SFWR 4C03: Computer Networks and Computer Security January 12-15 2004

Lecturer: Kartik Krishnan Lectures 4-6

Ethernet: An Introduction

Ethernet is a well known and widely used LAN network technology that employs the bus topology consisting of a single long cable (bus/channel/ether) to which multiple computers attach. It was invented in Xerox Corporation in the early 1970's. This is also the IEEE Standard 802.3. Any computer attached to the bus can send a signal down the cable, and all computers attached to the cable receive a copy of the signal. Every computer can send data to every other computer, thus the Ethernet is an example of an *broadcast* network. A given Ethernet is limited to 500 meters in length, with a minimum separation of 3 meters between each pair of connections. The cable could be extended in length using repeaters; the IEEE standard mandates that no more than four repeaters should be used resulting in a maximum length of 2500 meters. Section 2.4 of Comer [1] covers other hardware details of the Ethernet, and Sections 2.5 and 2.6 cover other network technologies for LANs. We will hereafter assume in class that the underlying network technology is the Ethernet bus topology, unless otherwise specified. It is in any case the most widely used.

Ethernet hardware operates at a bandwidth of 10 Megabits per second (Mbps); a newer version known as *Fast Ethernet* operates at 100 Mbps.

Since the ethernet is a bus topology network multiple computers share access to a single medium. A sender transmits an Ethernet frame in the form of an electric signal that propagates from the sender towards both ends of the cable. During the transmission of a frame, the sending computer has exclusive use of the entire cable, i.e., other computers must wait. After the end of this transmission the shared medium becomes available for other computers.

Carrier Sense Multiple Access with Collision Detect

An Ethernet network does not have a centralized controller that tells each computer how to take turns using the shared cable. Each computer sends out a frame, and when two frames try to occupy the channel at the same time, there is a collision and both frames will be garbled. An important parameter that determines the time taken to detect a collision is the round-trip propagation delay time (2τ), i.e., the time taken by the frame to go from one end of the cable to the other and back. For a 10 Mbps Ethernet with a maximum length of 2500 meters this round-trip time is determined to be about 50μ sec. In order to effectively communicate over the Ethernet, all attached computers participate in a distributed coordination scheme called *Carrier Sense Multiple Access with Collision Detect* (CSMA/CD). The scheme uses the electric activity on the cable to determine the status of the channel, and is based on the following set of rules for each computer:

1. if the channel is idle, transmit; otherwise, go to Step 2.
2. If the channel is busy, continue to listen until the channel is idle, then transmit immediately.
3. If a collision is detected during transmission, transmit a brief jamming signal to assure other computers sharing the medium that there has been a collision and then cease transmission.
4. After transmitting the jamming signal, wait a random amount of time, then attempt to transmit again.

The randomization in step 4 of the algorithm is performed using the *binary exponential backoff algorithm.* After a collision, time is divided into discrete slots whose length is equal to the round-trip propagation time (2τ). After each collision each station waits 0 or 1 slot times before trying again. If two computer frames collide initially, and each computer picks the same random number, their frames will collide again. After the second collision, each computer picks either $0, 1, 2$ or 3 at random and waits that number of slot times. If a third collision occurs (the probability of this happening is 0.25), then the next time the number of slots to wait is chosen at random from the interval 0 to $2^3 - 1$. In general, after i collisions, a random number between 0 and $2^i - 1$ is chosen, and that number of time slots is skipped. After 16 collisions, each computer simply gives up. Further recovery of the message to be sent is then up to the higher layers in the communication protocol suite.

Ethernet Hardware Addresses and Frame Format

Each computer attached to the Ethernet is assigned a unique 48 bit number called its *Ethernet* or *physical* address. Usually, the Ethernet address is fixed in machine readable form in the host machine interface hardware. Thus, replacing a hardware interface that has failed changes the machine's physical address.

The host interface hardware examines frames and determines the frames to be sent to the host. It uses the destination address field in the frame as a filter ignoring those that are meant for other machines, and passing to the host only those frames that are meant for it. The host interface operates independently of the computer's central processor, thereby preventing the traffic on the Ethernet from slowing the processing on the host computer.

A 48 bit address could also specify more than a single destination computer. For example, the broadcast address (all 1s) is reserved for sending to all computers simultaneously (for instance this is utilized in the ARP protocol discussed later in this Lecture).

The data transmitted between two computers attached to the Ethernet is encapsulated in an Ethernet frame. These frames are of variable length with no frame smaller than 64 bytes (octets) or larger than 1518 bytes. The format of a frame is shown in Figure 2.7 of Comer [1]. Each frame contains the following fields in the order specified:

1. **Preamble Field:** It consists of 8 bytes (64 bits) of alternating 0s and 1s to help the two communicating interfaces synchronize.
2. **Destination Address:** It contains the 48 bit address of the intended recipient.
3. **Source Address:** It contains the 48 bit address of the source computer.
4. **Frame Type:** It identifies the type of data being carried in the frame. The operating system on the machine uses the frame type to determine which software module should process the frame.
5. **Frame Data:** The data to be transmitted. A lower bound on the frame data is 46 bytes, and an upper bound is 1500 bytes. The factors determining these upper and lower bounds are mentioned below.
6. **Cyclic Redundancy Check (CRC):** The 32 bit CRC helps the interface detect errors: the sender computes the CRC as a function of the data in the frame, and the receiver recomputes the CRC to verify that the frame has been received intact.

