

Processing of Ethernet Frames in Linux

The operations at the physical layer and the MAC layer are generally implemented in hardware on the network interface card. The operations at the data link layer, such as the construction and demultiplexing of frames, are performed in the device drivers of the network interface cards.

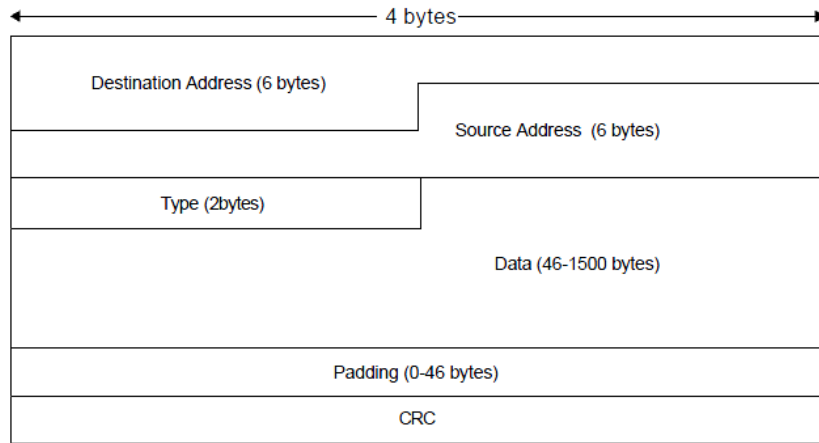
Each network interface is associated with a device driver. A device driver is responsible for the data exchange between the Linux operating system and the network interface card, generally via memory mapped I/O. In Figure 2.6 we show the operations of a device driver for an Ethernet card when it processes an IP datagram. The figure also includes the driver of the loopback interface. Recall that the loopback interface is a virtual interface, which is present on any Linux system and which is generally assigned the IP address 127.0.0.1.

If a host sends a packet to the loopback interface, the packet is immediately returned to the network layer of the host. Thus, the loopback interface permits a host to send packets to itself. When an IP module on a Linux host transmits an IP datagram on an Ethernet card, the device driver first tests if the sending host should also receive the IP datagram, and if so, passes the IP datagram to the loopback driver.

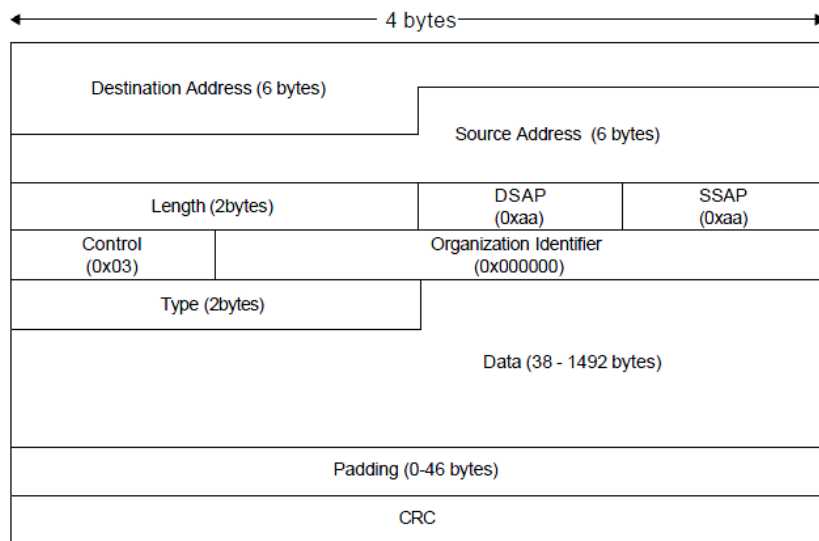
The sender of a datagram should receive the IP datagram, when the IP destination address is a multicast or broadcast address or when the IP destination address matches one of the IP addresses of sending host. If this is not the case, the Ethernet driver tests if it knows the MAC address for the destination IP address.

If not, the ARP protocol, is invoked to resolve the MAC address and the datagram is kept in a buffer until the MAC address becomes available. If the MAC address is known, the Ethernet driver constructs the Ethernet frame with the format shown in Figure 2.3.

Once the frame is constructed, the Ethernet device driver passes the frame to the Ethernet card. The MAC and physical layer are handled by the hardware on the Ethernet card, and are not part of the Linux operating system. When an Ethernet frame arrives at a network interface card, the interface cards assembles the frame, writes the frame into a memory location, and issues a signal to the Linux operating system. Then, the device driver for the appropriate network interface reads and demultiplexes the frame. As shown in Figure 2.3, if the packet is an ARP message the payload of the frame is passed to ARP, and if it is an IP datagram, the payload is passed to the IP module.



