Hello, everyone. Welcome back to the next lecture in the Computer Network series. We are currently in Part 1 of Framing. Framing is one of the services offered by the data link layer.

We will begin today's session by outlining the two outcomes. Upon completion of this session, the learner will be able to:

1. Understand framing.

2. Understand framing errors.

We will start with the question: What is framing? In the upcoming lecture, we will discuss framing in a formal way, but today, we will focus on understanding framing in a practical way through an animation.

Suppose there are two nodes. This node could be a computer, router, or switch. Let's say this is Node A, and this is Node B. Each node has an adapter, which sends data it receives from the data link layer as signals. These signals flow through a cable or other medium and are received by the physical layer of Node B.

Now, the problem arises because the application layer creates data in the form of zeros and ones. It is then passed to the transport layer, which adds its own set of zeros and ones. This is then handed over to the network layer, which adds its own header, creating what we call a packet (again, a set of zeros and ones). Finally, the data reaches the data link layer, which adds zeros and ones to both the header and trailer, forming a frame. This complete set of zeros and ones, packed into a frame, is understandable only by the data link layer.

As far as the physical layer is concerned, the frame is just a set of zeros and ones. The physical layer takes each individual bit (either a 1 or 0) and converts it into equivalent signals. Therefore, the physical layer knows nothing about the frame, but the data link layer deals exclusively with it.

When the zeros and ones are sent to the receiver, the receiver collects them. The physical layer of the receiver (the adapter) collects all the zeros and ones and must reconstruct the frame. But how does the receiver know which set of bits forms the frame? For example, if the computer sends 500 bits of data, with 50 bits belonging to Frame 1, how does the receiver know when Frame 1 ends? This is the problem we aim to solve.

We understand that between adapters, bits flow, and between nodes, frames flow. Bits flow between adapters, and frames flow between hosts or nodes.

Let’s see an animation. Imagine two computers. These two computers are "happy computers" because they are ready for data exchange. When the sender sends a sequence of bits, how does the receiver understand that these bits form Frame 1, Frame 2, Frame 3, etc.?

Before communication begins, both computers agree on a protocol. This protocol defines the rules and regulations governing the communication. Here, the protocol specifies that the start and end of the frame will be marked by the sequence "11011". When the sender computer receives a packet from the network layer, it adds a header and trailer, which contain this sequence—"11011" at both the start and the end of the frame.

Let's assume both computers agree on this protocol. Since a common protocol is established between the sender and receiver, communication can proceed effectively.

Now, the sender creates the data and the frame. This data, which comes from the network layer (the packet), will have a header and trailer. The start of the frame is "11011", and the end of the frame is also "11011". When the receiver accepts the data, it recognizes the end of the frame because it knows that "11011" marks both the start and end of the frame. Thus, the frame is accepted.

However, there is a serious problem. Let's examine it. Can you identify the issue? The problem arises when the data itself contains the sequence "11011". The sender adds the start and end sequences, but the data may also contain this same bit pattern. When the receiver sees "11011" in the data, it mistakenly assumes it marks the end of the frame.

As a result, the receiver may construct an incorrect frame. This situation is known as a framing error. We do not expect the receiver to create a frame like this, but it happens because the bit sequence "11011" appears in both the data and as a frame delimiter. This is the root of the problem.