Data Link Layer

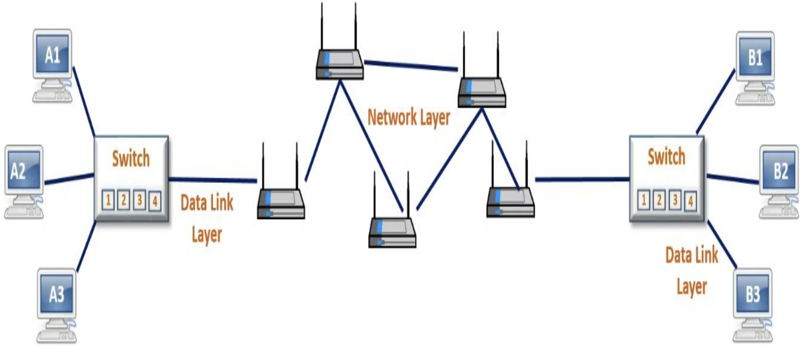
Er. Pratikshya Shrestha

Data Link Layer

- Data link layer takes packets from the network layer and encapsulates them into frames by adding some extra information like **MAC addresses** of the source and destination machines.
- When a data packet arrives at the destination network, then the data link layer is **responsible to direct the data to its correct host by using its MAC address.**
- There are two types of addressing used in computer networks, logical addressing and physical addressing.
- **Logical addressing** is used in the **network layer**, where IP addresses are used to find the host network.
- **Physical addressing** is used in the **data link layer**, where MAC addresses are used to find the host device on the network.

Data link layer

For example: Computer Al sends the data to computer B1 on another network. Now the router is used to find the host network by using its IP addresses (Logical addressing), after finding the host network the switch devices are used to find the correct host device on the network by using its MAC addresses.



Functions of data link layer

- Physical Addressing
- Framing
- Flow Control
- Error Control

Physical addressing

- Data link layer use physical addresses to transfer the data on the network.
- Physical addresses are the MAC addresses, it's a hardware identification number that uniquely identifies each device on a network.

Framing

- When a packet comes from the network layer to the data link layer, it adds header and tailor to the packet and encapsulated them into frames, then each frame sends bit-by-bit on the hardware. At the receiving end, the data link layer picks up the signals from the hardware and assembles them into frames.
- Header contains MAC addresses of source and destination NIC's devices and tailor contains some bits, which are used to detect errors on the network.

Flow control

- Data link layer is responsible for data flow control. Data transfer rate must be at the same speed else the data may be corrupted, the data link layer use methods to control the flow of data on both sides.

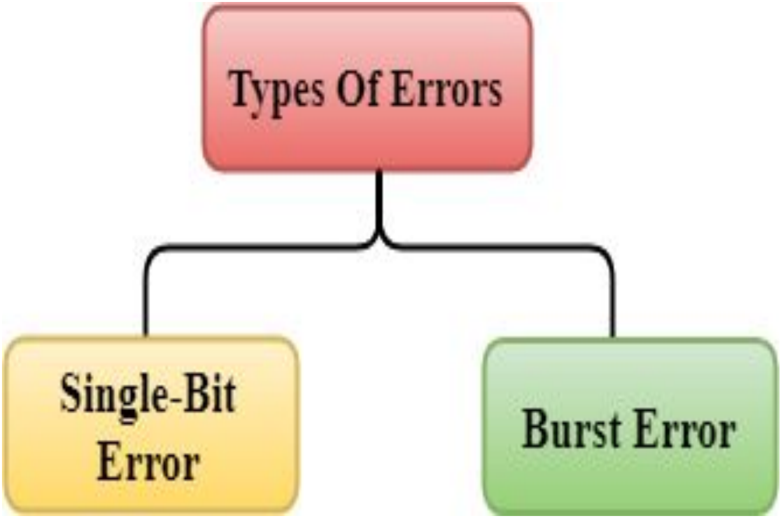
Error control

- This layer also provides an error control mechanism, it detects errors from the frames and retransmits frames that are damaged, duplicated, or lost.

Error control mechanism

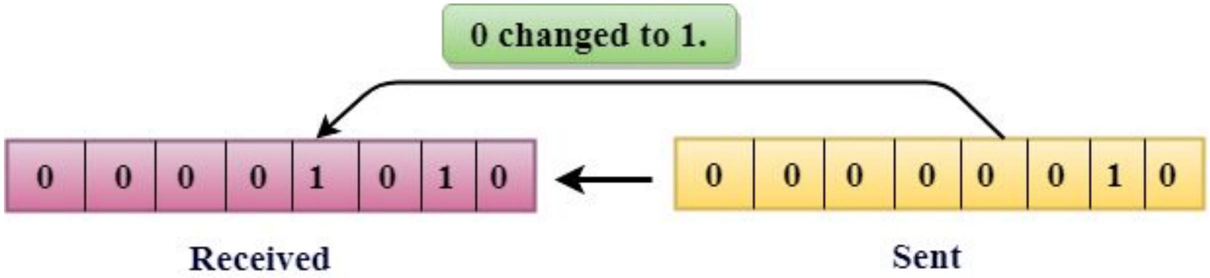
Error Detection

When data is transmitted from one device to another device, the system does not guarantee whether the data received by the device is identical to the data transmitted by another device. An Error is a situation when the message received at the receiver end is not identical to the message transmitted.



Single bit error

- The only one bit of a given data unit is changed from 1 to 0 or from 0 to 1.
- In the figure below, the message which is sent is corrupted as single-bit, i.e., 0 bit is changed to 1.



0 changed to 1.