(a) Ethernet Encapsulation



(b) IEEE 802.3 Encapsulation

Figure 2.3. Ethernet Frame Formats.

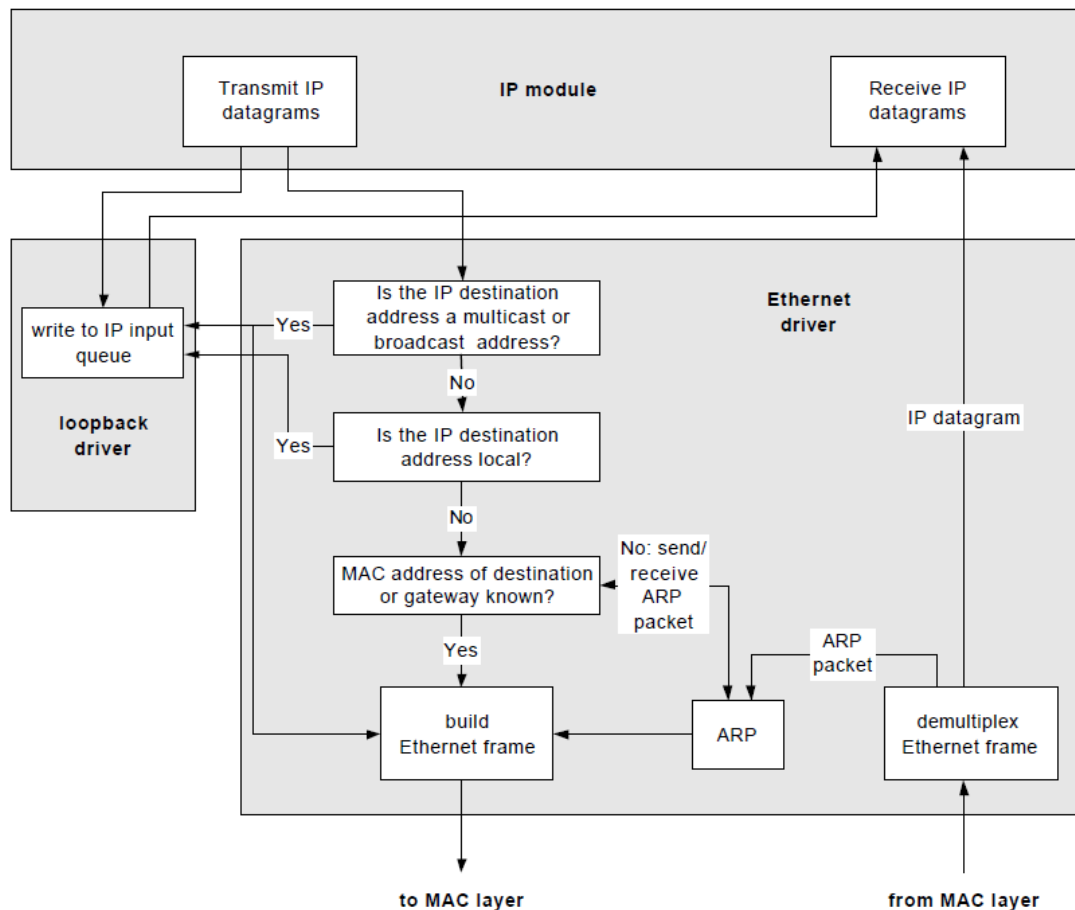


Figure 2.6. Ethernet and loopback device drivers (from W. Stevens: TCP/IP Illustrated, Volume 1).

When a datagram is transmitted it can be processed by either, the loopback driver, or the Ethernet driver, depending on the IP address of the outgoing datagram. If the IP address is not the local loopback address, the datagram is filtered one more time to determine if the address is a broadcast or multicast address, in which case it is sent directly to the Ethernet network with the appropriate hardware address and is also placed on the device's input queue for local processing.

If the IP address is none of the above, the driver checks the ARP cache to determine if the hardware address is available. If the ARP cache does not contain the destination IP address, the driver calls the ARP module to send out an ARP query. Once the ARP response comes back, it is processed and the outgoing packet is transmitted. All incoming packets are filtered to separate ARP frames from IP datagrams. The former are used to update the ARP cache and the latter are passed on to the input queue.

The 4.4BSD (UNIX like) implementation defines the variable `useloopback` and initializes it to 1. If this variable is set to 0, however, the Ethernet driver sends local packets onto the network instead of sending them to the loopback driver. This may or may not work, depending on your Ethernet interface card and device driver.