Link-State Routing : Idea

❖ The idea behind link state routing is simple and can be stated as five parts. Each router must do the following:

1. **Discover its neighbors and learn their network addresses.**
2. **Measure the delay or cost to each of its neighbors.**
3. **Construct a Link State Packet**
4. **Send this Link State Packet to all other routers.**
5. **Compute the shortest path to every other router.**

Step-1 Discover the Neighbors & Learn the Neighbors Information

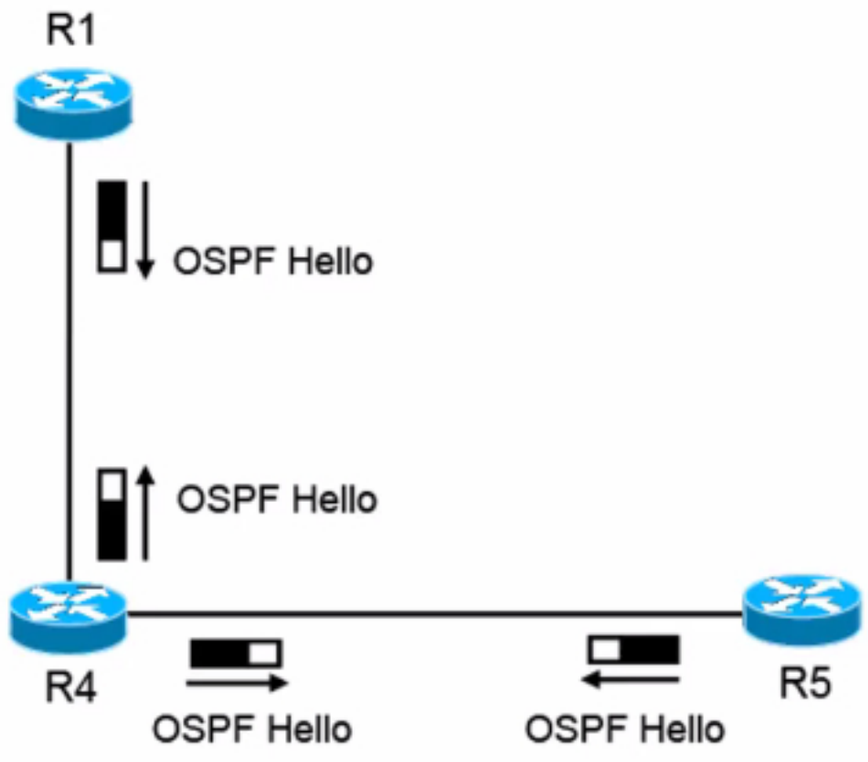
- All routers will maintain the routing tables
  - ✓ Contains **Information of entire network topology**
    - Lists of routers,
    - Information about each router's neighbors and
    - The connection between the neighbors
- Each router keeps track of its neighbors links
  - ✓ Whether the link is up or down
  - ✓ The cost on the link

Step-1 Discover the Neighbors & Learn the Neighbors Information

- Each router is responsible for
  - ✓ Meeting its neighbors
  - ✓ Learning their names (Network Addresses)
- The task to accomplish the above goal by sending a special **HELLO packet** on each point-to-point line.
- Receiving router on the other side is expected to send back a reply back by telling who it is.

Neighbors Discovery and Learning the Neighbors Information

- These names(Network Addresses) must be globally unique

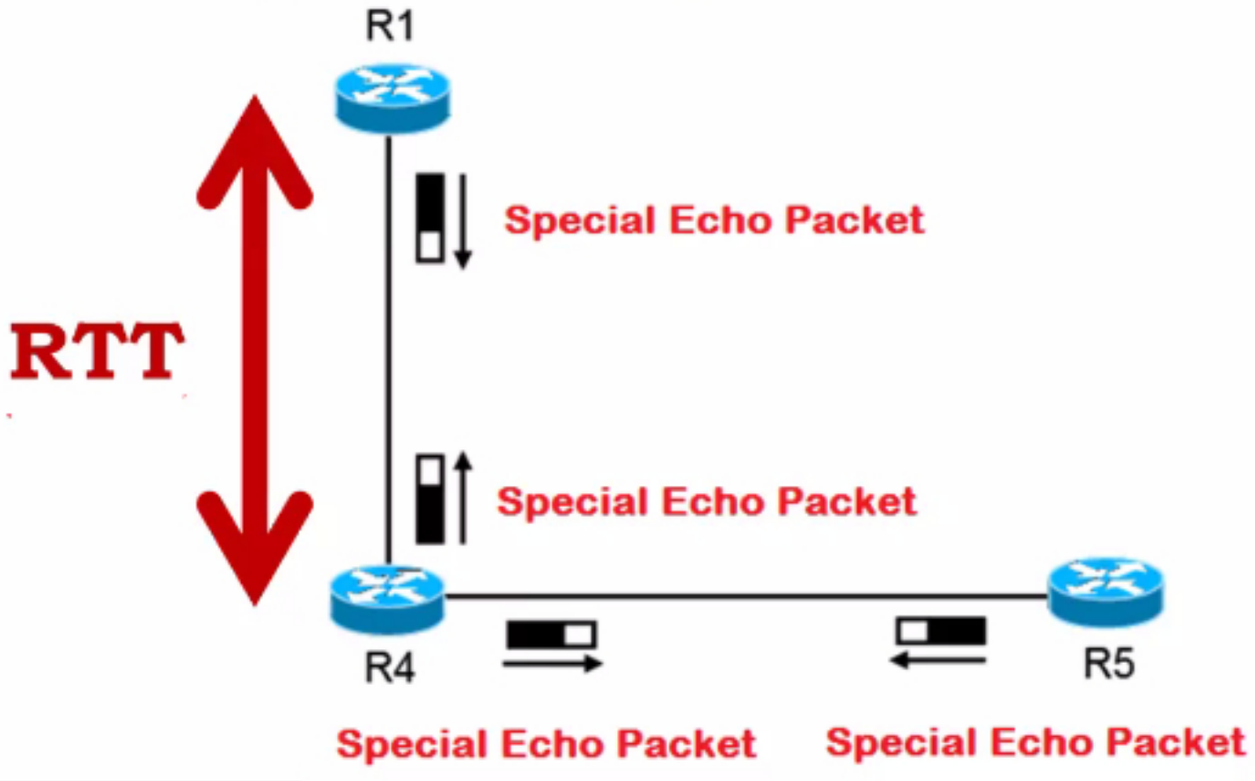


Step-2 Measure The Delay Or Cost To Each Of Its Neighbors.

- The link state routing algorithm requires each router to know the delay to each of its neighbors.
- To **determine the delay**
  - ✓ Intermediate Node/ Source Node will sent special ECHO packet to the neighbor node
  - ✓ On the other side neighbor node will reply with special ECHO packet back immediately.

Step-2 Measure The Delay Or Cost To Each Of Its Neighbors.

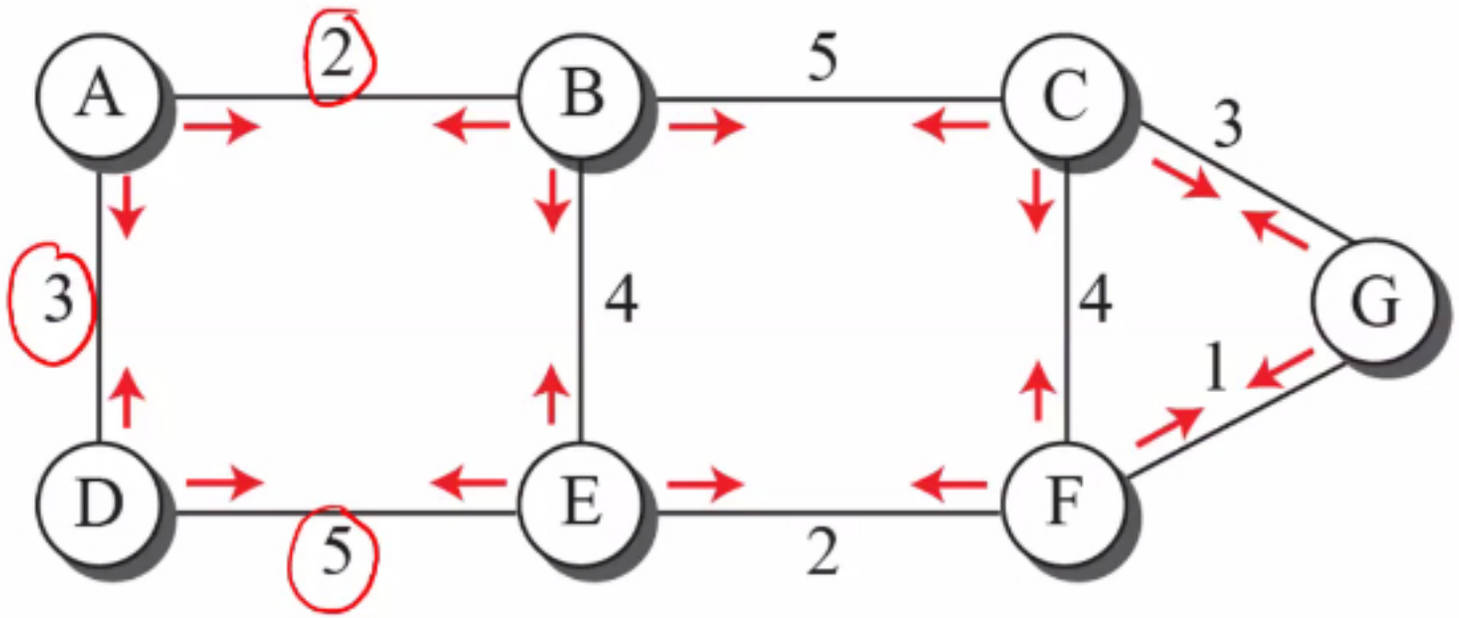
- By measuring the round-trip time and dividing it by two,
  - ✓ The sending router can get a reasonable estimate of the delay.