0 0 0 0 1 0 1 0

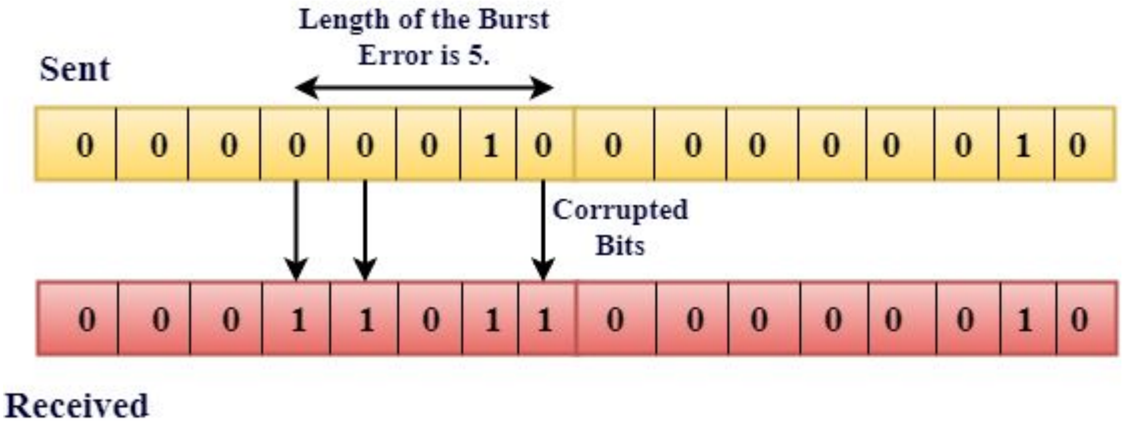
Received

0 0 0 0 0 0 1 0

Sent

Burst error

- The two or more bits are changed from 0 to 1 or from 1 to 0 is known as Burst Error.
- The Burst Error is determined from the first corrupted bit to the last corrupted bit.
- Note that burst error doesn’t necessary means that error occurs in consecutive bits. The length of the burst error is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not be corrupted.



Sent	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
Received	0	0	0	1	1	0	1	1	0	0	0	0	0	0	1	0

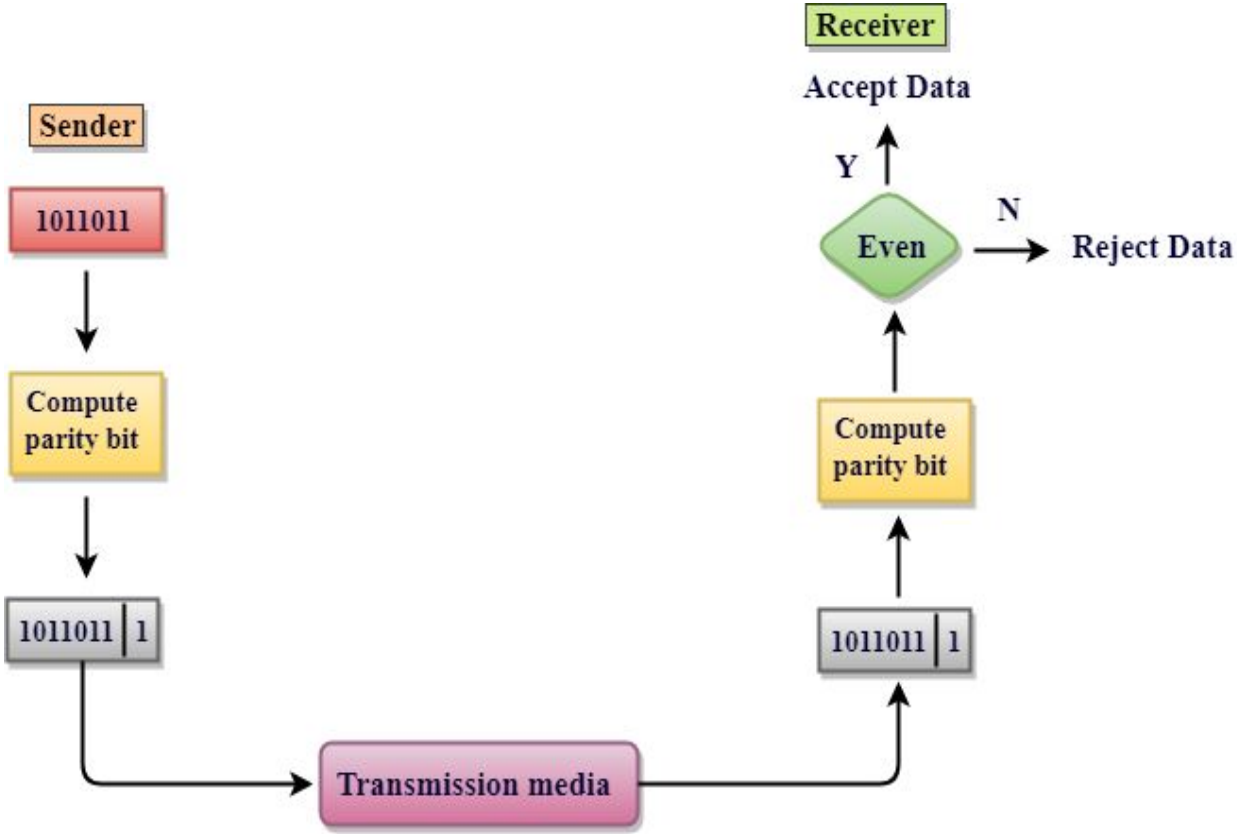
Error detecting techniques

The most popular Error Detecting Techniques are:

- Single parity check
- Two-dimensional parity check
- Checksum
- Cyclic redundancy check

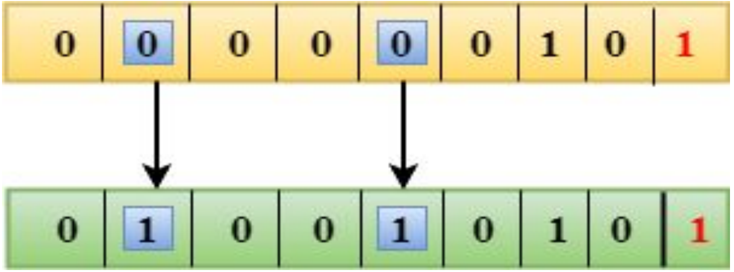
Single parity check

- Single Parity checking is the simple mechanism to detect the errors.
- In this technique, a redundant bit is also known as a parity bit which is appended at the end of the data unit so that the number of 1s becomes even.
- If the number of 1s bits is odd, then parity bit 1 is appended and if the number of 1s bits is even, then parity bit 0 is appended at the end of the data unit.
- At the receiving end, the parity bit is calculated from the received data bits and compared with the received parity bit.
- This technique generates the total number of 1s even, so it is known as even-parity checking.


