Name : Agung Wirayogi
NPM : 1306439094

SUMMARY/REPORT OF CHAPTER 5

Link layer its services, the principles underlying its operation, and a number of important specific protocols that use these principles in implementing link-layer services. Why we use link layer? We saw that the basic service of the link layer is to move a network layer datagram from one node (host, switch, router, WiFi access point) to an adjacent node. How to use link layer? We saw that all link-layer protocols operate by encapsulating a network-layer datagram within a link-layer frame before transmitting the frame over the link to the adjacent node.

The link layer is implemented in a network adapter, also sometimes known as a network interface card (NIC). At the heart of the network adapter is the link layer controller, usually a single, special purpose chip that implements many of the link layer services (such as framing, link access, error detection, and so on). Thus, much of a link-layer controller's functionality is implemented in hardware. For example, Intel's 8254x controller [Intel 2012] implements the Ethernet protocols, the Atheros AR5006 [Atheros 2012] controller implements the 802.11 WiFi protocols. Until the late 1990s, most network adapters were physically separate cards (such as a PCMCIA card or a plug-in card fitting into a PC's PCI card slot) but increasingly, network adapters are being integrated onto the host's motherboard—a so-called LAN-on-motherboard configuration.

Although the basic service of any link layer is to move a datagram from one node to an adjacent node over a single communication link, the details of the provided servicecan vary from one link-layer protocol to the next. Possible services that can be offered by a link-layer protocol include:

• Framing. Almost all link-layer protocols encapsulate each network-layer datagram within a link-layer frame before transmission over the link. A frame consists of a data field, in which the network-layer datagram is inserted, and a number of header fields. The structure of the frame is specified by the link layer protocol. We'll see several different frame formats when we examine specific link layer protocols.

• Link access. Amedium access control (MAC) protocol specifies the rules by which a frame is transmitted onto the link. For point-to-point links that have a single sender at one end of the link and a single receiver at the other end of the link, the MAC protocol is simple (or nonexistent) the sender can send a frame whenever the link is idle. The more interesting case is when multiple nodes share a single broadcast link the so called multiple access problem. Here, the MAC protocol serves to coordinate the frame transmissions of the many nodes.

• Reliable delivery. When a link layer protocol provides reliable delivery service, it guarantees to move each network-layer datagram across the link without error. Recall that certain transport layer protocols (such as TCP) also provide a reliable delivery service. Similar to a transport-layer reliable delivery service, a link-layer reliable delivery service can be achieved with acknowledgments and retransmission. A link layer reliable delivery service is often used for links that are prone to high error rates, such as a wireless link, with the goal of correcting an error locally on the link where the error occurs rather than forcing an end toend retransmission of the data by a transport or application-layer protocol. However, link layer reliable delivery can be considered an unnecessary overhead for low bit error links, including fiber, coax, and many twisted-pair copper links. For this reason, many wired link layer protocols do not provide a reliable delivery service.

• Error detection and correction. The link layer hardware in a receiving node can incorrectly decide that a bit in a frame is zero when it was transmitted as a one ,and vice versa. Such bit errors are introduced by signal attenuation and electromagnetic noise. Because there is no need to forward a datagram that has an error, many link-layer protocols provide a mechanism to detect such bit errors. This is done by having the transmitting node include error-detection bits in the frame, and having the receiving node perform an error check. Recall from Chapters 3 and 4 that the Internet's transport layer and network layer also provide a limited form of error detection the Internet checksum. Error detection in the link layer is usually more sophisticated and is implemented in hardware. Error correction is similar to error detection, except that a receiver not only detects when bit errors have occurred in the frame but also determines exactly where in the frame the errors have occurred (and then corrects these errors).

