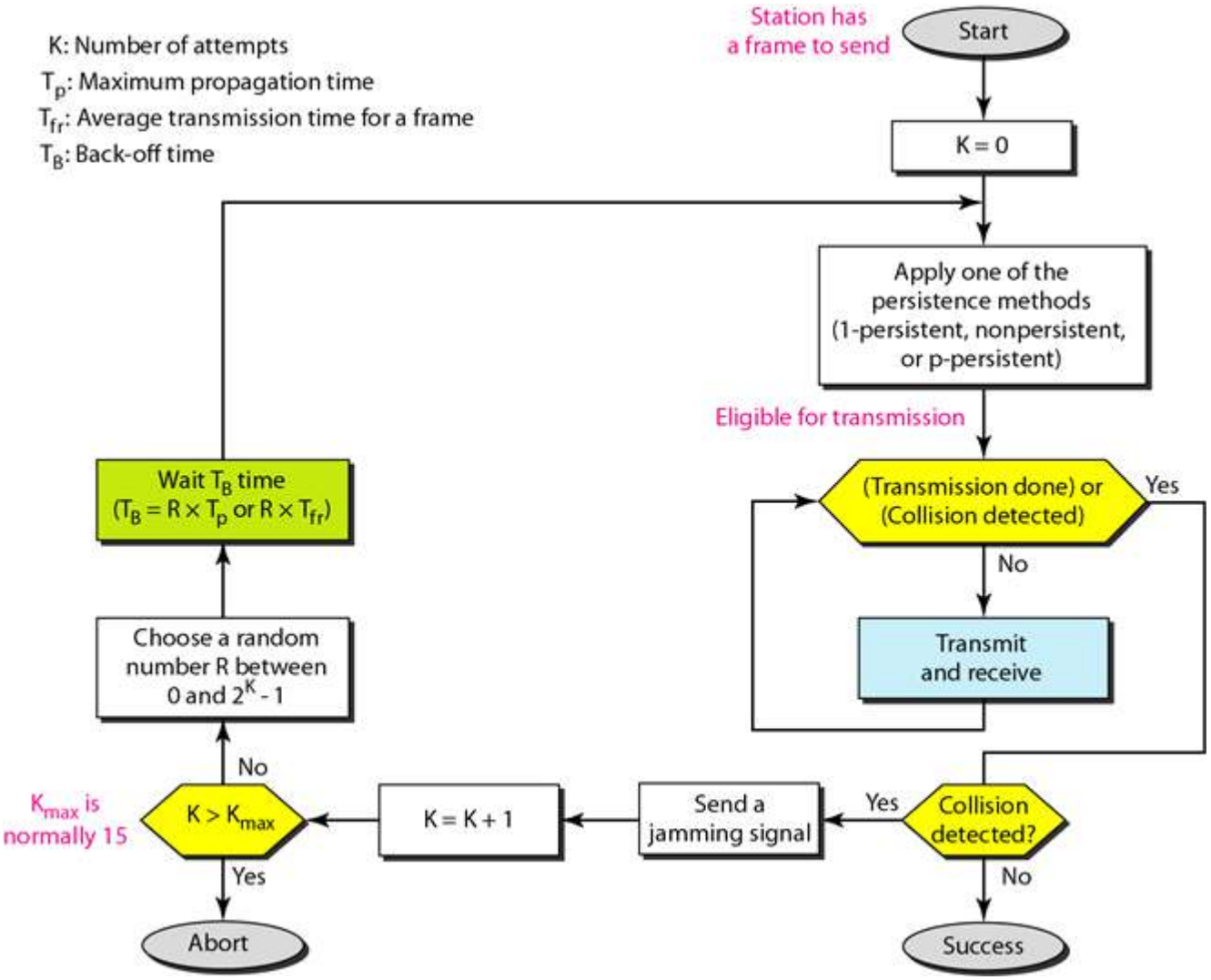




12.1.3.b Flow Diagram for CSMA/CD

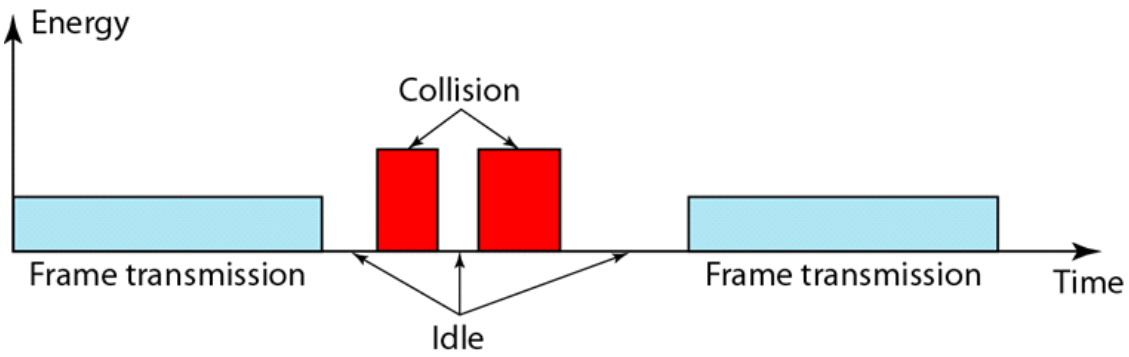
Monday, April 13, 2015 8:22 AM

ALOHA: first send entire frame, then wait for ACK
CSMA/CD: send and detect simultaneously. No ACK is needed.



12.1.3.c Energy Level in CSMA/CD

Monday, April 13, 2015 8:23 AM



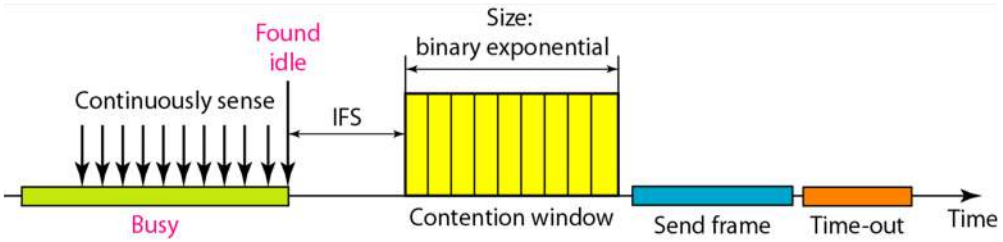
There are three energy levels on a channel: zero, normal, abnormal.

- **Zero:**
  - Channel is idle
- **Normal:**
  - A station has successfully captured the channel and is sending its frame.
- **Abnormal:**
  - Collision. --> Energy level is twice of normal.

Monitoring the channel is simply measuring the energy level on the channel.

However, the above method only works for **wired** networks. Wireless is different.

Figure 12.16 *Timing in CSMA/CA*



CSMA/CD: a collision will almost double the energy level.
This applies in wired networks.
However, in wireless networks, the case is different:
1)A collision adds only 5%~10% percent additional energy
2)Hard to send and monitor at the same time over the same frequency band.

12.27

12.2 Controlled Access

Monday, April 13, 2015 8:28 AM

The stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations.

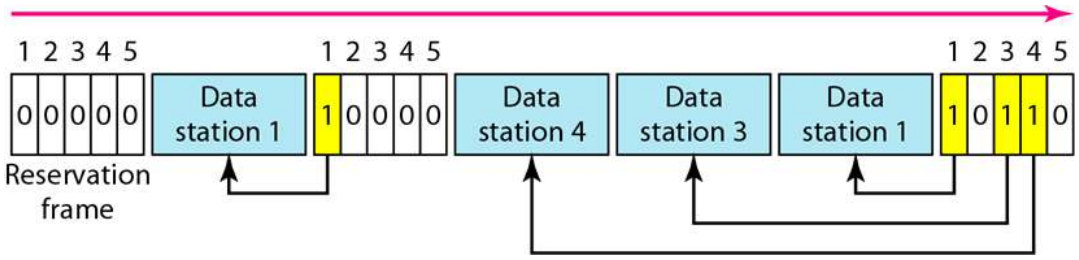
Topics discussed in this section:

Reservation
Polling
Token Passing

12.2.1 Reservation Access Method

Monday, April 13, 2015 8:31 AM

- A station needs to make a reservation before sending data.
- Time is divided into intervals.
  - In each interval, a reservation frame precedes the data frame sent in the interval.



If there are N stations in the system, there are exactly N reservation minislots in the reservation frame. Each minislot belongs to a station.

When a station needs to send a frame, it makes a reservation in its own minislot. The stations that have made reservations can send their data frames after the reservation frame.

In the diagram above, 5 stations, 5 minislot reservation frame.

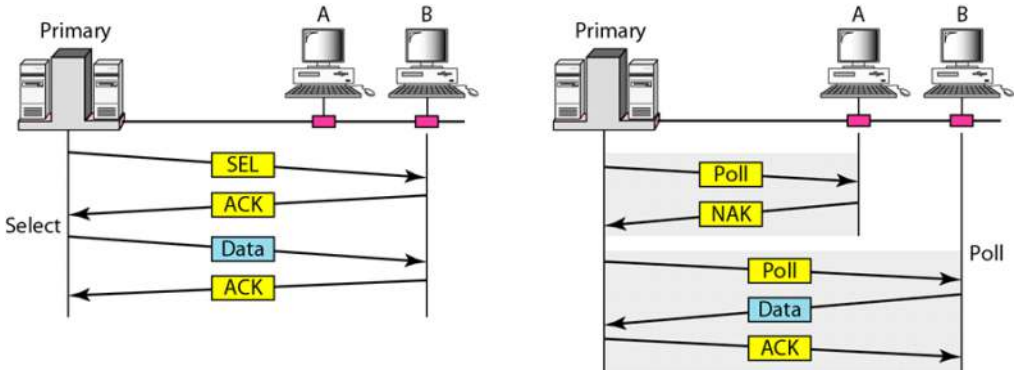
- At first interval,
  - Only stations 1,3,4 have made reservation.
- In second interval, only station 1 has made a reservation.

(Diagram in book/above might be backwards?)

12.2.2 Polling

Monday, April 13, 2015 8:35 AM

- Polling works with topologies in which one device is designated at the primary station, and the others are secondary stations.
- All data exchanges must be made through the primary device even when the ultimate destination is a secondary device.
  - The primary device controls the link, the secondary devices follow its instructions.
  - It is up to the primary device to determine which device is allowed to use the channel at a given time.
- The primary device, therefore, is always the initiator of a session.



If the **primary wants to receive** data,

- Ask the secondaries if they have anything to send --> **poll function**

If the **primary wants to send** data,

- Tells the secondaries to get ready to receive --> **select function.**

Select

- The primary alerts a secondary that data is coming.
- Needs an ACK from the secondary so that the receiver is ready. (The ACK is in response to a SEL frame).
- SEL is like "YOU'RE GONNA GET DATA WHEN YOU'RE READY. YOU READY BRO?"

Poll

- Primary asks everyone in turn if they have something to send. If NAK, then ask the next station.

12.2.3 Token Passing

Monday, April 13, 2015 8:43 AM

The stations are organized in a logical ring. Think of this as a doubly-linked list. Every node has a *prev and *next.
In Token passing, this is called the *predecessor* and *successor.* The current station is the one accessing the network.

A special packet called a **token** circulates through the ring. A station who has the token can access the channel and send data.

- When a station has data to send, it waits till it gets the token. (from *prev).
- When a station is done sending data, it passes the token along to *next.

Considerations

We need some token management.

- Token possession time must be limited.
- The token must be monitored to make sure it is not lost or destroyed.
- Need to assign priorities to the stations and to the types of data being transmitted.
- Token management is needed to make sure low-priority stations release the token to high-priority stations.

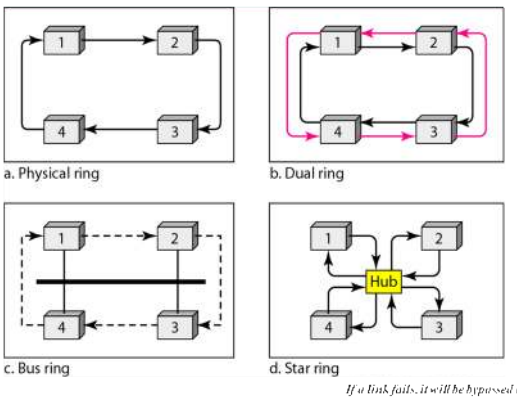
Logical vs. Physical Rings

The stations do not have to be physically connected in a ring.

In a **physical ring topology**, a single link failure brings down the whole network.

- **Dual-ring**: Auxiliary ring operates in reverse direction. Used for emergencies only. If one link fails, the secondary ring and part of the main ring form a weird-shaped "backup system"
- **Bus ring (token bus).** Stations connected to a bus. Logically arranged as a ring, however. When a station has finished sending data, inserts successor address into the token. Only the station with matching token destination address gets the token.
    - Token Bus LAN uses this.
- **Star-ring.** Physical topology is a star. There is a hub. The wiring inside the hub makes a ring. This topology is used by **IBM Token Ring**.

Figure 12.20 *Logical ring and physical topology in token-passing access method*



If a link fails, it will be bypassed by the hub

12.31

Lecture 6: Ethernet

Monday, April 13, 2015 8:59 AM

Chapter 13 of the textbook.

13.1 IEEE Standards

Monday, April 13, 2015 9:00 AM

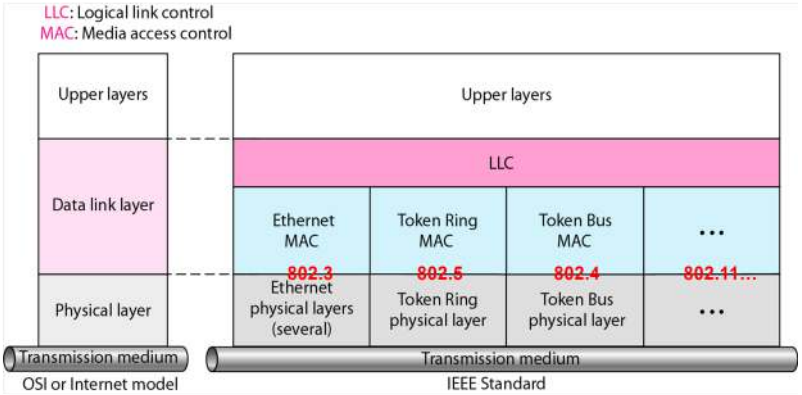
In 1985, IEEE had Project 802 to set standards to enable intercommunication among equipment from different manufacturers.

Topics discussed in this section:

Data Link Layer

Physical Layer

Figure 13.1 *IEEE standard for LANs*



Data Link Layer

Logical Link Control (LLC)

- Earlier, we discussed Data Link Control. (which handles framing, flow control, and error control).
  - Project 802, all of these are collected into one sublayer called the Logical Link Control.
  - Framing is handled in the LLC sublayer and the MAC sublayer.
- LLC provides one single data link control protocol for all IEEE LANs. LLC is different from MAC, which provides different protocols for different LANs. A single LLC protocol provides interconnectivity between different LANs because it makes the MAC sublayer transparent.