Step-3 Construct a Link State Packet

- The next step is for each router to build a **Link State Packet** containing all the data.
- The Link State Packet Contains
  - ✓ **Sender ID,**
  - ✓ **Sequence Number and**
  - ✓ **Age or Time-to-Live (TTL) for this packet**
  - ✓ **List of Neighbors and Cost of the link**

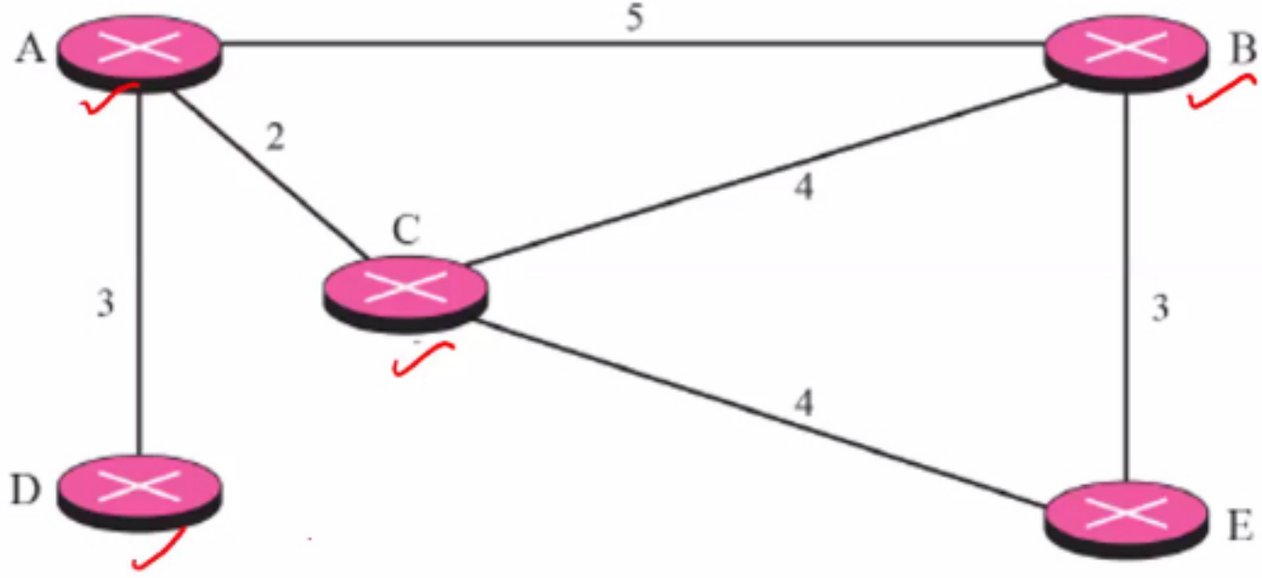
Step-3 Construct a Link State Packet



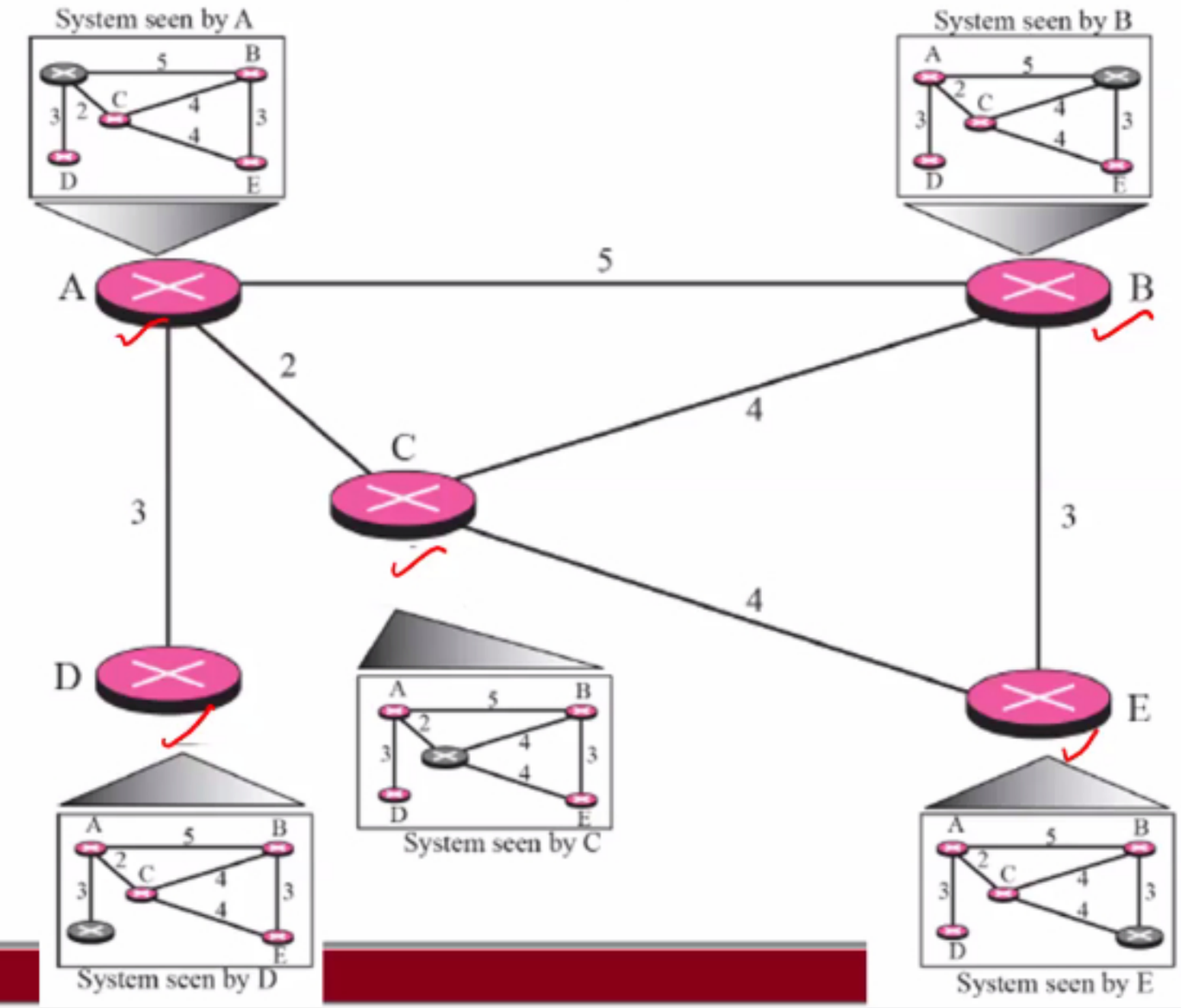
Step-3 Construct a Link State Packet

- Each router broadcasts the link state Packet
  - ✓ To give every router a **complete view of the graph**

Complete Topological Information



Complete Topological Information

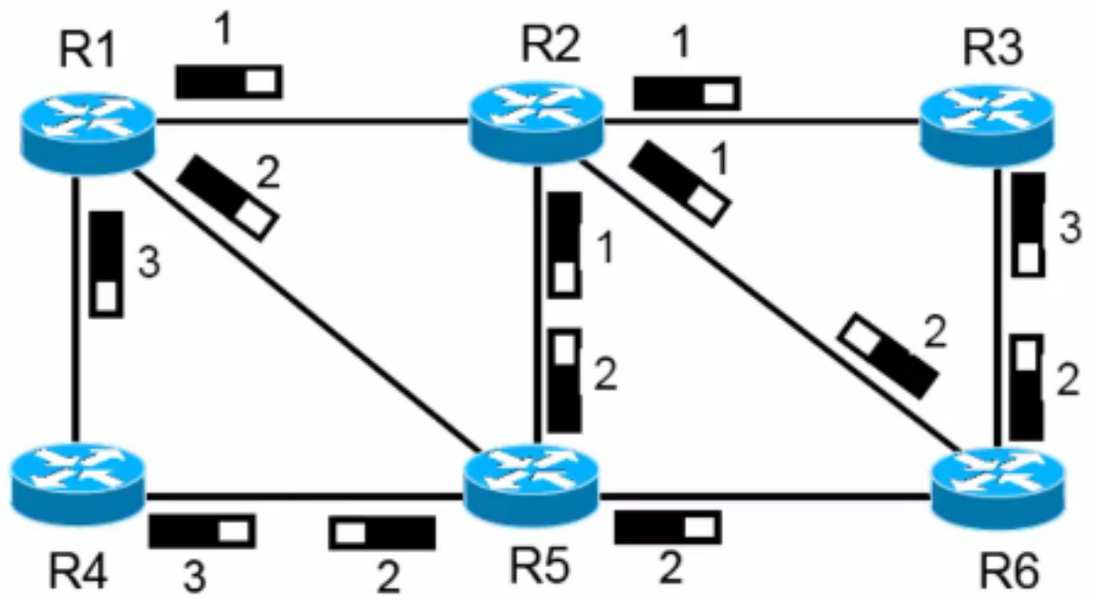


How to Distribute the Link State Packet

- The Important part of this algorithm is distributing the link state packets reliably
  - ✓ Distribution of Link State Packet (LSPs) can be difficult
- The fundamental idea to distribute the link state packets is to use **Flooding Technique**