We can learn a lot about Ethernet by examining the Ethernet frame. To give this discussion about Ethernet frames a tangible context, let's consider sending an IP datagram from one host to another host, with both hosts on the same Ethernet LAN. (Although the payload of our Ethernet frame is an IP datagram, we note that an Ethernet frame can carry other network-layer packets as well.) Let the sending adapter, adapter A, have the MAC address AA-AA-AA-AA-AA-AA and the receiving adapter, adapter B, have the MAC address BB-BB-BB-BB-BB-BB. The sending adapter encapsulates the IP datagram within an Ethernet frame and passes the frame to the physical layer. The receiving adapter receives the frame from the physical layer, extracts the IP datagram, and passes the IP datagram to the network layer. In this context, let's now examine the six fields of the Ethernet frame:

• Data field (46 to 1,500 bytes). This field carries the IP datagram. The maximum transmission unit (MTU) of Ethernet is 1,500 bytes. This means that if the IP datagram exceeds 1,500 bytes, then the host has to fragment the datagram, as discussed in Section 4.4.1. The minimum size of the data field is 46 bytes. This means that if the IP datagram is less than 46 bytes, the data field has to be "stuffed" to fill it out to 46 bytes. When stuffing is used, the data passed to the network layer contains the stuffing as well as an IP datagram. The network layer uses the length field in the IP datagram header to remove the stuffing.

• Destination address (6 bytes). This field contains the MAC address of the destination adapter, BB-BB-BB-BB-BB-BB. When adapter B receives an Ethernet frame whose destination address is either BB-BB-BB-BB-BB-BB or the MAC broadcast address, it passes the contents of the frame's data field to the network layer; if it receives a frame with any other MAC address, it discard the frame.

• Source address (6 bytes). This field contains the MAC address of the adapter that transmits the frame onto the LAN, in this example, AA-AA-AA-AA-AA-AA.

• Type field (2 bytes). The type field permits Ethernet to multiplex network-layer protocols. To understand this, we need to keep in mind that hosts can use other network layer protocols besides IP. In fact, a given host may support multiple network-layer protocols using different protocols for different applications. For this reason, when the Ethernet frame arrives at adapter B, adapter B needs to know to which network-layer protocol it should pass (that is, demultiplex) the contents of the data field. IP and other network-layer protocols (for example, Novell IPX or AppleTalk) each have their own, standardized type number. Furthermore, the ARP protocol (discussed in the previous section) has its own type number, and if the arriving frame contains an ARP packet (i.e., has a type field of 0806 hexadecimal), the ARP packet will be demultiplexed up to the ARP protocol.

• Cyclic redundancy check (CRC) (4 bytes). An error-detection technique used widely in today's computer networks is based on cyclic redundancy check (CRC) codes. CRC codes are also known as polynomial codes, since it is possible to view the bit string to be sent as a polynomial whose coefficients are the 0 and 1 values in the bit string, with operations on the bit string interpreted as polynomial arithmetic. CRC Example

Want:

D.2r XOR R = nG

equivalently:

D.2r = nG XOR R

equivalently:

if we divide D.2r by G,

want remainder R

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

• Preamble (8 bytes). The Ethernet frame begins with an 8-byte preamble field. Each of the first 7 bytes of the preamble has a value of 10101010; the last byte is 10101011. The first 7 bytes of the preamble serve to "wake up" the receiving adapters and to synchronize their clocks to that of the sender's clock. Why should the clocks be out of synchronization? Keep in mind that adapter A aims to transmit the frame at 10 Mbps, 100 Mbps, or 1 Gbps, depending on the type of Ethernet LAN. However, because nothing is absolutely perfect, adapter A will not transmit the frame at exactly the target rate; there will always be some drift from the target rate, a drift which is not known a priori by the other adapters on the LAN. A receiving adapter can lock onto adapter A's clock simply by locking onto the bits in the first 7 bytes of the preamble. The last 2 bits of the eighth byte of the preamble (the first two consecutive 1s) alert adapter B that the "important stuff" is about to come.

Ethernet technologies provide an unreliable service to the network layer. Specifically, when adapter B receives a frame from adapter A, it runs the frame through a CRC check, but neither sends an acknowledgment when a frame passes the CRC check nor sends a negative acknowledgment when a frame fails the CRC check. When a frame fails the CRC check, adapter B simply discards the frame. Thus, adapter A has no idea whether its transmitted frame reached adapter B and passed the CRC check. This lack of reliable transport (at the link layer) helps to make Ethernet simple and cheap. But it also means that the stream of datagrams passed to the network layer can have gaps.

Reference :

http://web.cs.ucdavis.edu/~prasant/ECS152A/NOTES/LinkLayer.pdf
book Computer Network A Top-Down Approach sixth edition by Kurose and Ross