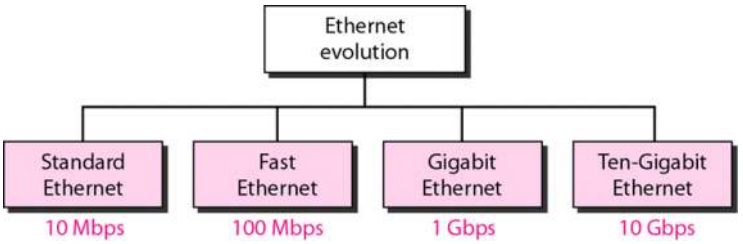
Media Access Control (MAC)

- In Lecture 5 (Chapter 12) we discussed multiple access methods such as random access (and others).
- Project 802 has a sublayer called MAC that defines the specific access method for Ethernet LANs and token-related mathods.
- In contrast to LLC, the MAC sublayer contains a number of distinct modules, each defining an access method and framing format specific to the corresponding LAN protocol.

13.2 Standard Ethernet

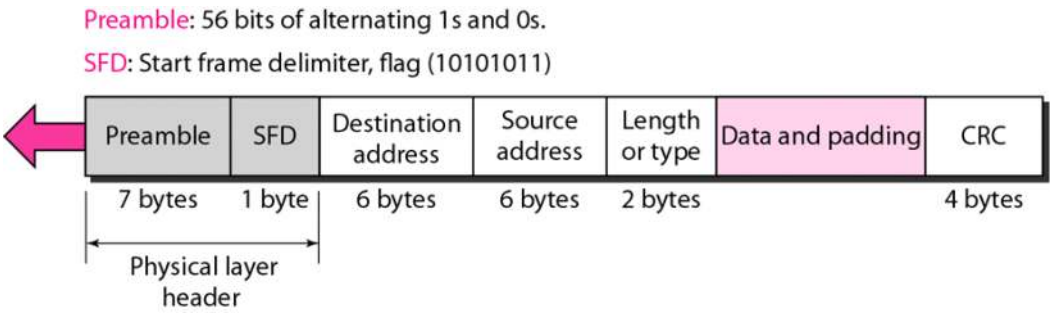
Monday, April 13, 2015 9:14 AM

Standard Ethernet originated at PARC in 1976.
Since then, there were several generations.



MAC Sublayer

The MAC sublayer governs the operation of the access method. It also frames data received from the upper layer and passes them onto the physical layer.



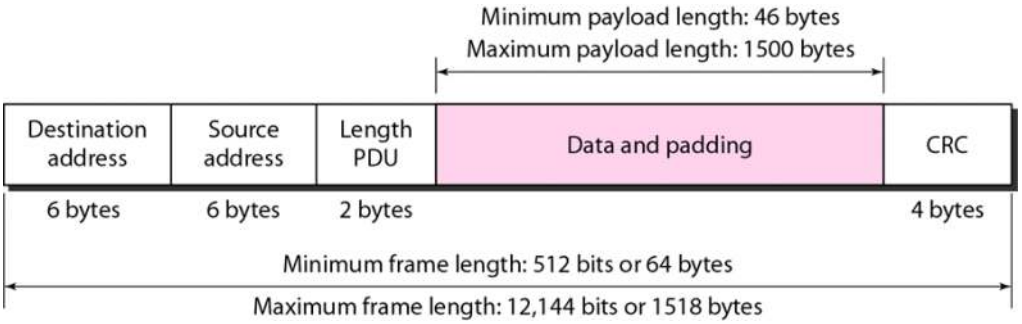
Frame Format

- **Preamble:**
  - 7 bytes of alternating 0s and 1s. Alerts the receiving system to the coming frame, enables it to sync input timing. The pattern provides only an alert and a timing pulse. The 56-bit pattern allows the receiver to miss some bits at the beginning of the frame.
  - Actually added at physicl later, not formally part of the frame.
- **Start Frame Delimiter (SFD):**
  - Signals the beginning of the frame. (Bits 10101011).
  - **Last Chance to sync**
  - The last 2 bits alerts the receiver that the next field is the destination address.
- **Destination Address (DA):**
  - 6 bytes. Contains address of destination stations or stations to receive the packet.
- **Source Address (SA):**
  - 6 bytes.
- **Length or Type**
  - Defined as a type field or length field.
  - Original Ethernet used this as a type field to define the upper layer protocol using the MAC frame.
  - IEEE Ethernet used it as a length field to determine **number of bytes in data field.**
    - Both uses common today.
- **Data.**
  - Contains the data encapsulated from the upper-layer protocols.
  - Minimum 46, Maximum 1500 bytes.
- **CRC**
  - CRC-32 check.

13.2.1 Frame Length

Monday, April 13, 2015 9:24 AM

There are some restrictions on the min/max lengths of the frame.



Minimum Length:

- Required for correct operation of CSMA/CD.
- 512 bites / 64 bytes.
  - Part of this is the header/trailer. (18 bytes)
  - Minimum length of data is 64-18 = 46 bytes. If the data is shorter than this, **padding** is added to make up the difference.

Maximum Length:

- The standard defines the max length of a frame (without preamble and SFD) as 1518 bytes.
  - Subtract the 18 bytes of header/trailer to get **max length of payload = 1500 bytes.**
- Historical reasons for this:
  - Memory used to be expensive, reduce buffer sizes.
  - Prevents one station from monopolizing the medium.

Frame length:
Minimum: 64 bytes (512 bits)
Maximum: 1518 bytes (12,144 bits)

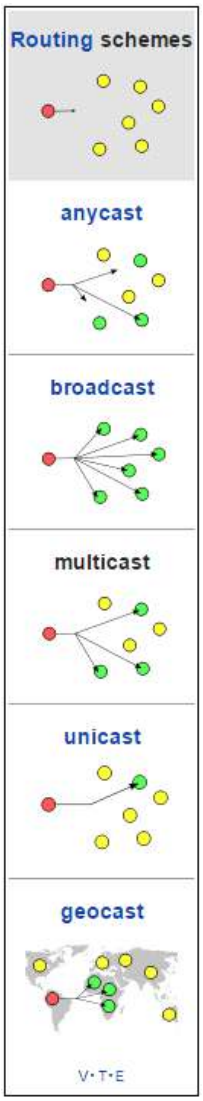
13.2.2 Addressing

Monday, April 13, 2015 9:29 AM

Each station has it's own Network Interface (NIC) Card providing the station with a 6-byte physical address.

06 : 01 : 02 : 01 : 2C : 4B

6 bytes = 12 hex digits = 48 bits



Unicast, Multicast, Broadcast Addresses

- A source address is always unicast. (comes from one station).
- Destination can be unicast, multicast, or broadcast.
- **Unicast Destination:** One recipient. (Senders:receiver is one-to-one).
- **Multicast Destination:** Group of addresses (Senders:receivers many-to-many)
- **Broadcast:** Recipients are all stations on the LAN.

The least significant bit of the first byte defines the type of address. If the bit is 0, the address is unicast; otherwise, it is multicast.

The broadcast destination address is a special case of the multicast address in which all bits are 1s.

Examples

Define the type of the following destination addresses:

a. 4A:30:10:21:10:1A b. 47:20:1B:2E:08:EE c. FF:FF:FF:FF:FF:FF

Solution

To find the type of the address, we need to look at the second hexadecimal digit from the left. If it is even, the address is unicast. If it is odd, the address is multicast. If all digits are F's, the address is broadcast. Therefore, we have the following:

a. This is a unicast address because A in binary is 1010.

b. This is a multicast address because 7 in binary is 0111.

c. This is a broadcast address because all digits are F's.

Example

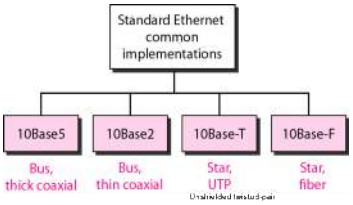
Show how the address 47:20:1B:2E:08:EE is sent out on line.

Solution

The address is sent left-to-right, byte by byte; for each byte, it is sent right-to-left, bit by bit, as shown below:

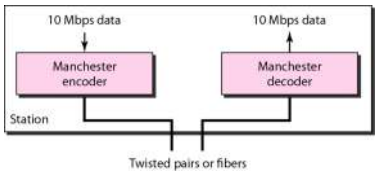
13.3.3 Physical Layer

Monday, April 13, 2015 9:38 AM



Standard Ethernet defines several physical layer implementations.

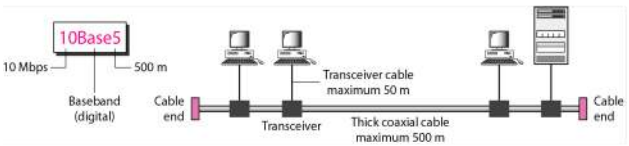
Encoding and Decoding



All standard implementations use digital signalling (baseband) and 10Mbps. Manchester encoding is self-synchronous, providing a transition at each bit interval.

10Base5 Implementation

Figure 13.10 *10Base5 implementation*



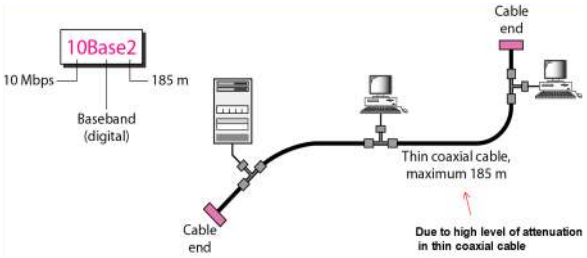
Thick cable: was the first Ethernet specification to use a bus topology. Stiff to bend with your hand.

Transceiver: responsible for transmitting, receiving, and detecting collisions. It has separate paths for transmitting to and receiving from the station.

Collisions only happen in the coaxial cable.

10Base2 Implementation

Figure 13.11 *10Base2 implementation*

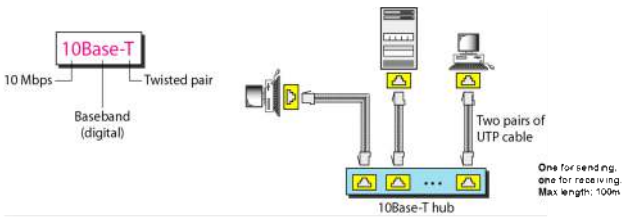


Because the cable can be bent to pass very close to the stations

Transceiver is normally part of the network interface card (NIC), which is installed inside the station.
Thin cable: cheaper and more flexible.

10Base-T Implementation

Figure 13.12 *10Base-T implementation*



10Base-F Implementation

Figure 13.13 *10Base-F implementation*

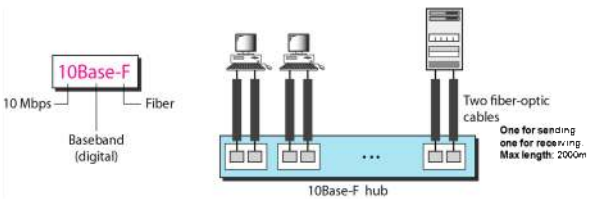


Table 13.1 *Summary of Standard Ethernet implementations*

Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Media	Thick coaxial cable	Thin coaxial cable	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m
Line encoding	Manchester	Manchester	Manchester	Manchester

13.3 Changes In The Standard

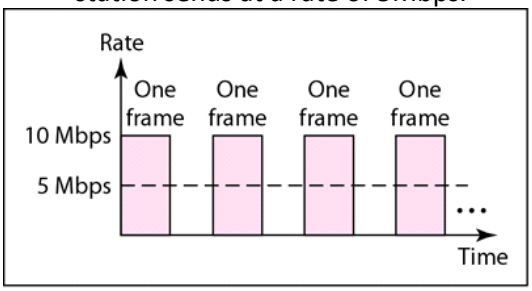
Monday, April 13, 2015 9:45 AM

10-Mbps Standard Ethernet has changed before moving to higher data rates. These changes have made it compatible with other high-data-rate LANs.

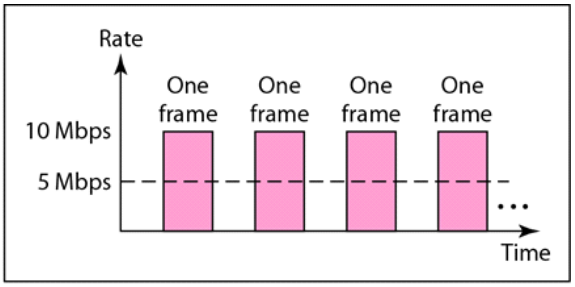
Sharing Bandwidth

In an unbridged Ethernet network, the total capacity (10Mbps) is shared among all stations with a frame to send. The stations share the bandwidth of the network.

- If only one station has frames to send, it benefits from the total capacity (10Mbps)
- If two stations have a lot of frames to send, they probably alternate in usage. On average, each station sends at a rate of 5Mbps.



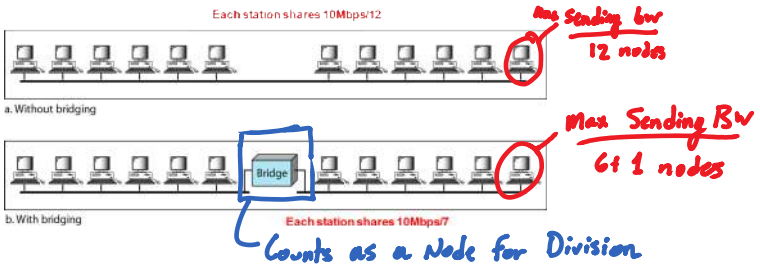
a. First station



b. Second station

The Bridge

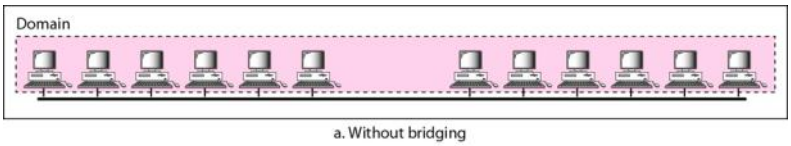
We can divide the network into two or more subnetworks. Bandwidth-wise, each one is independent.



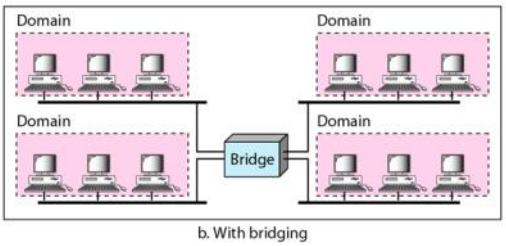
Separating Collision Domains

The bridge separates the collision domain.

Without bridging, 12 stations contend for access to the medium. With bridging, only 3 stations contend for access to the medium.



a. Without bridging



b. With bridging

13.3.1 Bridges

Monday, April 13, 2015 10:10 AM

A bridge operates in both the Physical and Data Link Layer.