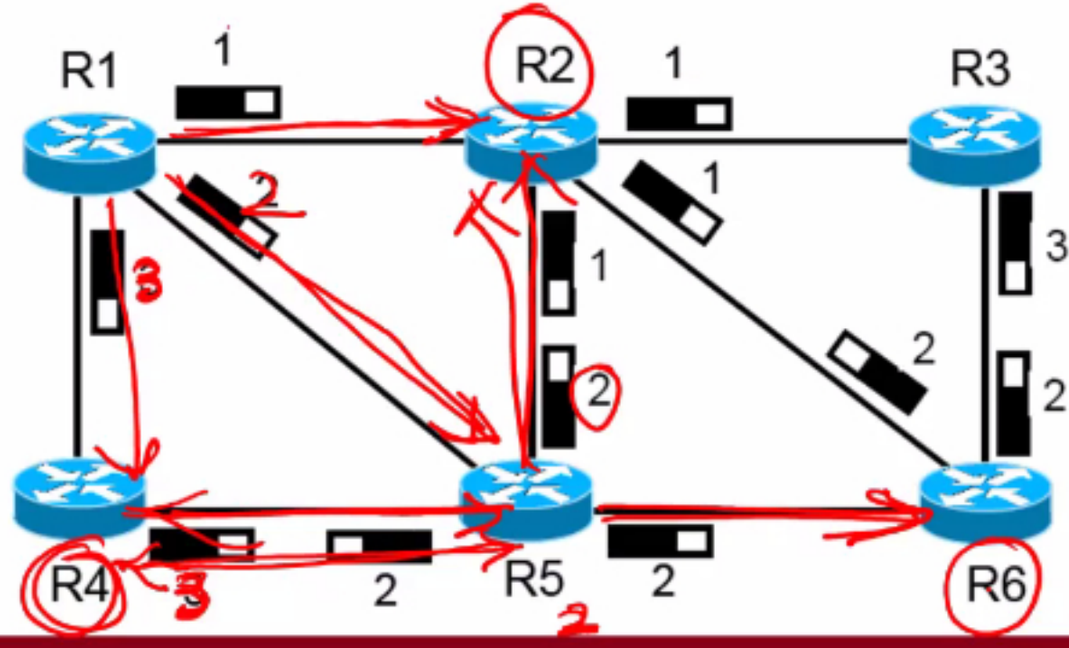
Flooding

- Each Node sends link-state information to all other links and then the next node sends out all of its links
  - ✓ Except the **one(s) where the information arrived**



Flooding

- Each Node sends link-state information to all other links and then the next node sends out all of its links
  - ✓ Except the **one(s) where the information arrived**



Reliable Flooding

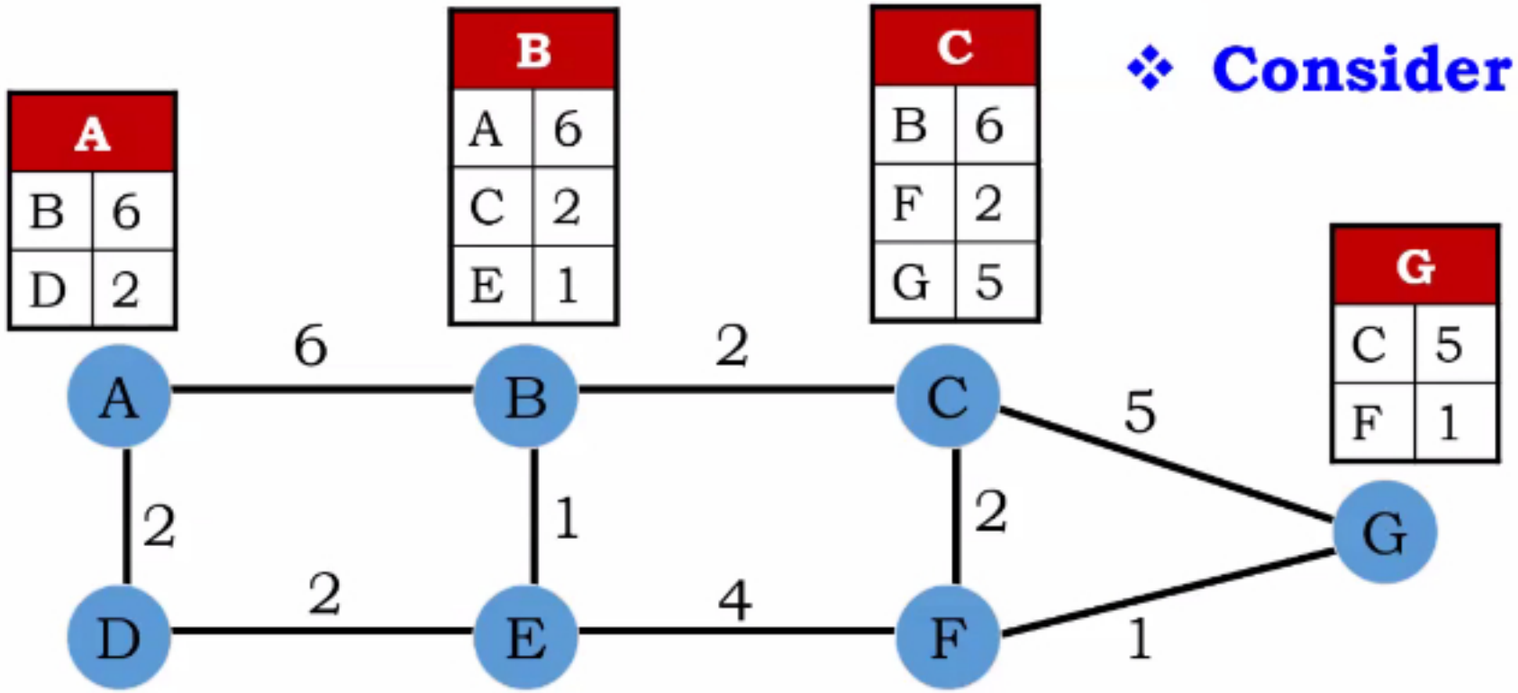
- Reliable Flooding ensure all nodes receive link-state information that has latest version
- Challenges to Reliable Flooding
  - ✓ **Packet loss**
  - ✓ **Out-of-order arrival**

Solutions: Reliable Flooding

- Sequence numbers
- Time-to-live for each packet
- Acknowledgments and Retransmissions

Dijkstra's LSR Algorithm:

❖ **Consider the following network:**



A	
B	6
D	2

B	
A	6
C	2
E	1

C	
B	6
F	2
G	5

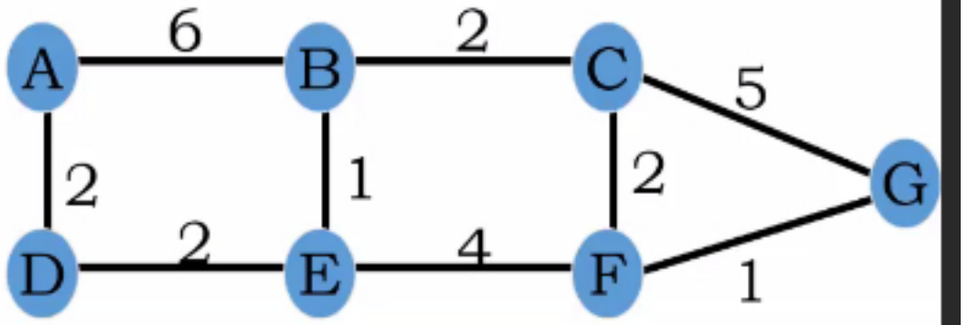
G	
C	5
F	1

D	
A	2
E	2

E	
B	1
D	2
F	4

F	
C	2
E	4
G	1

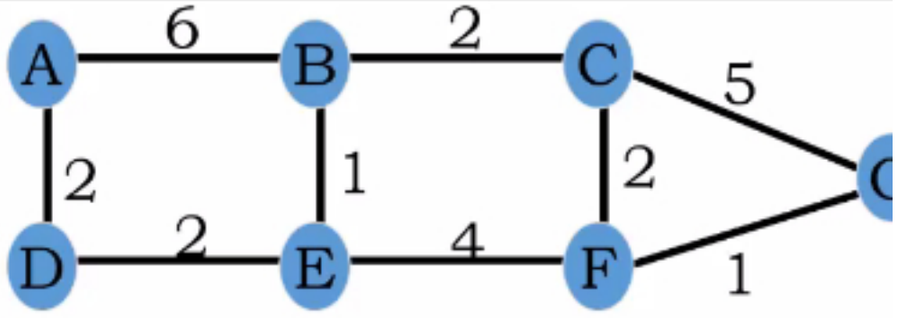
Dijkstra's LSR Algorithm:



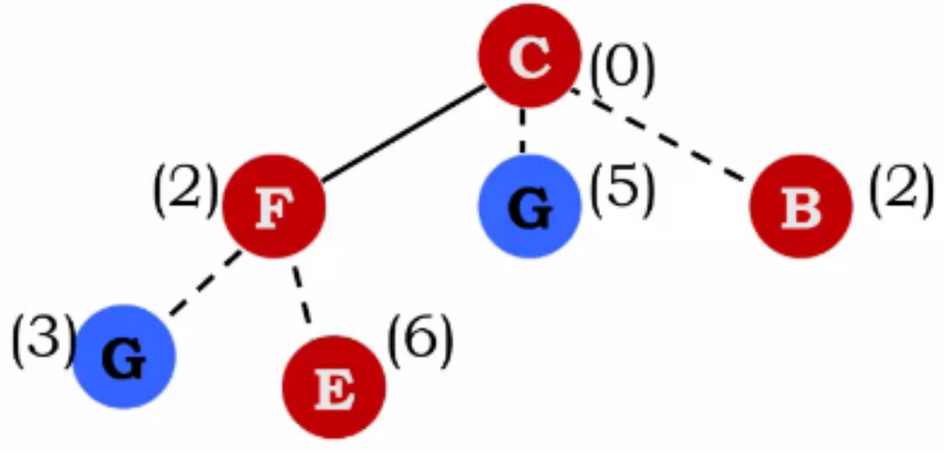
- Now, if we want to generate a PATH for C:

C (0)

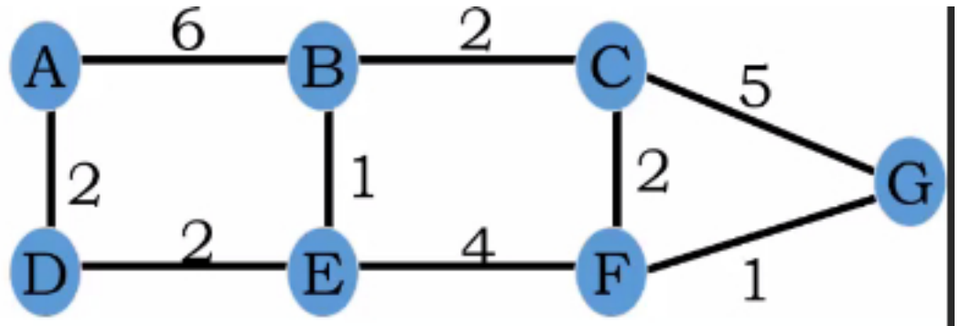
Dijkstra's LSR Algorithm:



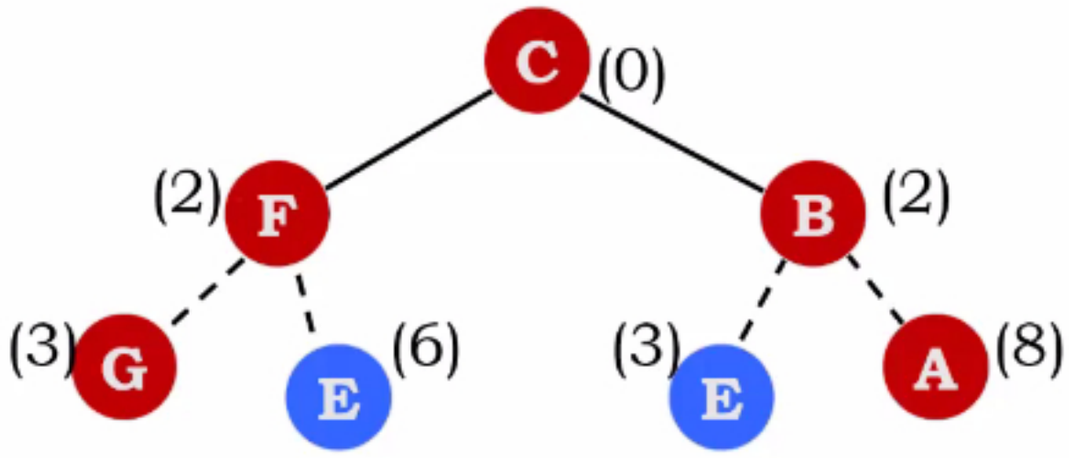
- G exists in Temp twice, keep only the best
- The **new G is a better path than the old (3 < 5)**



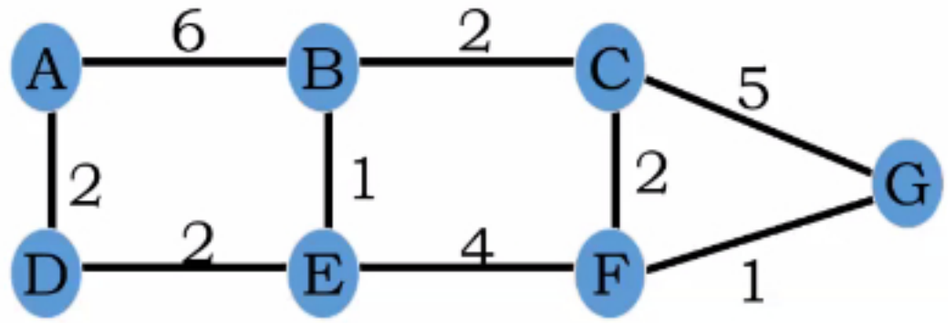
Dijkstra's LSR Algorithm:



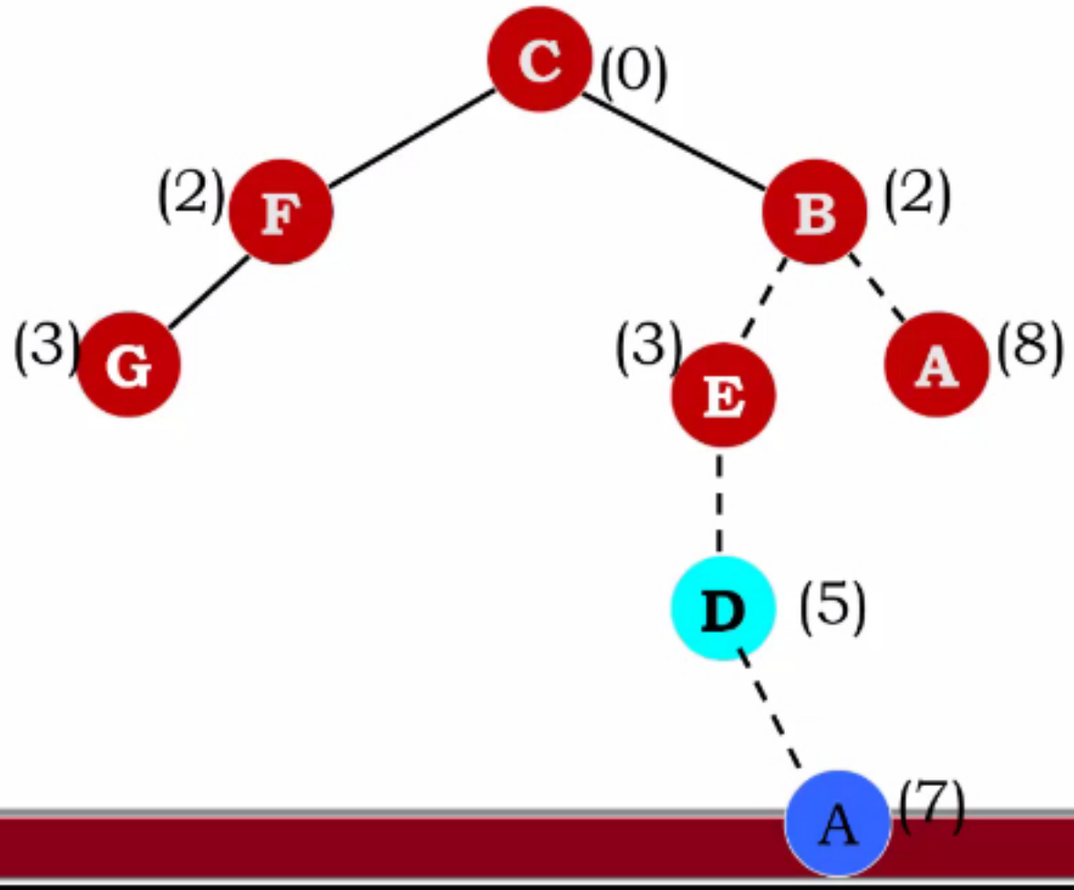
- E exists in Temp twice, keep only the best
- The new E is better than the old (3 < 6)



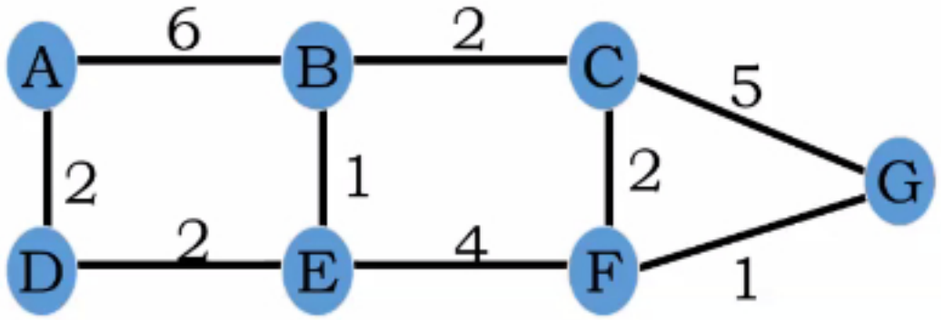
Dijkstra's LSR Algorithm:



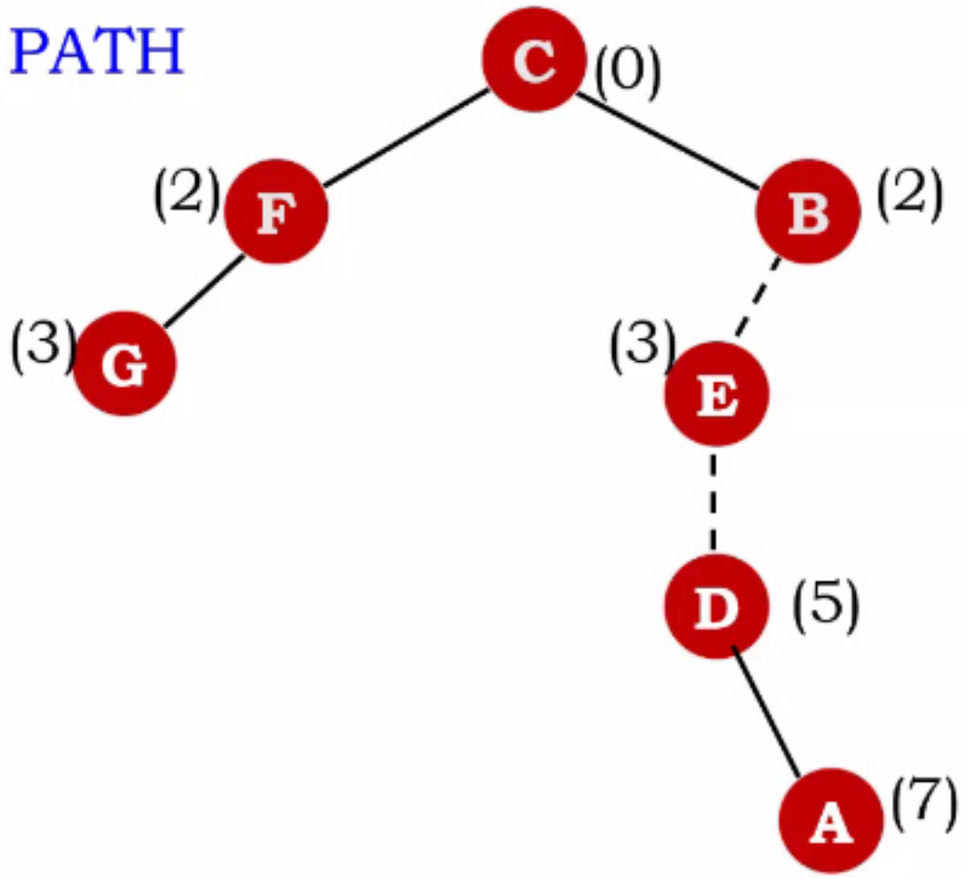
- Place D in PATH (shown as solid line)
  - ✓ Add path to A since it is better than old A



Dijkstra's LSR Algorithm:



- We are done since all routes from Temp were placed into PATH



When LSPs Are Generated And Distributed

- A time period passes
- New neighbors connect to the router
- The link cost of a neighbor has changed
- A link to a neighbor has failed (link failure)
- A neighbor has failed (node failure)