Sender
1011011
Compute parity bit
1011011 | 1
Transmission media
1011011 | 1
Compute parity bit
Even
Y
Accept Data
N
Reject Data
Receiver

DRAWBACKS

- It can only detect single-bit errors which are very rare.
- If two bits are interchanged, then it cannot detect the errors.



0 0 0 0 0 0 1 0 1

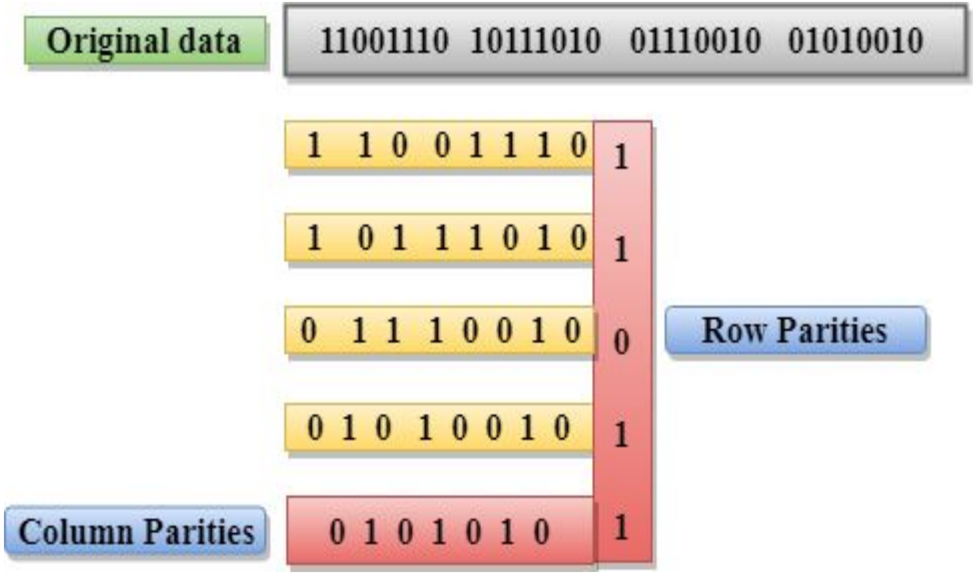
0 1 0 0 1 0 1 0 1

Two dimensional parity check

- Performance can be improved by using **Two-Dimensional Parity Check** which organizes the data in the form of a table.
- Parity check bits are computed for each row, which is equivalent to the single-parity check.
- In Two-Dimensional Parity check, a block of bits is divided into rows, and the redundant row of bits is added to the whole block.
- At the receiving end, the parity bits are compared with the parity bits computed from the received data.

DRAWBACKS

- If two bits in one data unit are corrupted and two bits exactly the same position in another data unit are also corrupted, then 2D Parity checker will not be able to detect the error.



Original data

11001110 10111010 01110010 01010010

1 1 0 0 1 1 1 0 1

1 0 1 1 1 0 1 0 1

0 1 1 1 0 0 1 0 0

0 1 0 1 0 0 1 0 1

0 1 0 1 0 1 0 1

Row Parities

Column Parities

CHECKSUM

A Checksum is an error detection technique based on the concept of redundancy.

It is divided into two parts:

- **Checksum Generator**
- **Checksum Checker**

Checksum generator

A Checksum is generated at the sending side. Checksum generator subdivides the data into equal segments of n bits each, and all these segments are added together by using one's complement arithmetic. The sum is complemented and appended to the original data, known as checksum field. The extended data is transmitted across the network.

The Sender follows the given steps:

1. The block unit is divided into k sections, and each of n bits.
2. All the k sections are added together by using one's complement to get the sum.
3. The sum is complemented and it becomes the checksum field.
4. The original data and checksum field are sent across the network.

Checksum checker

A Checksum is verified at the receiving side. The receiver subdivides the incoming data into equal segments of n bits each, and all these segments are added together, and then this sum is complemented. If the complement of the sum is zero, then the data is accepted otherwise data is rejected.

The Receiver follows the given steps:

1. The block unit is divided into k sections and each of n bits.
2. All the k sections are added together by using one's complement algorithm to get the sum.
3. The sum is complemented.
4. If the result of the sum is zero, then the data is accepted otherwise the data is discarded.

Cyclic redundancy check(crc)

- An error detection mechanism in which a special number is appended to a block of data in order to detect any changes introduced during storage (or transmission).
- The CRe is recalculated on retrieval (or reception) and compared to the value originally transmitted, which can reveal certain types of error.
- For example, a single corrupted bit in the data results in a one-bit change in the calculated CRC, but multiple corrupt bits may cancel each other out.

Requirements of CRC

- A CRC will be valid if and only if it satisfies the following requirements:

  1. It should have exactly one less bit than divisor.

  2. Appending the CRC to the end of the data unit should result in the bit sequence which is exactly divisible by the divisor.

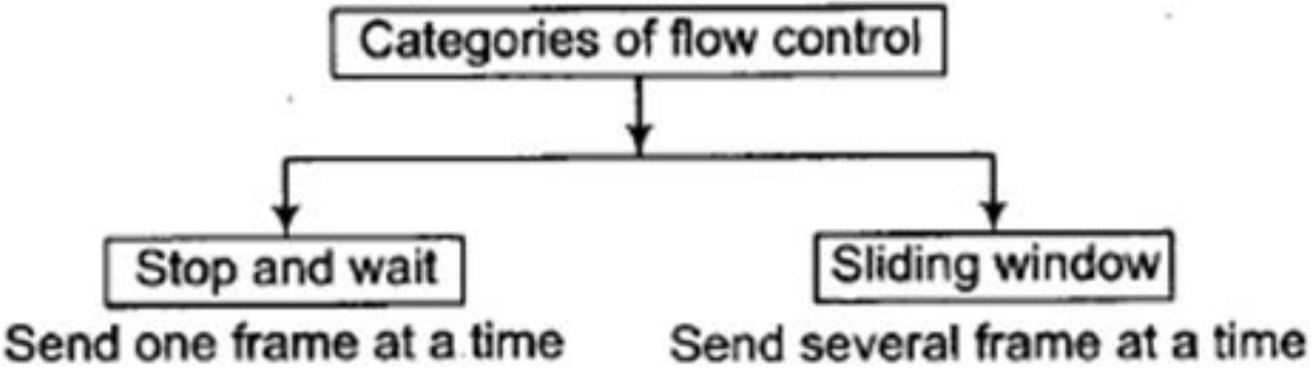
Error Detection and correction

Hamming code

- Hamming code in computer networks is linear block code for error correction and was developed by R.W. Hamming.
- This mechanism is used to identify and correct errors which can occur during the data transmission. It is possible to detect up to 2 errors and correct only one using hamming code technique.