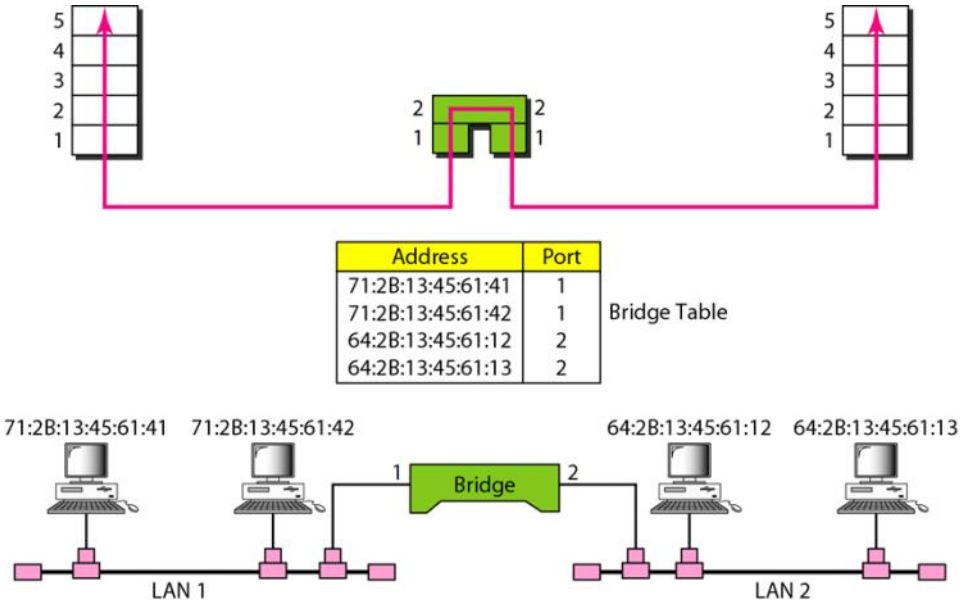
- **Physical Layer Device:**
  - Regenerates the signal it receives.
- **Data Link Layer Device:**
  - Check the physical (MAC) addresses (source and destination contained in the frame.

Filtering

- A bridge has filtering capability. It can check the destination address of a frame and decide if the frame should be forwarded or dropped. If the frame has to be forwarded, the decision must specify the port.
- A bridge has a table that maps addresses to ports.

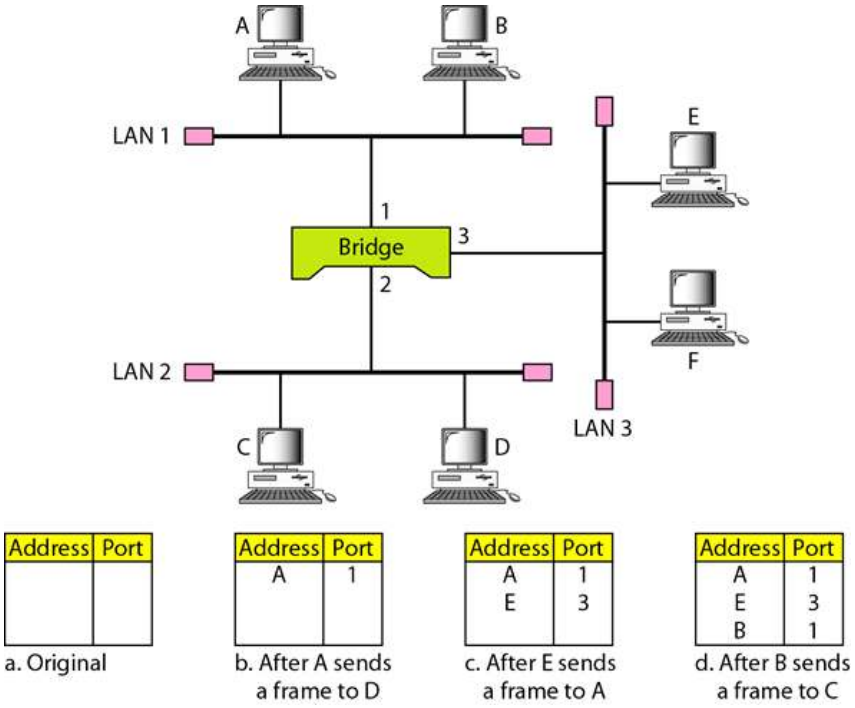
A bridge has a table used in filtering decisions.

A bridge does not change the physical (MAC) addresses in a frame.



13.3.2 Learning Bridges

Monday, April 13, 2015 10:23 AM



Static Bridges

The earliest bridges has static forwarding tables. The sysadmin would manually enter each table entry during bridge setup. If a station was added or deleted, or even when MAC addresses changed, the table would have to be modified.

Learning Bridges

A better solution is a dynamic table that maps addresses to ports automatically, gradually learning from frame movements.

- Bridge inspects destination and source addresses.
  - Destination is used for forwarding decision (table lookup)
  - Source is used for adding entries to the table and updating.

Procedure

Using the above figure:

1. When station A sends a frame to Station D, the bridge does not have an entry for D or A. The frame goes out from all 3 ports, flooding the network.
   a. However, the bridge has learned that Station A is connected to port 1. All future traffic to station A goes through port 1.
2. Afterwards, Station E sends a message to Station A. The Bridge knows A is on port 1. It also learns E is on Port 3.
3. The process continues.

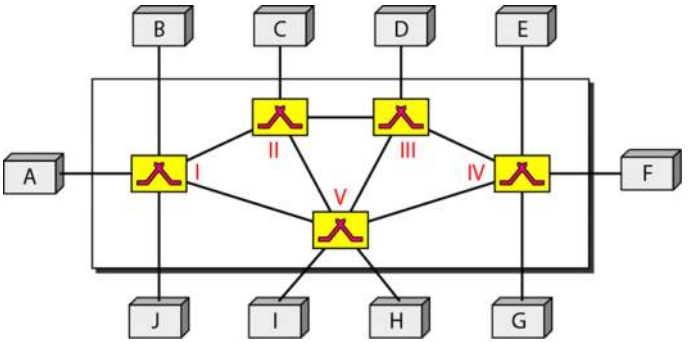
Lecture 8: Packet Switched Networks

Monday, April 13, 2015 10:34 AM

8.1 Switching

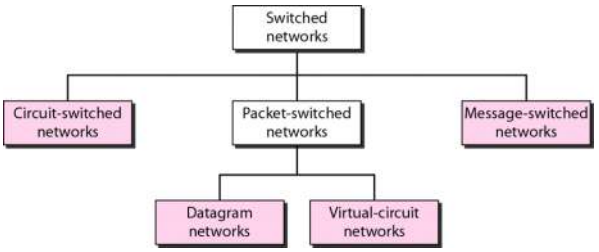
Monday, April 13, 2015 10:38 AM

When we have multiple devices, we have the problem of making one-to-one connections. One solution is either a mesh topology or a star topology. These methods are very wasteful for large networks. The number of links and lengths is very high. Most links would be idle.



A better solution is switching.
Switches: Devices capable of creating temporary connections between two or more devices linked to the switch.

Taxonomy of Switched Networks



- **Circuit-switched network:**
  - Switches connected by links. A connection between two stations is a dedicated path made of one or more links.
  - Each connection uses only one dedicated channel on each link. These channels are divided either by Time Division Multiplexing or Frequency Division Multiplexing.
- **Message switching:**
  - Each switch stores the whole message and forwards it to the next switch.
- **Packet-Switched Networks:**
  - **Virtual-circuit networks**
  - **Datagram Networks**

8.2 Datagram Networks

Monday, April 13, 2015 10:52 AM

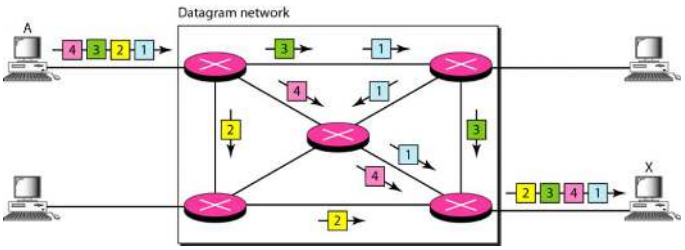
If a message goes through a packet-switched network, it needs to be divided into **packets.**

- The packet size is determined by the network and governing protocol.
- In **Packet-switching:**
  - There is no resource allocation for a packet.
    - -> No reserved bandwidth on links,
    - No scheduled processing time for each packet
  - Resources allocated on First-Come, First-Served basis.
    - When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed.
  - Each packet is treated independently of all others, even if it's part of a multi-packet transmission.
    - Packets in this approach are referred to as **datagrams.**

In a packet-switched network, there is no resource reservation; resources are allocated on demand.

Routers

Switches in a datagram network are typically referred to as **routers**.



A switch in a datagram network uses a routing table that is based on the destination address.

The destination address in the header of a packet in a datagram network remains the same during the entire journey of the packet.

Above, all four packets belong to the same message, but may travel different paths to their destinations. This approach can cause packets to arrive out of order. Packets may also be lost or dropped.

- It is typically the responsibility of an upper-layer protocol to reorder the datagrams or ask for lost datagrams before passing them along to the application.

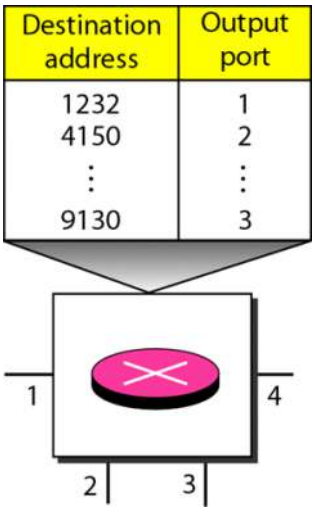
Connectionless Networks

- **Connectionless:** the switch does not keep any information about the connection state.
  - No setup/teardown phases. Each packet is treated the same by a switch regardless of its source or destination.

Routing Tables

Each switch has a routing table which is based on the destination address. The routing tables are dynamic and updated periodically.

- The destination addresses and the corresponding forwarding output ports are recorded in the tables.
  - This is different from the table of a circuit-switched network in which each entry is created when the setup phase is completed, and deleted when the teardown is complete.



Destination Addresses

Every packet in a datagram network carries a header that contains, among other info, the destination address of the packet. When the switch receives a pcaket, the destination address is examined, the routing table is consulted. This address remains the same during the entire journey.

8.2.1 Efficiency and Delay

Monday, April 13, 2015 11:12 AM

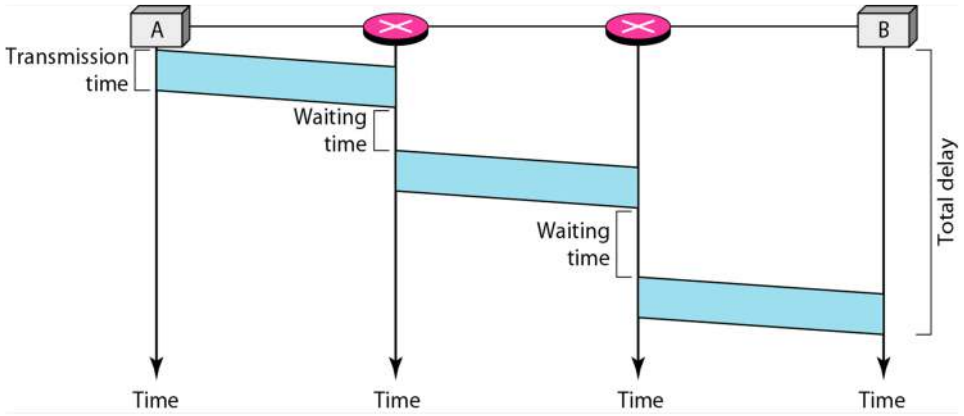
Efficiency

- The efficiency of a datagram network is better than circuit-switched network.
  - Resources allocated only when packets need to be transferred.

Delay

- Each packet may experience a wait at a switch before it is forwarded.
- Since not all packets in a message travel through the same switches, the delay is not uniform for the packets in a message.

Figure 8.9 *Delay in a datagram network*



Total delay = 3 transmission times + 3 propagation times + 2 waiting times

Datagram Networks in the Internet

Switching in the Internet is done by using the datagram approach to packet switching at the network layer.

8.3 Virtual-Circuit Networks

Monday, April 13, 2015 11:16 AM

A virtual-circuit network is a cross between a circuit-switched network and a datagram network.

1. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
2. Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a packet-switched network.
3. Data are packetized and each address carries an address in the header. However, the address has local jurisdiction (it defines what should be the next switch and the channel on which the packet is being carried), not end-to-end jurisidction.
    a. How do intermediate switches know where to send the packet?
        i. Virtual-circuit identifiers
4. All packets follow the same path established during the connection.
5. Virtual-circuit network typically implemented in data link layer. Circuit-switched typically implemented in the physical layer. Datagram network typically in network layer.

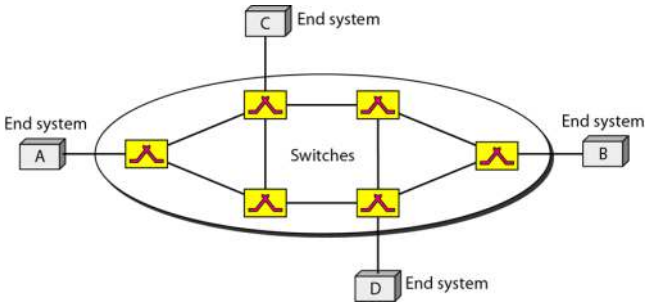
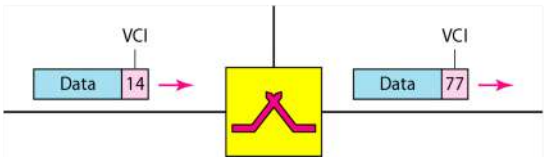


Figure 8.11 *Virtual-circuit identifier*



Addressing

Two types of addressing: global and VCI

- **Global**
    - Unique in the scope of the network, or internationally if part of an international network.
- **Virtual-Circuit Identifier**
    - Small number that only has switch scope. (used by a frame between two switches).
    - When the frame arrives at a switch, it has a VCI. When it leaves, it has a different VCI.

8.3.1 Three Phases: Setup, Transfer, Teardown

Monday, April 13, 2015 11:25 AM

Three phases:

- **Setup**
  - Source and destination use global addresses to help switches make table entries for the connection.
- **Data transfer**

- **Teardown**
  - Source and destination inform the switches to delete the corresponding entry

Data Transfer Phase

To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit.

- Active until the source sends all its frames to the destination. The procedure at the switch is the same for each frame of a message. The process creates a virtual circuit, not a real circuit, between the source and destination.