Link-State Summary

- **Good**
  - ✓ Converges relatively quickly
- **Bad**
  - ✓ Lots of information stored at each node because LSP for each node in network must be stored at each node (**scalability problem**)
  - ✓ Flooding of LSPs uses bandwidth
  - ✓ Potential security issue (if false LSP propagates)

Link-State Summary

- **Good**
  - ✓ Converges relatively quickly
- **Bad**
  - ✓ Lots of information stored at each node because LSP for each node in network must be stored at each node (**scalability problem**)
  - ✓ Flooding of LSPs uses bandwidth
  - ✓ Potential security issue (if false LSP propagates)

Goals:

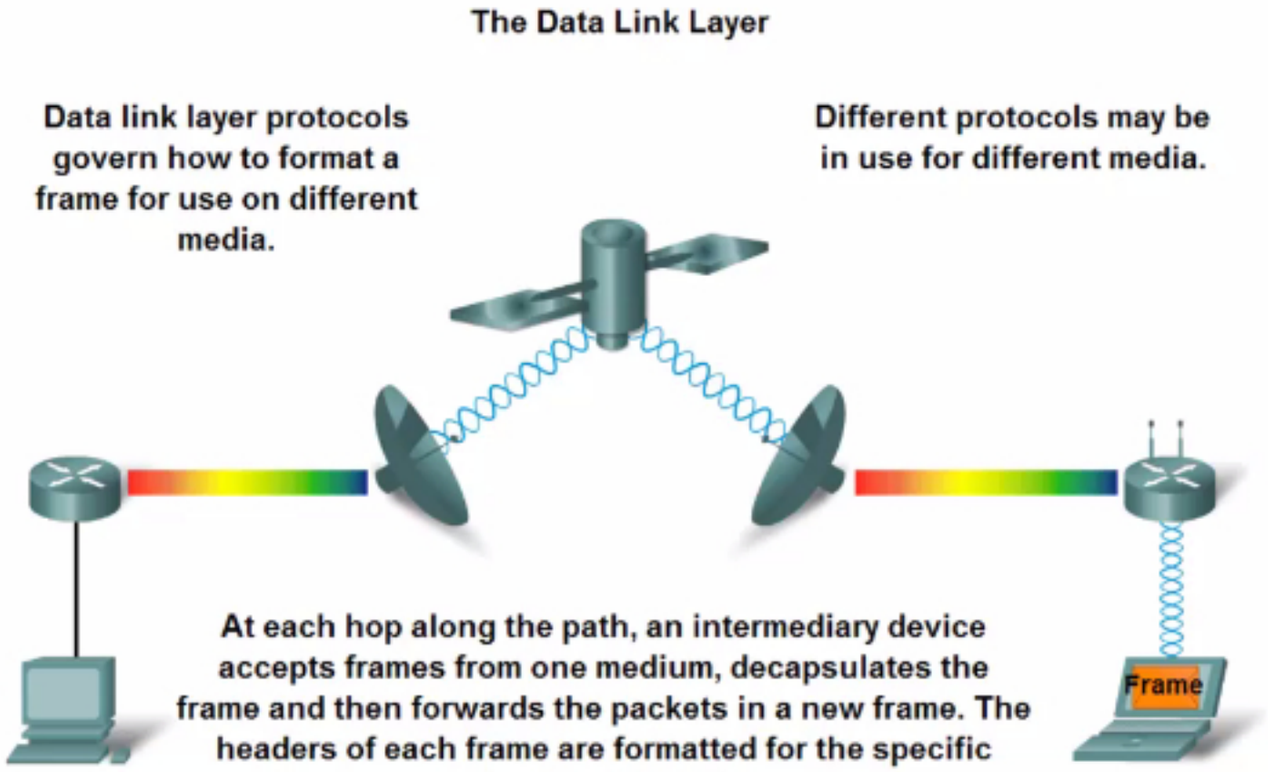
- Introduction to the data link layer
  - ✓ Framing- Determining message boundaries
  - ✓ Error control
  - ✓ Error detection
    - Parity
    - Cyclic redundancy checks (CRC)
  - ✓ Error correction
    - Row/column parity
    - Hamming codes

Goals:

- ❖ Introduction to the data link layer
  - ✓ Framing- Determining message boundaries
  - ✓ Error control
  - ✓ Error detection
    - ○ Parity
    - ○ Cyclic redundancy checks (CRC)
  - ✓ Error correction
    - ○ Row/column parity
    - ○ Hamming codes

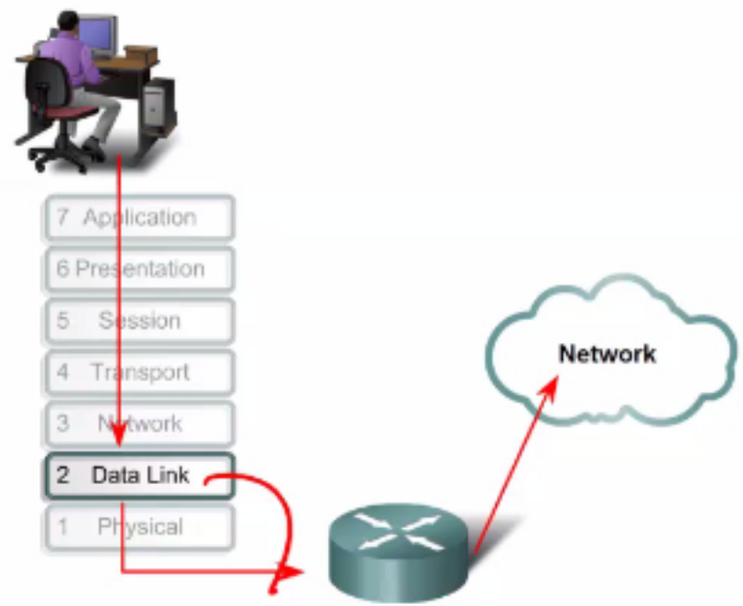
Data Link Layer

- **Service:** To reliably deliver packets between two physically connected machines.



Data Link Layer

- The Data Link layer performs two basic services:
  - ✓ Allows the upper layers to access the media using techniques such as framing
  - ✓ **Controls** how data is placed onto the media and is received from the media using techniques such as media access control and error detection

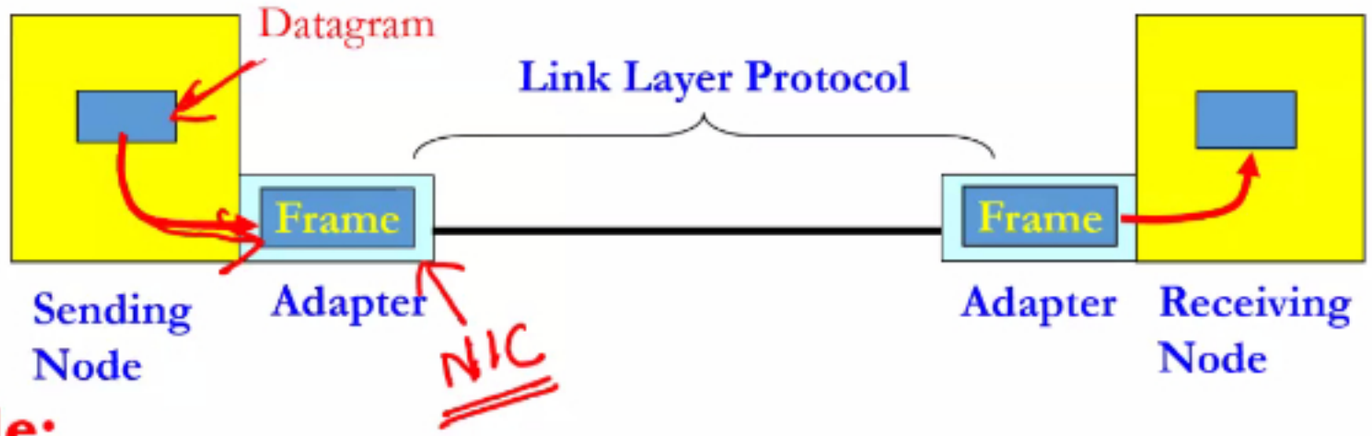


The Data Link layer prepares network data for the physical network.

Functions of Data Link Layer

- Providing a well-defined service interface to the network layer
- Determining how the bits of the physical layer are grouped into frames
- Dealing with transmission errors
- Regulating the flow of frames so that **slow receivers are not flooded by fast senders**

Digital Adaptors Communicating



- **Sending side:**
  - ✓ Encapsulates datagram in a frame
  - ✓ Adds error checking bits, flow control, etc.
- **Receiving side**
  - ✓ Looks for errors, flow control, etc.
  - ✓ Extracts datagram and passes to receiving node

Data Link Layer: Design Issues

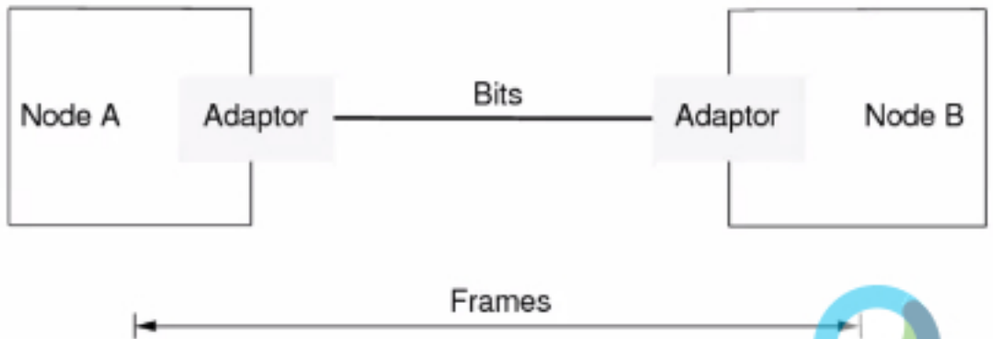
- **Services provided to network layer:**
  - ✓ Unacknowledged connectionless service.
  - ✓ Acknowledged connectionless service.
  - ✓ Acknowledged connection-oriented service.
- **Framing:**
  - ✓ How to determine the start and end of a bit stream on the physical link?

