**Chapter 9: Introduction to Data-Link Layer**

Table of Contents

[9.1 Introduction 3](#_Toc63188587)

[Nodes and Links 3](#_Toc63188588)

[Framing 3](#_Toc63188589)

[Error Control 4](#_Toc63188590)

[Flow Control 4](#_Toc63188591)

[Access Control 5](#_Toc63188592)

[9.2 Link-Layer Addressing 6](#_Toc63188593)

[Types of Addressing 8](#_Toc63188594)

[Address Resolution Protocol 9](#_Toc63188595)

For this chapter, make sure to follow the 5th edition of the book. The 4th edition has different content in this chapter. There are no slides available for this chapter.

## 9.1 Introduction

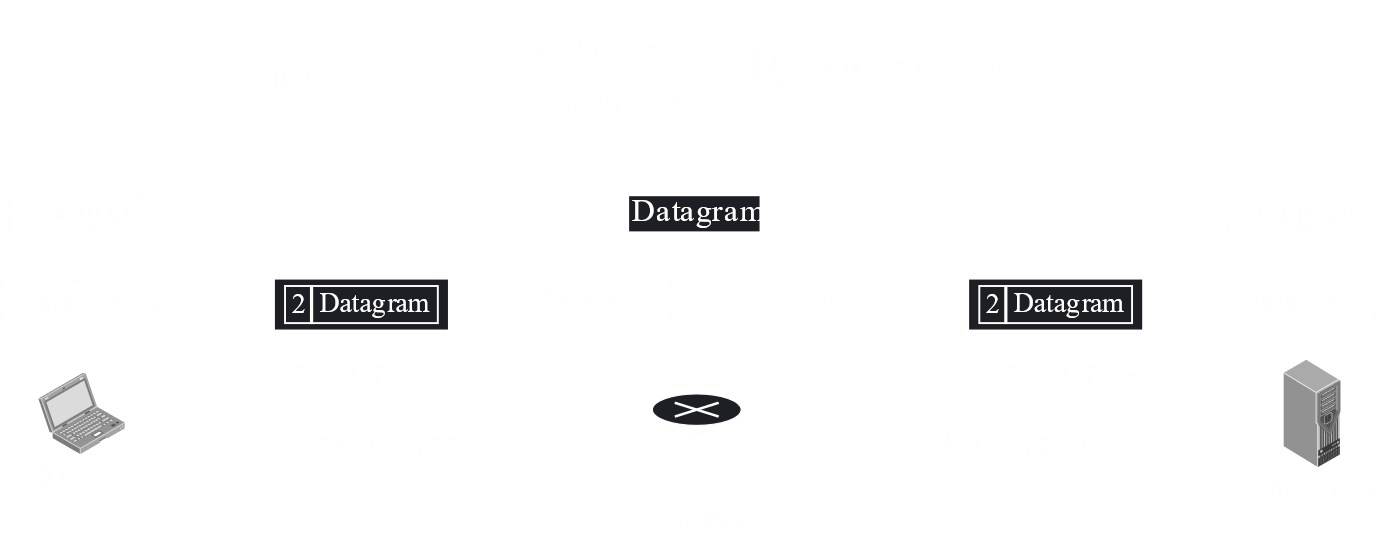
The main work of the data-link layer involves framing, access control, flow control and error control.

### Nodes and Links

The data-link layer is also responsible for physical addressing, done using MAC addressing in ethernet. It is responsible for node-to-node delivery. Nodes refer to any devices on the network, including any routers. There are links in between the nodes, referring to the physical networks between them.

### Framing

The physical address changes from one hop to another, since the jurisdiction of the physical address is limited to that scope. This needs to be done since the global IP addressing that can identify any device on the internet is too large for routers to handle.



In the above diagram, we can see that the source data link layer encapsulates one frame, the intermediate device decapsulates it and encapsulates a different frame and that is finally decapsulated at the destination. The encapsulation and decapsulation may seem like overkill for a process in which just the physical address is changing, but in reality, this needs to be done because the entire frame changes. The link between the source and the intermediate device is not necessarily the same as that between the intermediate device and the destination. For example, maybe the first link is an ethernet cable while the second is a wireless connection. The frame created at each hop is created based on the physical connection the frame will be travelling through, so each frame is different.