Setup Phase

A switch creates an entry for a virtual circuit.
Two steps required:

- Setup request
- Setup acknowledgment

Source A needs to create a virtual circuit to B.
Setup Request:

1. Source A sends setup frame to Switch 1.
2. Switch 1 receives setup request frame. It knows a frame going from A to B goes out through port 3.
   a. Assume the switch knows the output port:
   b. Creates an entry in the table for the virtual circuit
   c. The switch assigns the incoming port (1) and chooses the incoming VCI (14) and the outgoing port (3). It does not yet know the outgoing VCI, which is found during the ack step. Switch forwards setup frame through port 3 to switch 2.
3. Switch 2 receives the setup request frame. Same seq of events.
4. Switch 3 receives the setup request frame.
5. Destination B receives the setup frame. If it is ready to receive frames from A, assignes a VCI to the incoming frames that come from A (77).

Figure 8.12 *Switch and tables in a virtual-circuit network*

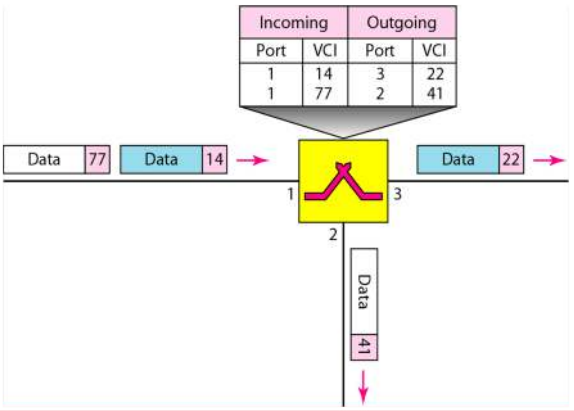
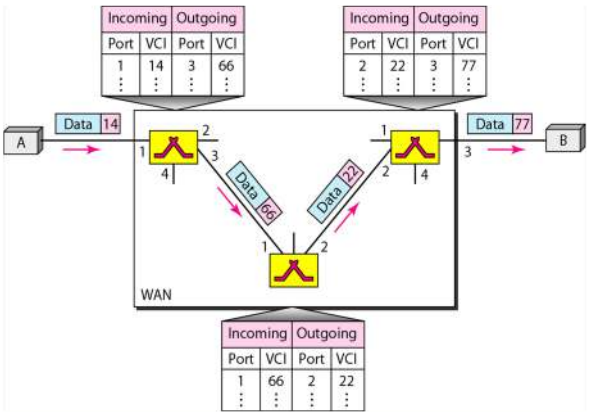


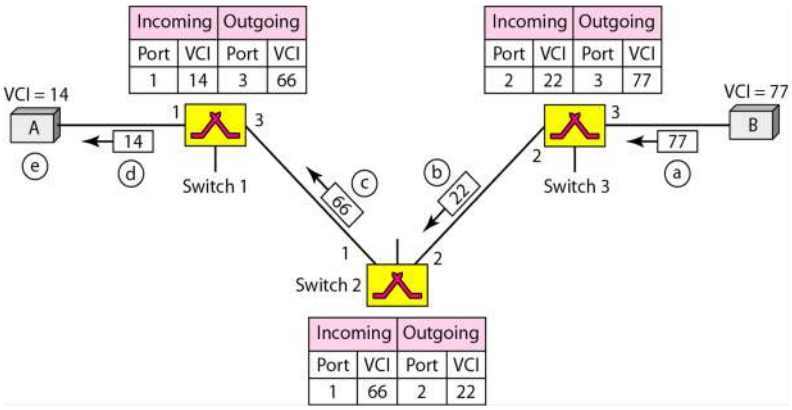
Figure 8.13 *Source-to-destination data transfer in a virtual-circuit network*



8.3.1.a Setup ACK Frames / Teardowns

Monday, April 13, 2015 1:13 PM

Figure 8.15 *Setup acknowledgment in a virtual-circuit network*



Acknowledgment

The acknowledgment frame, completes the entries in the switching tables.

1. The destination sends an ACK to switch 3. This carries the global source and destination addresses so the switch knows which entry in the table is to be completed.
   a. It also carries VCI 77, chosen by the the destination as the incoming VCI for frames from A.
   b. Switch 3 also uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but the outgoing VCI for switch 3.
2. Switch 3 sends an ACK to switch 2 that contains its incoming VCI in the table, chosen in the previous step. Switch 2 uses this as the outgoing VCI in the table.
3. Switch 2 sends an acknowledgment to switch 1 that contains its incoming VCI in the table, chosen in the previous step. Switch 1 uses this as the outgoing VCI at the table.
4. Switch 1 sends an ACK to source A that contains its incoming VCI in the table, chosen in the previous step.
5. The sources uses this as the outgoing VCI for the data frames to be sent to destination B.

Teardown

Source A, after sending all frames to B, sends a special frame called a ***teardown request***. Destination B responds with a teardown confirmation . All switches delete the corresponding entry from their tables.

8.3.2 Efficiency and Delay

Monday, April 13, 2015 11:25 AM

Efficiency

Resource reservation can be made during setup, or on demand during data transfer phase.

Resource reservation during setup:

- Delay for each packet is the same.

Resource reservation during data transfer:

- Each packet may encounter different delays.

Large advantage to VCI:

- Even if resource allocation on demand, the source can check for availability of resources without actually reserving it.
- Like a family who phones a restaurant. The restaurant may not accept reservations (allocation of tables is on demand), but the family can still find out the waiting time.

In virtual-circuit switching, all packets belonging to the same source and destination travel the same path; but the packets may arrive at the destination with different delays if resource allocation is on demand.

Delay

There is a one-time delay for setup, and a one-time delay for teardown. If resources allocated during setup phase, no wait time for individual packets

If 3 packets travelling along:

$$Total\ Delay = \sum transmission\ times + \sum propigation\ times + setup\ delay + teardown\ delay$$

22.3 Unicast Routing Protocols

Monday, April 13, 2015 1:28 PM

A routing table can be *static* or *dynamic.*

Static table: manual entries
Dynamic table : Updated automatically when there is a change somewhere in the internet.

A routing protocol is a combination of rules and procedures that lets routers in the Internet inform each other of changes.

Optimization

A router recieves a packet from a network and passes it to another network. How does a router know to which network to pass a packet?

One approach is to assign a cost for passing through a network. Some simple protocols, such a ***Routing Information Protocol (RIP)*** treat all networks as equals: the cost os passing through a network is the same --> one hop count.

Open Shortest Path First (OSPF): allow the administrator to assign a cost for passing through a network based on the type of service required. A route through a network can have different costs (metrics).

Routers use routing tables to help decide the best route. OSPF allows different tables for different levels of service.

Border Gateway Protocol (BGP) the criterion in the policy, set by the administrator. The policy defines what paths should be chosen.

22.3.1 Forwarding

Monday, April 13, 2015 3:33 PM

Forwarding: to place a packet in its route to its destination.

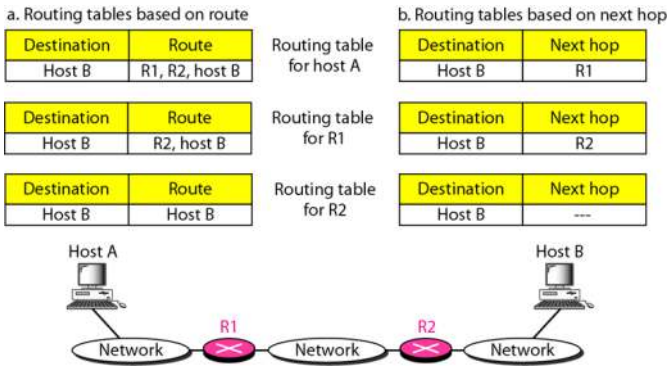
- Requires a host or router to have a routing table.
    - When a host has a packet to send, or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.
        - Impossible in the Internet, since the routing table would be huge.

Forwarding Techniques

Several techniques exist.

Next-Hop Method vs. Route Method.

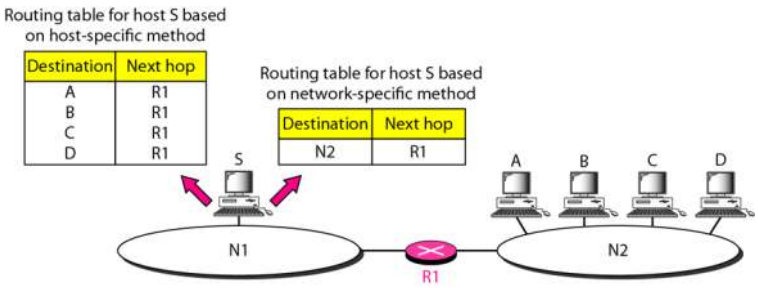
Next-Hop Method: The contents of the routing table is only the address of the next hop, instead of the whole route.



Network-Specific Method vs. Host-specific Method

Instead of having an entry for every destination host connected to the same physical network (host-specific method), we have only one entry that defines the address of the destination network itself.

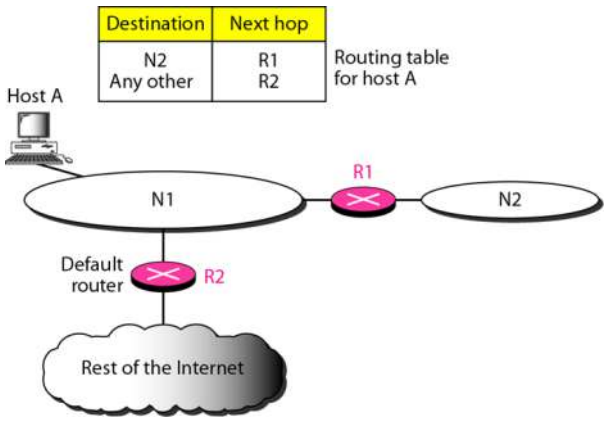
- We treat all hosts connected to the same network as one single entity.



Default Method

At right, host A is connected to a network with two routers. Router R1 routes the packets to hosts connected to network N2. However, for the rest of the internet, router R2 is used.

- Instead of listing all the networks in the entire Internet, host A can have just one entry called the default (typically address 0.0.0.0).



22.3.2 Intra- and Inter- Domain Routing

Monday, April 13, 2015 3:29 PM

An Internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers.

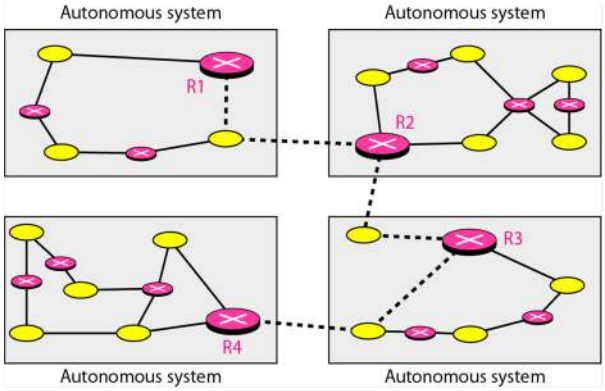
The internet is divided into ***autonomous systems.***

Autonomous system: A group of networks and routers under the authority of a single administration.

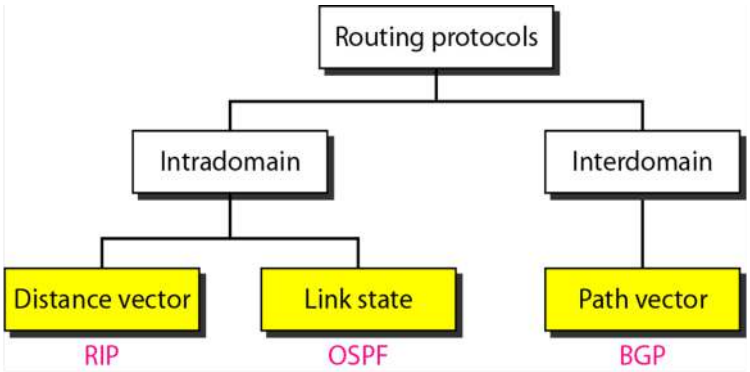
Intra-domain routing: within an autonomous system.

Inter-domain Routing: between autonomous systems.

Figure 22.12 *Autonomous systems*



22.21 Autonomous system: a group of networks and routers under the authority of a single administration.



Routing information protocol Open shortest path first Border gateway protocol

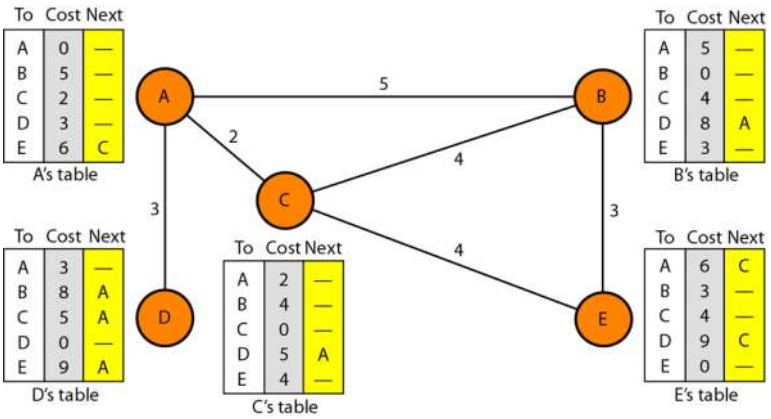
22.3.2.1 Distance Vector Routing

Monday, April 13, 2015 3:32 PM

The least-cost route between any two nodes is the route with minimum distance.

Each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).

In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.



Initialization

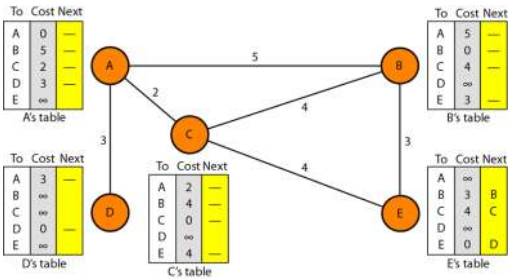
At the beginning, the tables are not initialized. Each node only knows the distance to its immediate neighbors.

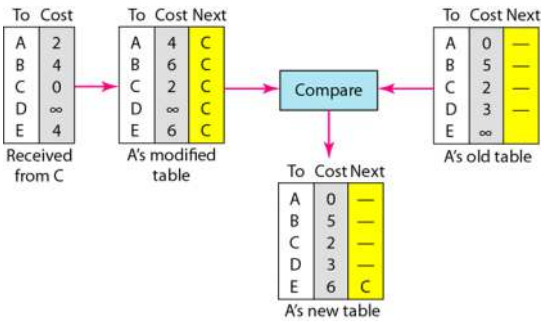
Sharing

The whole idea of distance vector routing is the sharing of information between neighbours. Nodes A and C can improve their routing tables if they help each other.

Problem?

- *How much of a table must be shared with each neighbor?*
  - A node is not aware of its neighbor's table. The best solution is for each node to send its entire table to the neighbor and let the neighbor decide what part to use/discard.
    - The 3rd column (next) is not useful for the neighbor. When the neighbor receives a table, this column needs to be replaced with a sender's name.
    - Sharing means only sharing the first two clolumns.





Updating.

The figure shows how node A updates its routing table after receiving a partial table from Node C.