Flow control

- When a data frame is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed. That is, sender sends at a speed on which the receiver can process and accept the data.
- What if the speed (hardware/software) of the sender or receiver differs? If sender is sending too fast the receiver may be overloaded, and data may be lost.
- Two types of mechanisms can be deployed to control the flow:
  - Stop and Wait
  - Sliding window



Categories of flow control

Stop and wait

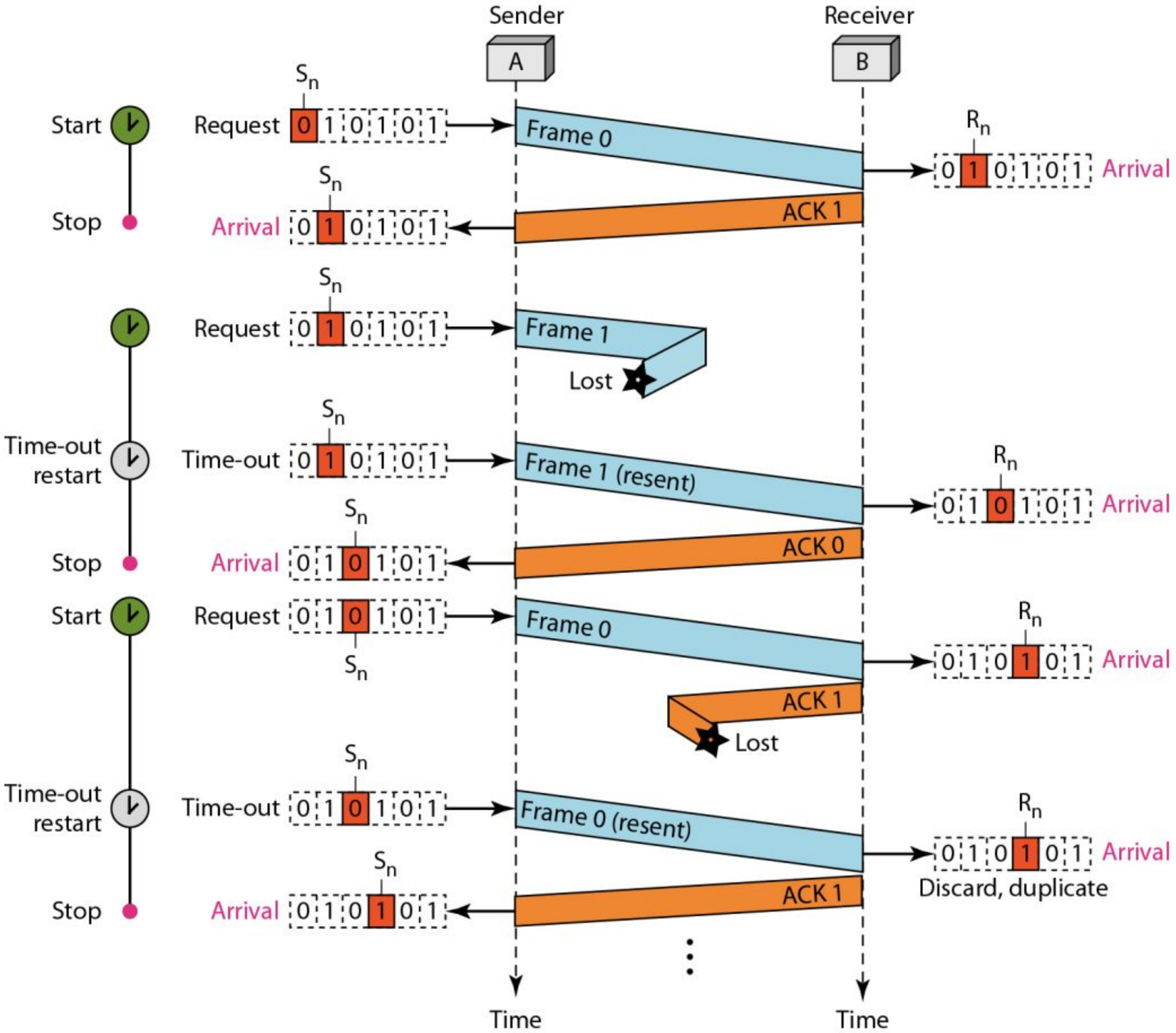
Send one frame at a time

Sliding window

Send several frame at a time

Stop and wait

- This flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is received.
- Suppose if any frame sent is not received by the receiver and is lost. So the receiver will not send any acknowledgment as it has not received any frame. Also, the sender will not send the next frame as it will wait for the acknowledgment for the previous frame which it had sent. So a deadlock situation can be created here. To avoid any such situation there is a time-out timer. The sender will wait for this fixed amount of time for the acknowledgment and if the acknowledgment is not received then it will send the frame again.