The upper bound on the frame length was set arbitrarily at 1500 bytes, mostly based on the fact that the network interface hardware requires enough RAM to hold an entire frame and RAM was expensive in the 1970's. Another contention was that all computers have to share the channel, and thus an upper bound on the frame length does not allow a user to monopolize the entire channel for more than specified period of time. The reason for having a lower bound on the frame length is more interesting, and it is to prevent a computer from completing the transmission of a short frame before the first bit of the frame has reached the far end of the cable, where it may collide with another frame. Consider the following problem: At time 0, station A at one end of the network sends out a frame intended for station B at the other end of this LAN network. Let us call the propagation time for the frame to reach the other end of the cable as τ (this is one half of the round trip propagation time 2τ discussed earlier). Let's say that just before this frame gets to the other end, i.e., at time $\tau - \epsilon$, the most distant station B starts transmitting. When B detects the leading edge of A's frame it aborts its transmission (Step 3 in the CSMA protocol), and generates a 48 bit noise burst to warn other stations. At about time 2τ, station A sees this noise burst and aborts its transmission too. Now, if the station were transmitting a very short frame, it is conceivable that a collision occurs, but the transmission is completed before the acknowledgement (noise burst) gets back at 2τ. The station A will then incorrectly conclude that the frame was successfully sent. On a 10 Mbps LAN, it takes 10^{-7} sec to transmit each bit of the frame. Keeping in mind that $2\tau = 50\mu$ sec, 5000 bits is the smallest frame that is guaranteed to work. To add some margin of safety, this number was rounded to 512 bits or 64 bytes. Frames will fewer than 64 bytes are padded out to 64 bytes with the Pad field.

Internet Protocol Addresses

Addressing is a critical component of the internet abstraction. To give the appearance of a single, uniform system, all host computers must use an uniform addressing scheme. Unfortunately, physical network addresses (discussed in the previous section) do not suffice because an internet can include multiple network technologies, with its own address format. Thus, the addresses used by two technologies may be incompatible because they are of different sizes or have different formats.

Each host on the internet is assigned a unique 32-bit internet address (IP address) that is used in all communication with that host.

An IP address does not identify a specific computer. Instead, each IP address identifies a connection between the computer and a network. A computer with multiple network connections, e.g., a router must be assigned one IP address for each connection.

Users, application programs, and the higher layers of the protocol software (TCP, IP) use IP addresses to communicate. On the other hand physical addresses are used by the lower layers of the protocol software such as the network interface layer.

Conceptually, each 32 bit IP address is divided into two parts: a prefix and a suffix. This two level hierarchy is designed to make routing efficient. The address prefix is some sort of a network id, and it identifies the physical network to which the computer is attached, while the suffix (host id) identifies an individual computer on that network. Each physical network in the internet is assigned a unique prefix, and each computer on a given physical network is assigned a unique address suffix. Routing through the internet is based on the network portion of the address. Once the packet reaches the destination network, the host id is used to direct the frame to the appropriate destination machine.

In a classful addressing scheme, each IP address is said to be *self-identifying* because the boundary between prefix and suffix can be computed from the address alone, without reference to external information. In particular, the class of an address can be determined from the three higher order bits. There are five classes of 32 bit IP addresses.

1. **Class A:** Class A addresses start with a 0 in the first bit and use the first octet for the network address, leaving three octets for the host address. Hence, the first octet of a class A address has a value between 0 and 127 (i.e., binary numbers 00000000 and 01111111, respectively). A class A network consists of (2^{24}), i.e., 16,777,216 host computers.

2. **Class B:** Organizations that did not require such a large number of hosts could be allocated a Class B address. A Class B address starts with 10 in the first two bits and uses the first two octets for the network address and the last two octets for the host address. A Class B network consists of 2^{16}, i.e., 65536 host computers.

3. **Class C:** Even smaller organizations could be allocated Class C addresses that start with 110 in the first three bits and use the first three octets for the network address and only the last octet for the host address. Each Class C network has 2^8, i.e. 256 hosts.

4. **Class D:** Class D addresses begin with 1110 are used for multicast traffic sent to a collection of machines.
5. **Class E:** Class E addresses starting with 11110 are reserved for future use.

. The division along octet boundaries motivated the representation of IP addresses in dotted-decimal notation, which represents each octet as a decimal number ranging from 0 to 255.