Data Link Layer : Design Issues

- **Error control:**
  - ✓ Error correcting codes.
  - ✓ Error detecting codes.
- **Flow control:**
  - ✓ Ensuring that receiver can keep up with transmission.

Why Framing?

- The **physical layer** provides bit synchronization to ensure that the sender and receiver use the same bit duration and timing.
- Framing is necessary for the purpose of synchronization and data control functions.
- Framing is the process of grouping the bits into frames (messages or packets)



Why Framing?

- How to interpret a continuous stream of bits as a series of frames?

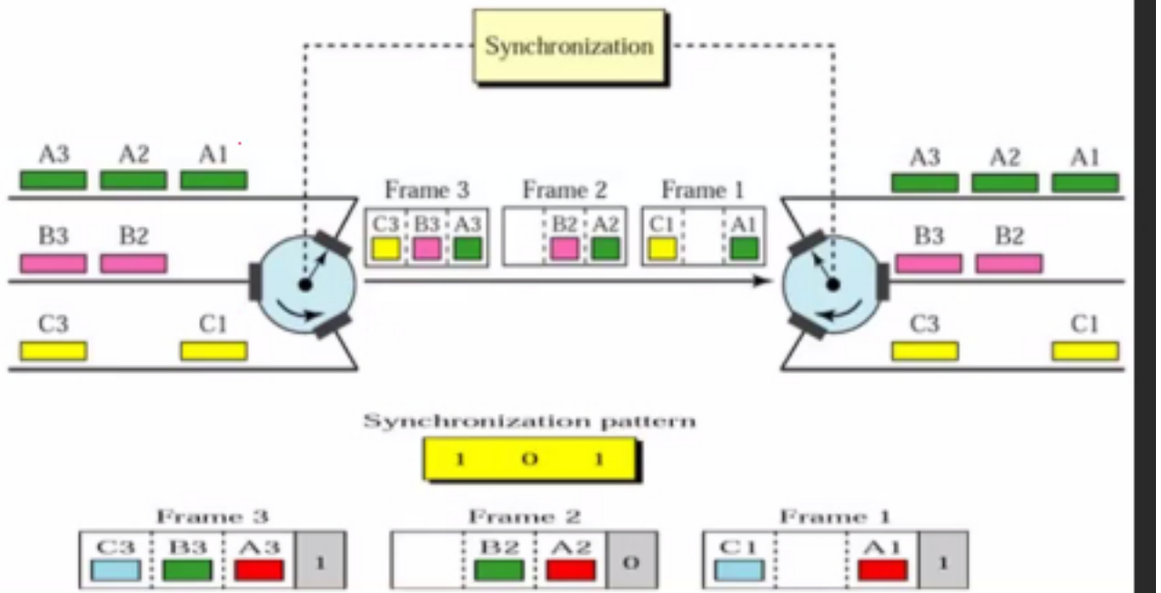
From Continuous Stream Of Bits

01111000010111001111011110001000010

To Series Of Frames

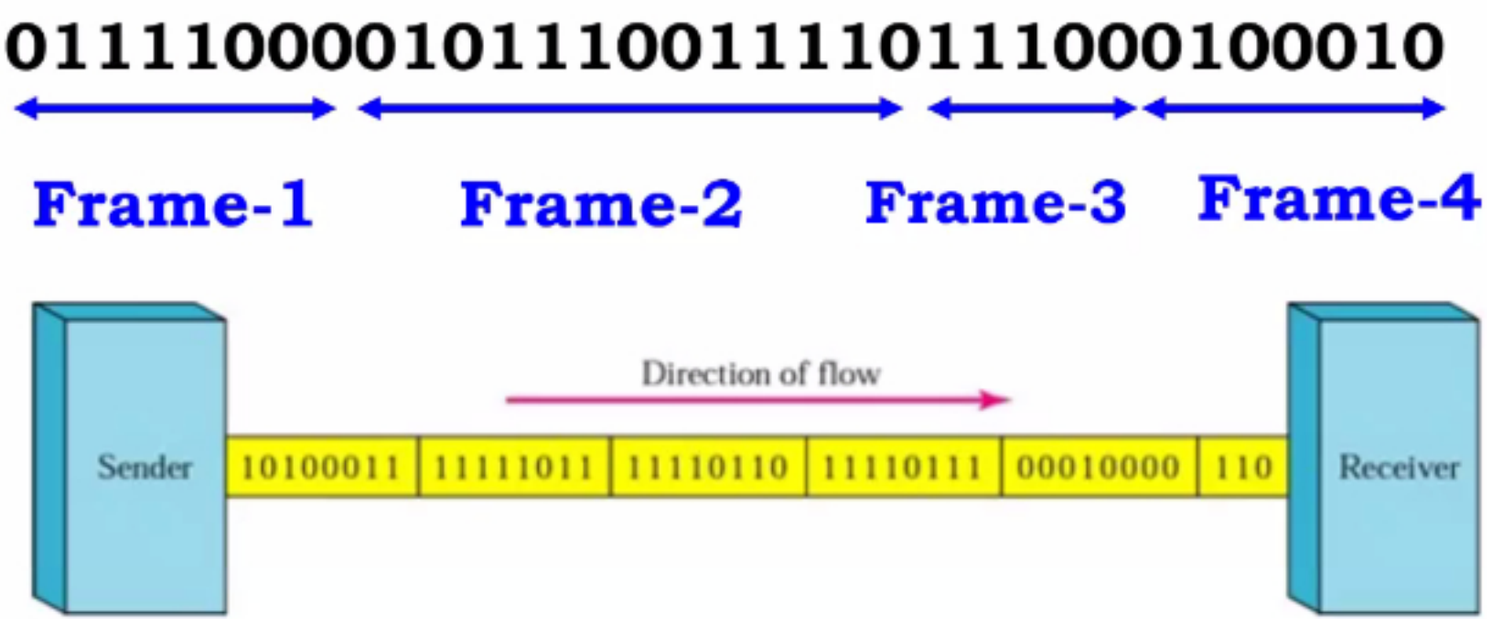
Synchronization

- Data **sent by a sender** in bit-serial form through a medium must be correctly interpreted at the **receiving end.**
  - Requires the **beginning, the end, logic level and duration of each bit** as sent at the transmitting end must be recognized at the receiving end.



Synchronization

- One basic question is **how the receiver knows the Beginning and End of a Frame.**

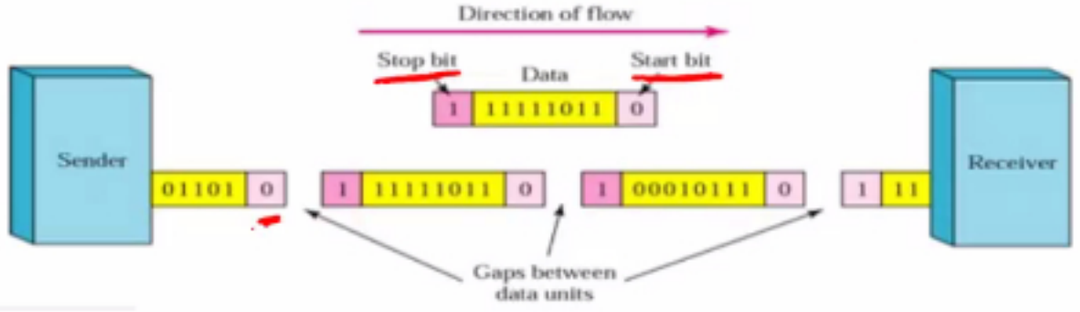


Synchronization

- One basic question is **how the receiver knows the Beginning and End of a Frame.**

01111000010111001111011110001000010

Frame-1 **Frame-2** **Frame-3** **Frame-4**

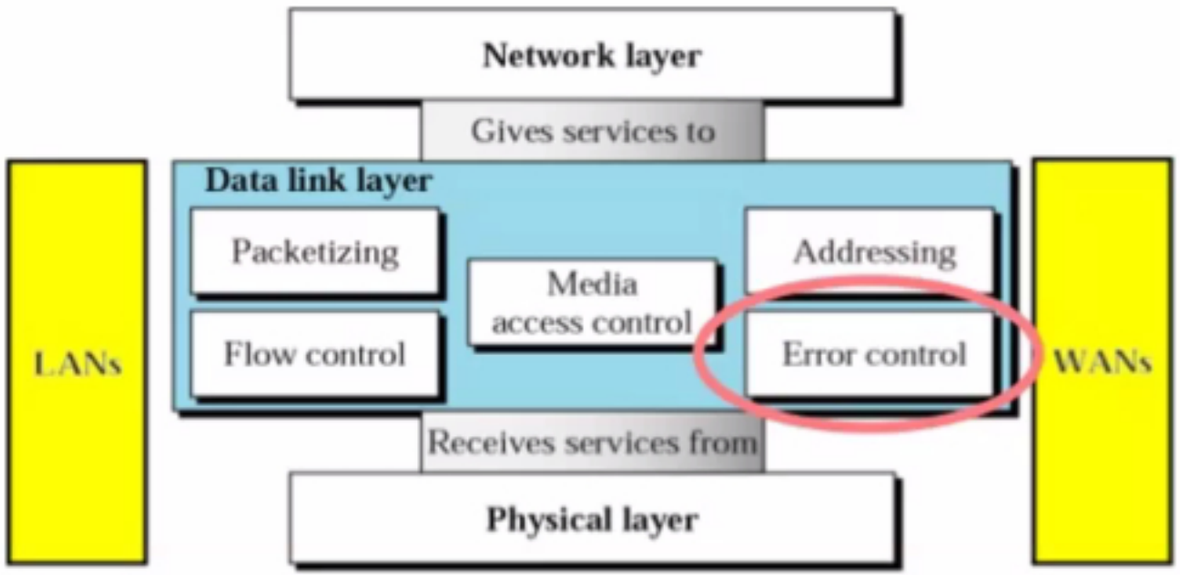


Solution

- Framing will encapsulate the packets
  - ✓ Frame are separated from one another
  - ✓ Identifies the idle period of a data link transmission.
  - ✓ Adds checksum at the end of each frame such that the receiver can detect errors