Sender
Receiver
A
B
Start
Request
S_n
0 1 0 1 0 1
Frame 0
R_n
0 1 0 1 0 1
Arrival
ACK 1
Stop
Arrival
S_n
0 1 0 1 0 1
S_n
Request
0 1 0 1 0 1
Frame 1
Lost
Time-out restart
Time-out
S_n
0 1 0 1 0 1
Frame 1 (resent)
R_n
0 1 0 1 0 1
Arrival
ACK 0
Stop
Arrival
S_n
0 1 0 1 0 1
Start
Request
0 1 0 1 0 1
S_n
Frame 0
R_n
0 1 0 1 0 1
Arrival
ACK 1
Lost
Time-out restart
Time-out
S_n
0 1 0 1 0 1
Frame 0 (resent)
R_n
0 1 0 1 0 1
Arrival
Discard, duplicate
ACK 1
Stop
Arrival
S_n
0 1 0 1 0 1
Time
Time

Disadvantages of stop and wait

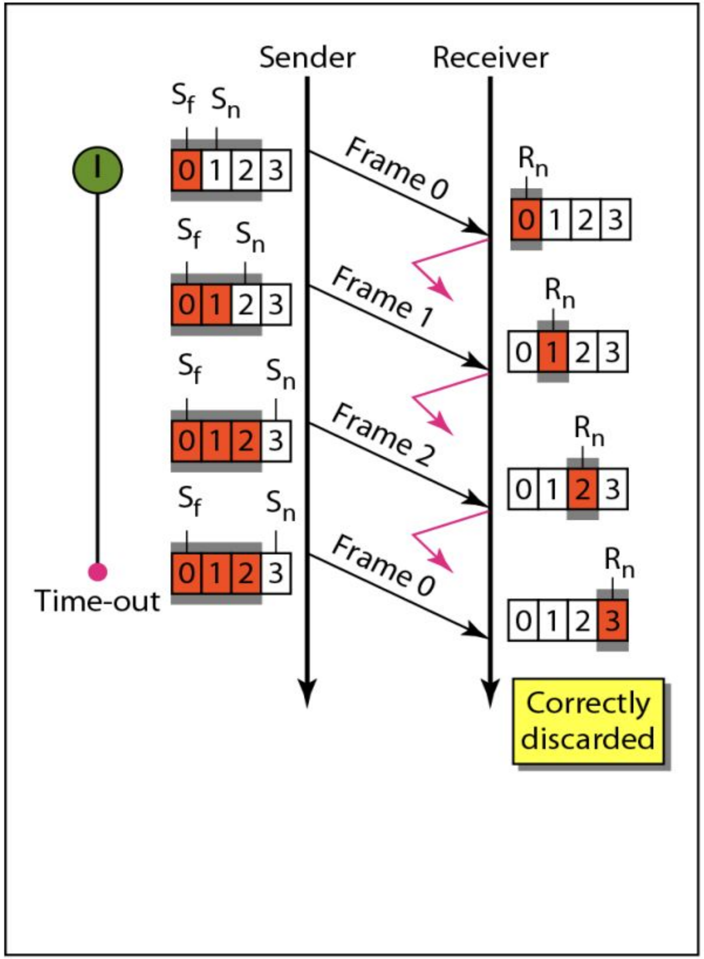
1. We can send only one packet at a time.
2. If the distance between the sender and the receiver is large then the propagation delay would be more than the transmission delay. Hence, efficiency would become very low.
3. After every transmission, the sender has to wait for the acknowledgment and this time will increase the total transmission time.

Sliding window

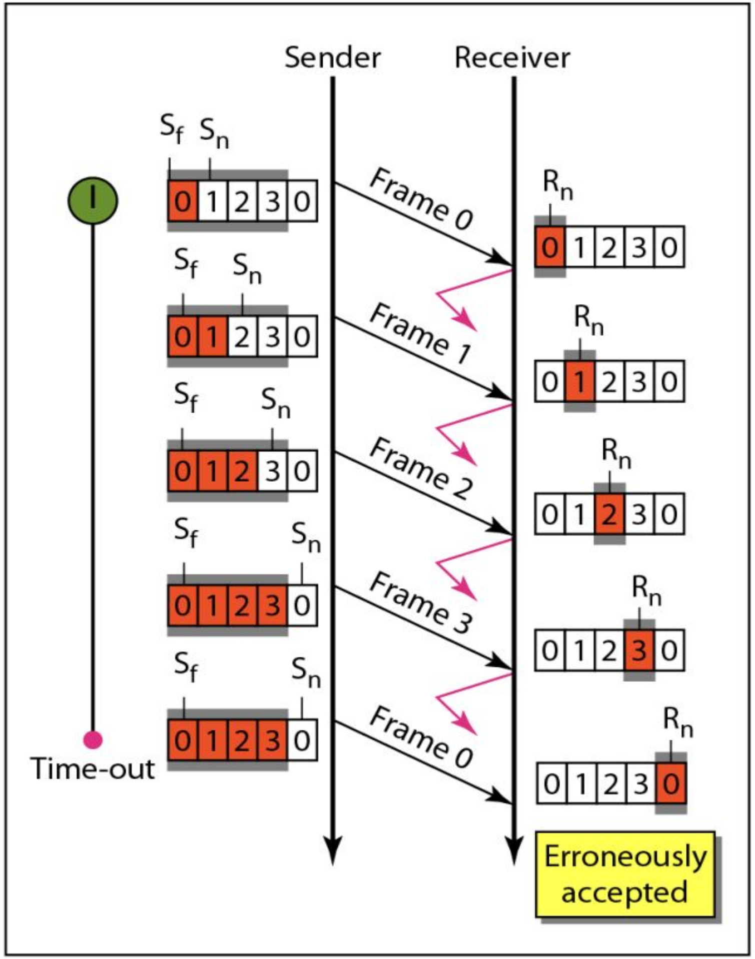
- The Sliding Window is a method of flow control in which a sender can transmit the several frames before getting an acknowledgement.
- In Sliding Window Control, multiple frames can be sent one after the another due to which capacity of the communication channel can be utilized efficiently.
- When the receiver sends the ACK, it includes the number of the next frame that it wants to receive. For example, to acknowledge the string of frames ending with frame number 4, the receiver will send the ACK containing the number 5. When the sender sees the ACK with the number 5, it got to know that the frames from 0 through 4 have been received.

Go-back-N

- The send window is an abstract concept defining an imaginary box of size 2^m − 1.
- The receive window is an abstract concept defining an imaginary box of size 1.
- The window slides when a correct frame has arrived; sliding occurs one slot at a time.
- If m=2 then window size=3



a. Window size $< 2^m$



b. Window size $= 2^m$

Figure shows an example of Go-Back-N. This is an example of a case where the forward channel is reliable, but the reverse is not. No data frames are lost, but some ACKs are delayed and one is lost. The example also shows how **cumulative acknowledgments** can help if acknowledgments are delayed or lost. After initialization, there are seven sender events. Request events are triggered by data from the network layer; arrival events are triggered by acknowledgments from the physical layer. There is no time-out event here because all outstanding frames are acknowledged before the timer expires. Note that although ACK 2 is lost, ACK 3 serves as both ACK 2 and ACK 3.