There are some special address conventions:

1. An IP address consisting of 32 zeros refers to the concerned host computer.
2. An IP address with the net id of zeros refers to a particular host computer on a network.
3. An IP addresses consisting of all ones is used for limited broadcast.
4. An IP address with a valid net id, and a host id of all ones is intended as a directed broadcast for all the hosts on that network.
5. Finally, the address 127.0.0.1 is intended as a loopback address. This is used for testing TCP/IP and inter-process communications on the local computer.

The *Internet Corporation for Assigned Names and Numbers* (ICANN) assigns net id's, while the local Internet Service Provider or the System Administrator on an University network typically assign the host id's.

Address Resolution Protocol

Although each machine has one or more IP addresses, these cannot be used for sending frames because the network interface layer (hardware) does not understand IP addresses. Mapping between a protocol (IP address) and a hardware (Ethernet address) is called address resolution. A host or a router uses address resolution when it needs to send to another computer on the same physical network, and also knows the destination computer's IP address.

There are three types of address resolution algorithms:

1. **Table lookup:** Address bindings are stored in a table in memory, which the software searches when it needs to resolve an address.
2. **Closed-form computation:** The protocol address assigned to a computer is carefully chosen so the computer's hardware address can be computed from the protocol address using basic Boolean and arithmetic operations.
3. **Message exchange:** Computers exchange messages across the network to resolve an address.

The last approach is generally implemented on Ethernet networks and utilizes its broadcast capability. To guarantee that all computers agree on the exact format and meaning of messages used to resolve addresses, the TCP/IP protocol suite includes an *Address Resolution Protocol* (ARP). There are two ARP message types: a request and a response. The request message contains an IP address and requests the corresponding hardware address. This message is broadcast, i.e., all computers on the network receive this message. The intended recipient sends out a reply (only to the sender) and the reply contains both the IP address sent in the request and the hardware address.

Essential to the efficient operation of ARP is the maintenance of an *ARP cache* on each host. The cache maintains the recent mappings from Internet addresses to hardware addresses. The normal expiration time of an entry is 20 minutes from the moment the entry was created. A host computer normally examines its ARP cache, and if unable to find the binding sends out an ARP request. Since the ARP request is broadcast, all hosts on the network can update their cache accordingly.

ARP Packet format

The ARP message format is given in Figure 5.3 of Comer [1]. It contains the following fields:

1. **Hardware Type:** This is a 2 octet field that specifies the type of the hardware address employed in the physical network; it contains the value 1 for Ethernet.

2. **Protocol Type:** A 2 octet field that specifies the high level protocol employed, i.e., the type of the high level protocol address the sender has supplied. It contains $(0800)_{16}$ for IP addresses.

3. **HLEN:** This is a one octet field that specifies the length of the hardware address in bytes. This is 6 for Ethernet addresses.

4. **PLEN:** This is also a one octet field that specifies the length of the protocol address which is 4 for IP addresses.

5. **Operation:** This is 2 octet field that specifies whether the operation is an ARP request (1), ARP response (2), RARP request (3), or RARP response (4).

6. Finally, the last four fields contain the sender's hardware and protocol addresses, and the target's hardware and protocol addresses.

For an ARP request all the fields are filled in except the target hardware address. When a system receives an ARP request directed to it, it fills in its hardware address, swaps the two sender addresses with the two target addresses, sets the *Operation* field to 2 and sends the reply.

The ARP message is sent in the data portion of the hardware frame. The frame header contains the 6 byte Ethernet destination address (for an ARP request this is a 6 byte all ones broadcast address), the 6 byte physical address of the source. Finally, the 2 byte *Frame* type is $(0806)_{16}$ for ARP requests/replies.

Proxy ARP

Proxy ARP lets a router answer ARP request on one of its networks for a host on another of its networks. This fools the sender of the ARP request into thinking that the router is the destination host, when in fact the destination host is *on the other side* of the router. The router, here, is acting as a proxy agent for the destination host, relaying packets to it from other hosts. There is a nice discussion on proxy ARP in Section 5.6.3. of Tanenbaum [4], and Section 4.6 of Stevens [3]

Recommended Reading

1. Sections 2.4-2.6 of Comer [1], Chapter 14 of Stallings [2], and Section 4.3 of Tanenbaum [4] for various LAN technologies including Ethernet.
2. Chapter 4 of Comer [1], Chapter 3 of Stevens [3], and Section 5.6.2 of Tanenbaum [4] for a discussion of IP addresses.
3. Chapter 5 of Comer [1], Chapter 4 of Stevens [3], and Section 5.6.3 of Tanenbaum [4] for a discussion of the ARP protocol.

References

[1] D.E. Comer, *Internetworking with TCP/IP: Principles, Protocols, and Architectures*, 4th edition, Prentice Hall, NJ, 2000.

[2] W. Stallings, , *Data & Computer Communications*, 6th edition, Prentice Hall, NJ, 2002.

[3] W. Richard Stevens, *TCP/IP Illustrated, Volume I: The Protocols*, Addison Wesley Professional Computing Series, 1994.

[4] A. Tanenbaum, *Computer Networks*, 4th edition, Prentice Hall, NJ, 2003.