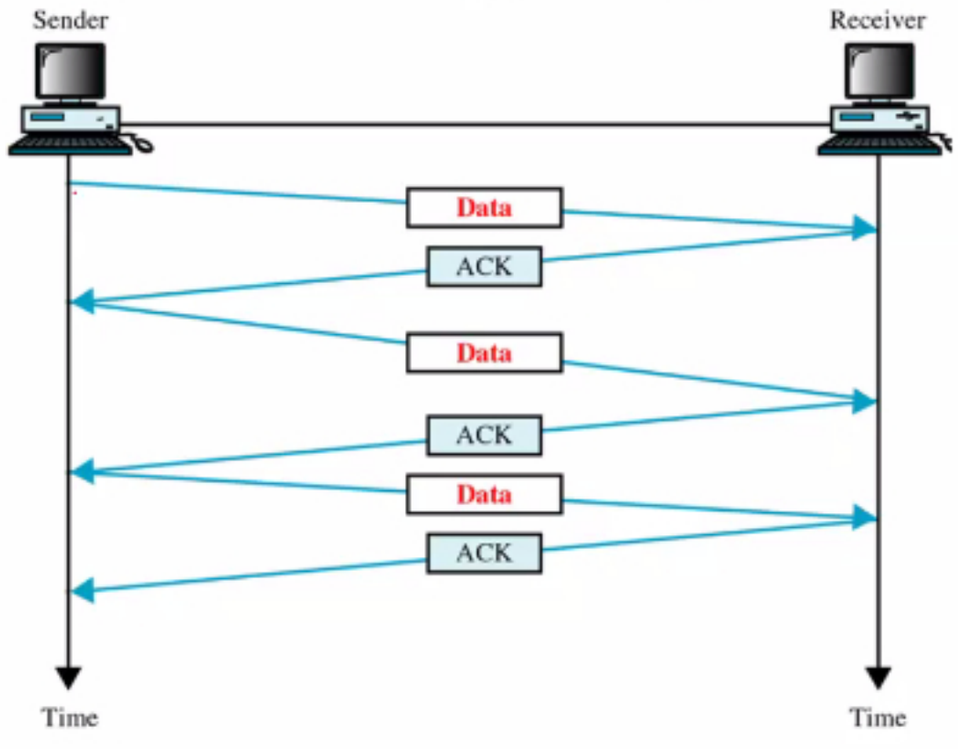
Data Link Control

- The most important responsibilities of the data link layer are flow control and error control.
  - ✓ Collectively, these functions are known as **Data Link Control.**



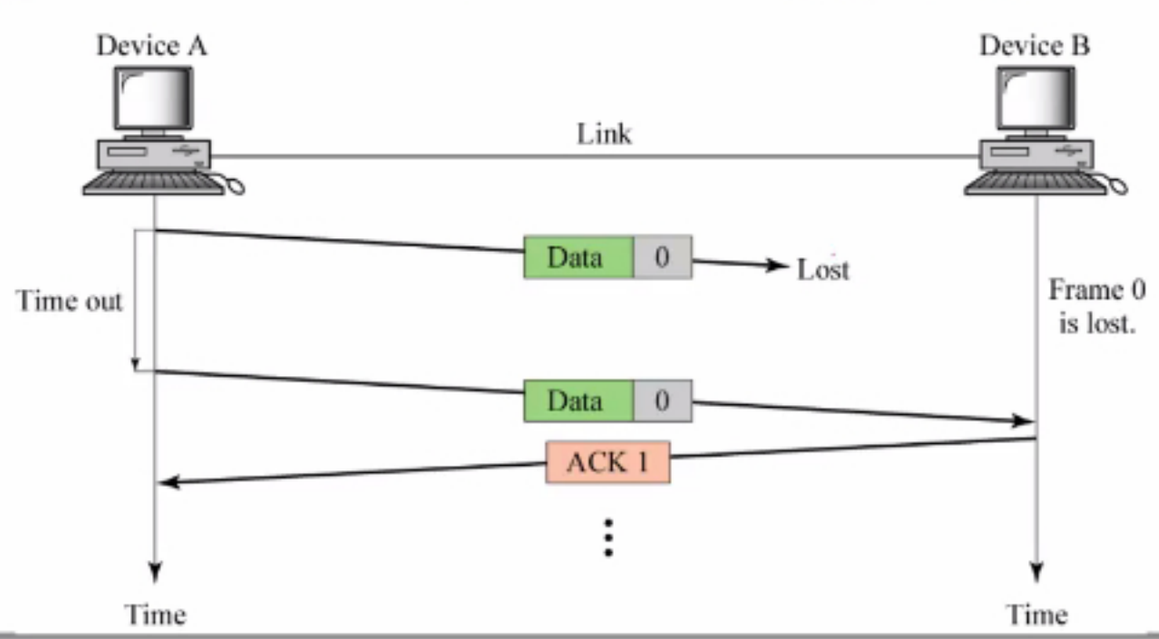
Error Control

- The usual way to ensure reliable delivery is to provide the sender with some feedback about what is happening at the destination side.



Error Control : Timers

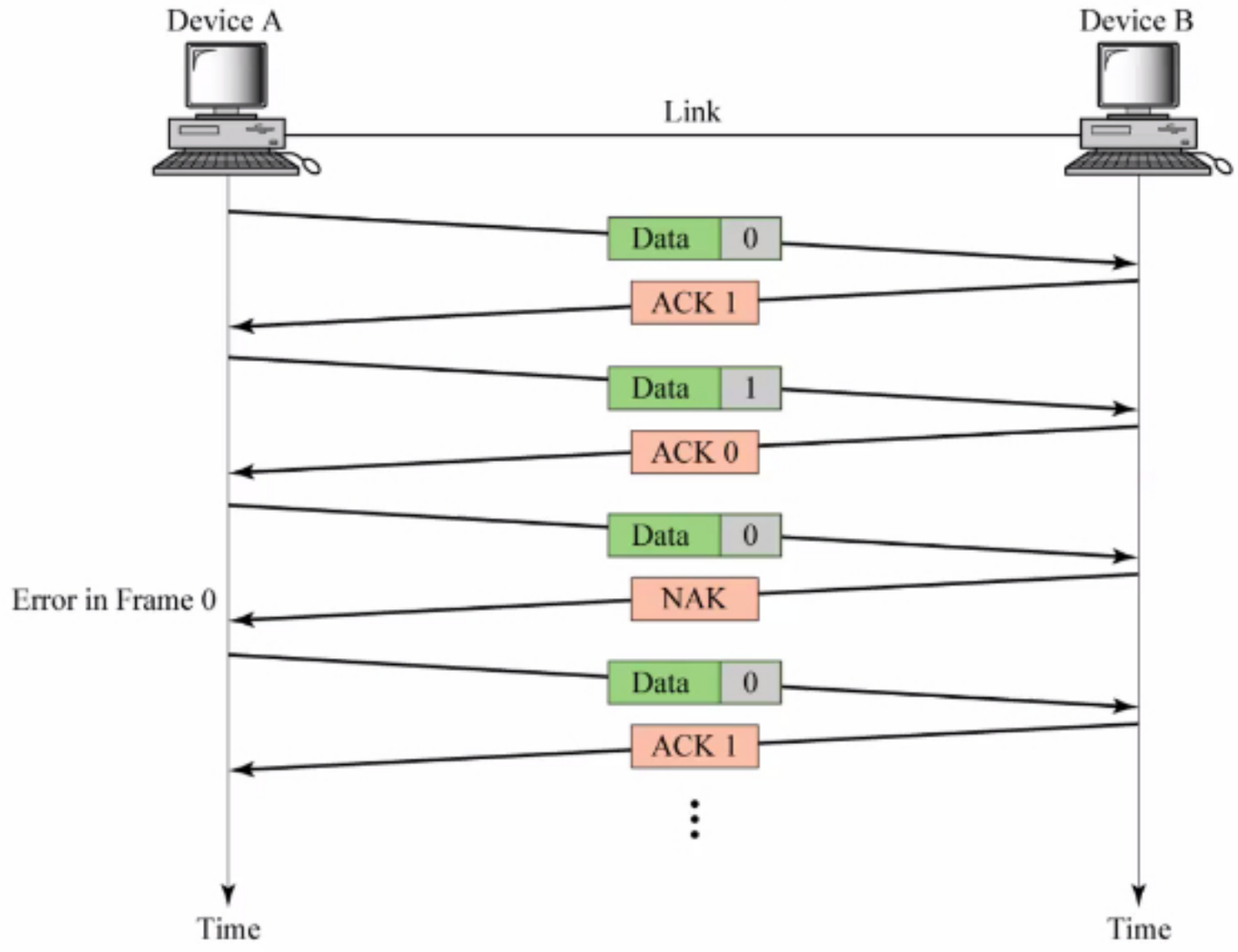
- What happens if frame to vanish completely due to hardware troubling
  - ✓ The **receiver will not react at all.**
- Data Link Layer should introduce the **TIMERS**



Error Control

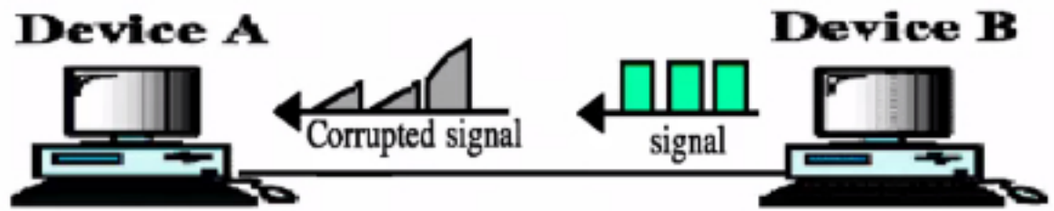
- When will happen **when the frames are transmitted multiple times by the sender and that the receiver will accept the same frame two or more times**
  - **Assign SEQUENCE NUMBERS to outgoing frames,** so that the receiver can distinguish retransmissions from originals.

Error Control: Sequence Number



Error Detection And Error Correction

- Data can be corrupted during transmission.
- For reliable communication
  - ✓ Errors must be detected and corrected.