1. Receiving node adds the cost between itself and the sending node.
2. The receiving node adds the name of the sending node to each row as the 3rd column if the receiving node uses information from any row. The sending node is the next node in the route.
3. Compare each row of the old table to the new one.
   a. Receiving node choses the row with the smaller cost. If tied, keep the old one.
   b. If the next-node entry is the same, the receving node choses the new row.
      i. Suppose node C previously advertized a route to node X with distance 3.
      ii. Suppose now there is no path between C and X; node C advertises this route with distance infinity.
      iii. Node A must not ignore this value even though the old entry is smaller. The old route does not exist anymore. The new route has a distance of infinity.

Infinity + infinity = infinity.

22.3.2.2 When to share

Monday, April 13, 2015 3:59 PM

When does a node send its partial routing table to its immediate neighbors?

It is sent periodically and when there is a change in the table.

Periodic update:

A node sends its routing table, normally every 30s, in a periodic update. Protocol dependent.

Triggered Update

Sends its two-column routing table to its neighbors any-time there is a change in its routing table. The change can result from:

- Node receives a table from a neighbor, resulting in changes in its own table after updating.
- Node detects some failure in the neighboring links which results in a distance change to infinity.

22.3.2.3. Two-node instability

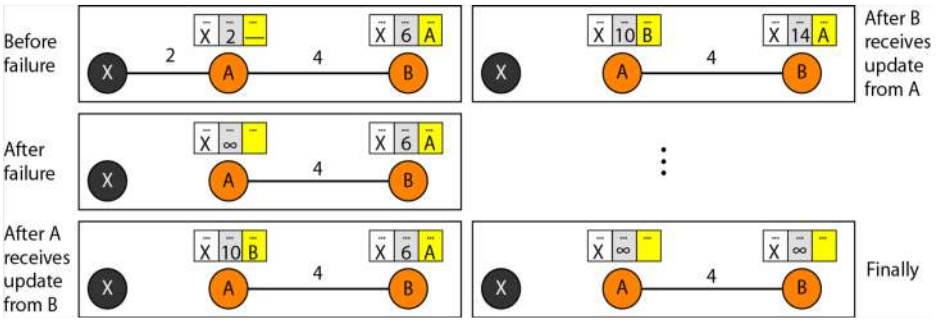
Monday, April 13, 2015 3:57 PM

We know infinity + infinity = infinity.

The modified table shows how to reach A from A via C. If A needs to reach itself via C, it goes to C and comes back, distance of 4.

The only benefit from this updating of node A is the last entry, how to reach E.

Figure 22.17 *Two-node instability*



One solution: redefine infinity to a smaller number, e.g., 16 (cannot used in large networks)

22.31

Two-node Loop Instability

We have 3 nodes. At the beginning, both nodes A and B know how to reach node X. Suddenly, the link between A and X fails.
Node A changes its table. If A can send its table to B immediately, everything is fine.
The system can become unstable if B sends its routing table to A before receiving A's routing table.

- A gets a table from B, assumes B has found a way to X.
- A sends new update to B.
    - B thinks something has been changed around A and updates its tables.
    - ***<u>The cost of reaching X increases gradually until it reaches infinity.</u>***
        - Until we get to distance=infinity, the system is unstable.
            - A thinks A->X is via B. B thinks B->X is via A.
                - If A gets a packet for X it goes to B then comes back to A.
                - If B gets a packet for X, goes to A and comes back to B.
                    - Packets bounce around.

Solution

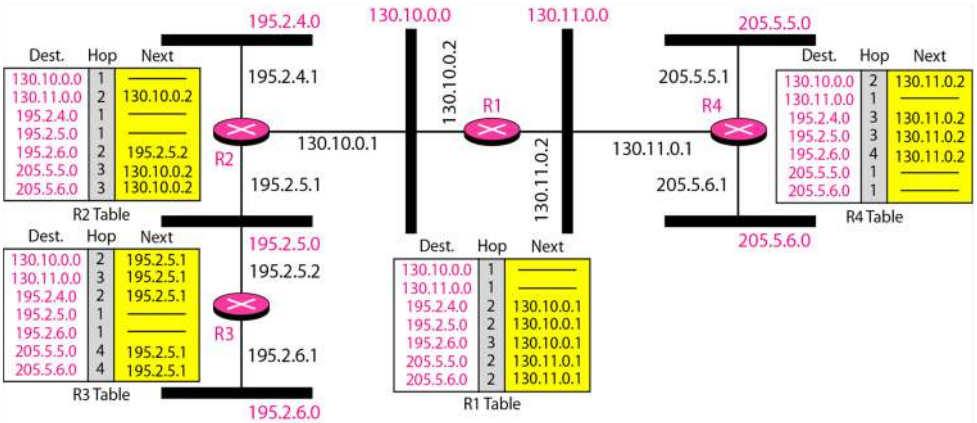
Define infinity as a smaller number, such as 16 or 100. If infinity=100, the system will be stable in less than 20 updates.
Most Distance Vector Protocols define distance between nodes to be 1, Infinity = 16. This means the size of the network, in any direction cannot exceed 15 hops.

22.3.3 Routing Information Protocol (RIP)

Monday, April 13, 2015 4:06 PM

- Distance:
  - Number of links to reach destination: hop count
- Infinity = 16
  - No route can have > 15 hops
- Next-node column defines the address of the next router.
  - The destination in a routing table is a network, meaning the first column is a network address.

Figure 22.19 *Example of a domain using RIP*



22.3.4 Link State Routing

Monday, April 13, 2015 4:10 PM

22.3.4.1 Dijkstra's Algorithm

Monday, April 13, 2015 4:12 PM

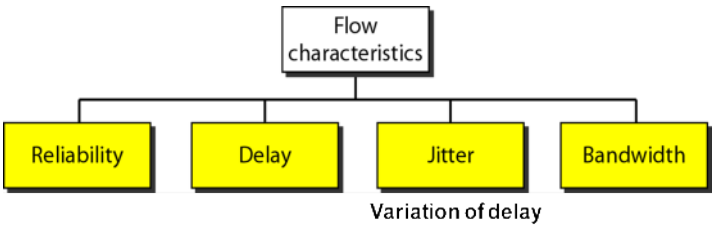
22.3.4.2 Assignment Solutions

Monday, April 13, 2015 4:12 PM

24.5 Quality of Service

Monday, April 13, 2015 4:14 PM

Quality of Service (QoS) has fuzzy definitions. We can informally define the QoS as something a flow seeks to attain.



Flow Characteristics

- Reliability
- Delay
- Jitter
- Bandwidth

- **Reliability**
  - Lack of reliability means losing a packet or acknowledgment, which entails retransmission.
  - Some things need to be more reliable than others.
- **Delay**
  - Source-to-destination delay is important. Some things, like video calls can tolerate less than others.
- **Jitter**
  - Variation in delay.
  - High jitter means the difference between delays is large. Low jitter means the variation is small.
  - **Assume 4 packets departing at times 0, 1, 2, and 3. If they arrive at times 20, 21, 22, and 23, then delay is 20 for each packet. Jitter is zero. It is acceptable. If they arrive at times 21, 23, 21, and 28. Delay is 21, 22, 19, and 25 for the four packets. It is unacceptable.**

- **Bandwidth**
  - Video conferencing -> millions bps
  - Email -> may not even reach a million total bits.

24.6 Techniques to Improve QoS

Monday, April 13, 2015 4:14 PM

There are four common methods of improving QoS:

1. Scheduling
2. Traffic Shaping
3. Admission Control
4. Resource reservation

Scheduling

Packets from different flows arrive at a switch/router for processing. A good algorithm treats them differently and fairly.

FIFO Scheduling

Priority Scheduling

- Packets are first assigned to a priority class.
- Each class has its own queue.
- High priority packets processed first.
- The system does not stop serving a queue until its empty.

Weighted Fair Scheduling

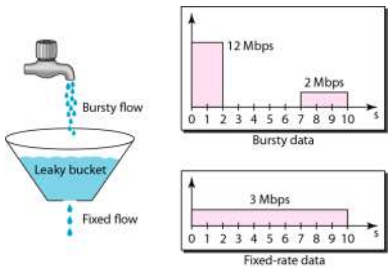
- Packets assigned to different classes and admitted to different queues.
- The wheel moves round. 3 packets at a time from hi-prio, 2 from mid, 1 from low. Just keep going round.

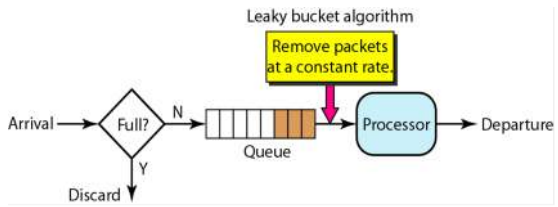
Traffic Shaping

Control the amount and rate of traffic sent to the network.

Leaky Bucket

Water leaks out of a bucket. The rate at the bottom is steady. The input rate can vary.





FIFO with fixed service rate

Token Bucket allows idle hosts to accumulate credits for priority flow.

Figure 24.16 *FIFO queue*

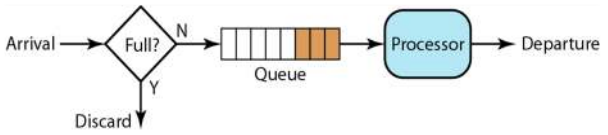


Figure 24.17 *Priority queuing*

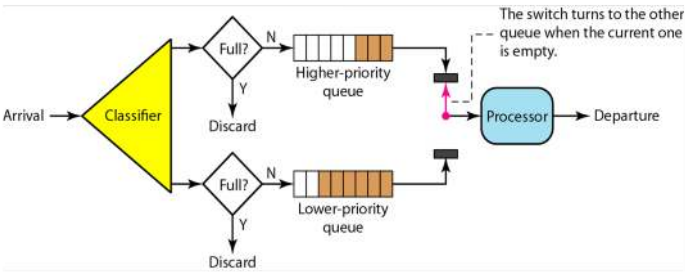
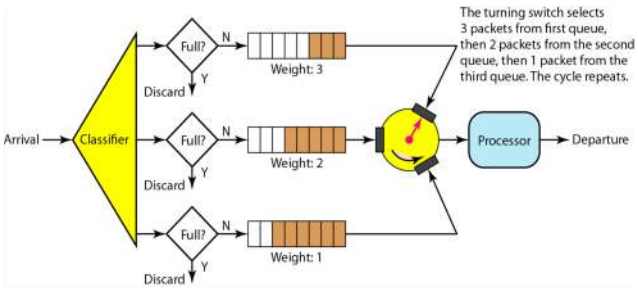


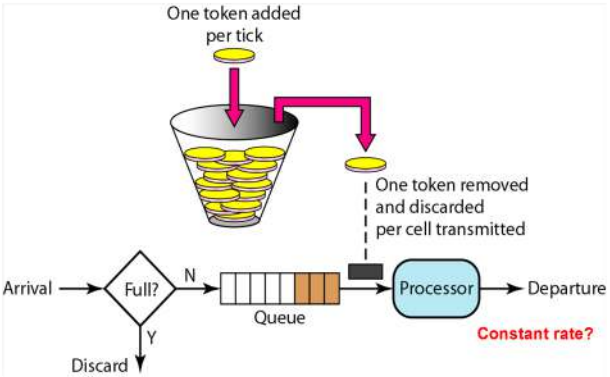
Figure 24.18 *Weighted fair queuing*



A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

The token bucket allows bursty traffic at a regulated maximum rate.

Figure 24.21 *Token bucket*



24.7 Integrated Services

Monday, April 13, 2015 4:15 PM

IntServ, Integrated Services is a flow-based QoS model designed for IP.

Integrated Services is a flow-based QoS model designed for IP.

24.8 Differentiated Services

Monday, April 13, 2015 4:15 PM

Lecture 9: Addressing

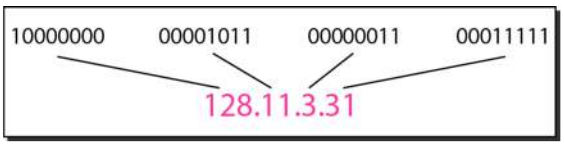
Monday, April 13, 2015 2:07 PM

19.1 IPv4 Addresses

Tuesday, April 14, 2015 11:25 AM

An IPv4 is a 32 bit address that uniquely and univerally defines the connection of a device to the internet.

- **They are unique in the sense that each address defines one, and only one, connection to the Internet.**
- **Two devices on the Internet can never have the same address at the same time.**
  - **By using some strategies, an address may be assigned to a device for a time period and then taken away and assigned to another device.**
- **If a device operating at the network layer has m connections to the Internet, it needs to have m addresses.**
  - **A router is such a device.**
- **They IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.**



Dotted-decimal notation is compact and easier to read, in which each number is a value ranging from 0 to 255

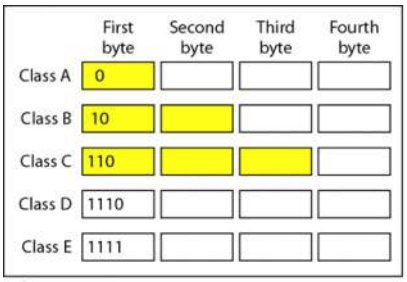
The address space of IPv4 is 2^{32} or 4,294,967,296.

19.1.1 Classful Addressing

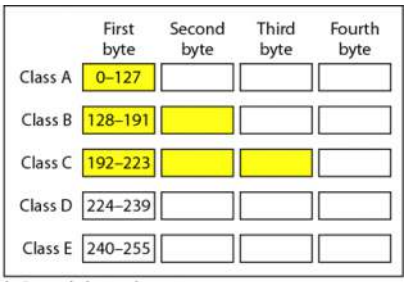
Tuesday, April 14, 2015 11:29 AM

At its inception, IPv4 used classes for addressing. It is obsolete. Divided into 5 classes: A, B, C, D, E.

We can find the class of the address based on binary or dotted decimal notation.



a. Binary notation



b. Dotted-decimal notation

Classes and Blocks

Each class is divided into a fixed number of blocks, each with fixed size.
Because of the block size, there are many addresses wasted.
Class D was designed for multicasting. Each address in this group is used to define one group of hosts on the internet.
Class E was designed for future use. These addresess are wasted too.
CIDR Notation

- mask in the form "/n" where n can be 8, 16, or 24 in classful addressing.
- This notation is also called slash notation or **Classless Interdomain Routing (CIDR) notation.**

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

The Netid is in color. The hostid is in black.

19.1.2 Class-less Addressing

Tuesday, April 14, 2015 11:37 AM

Class-less addressing

No classes, but the addresses are still granted in blocks.

Address Blocks

In classless addressing, when an entity (small or large) needs to be connected to the Internet, it is granted a block of addresses.

The size varies: household may be given 2 addresses. An organization may get thousands.

Restrictions

1. The addresses in a block most be contiguous, one after another,
2. The number of addresses in a block must be a power of 2
3. The first addresses must be evenly divisible by the number of addresses.