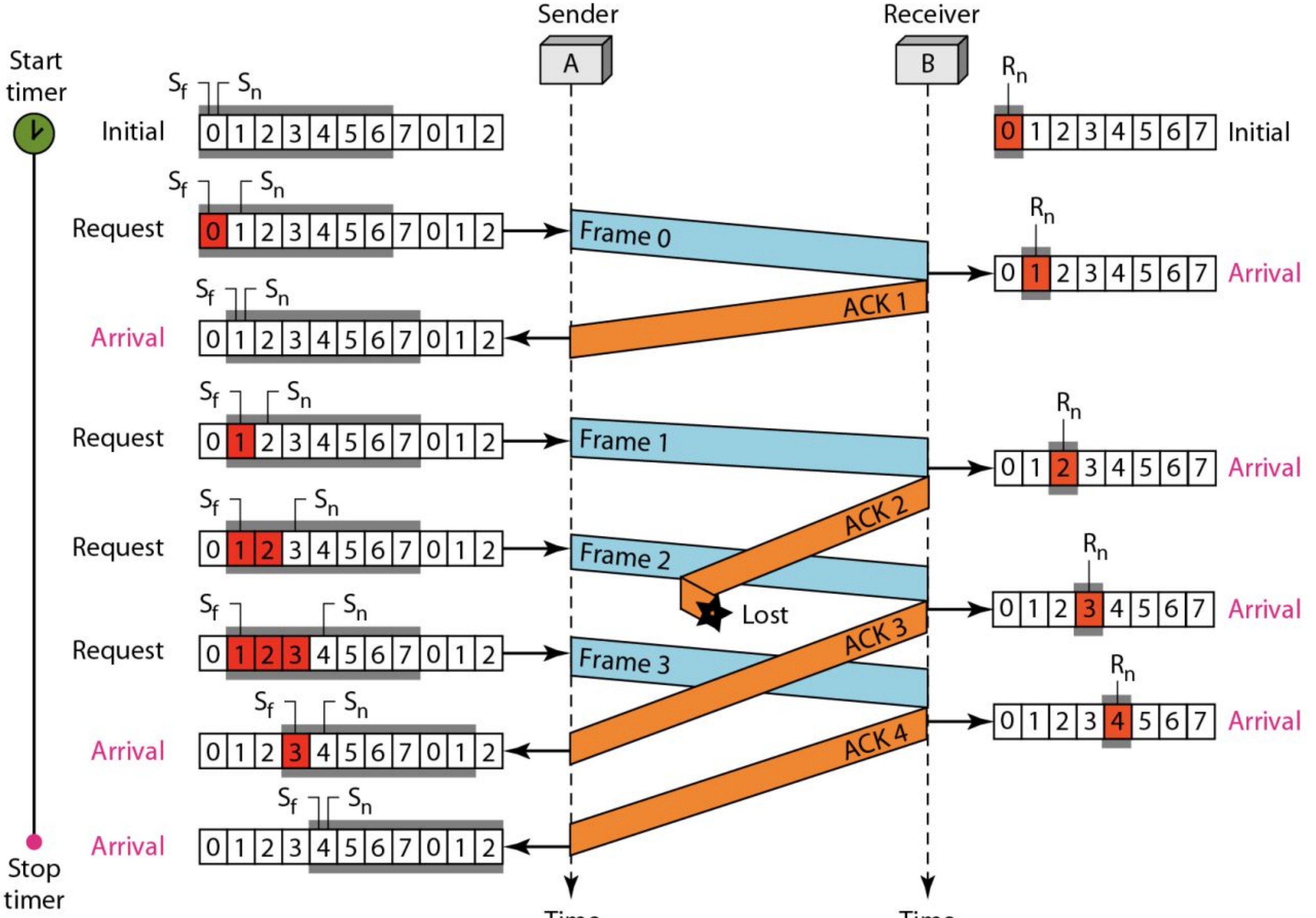
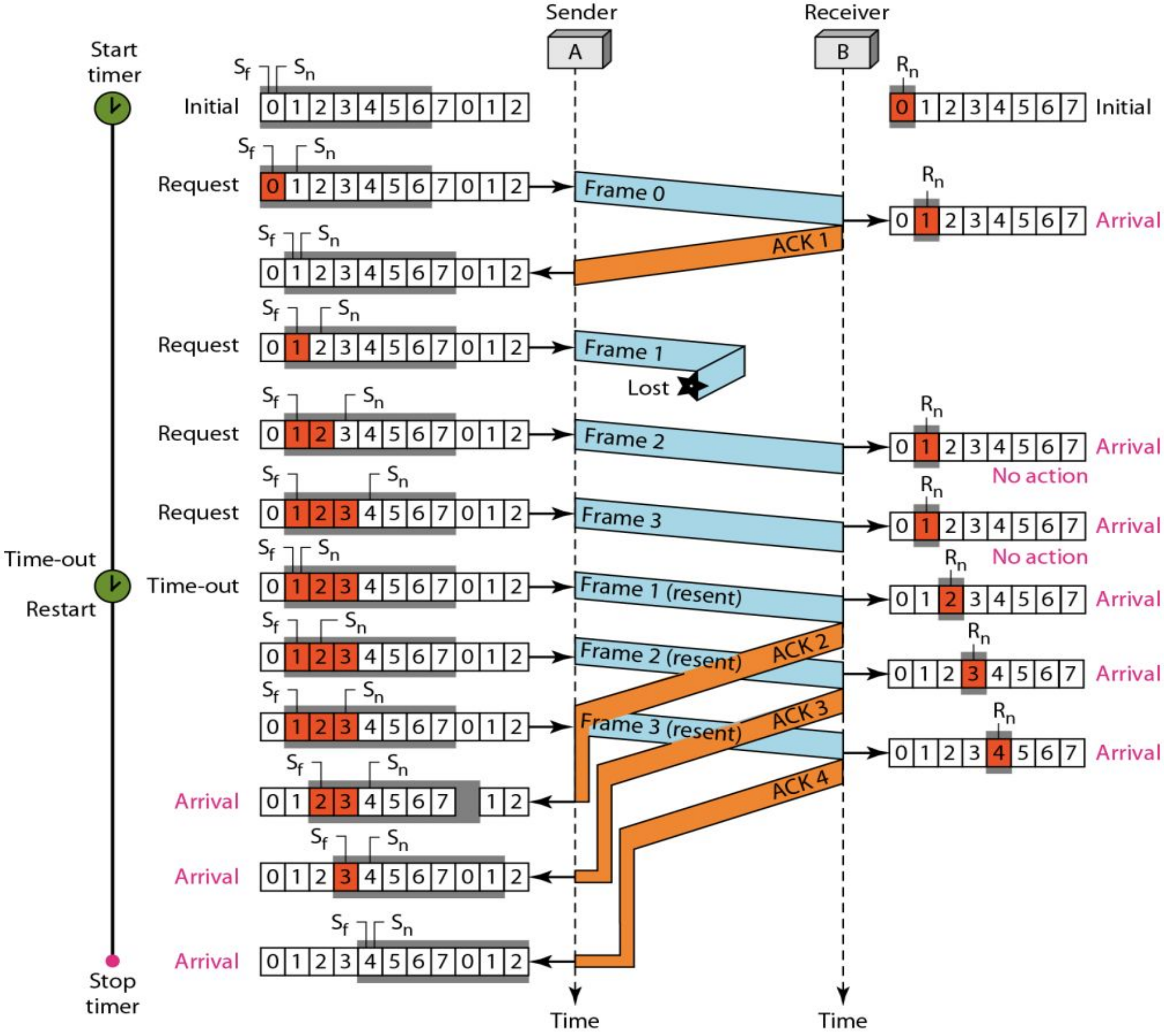