### Error Control

When being transported, some of the data may become corrupted due to a variety of reasons. This is handled by error control. In data communication, when errors are detected, the data is just retransmitted since that is easier than actually correcting the data. Sometimes however, especially in real-time communication, it might not be feasible to retransmit data since that would cause problems like jitter. In those cases, error correction is attempted.

### Flow Control

It is possible that the sender is sending data so fast that the receiver cannot process it in time. This would cause the receiver’s buffer to overflow leading to data loss. To ensure this does not happen, flow control balances the rate of data transmission. Flow control needs assistance from the receiver’s side to work properly

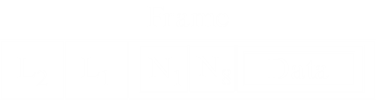
### Access Control

Access control may seem a little unfamiliar. This essentially means that if multiple users are using the network at the same time, as happens in multidrop networks, it is the data-link layer’s job to ensure everyone is able to mutually exclusively use the network. To do this, the data-link layer is divided into two sublayers, medium access control (MAC) and logical link control or data link layer. The MAC layer deals with access control.

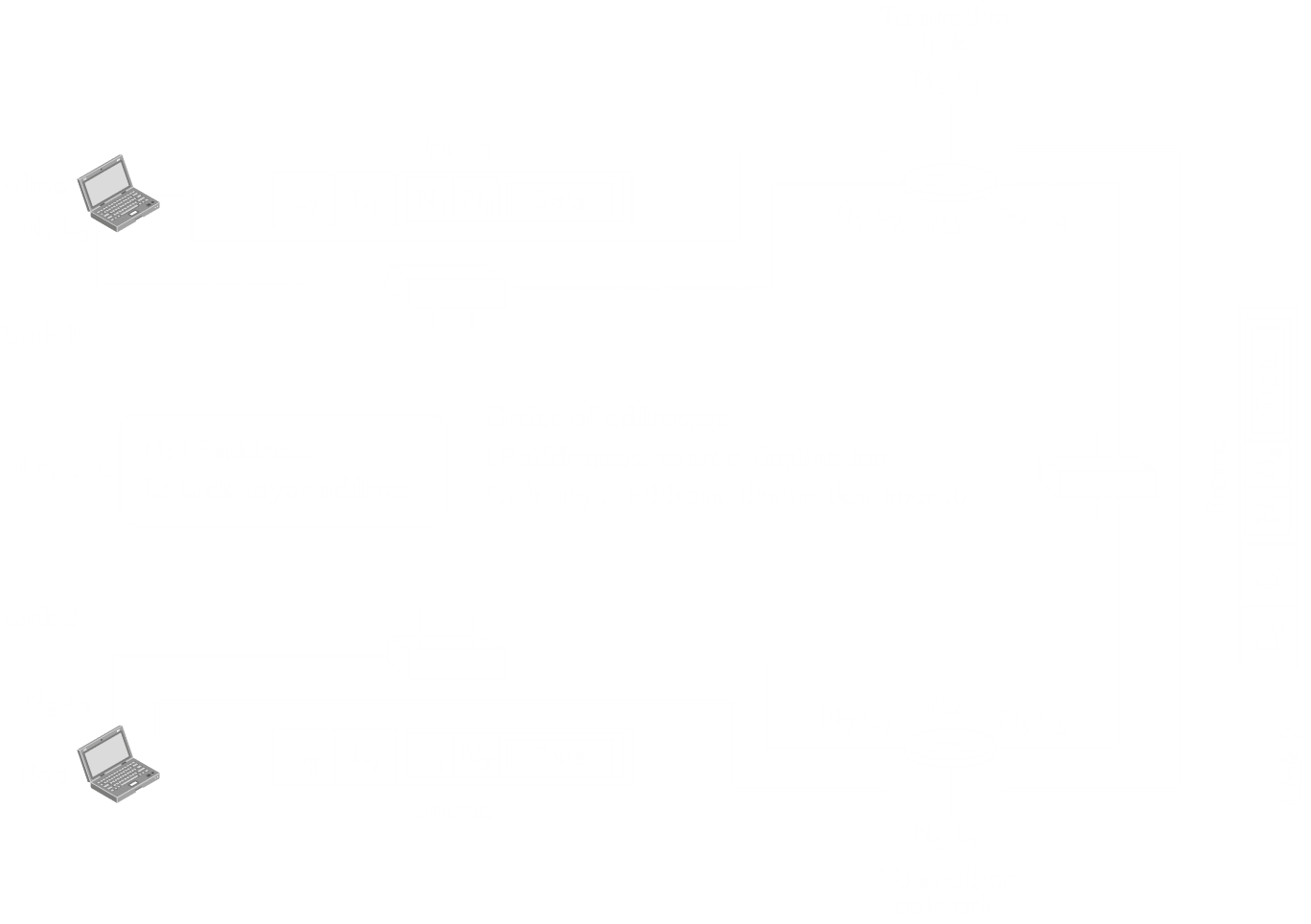
## 9.2 Link-Layer Addressing

In a frame, the way addresses are setup follows this pattern:

Destination Physical Address 🡪 Source Physical Address 🡪 Source Logical Address 🡪 Destination Logical Address 🡪 Data



The inner portion of the frame, the part with the two logical addresses and the data, is called the datagram. This portion was created by the network layer. It remains unchanged between hops.



This diagram shows an example of the path a datagram takes. It simply shows that the frames are changing at every hop.

There is one important thing to notice. Consider the router R1. When a frame was arriving at R1, the destination for the frame was L2. However, when leaving R1, the source is not L2, but rather L4. This is again because a completely different connection is being entered now and a completely new frame must be created. This frame cannot use the destination address for the previous frame as its source address because the type of connection is different. If the same address was used, the new connection would be unable to use it.

Also notice that the routers have multiple IP addresses, one at each interface. This is because an IP address or a logical address defines a point of connection in the internet. Since the router is connected at multiple points, it gets multiple IP addresses. This is like having a house at the corner of two streets with a gate at each street. To be able to identify which gate to use, we need a different address at each gate related to the street it is on.