Mask

A **mask** is a 32-bit number in which the n leftmost bits are 1's and the 32-n rightmost bits are 0s. In classless addressing, the n value can take any number from 0 -> 32.

n bits in netid; 32-n bits in hostid. Total address: 2^{32-n}

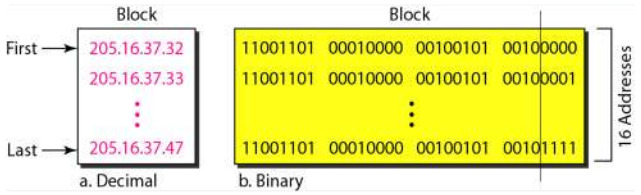
→ 32-n = $\log_2$ (number of addresses in the block).

For the block of addresses, the rightmost 32-n bits varies from all 0s to all 1s.

The **first address in the block** can be found by setting the rightmost 32 − n bits to 0s.

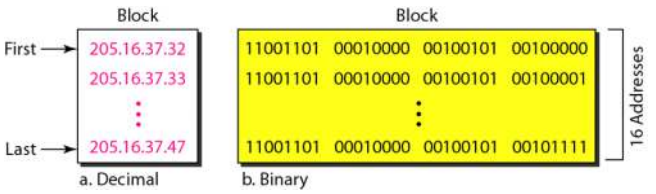
The **last address in the block** can be found by setting the rightmost 32 − n bits to 1s.

Figure 19.3 *A block of 16 addresses granted to a small organization*



16 addresses: the last four bits change from 0000 to 1111

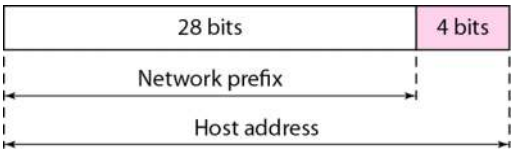
In IPv4 addressing, a block of addresses can be defined as x.y.z.t /*n* in which x.y.z.t defines one of the addresses and the /*n* defines the mask.



The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

19.1.3 IPv4 Address Hierarchy

Tuesday, April 14, 2015 11:47 AM



Each address in the block can be considered as a two-level hierarchical structure: the leftmost *n* bits (prefix) define the network; the rightmost 32 − n bits define the host.

Subnetting

An organization that is granted a large block of addresses may want to create subnets.

The rest of the world still sees the organization as one entity.

- The Router sends incoming messages to the appropriate subnets.

The organization needs to create small subblocks of addresses.

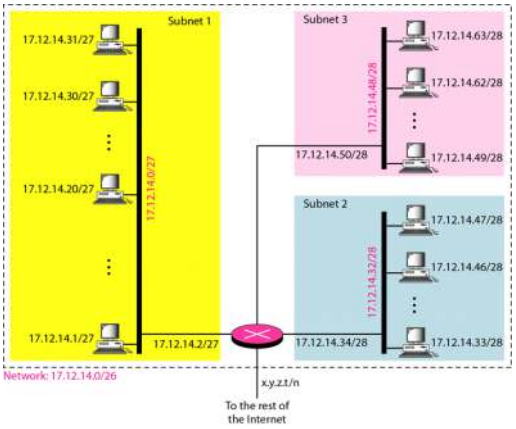
The organizaztion already has a mask --> it needs to create a **subnet mask.**

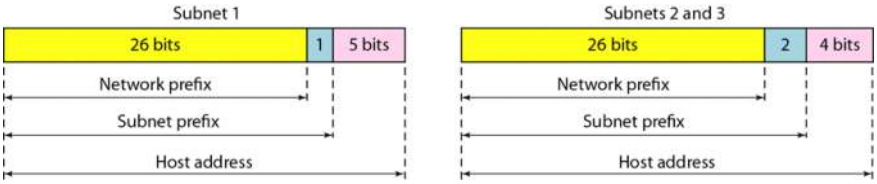
Example

As an example, suppose an organization is given the block 17.12.14.0/26, which contains 2^{32-26}=64 addresses. The organization has three offices and needs to divide the addresses into three subblocks of 32, 16, and 16 addresses.

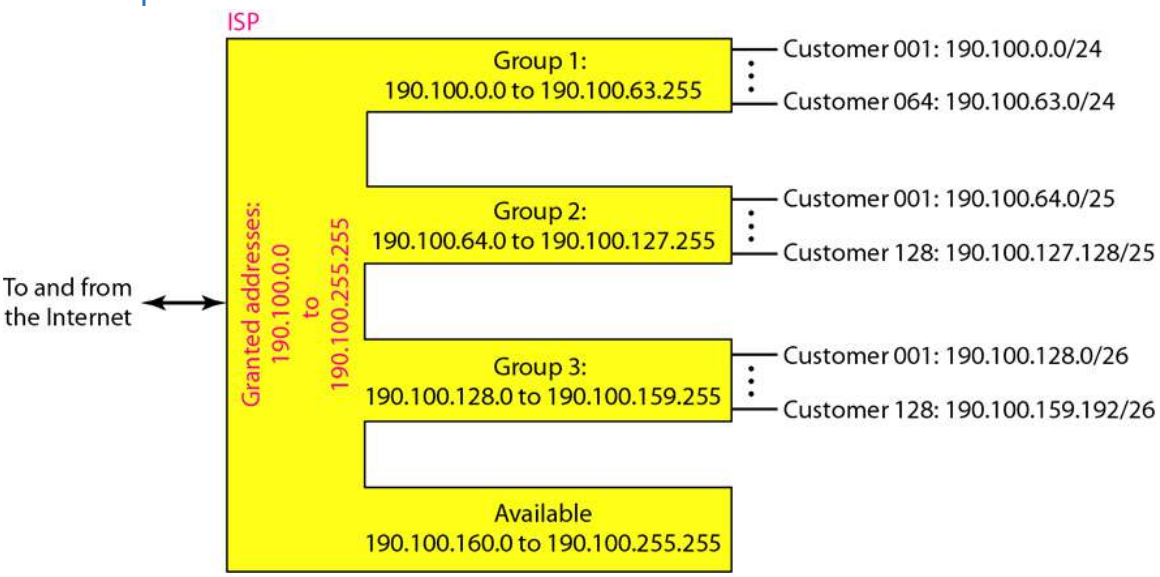
We can find the new masks as follows:

- the first subnet has 32 addresses. Assume its mask is n1.
  2^{32-n1}=32 → Then 32-n1 = $\log_2 32$, and we get n1=27.
- the second subnet has 16 addresses. Assume its mask is n2.
  2^{32-n2}=16 → Then 32-n2 = $\log_2 16$, and we get n2=28.
- the third subnet has 16 addresses. Assume its mask is n3.
  2^{32-n3}=16 → Then 32-n3 = $\log_2 16$, and we get n3=28.





ISP Example



19.1.3.1 Assignment Solutions

Tuesday, April 14, 2015 11:55 AM

1. Find the class of the following classful IP addresses. (4 point)
a) 11110111 11110011 10000111 11011101
b) 10101111 11000000 11110000 00011101
c) 11011111 10110000 00011111 01011110
d) 11101111 11110111 11000111 00011101

solution:
a) Class E
b) Class B
c) Class C
d) Class D

2. In a block of addresses with mask "/16", we know the IP address of one host is 25.34.12.56. What are the first address and the last address in this block? (8 points)

Solution:
The mask is n=16.
We write the given IP address in binary: 00011001.00100010.00001100.00111000

We change the 32-n=16 rightmost bits to all 0s, and get the first address in the block: 00011001.00100010.00000000.00000000 (25.34.0.0).

We change the 32-n=16 rightmost bits to all 1s, and get the last address in the block: 00011001.00100010.11111111.11111111 (25.34.255.255).

3. An organization is granted the address block 16.0.0.0/8. The administrator wants to create 512 equal-size subnets. Assume no address is left unused after the 512 subnets are created. (8 points)
a) Find the subnet mask.
b) Find the number of addresses in each subnet.
c) Find the first and last addresses in the first subnet.
d) Find the first and last addresses in the last subnet.

Solution:
a)&b): the total number of addresses of the organization is: $2^{32-8} = 2^{24}$.
The number of addresses in each subnet is $2^{24}/512 = 2^{15}$. Therefore, the subnet mask is 32-15=17.
c)&d): Since the mask for the organization is 8, the first 8 bits of all addresses should be 0b00010000. Since the subnet mask is 17, then the 9 bits following the first 8 bits will distinguish the 512 subnets: the 9 bits are (0,0,0,0,0,0,0,0,0) for the first subnet, the 9 bits are (0,0,0,0,0,0,0,0,1) for the second subnet, …, and the 9 bits are (1,1,1,1,1,1,1,1,1) for the last subnet. In the following, underscore means subnet prefix
Therefore, for the first subnet, the first address is 0b 00010000 00000000 00000000 00000000 (16.0.0.0), and the last address is 0b 00010000 00000000 01111111 11111111 (16.0.127.255).
For the last subnet, the first address is 0b 00010000 11111111 10000000 00000000 (16.255.128.0), and the last address is 0b 00010000 11111111 11111111 11111111 (16.255.255.255).

19.1.4 (NAT) Network Address Translation

Tuesday, April 14, 2015 11:58 AM

A user can have a large set of ***private*** addresses internally, and one ***universal*** (or small amount) address externally.

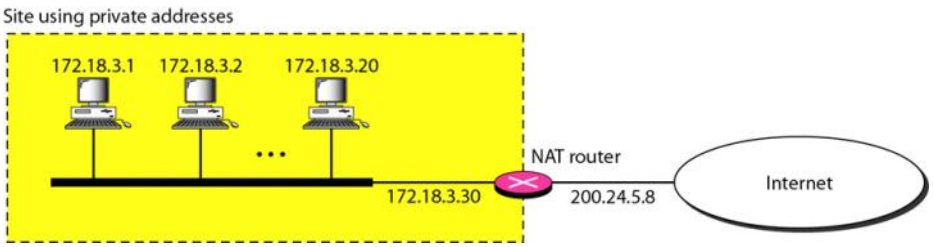
The Internet authorities have reserved **three sets of addresses as private**. Any organization can use these without permission. They are unique inside the organization, but not unique globally.

- No router will forward a packet that has one of these addresses as the destination.

Table 19.3 *Addresses for private networks*

Range			Total
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

The router that connects this network to the internet has one private address, and one global address. The rest of the internet only sees the NAT router.



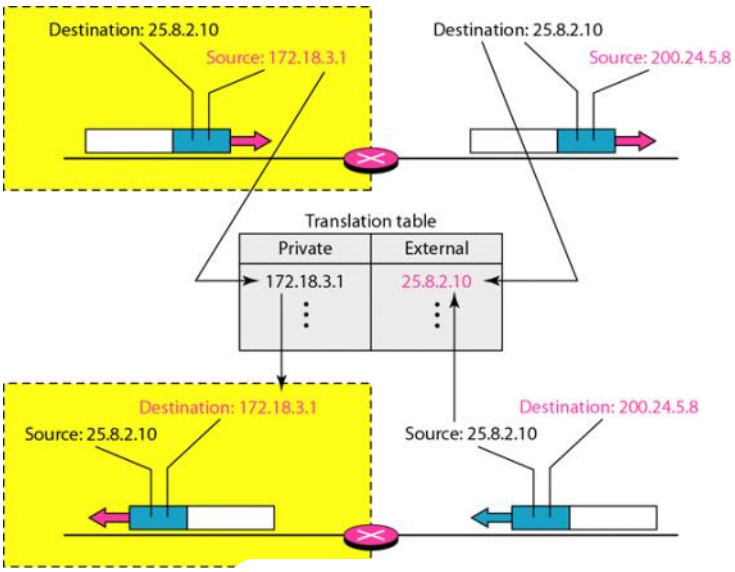
Address Translation

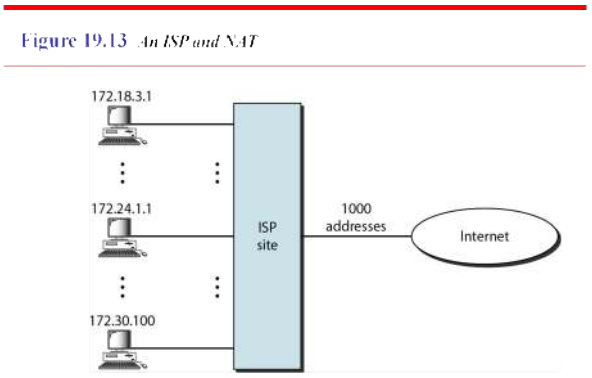
For **Outgoing Packets:**

- Replace the *source address* with the global NAT address.

For **Incoming Packets:**

- Replace the *destination address* with the private address.
- Need to use a **translation table**
  - When a router grabs an outgoing packet, it keeps track of who sent it and to where externally.
  - **Restriction:** Only one private network host can access the same external host.





19.40

Lecture 10: Network Layer: Internet Protocol

Tuesday, April 14, 2015 12:14 PM

Chapter 20 of the textbook.

20.2 IPv4

Tuesday, April 14, 2015 12:17 PM

IPv4 is used by TCP/IP these days. It is unreliable and connectionless datagram protocol. It is best effort.

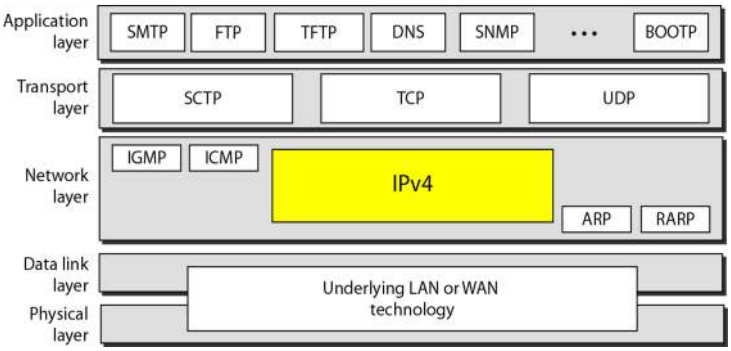
- **Best-effort:** No error control or flow control.

IPv4 assumes unreliability of the underlying layers, does its best to get the message across but with no guarantees.
IPv4 is a connectionless protocol for a packet-switching network that uses the datagram approach.

- Each datagram handled independently, can follow different route to destination (and/or out of order). Some can be lost/corrupted.

If reliability is important, IPv4 must be paired with a reliable protocol such as TCP.