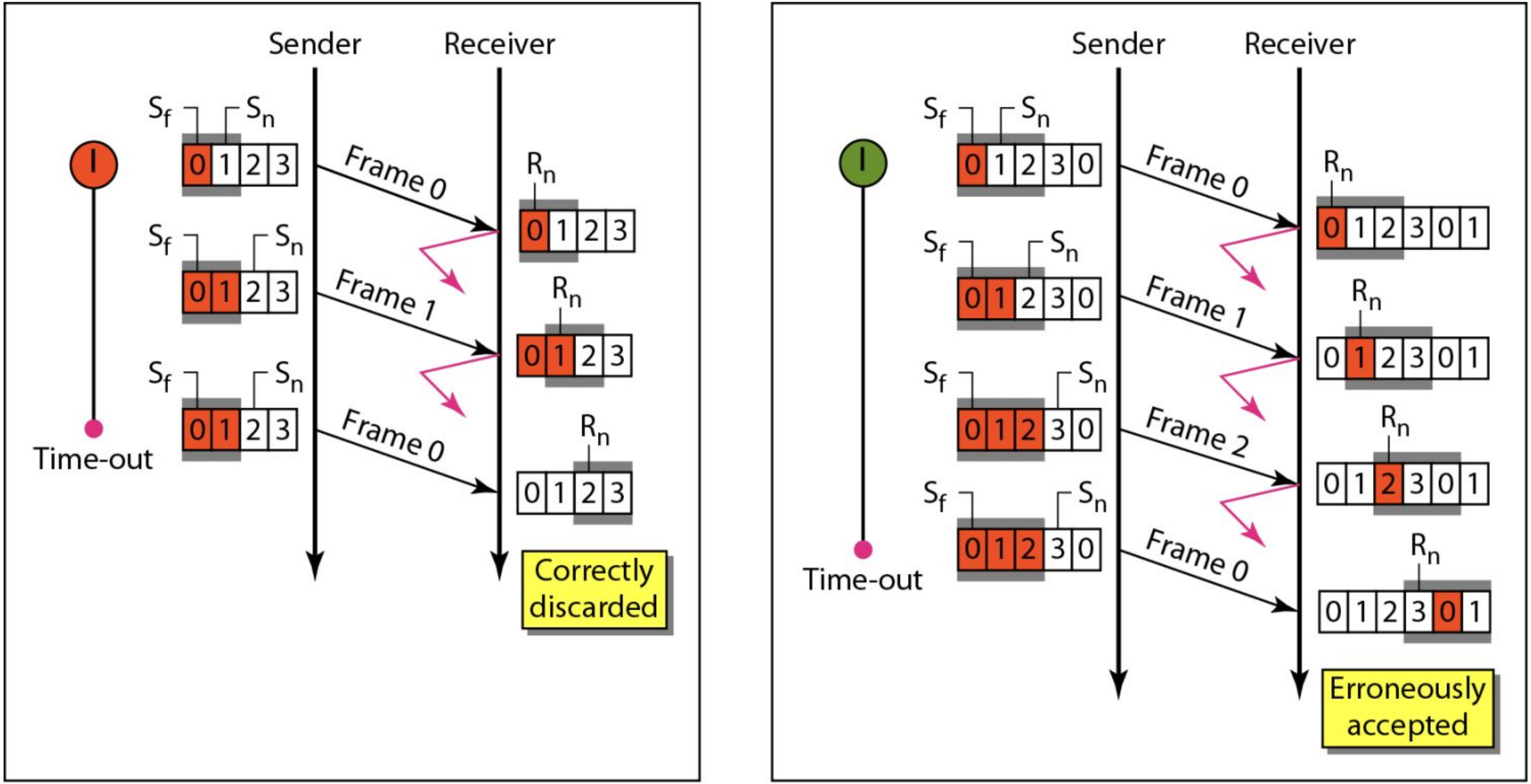

Sender
Receiver
A
B
Start timer
Stop timer
Initial
Request
Arrival
Request
Request
Request
Arrival
Arrival
Sf
Sn
Rn
0 1 2 3 4 5 6 7 0 1 2
0 1 2 3 4 5 6 7
Initial
Arrival
Arrival
Arrival
Arrival
Frame 0
ACK 1
Frame 1
ACK 2
Frame 2
Lost
Frame 3
ACK 3
ACK 4
Time
Time

Figure shows what happens when a frame is lost. Frames 0, 1, 2, and 3 are sent. However, frame 1 is lost. The receiver receives frames 2 and 3, but they are discarded because they are received out of order. The sender receives no acknowledgment about frames 1, 2, or 3. Its timer finally expires. The sender sends all outstanding frames (1, 2, and 3) because it does not know what is wrong. Note that the resending of frames 1, 2, and 3 is the response to one single event. When the sender is responding to this event, it cannot accept the triggering of other events. This means that when ACK 2 arrives, the sender is still busy with sending frame 3.


Sender
Receiver
A
B
Start timer
Initial
S_f
S_n
0 1 2 3 4 5 6 7 0 1 2
R_n
0 1 2 3 4 5 6 7
Initial
Request
Frame 0
Arrival
ACK 1
Request
Frame 1
Lost
Request
Frame 2
Arrival
No action
Request
Frame 3
Arrival
No action
Time-out
Restart
Time-out
Frame 1 (resent)
Arrival
Frame 2 (resent)
ACK 2
Arrival
Frame 3 (resent)
ACK 3
Arrival
ACK 4
Arrival
Arrival
Arrival
Stop timer
Time
Time

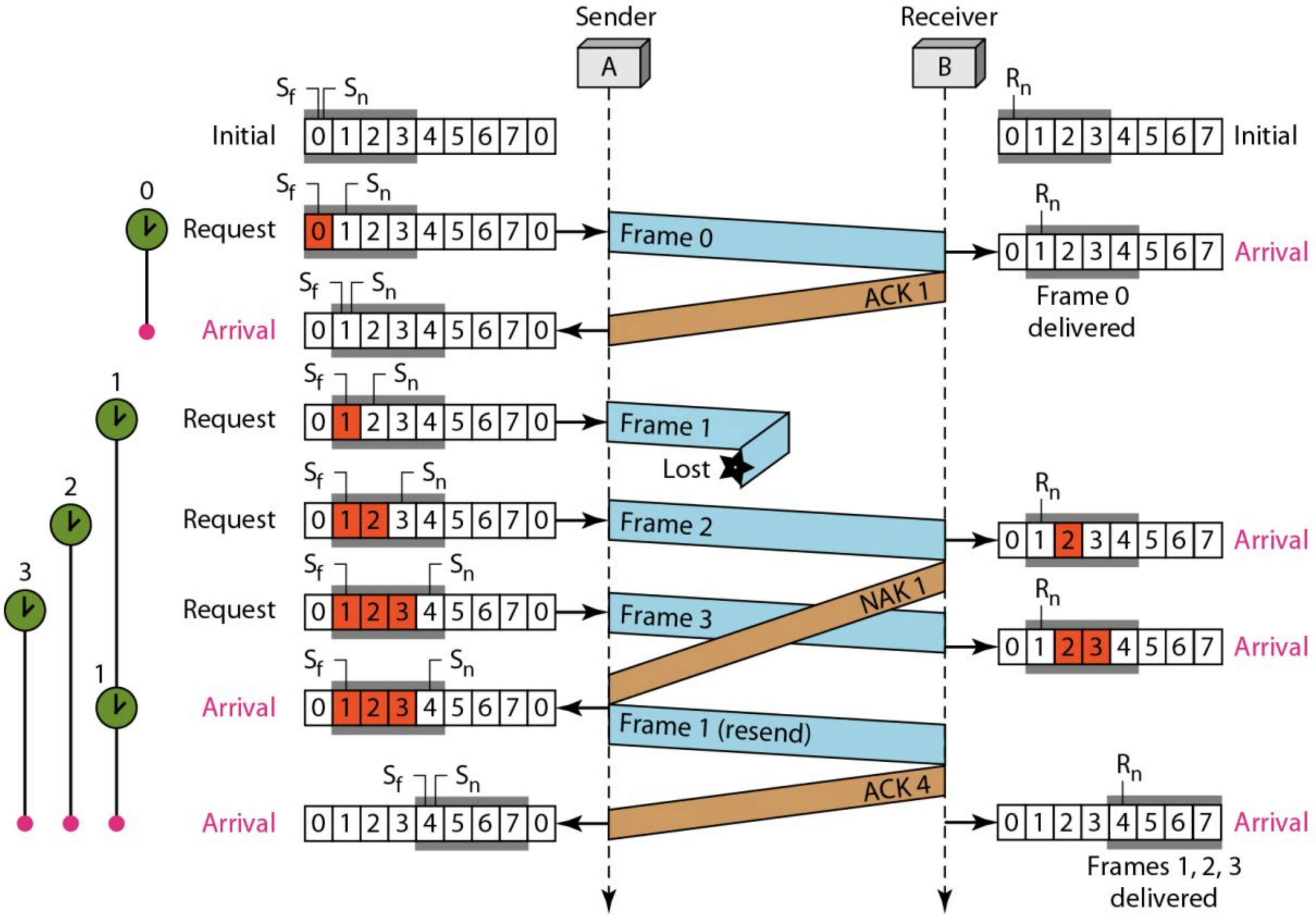
SELECTIVE-REPEAT-REQUEST

- The selective repeat overcomes the inefficiency of the go-back-n protocol. In go-back-n, if the sender does not receive acknowledgement it retransmits all the frames from the current sender sliding window. But what if there is a frequent loss of frames or if there is a lot of error in the frame a lot of bandwidth is wasted.



a. Window size = 2^{m-1}

b. Window size > 2^{m-1}


Sender
Receiver
A
B
S_f
S_n
R_n
Initial
0 1 2 3 4 5 6 7 0
0 1 2 3 4 5 6 7
Initial
0
Request
Frame 0
Arrival
ACK 1
Arrival
Frame 0
delivered
1
Request
Frame 1
Lost
2
Request
Frame 2
Arrival
3
Request
Frame 3
NAK 1
Arrival
1
Arrival
Frame 1 (resend)
ACK 4
Arrival
Arrival
Frames 1, 2, 3
delivered