Some basic responsibilities of the data link layer is to **provide a reliable communication for the upper layer**

Why Error Detection/Error Correction

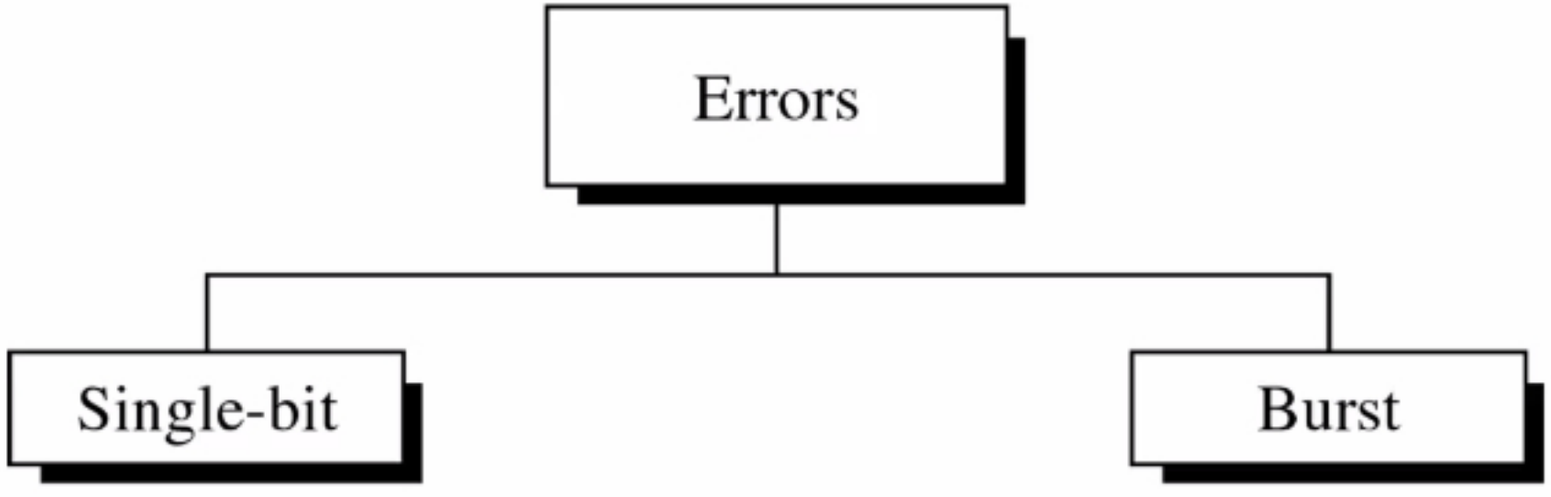
- Environmental interference and physical defects in the communication medium
  - ✓ Cause **RANDOM BIT ERRORS** during data transmission.
- Some applications require that errors be **detected and corrected.**
  - ✓ **Detection:** determining if an error has occurred.
  - ✓ **Correction:** actually fixing errors.

Types of Errors

- There are TWO types of Bit Errors in a Link

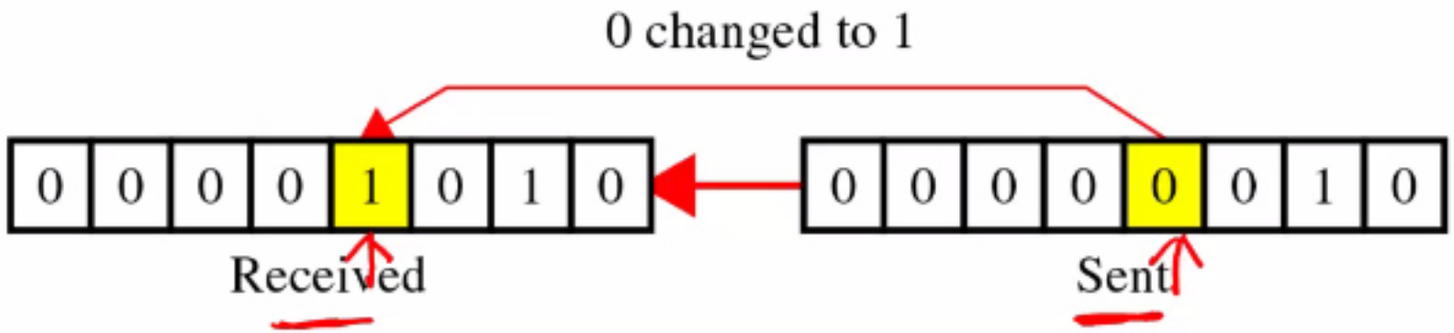
1. **Single-Bit Error**

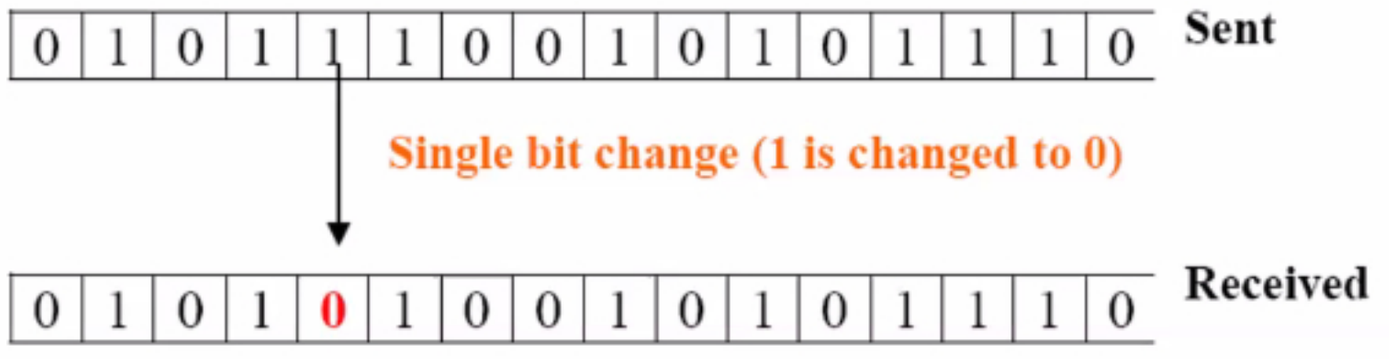
2. **Burst Error**



Single-Bit Error

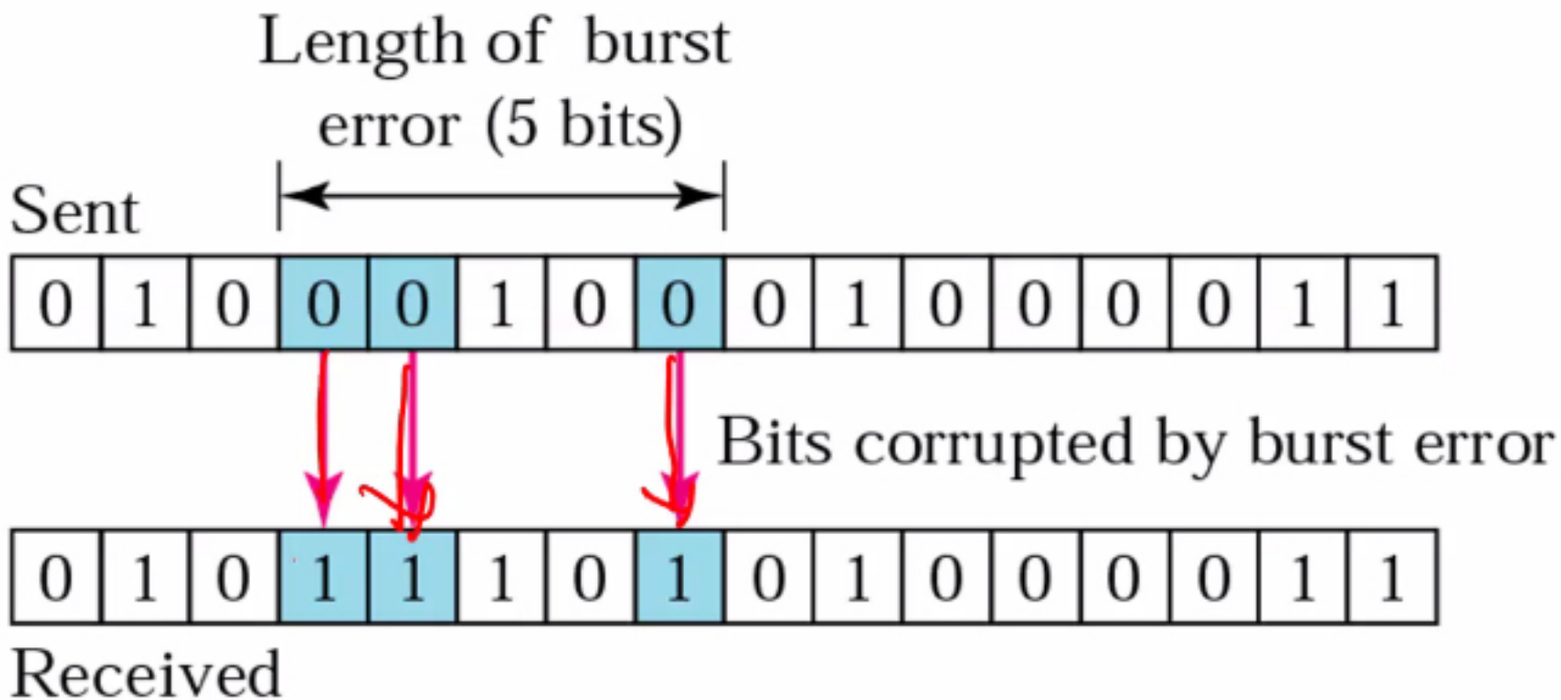
- In a single-bit error, only one bit in the data unit has changed.





Burst Error

- A burst error means that 2 or more bits in the data unit have changed



Dealing with Errors

- Receiver must be aware that an error occurred in a frame
  - ✓ Need to have an **ERROR DETECTION MECHANISM**
- Add some information to correct errors
  - ✓ **ERROR CORRECTING MECHANISM**
- Ask sender to re-send frame (retransmission strategies)

In practice all are employed