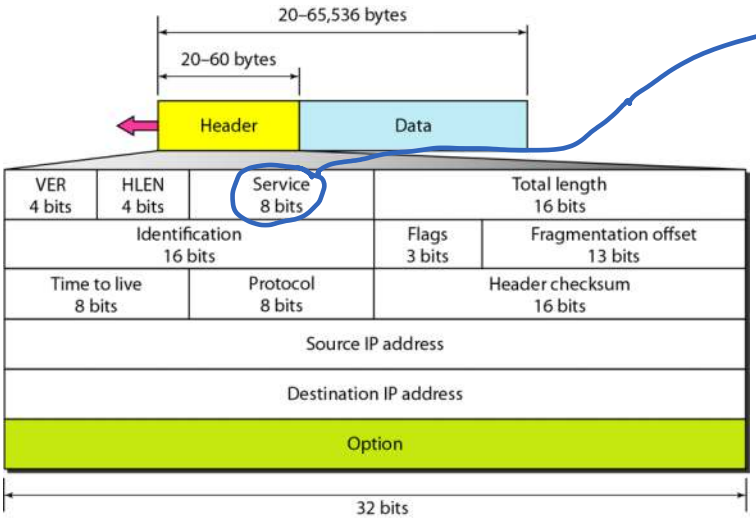
Figure 20.4 *Position of IPv4 in TCP/IP protocol suite*

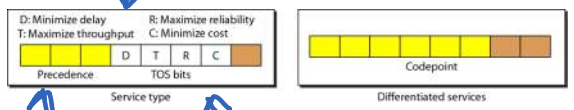


SMTP: simple mail transfer protocol
FTP: file transfer protocol
TFTP: trivial file transfer protocol
DNS: domain name system
SNMP: simple network management protocol
BOOTP: bootstrap protocol
SCTP: stream control transmission protocol
TCP: transmission control protocol
UDP: user datagram protocol
IGMP: Internet group management protocol
ICMP: Internet control message protocol
ARP: address resolution protocol
RARP: reverse address resolution protocol

20.3 IPv4 Datagram Format

Tuesday, April 14, 2015 12:25 PM





The precedence subfield was part of version 4, but never used.

Table 20.1 *Types of service*

TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

At most one bit is one

- **Version (VER):** 4-bit field defines version. Currently 4.
- **Header Length (HLEN):** 4-bit field.
  - 4-bit number of how many 4-byte words.
  - Header length is variable.
    - No options, 20 bytes total. Value of field = 4.
    - Option field at max size, value of field is 15. (15x 4 = 60).
- **Services:** Used to be called service type. Now its differentiated services.
  - **Service type:**
    - 3 precedence bits.
      - Decimal 0->7. Defines priority in congestion
    - TOS bits. One and only one of the bits can have a 1 value.
  - **Differentiated Services:** First 6 bits make up the codepoint subfield. Last two not used.
    - **Code-point subfield.** Two usages.
      - *3 right-most bits are 0s.* **->** 3 left-most bits interpreted as precedence bits. (For compatibility).
      - *3 right-most bits 1s.* -> 6 left-most bits define 64 servies based on a magic priority table.
- **Total Length:**
  - Length of data = total length - header length.
- **Identification, Flags, fragment offset** used in fragmentation.
- **Time to live:** Maximum number of hops visited by the datagram. Each router that processes the datagram decrements this number by 1.

20.3.1 Protocol Field & Examples

Tuesday, April 14, 2015 12:43 PM

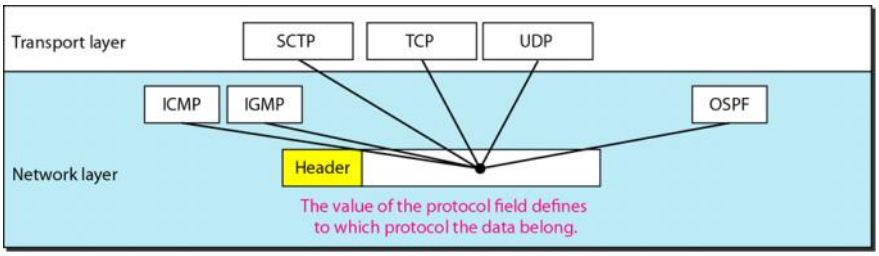


Table 20.4 *Protocol values*

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Example

An IPv4 packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 leftmost bits (0100) show the version, which is correct. The next 4 bits (0010) show an invalid header length (2 × 4 = 8). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example

In an IPv4 packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8 × 4, or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

Example

In an IPv4 packet, the value of HLEN is 5, and the value of the total length field is 0x0028. How many bytes of data are being carried by this packet?

Solution

The HLEN value is 5, which means the total number of bytes in the header is 5 × 4, or 20 bytes (no options). The total length is 40 bytes, which means the packet is carrying 20 bytes of data (40 − 20).

Example

An IPv4 packet has arrived with the first few hexadecimal digits as shown.

0x4500002800010000|0102 . . .

How many hops can this packet travel before being dropped? The data belong to what upper-layer protocol?

Solution

To find the time-to-live field, we skip 8 bytes. The time-to-live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper-layer protocol is IGMP.

20.4 Fragmentation

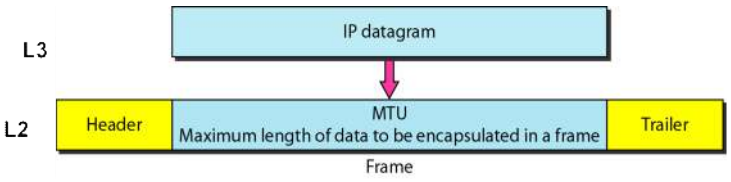
Tuesday, April 14, 2015 12:47 PM

A datagram can travel through different networks. Each router decapsulates the IPv4 datagram from the frame it receives, processes it, then encapsulates it in another frame.

The format and size of the {received/sent} frame depend on the protocol used by the physical network through which the frame {just traveled/is about to travel to}.

Maximum Transfer Unit (MTU)

Each data link layer protocol has its own frame format in most protocols.
IPv4 designers made the maximum length of the IPv4 datagram = 65535 bytes.



Each data link layer protocol has its own frame format in most protocols. One of the fields in the format defines the maximum size of the data field in an L2 frame.

Table 20.5 *MTUs for some networks*

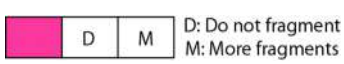
Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

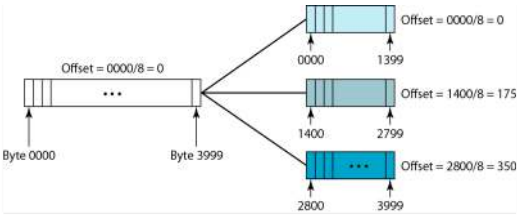
Datagrams are fragmented at the source, or anywhere the MTU decreases. They are put together only at the destination.

Fields Related to Fragmentation

On the previous page we mentioned some fields used for fragmentation.

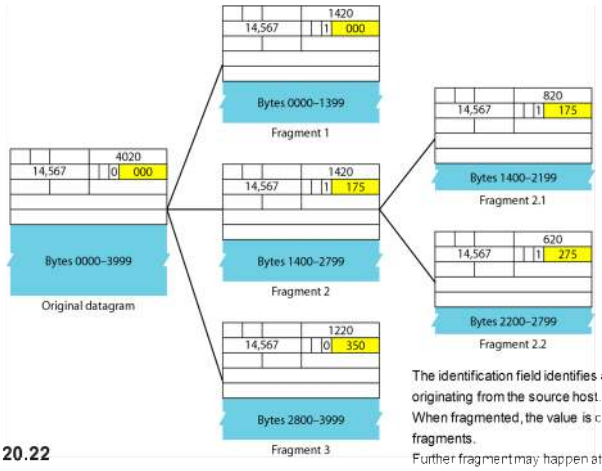
- **Identification.** 16-bit field identifies a datagram originating from the source host. All fragments have same id number, same as original. Helps for reassembly.
- **Flags:** 3-bit field. 1st bit is reserved.
  - 2nd bit is the ***do not fragment*** bit. If 1, do not fragment the datagrame. If 0, fragment as required.
  - 3rd bit is **more fragments** bit. If 1, this datagram is the last/only fragment.
- **Fragmentation Offset.** 13-bit field shows the relative position of the fragment with respect to the whole datagram.





Offset is measured in the units of 8 bytes.

Figure 20.12 *Detailed fragmentation example*



The identification field identifies a datagram originating from the source host.
When fragmented, the value is copied to all fragments.
Further fragment may happen at a router.

20.22

20.4.1 Fragmentation Examples.

Tuesday, April 14, 2015 1:11 PM

Example

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 0, it means that there are no more fragments; the fragment is the last one. However, we cannot say if the original packet was fragmented or not. A non-fragmented packet is considered the last fragment.

Example

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

If the M bit is 1, it means that there is at least one more fragment. This fragment can be the first one or a middle one, but not the last one. We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset).

Example

A packet has arrived with an M bit value of 1 and a fragmentation offset value of 0. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is 1, it is either the first fragment or a middle one. Because the offset value is 0, it is the first fragment.

Example

A packet has arrived in which the offset value is 100. What is the index of the first byte? Do we know the index of the last byte?

Solution

To find the index of the first byte, we multiply the offset value by 8. This means that the first byte index is 800. We cannot determine the index of the last byte unless we know the length.

Example

A packet has arrived in which the offset value is 100, the value of HLEN is 5, and the value of the total length field is 100. What are the indices of the first byte and the last byte?

Solution

The first byte's index is 100 × 8 = 800. The total length is 100 bytes, and the header length is 20 bytes (5 × 4), which means that there are 80 bytes in this datagram. If the first byte's index is 800, the last byte's index must be 879.

20.5 IPv4 Assignment Examples

Tuesday, April 14, 2015 1:13 PM

3. An IPv4 datagram arrives at a router with the following information as the first 20 bytes in the datagram (in hexadecimal)

0x45 04 00 A0 01 02 00 A0 10 01 ?? ?? 0A 0C 0E 05 0C 06 07 09

in which a "?" means a hexadecimal digit to be determined by you.
a) Are there any options in the header?
b) Is the packet fragmented?
c) What is the size of the datagram (not including header)?
d) How many more routers can the datagram travel to?
e) What is the identification number of the datagram?
f) What is the type of service?
g) Which higher layer protocol is used for the encapsulated data in the packet?
h) Please fill the four hexadecimal digits marked as "?".

Solution:
VER=4. HLEN=5. Service = 0x04. Total length= 0x00A0 = 160
Identification=0x0102. Flag bits=0b000. Fragmentation offset=160
Time to live = 0x10 = 16. Protocol=1
a) HLEN = 5. No options.
b) Yes, since the fragmentation offset is nonzero
c) 160 - 5*4 = 140 bytes.
d) 16
e) 0x0102
f) maximize reliability
g) ICMP
h) 0x4504 + 0x00A0 + 0x0102 + 0x00A0 + 0x1001 +0x0A0C + 0x0E05 + 0x0C06 + 0x0709
= 0x8267
So the checksum is 0x7D98.

1. For each field within the first 20 bytes of an IPv4 header, please indicate whether it may change from router to router. Please also briefly give your reason.

Solution:
VER: always 4
HLEN: The option part may be changed by a router. So the HLEN may change.
Service: not change.
Total length: may change (e.g., if fragmentation happens at a router)
Identification: not change
Flags/fragmentation offset: may change (e.g., if fragmentation happens at a router)
Time to live: reduced by one after each router
Protocol: not change
Header chechsum: change after each router (change in any other field will make checksum change)
Source IP address and Destination IP address: not change.

2. In an IPv4 datagram, the M bit is 1, the value of HLEN is 15 (in decimal), the value of total length is 300 (in decimal), and the offset value is 150 (in decimal). What are the index of the first (data) byte and the index of the last (data) byte in this datagram? Is this the last fragment, the first fragment, or a middle fragment? Please explain.

Solution:
Index of first byte: 150*8=1200
Index of last byte: 1200 + (300 - 15*4) - 1 = 1439.
It is a middle fragment. A nonzero offset means it is not the first fragment. M=1 means it is not the last fragment.

20.6 Checksum

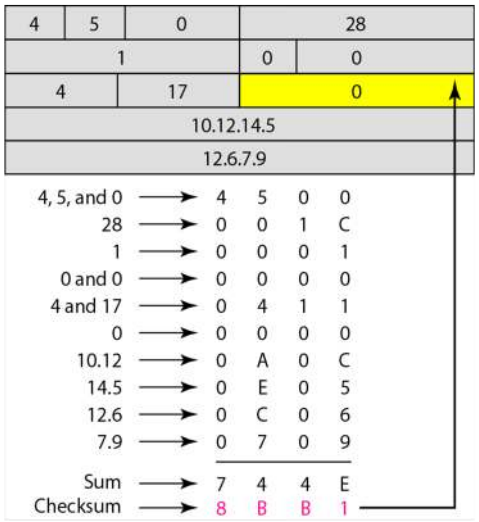
Tuesday, April 14, 2015 1:15 PM

Figure 20.13 shows an example of a checksum calculation for an IPv4 header without options.

The header is divided into 16-bit sections. All the sections are added and the sum is complemented. The result is inserted in the checksum field.

The checksum covers only the header, not the data. *One reason is because all higher-level protocols that encapsulate data in the IPv4 datagram have a checksum field that covers the whole data.*

Figure 20.13 *Example of checksum calculation in IPv4*



When a router receivers a packet, it verifies the checksum. If the result is not all bit 1's, the router discard the packet.

When the router forwards the packet, since some fields of the header are changed (such as Time to Live), it calculates a new checksum.

Ones' complement

the checksum is verified at each router. The result should be all 1's in binary if there is no corruption

20.29

Lecture 11: Transport Layer

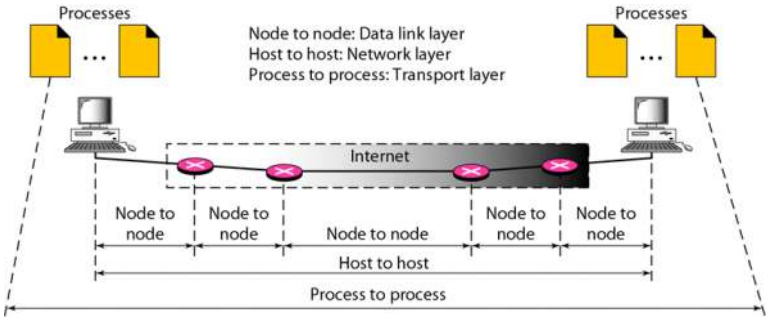
Tuesday, April 14, 2015 1:19 PM

Chapter 23 of the textbook PDF.

23.1 Process-to-Process Delivery

Tuesday, April 14, 2015 1:20 PM

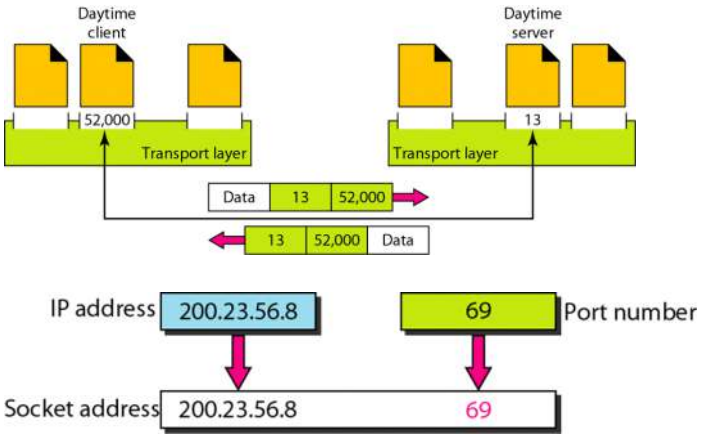
- **Data Link Layer:** delivery of frames between two neighboring nodes. Node-to-node delivery.
- **Network Layer:** datagram delivery, *host-to-host.*
- **Transport Layer:** process-to-process delivery (of a packet, part of a message, from one process to another). This is the typical **client-server** relationship.



Port Numbers

Transport Layer addresses (remember how all the things had addresess?) are actually **port numbers.**

- In the transport layer, we need a transport layer address, called a port number, to choose among multiple processes running on a host. The destination port number is for delivery, while the source port number is needed for reply.
- In the Internet model, the port numbers are 16-bit integers between 0 and 65535. The client program defines itself with a port number, chosen randomly by the transport layer software, referred to as ephemeral port number.
- The Internet has decided to use universal port numbers for servers, called well-known port numbers, such that a client knows its corresponding port number at the server.



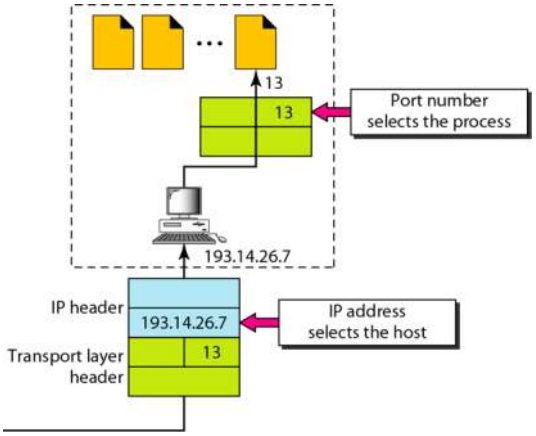
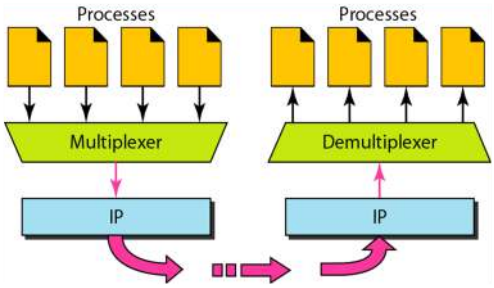
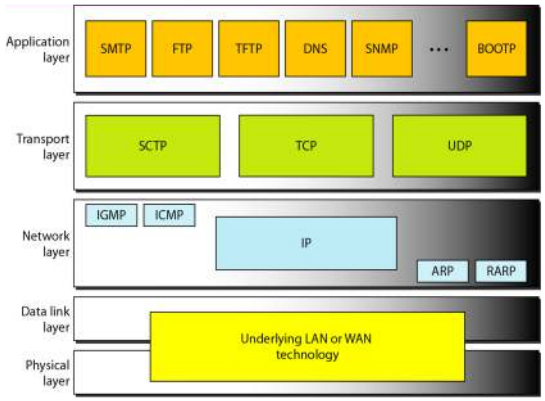


Figure 23.6 *Multiplexing and demultiplexing*



At each host, there are several processes that need to send/receive packets. However, there is only one transport layer protocol.
Multiplexing: the transport layer protocol accepts messages from different processes, differentiated by their assigned port numbers. After adding header, the transport layer passes the packets to L3.

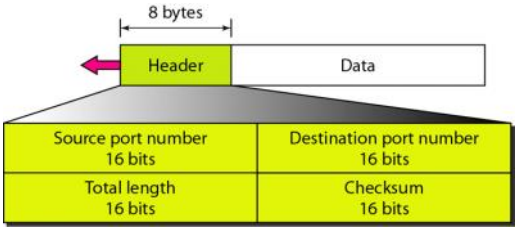
Figure 23.8 *Position of UDP, TCP, and SCTP in TCP/IP suite*



23.2 (UDP) User Datagram Protocol

Tuesday, April 14, 2015 1:44 PM

Connectionless, unreliable transport protocol. Does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.
Very limited error checking.
It has some advantages: very low overhead.



UDP length
= IP length – IP header's length

Checksum

Checksum for UDP includes a pseudoheader, UDP header, and data coming from application.

Example

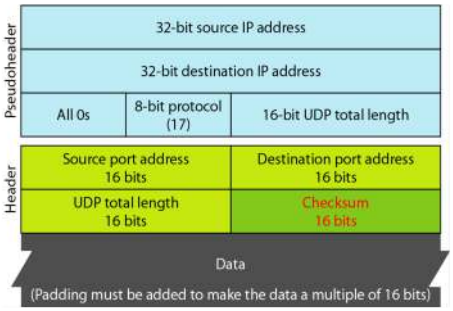
Figure 23.11 shows the checksum calculation for a very small user datagram with only 7 bytes of data. Because the number of bytes of data is odd, padding is added for checksum calculation. The pseudoheader as well as the padding will be dropped when the user datagram is delivered to IP.

Table 23.1 *Well-known ports used with UDP*

Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

23.11

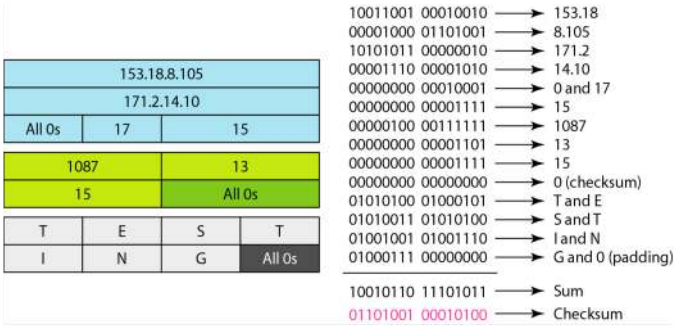
Figure 23.10 *Pseudoheader for checksum calculation*



Purpose of pseudoheader: provide protection against misrouted datagrams.

23.14

Figure 23.11 *Checksum calculation of a simple UDP user datagram*



Carry is dropped at the most significant bit.
Inclusion of the checksum in a UDP datagram is optional.
If the checksum is not calculated, fill the field by all 1s.

23.16

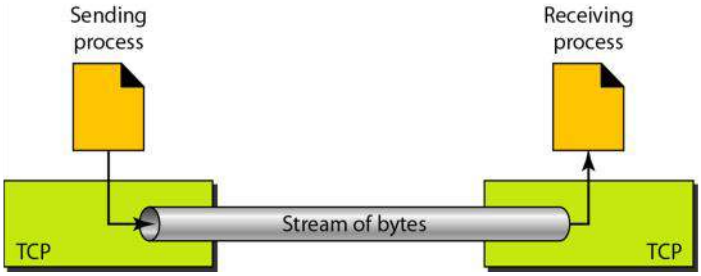
23.3 TCP

Tuesday, April 14, 2015 2:44 PM

TCP is connection-oriented. It creates a virtual connection between two TCPs to send data. It also has flow and error control mechanisms at the transport level. **Reliable.**

Stream delivery

TCP creates an environment in which the sending and receiving processes seem to be connected by an imaginary "tube" that carries their data across the Internet.
In the transport layer, TCP groups a number of bytes together into a packet called segment. TCP adds a header to each segment and delivers the segment to the IP layer for transmission.



Each byte in TCP receives has a **byte number.** For the first , TCP generates a random number between 0 and 2^{32}-1.

- For example, if the number is 1057 and we're sending 6000 bytes: the numbers from 1057 to 7056.
- Used for flow and error control.

The **sequence number** is the number of the first byte carried in the segment.

Example

We are transferring a file of 5000 bytes.

Segment 1	→	Sequence Number: 10,001 (range: 10,001 to 11,000)
Segment 2	→	Sequence Number: 11,001 (range: 11,001 to 12,000)
Segment 3	→	Sequence Number: 12,001 (range: 12,001 to 13,000)
Segment 4	→	Sequence Number: 13,001 (range: 13,001 to 14,000)
Segment 5	→	Sequence Number: 14,001 (range: 14,001 to 15,000)

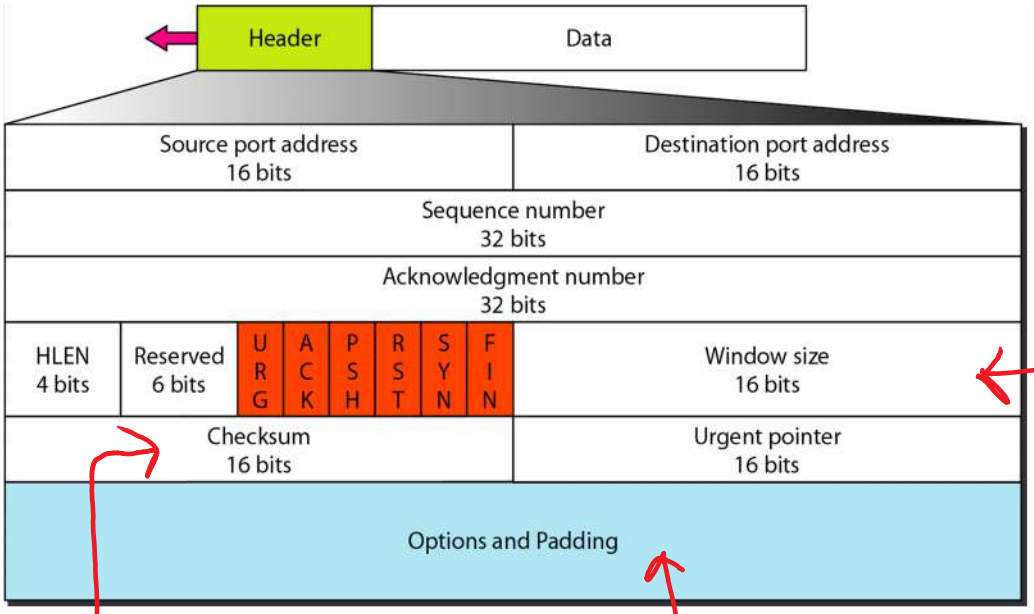
Table 23.2 *Well-known ports used by TCP*

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP, Data	File Transfer Protocol (data connection)
21	FTP, Control	File Transfer Protocol (control connection)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

23.3.1 TCP Header

Tuesday, April 14, 2015 2:57 PM

Figure 23.16 *TCP segment format*



23.23 *HLEN: the number of 4-byte words in the TCP header. 5~15*

Checksum
Follows the same procedure as for UDP. Inclusion of the checksum for TCP is mandatory. The same pseudoheader, serving the same purpose, is added to the segment. For the TCP pseudoheader, the value for the protocol field is 6.

Options: upto 40 bytes.

Window size: Define the size of the window, in bytes, that the other part must maintain.

- The length of the field is 16 bits --> max window size is 65526 bytes.
- Normally referred to as the **receiving window (rwnd)** and is determined by the receiver.
- The sender must obey the dictation of the receiver in this case.

23.3.2 TCP Exchange

Tuesday, April 14, 2015 3:08 PM

In modern implementations, a retransmission occurs if the retransmission timer expires or three duplicate ACK segments have arrived.

Data may arrive out of order and be temporarily stored by the receiving TCP, but TCP guarantees that no out-of-order segment is delivered to the process.

Figure 23.24 *Normal operation*

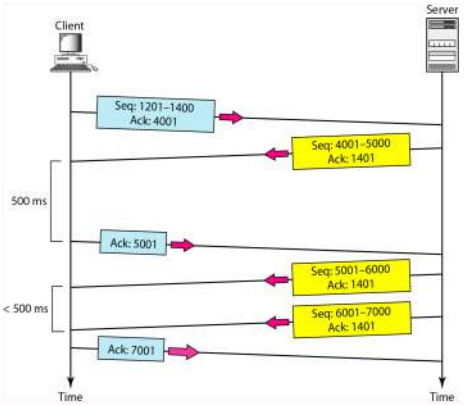
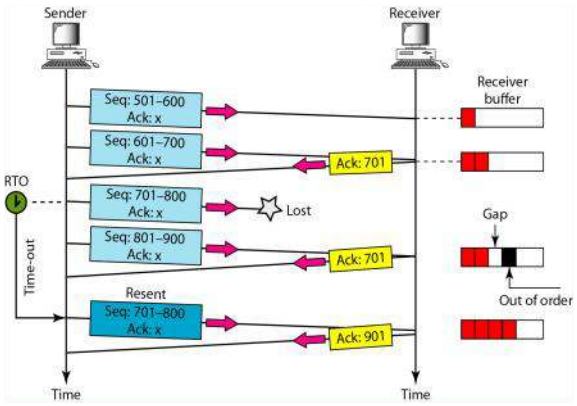


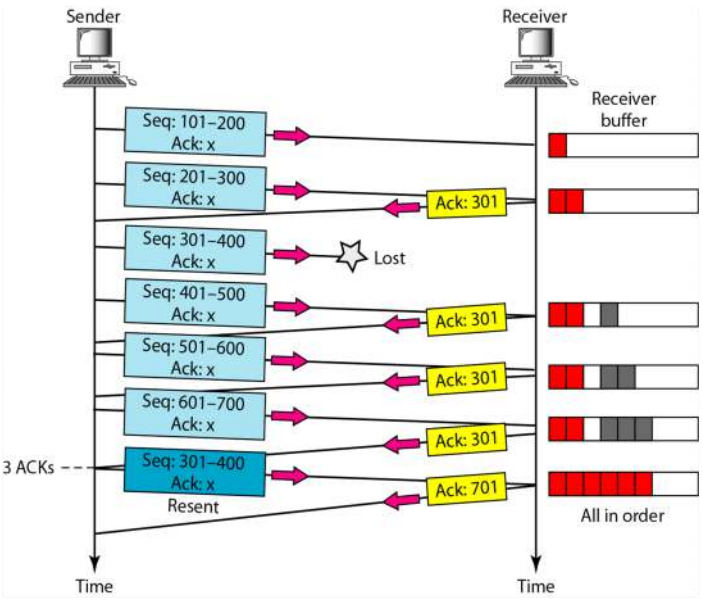
Figure 23.25 *Lost segment*



23.28

The receiver TCP delivers only ordered data to the process.

Figure 23.26 *Fast retransmission*



23.30

Assignment #9

Stefan Martynkiw
1296154

1) a) Class A
   B) Class B
   c) Class C
   d) Class D

2) If we have a /16 block, then the first address must be

   25.34.0.0

   and the last address is

   25.34.255.255

3) We need $\log_2(512) = 9$ bits per subnet for the host address.

The network prefix length is then 32-9 = 25 bits, making the subnet mask

1111 1111 1111 1111 1111 1111 1000 0000

255. 255. 255. 128

a) Subnet mask: 255.255.255.128

B) 7 bits of host address make

$2^7 = 128$ addresses per subnet

c) In the first subnet! the first address is

16.0.0.0

The last address is

16.0.0.128

d) In the last subnet, the first address is

16.1.255.0

the last address is

16.1.255.128