Thus, the router has a different physical and logical address at every interface.

Another important question to answer is, how does the source know the IP address of the destination and the physical address of the next hop in the first place? The IP address of the destination is collected through the DNS service, which will be discussed in detail later on. The next hop is identified using the routing table available at the source machine, but this routing table only provides the IP address of the next hop. We still need the physical address to be able to create the frame. The physical address of the next hop is determined using the address resolution protocol (ARP), which we will discuss soon. This process occurs at every hop.

### Types of Addressing

There are three types of addressing used in the data-link layer:

1. Unicast Address
2. Multicast Address
3. Broadcast Address

All of these are physical addresses obviously. They are all 12-digit hexadecimal codes, with every 2 digits separated by colons. Thus, there are a total of 48 bits.

A unicast address is used with a unicast packet, which is intended for one entity in a link. This is one example:

A3:34:45:11:92:F1

A multicast address is used with a multicast packet, which is intended for more than one entity in a link. This is an example:

A2:34:45:11:92:F1

A question may now arise as to how we would differentiate a multicast address from a unicast one, since they are similar. If the second digit from the left is an even number, then this is a multicast address. Every two digits represents 8 bits. Thus, there are 6 octets. In the first octet, if the LSB is 0, it always produces an even number in the second digit.

A broadcast address is used with a broadcast packet, intended for every single entity on the link. The broadcast address is unique. It is all 1s. In hexadecimal, this means

FF:FF:FF:FF:FF:FF

### Address Resolution Protocol

As discussed previously, once a node has identified the next hop’s IP address using the routing table, it still needs to know its physical address. This is where the ARP comes in. The ARP sends out a broadcast packet with a request for the IP address to identify itself. Every hop on the link gets the request, but only the one who has the specified IP address responds with its physical address. This response is a unicast packet.

An example for the entire process described in this chapter is given below:

