Computer Network

Lecture-21

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

NOISY CHANNELS

Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ)

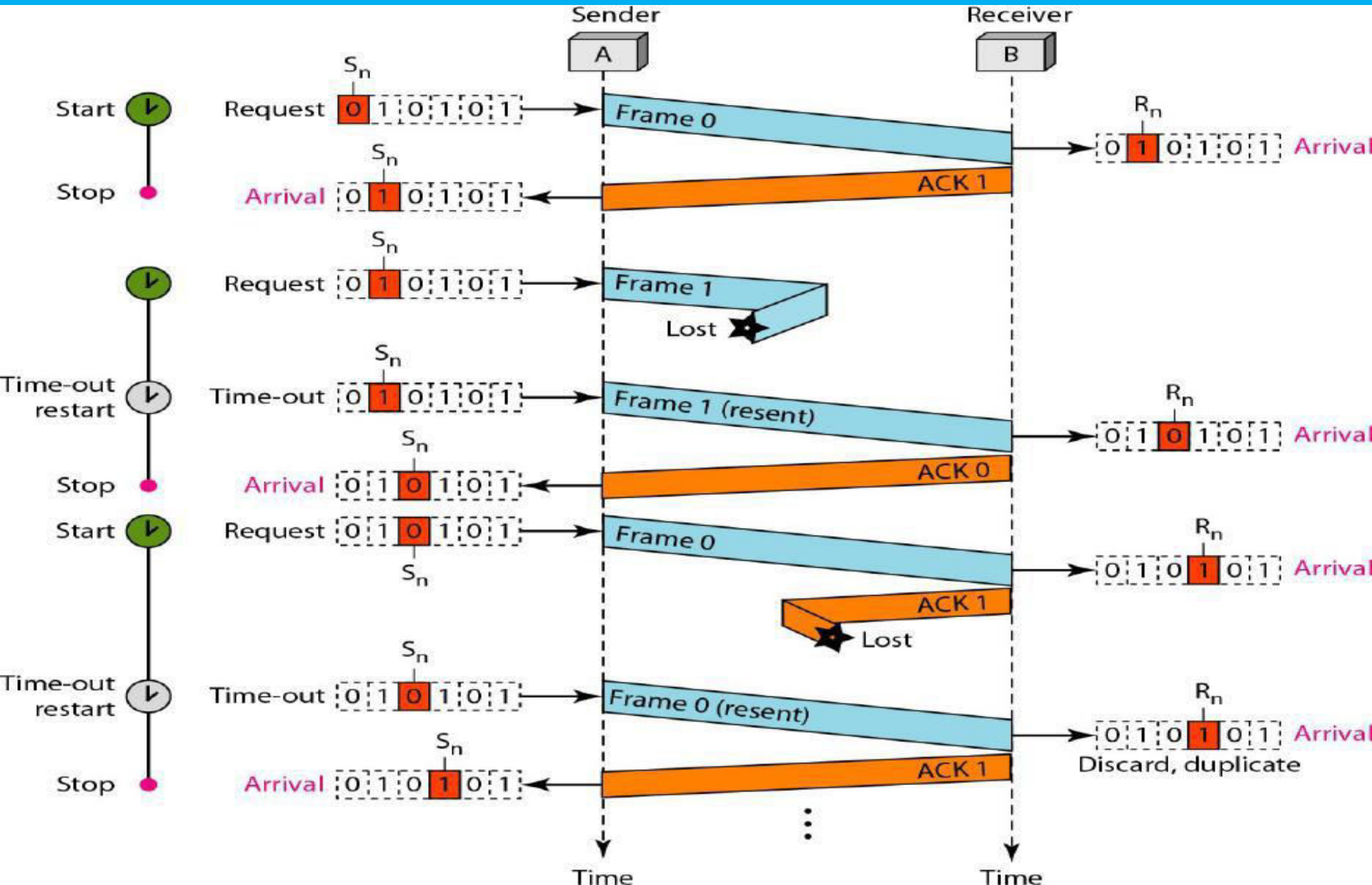
- This protocol adds a simple error control mechanism to the Stop-and-Wait Protocol.
- To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded.
- Lost frames are more difficult to handle than corrupted ones.
- To handle lost frames, this protocol uses sequence number.

Stop-and-Wait ARQ

Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ)

- When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated.
- The corrupted and lost frames need to be resent in this protocol.
- When the sender sends a frame, it keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted.

Stop-and-Wait ARQ



Stop-and-Wait ARQ

- Since an ACK frame can also be corrupted and lost, it too needs redundancy bits and a sequence number.
- The ACK frame for this protocol has a sequence number field.
- In this protocol, the sender simply discards a corrupted ACK frame or ignores an out-of-order one.
- A field is added to the data frame to hold the sequence number of that frame.

Stop-and-Wait ARQ

- If the number of bits used for sequence number is m, then range of sequence numbers is 0 to 2^m-1.
- This protocol uses 1 bit for sequence number i.e. m=1.
- This protocol uses the following sequence numbers 0, 1, 0, 1, 0, 1, 0, 1, 0, 1……
- This protocol uses the sliding window concept.
- The size of both sender and receiver window in this protocol is 1.

Stop-and-Wait ARQ

Efficiency

- The Stop-and-Wait ARQ discussed in the previous section is very inefficient if our channel is thick and long. By thick, we mean that our channel has a large bandwidth; by long, we mean the round-trip delay is long. The product of these two is called the bandwidth delay Product.
- The channel is always there. If we do not use it, we are inefficient. The bandwidth-delay product is a measure of the number of bits we can send out of our system while waiting for news from the receiver.

Stop-and-Wait ARQ

Example:

Assume that, in a Stop-and-Wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20 ms to make a round trip. What is the bandwidth-delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?

Stop-and-Wait ARQ

Solution:

The bandwidth-delay product is

$$= 1*10^{6} * 20*10^{-3} = 20*10^{3}$$

$$= 20000 \text{ bits}$$

- The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and then back again. However, the system sends only 1000 bits. We can say that the link utilization is only 1000/20,000, or 5 percent.
- For this reason, for a link with a high bandwidth or long delay, the use of Stop-and-Wait ARQ wastes the capacity of the link.

Stop-and-Wait ARQ

Example:

What is the utilization percentage of the link in the previous example if we have a protocol that can send up to 15 frames before stopping and worrying about the acknowledgments?

Solution:

The bandwidth-delay product is still 20,000 bits. The system can send up to 15 frames or 15,000 bits during a round trip. This means the utilization is 15,000/20,000, or 75 percent.

Note: Of course, if there are damaged frames, the utilization percentage is much less because frames have to be resent.

Go-Back-N Automatic Repeat Request

- To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment.
- In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.
- In the Go-Back-N Protocol, the sequence numbers are modulo 2^m, where m is the size of the sequence number field in bits.
- In this protocol, the sliding window is an abstract concept that defines the range of sequence numbers that is the concern of the sender and receiver.

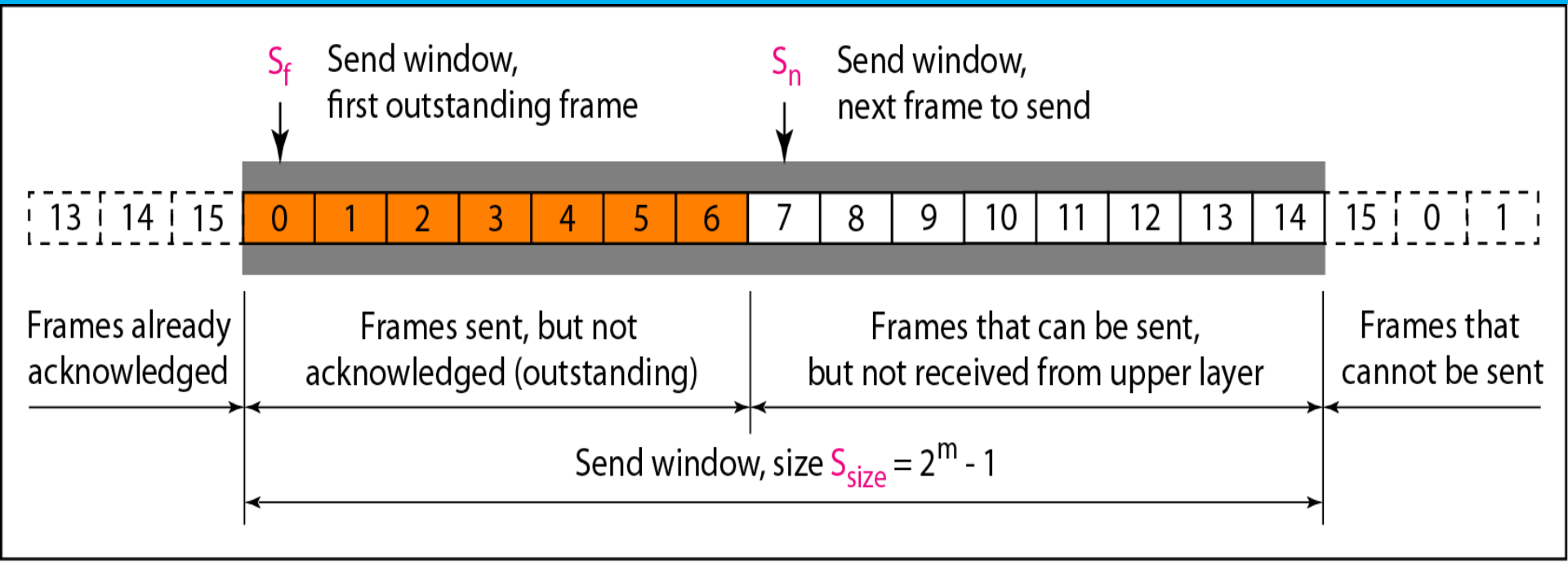
Go-Back-N Automatic Repeat Request

- If m=4 bits are used for sequence number, then the only sequence numbers are 0 through 15 inclusive. So the sequence numbers are

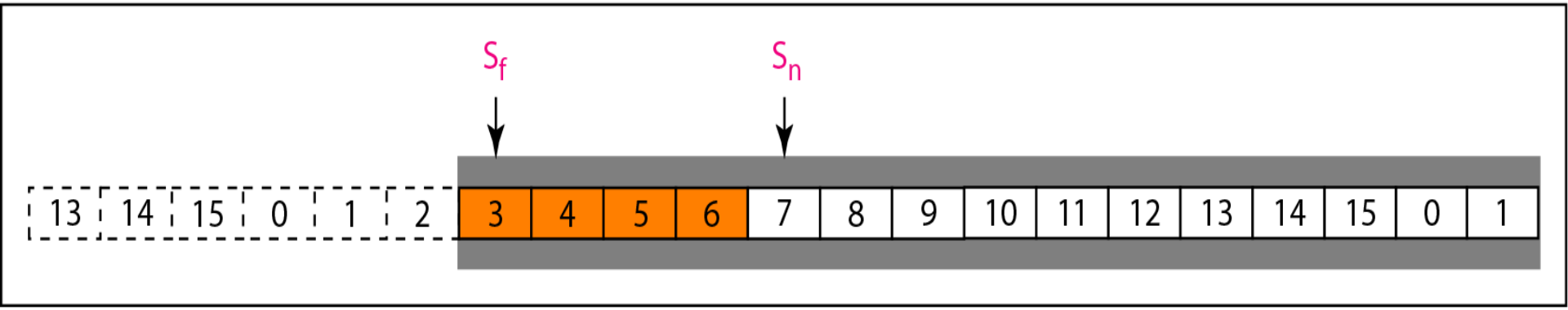
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...

- The maximum size of the sender window is 2^m-1.
- The size of the receiver window is 1.
- The sender window at any time divides the possible sequence numbers into four regions.

Go-Back-N Automatic Repeat Request

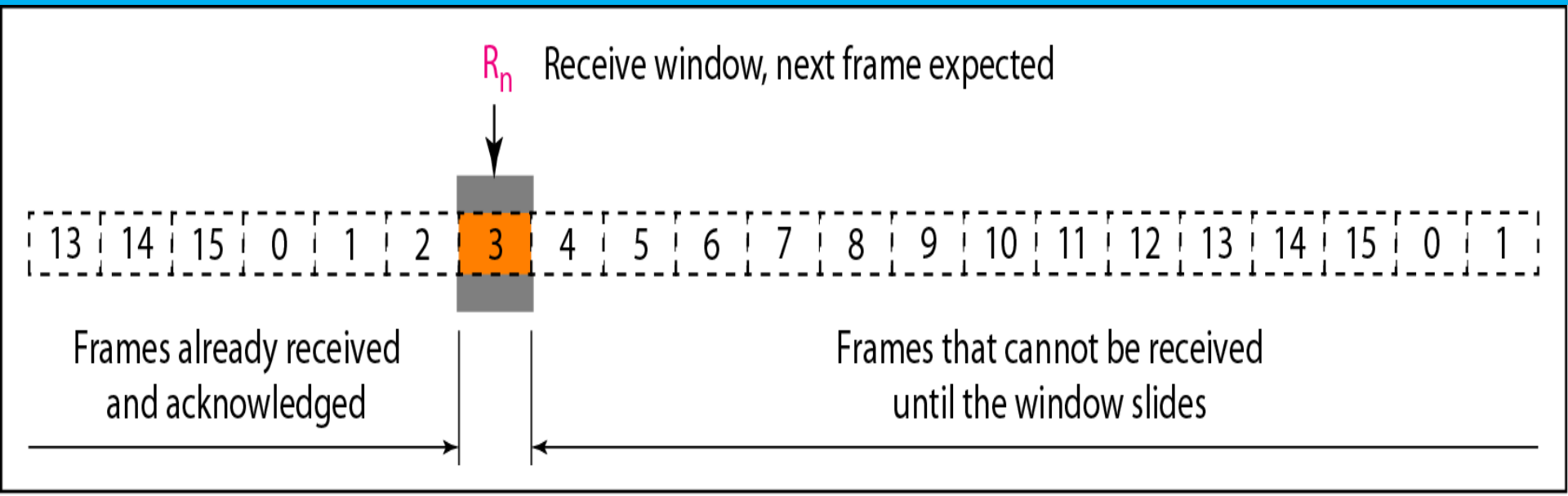


a. Send window before sliding

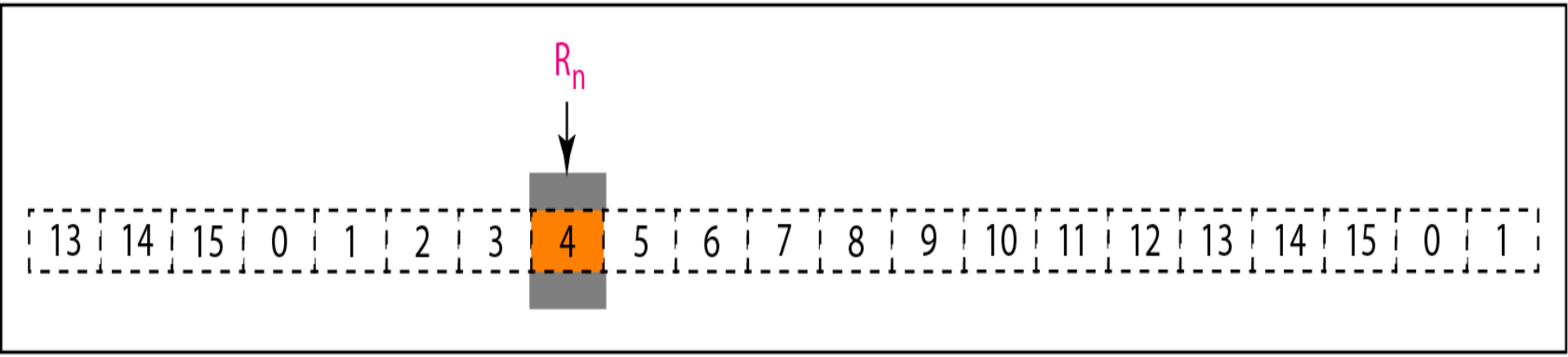


b. Send window after sliding

Go-Back-N Automatic Repeat Request



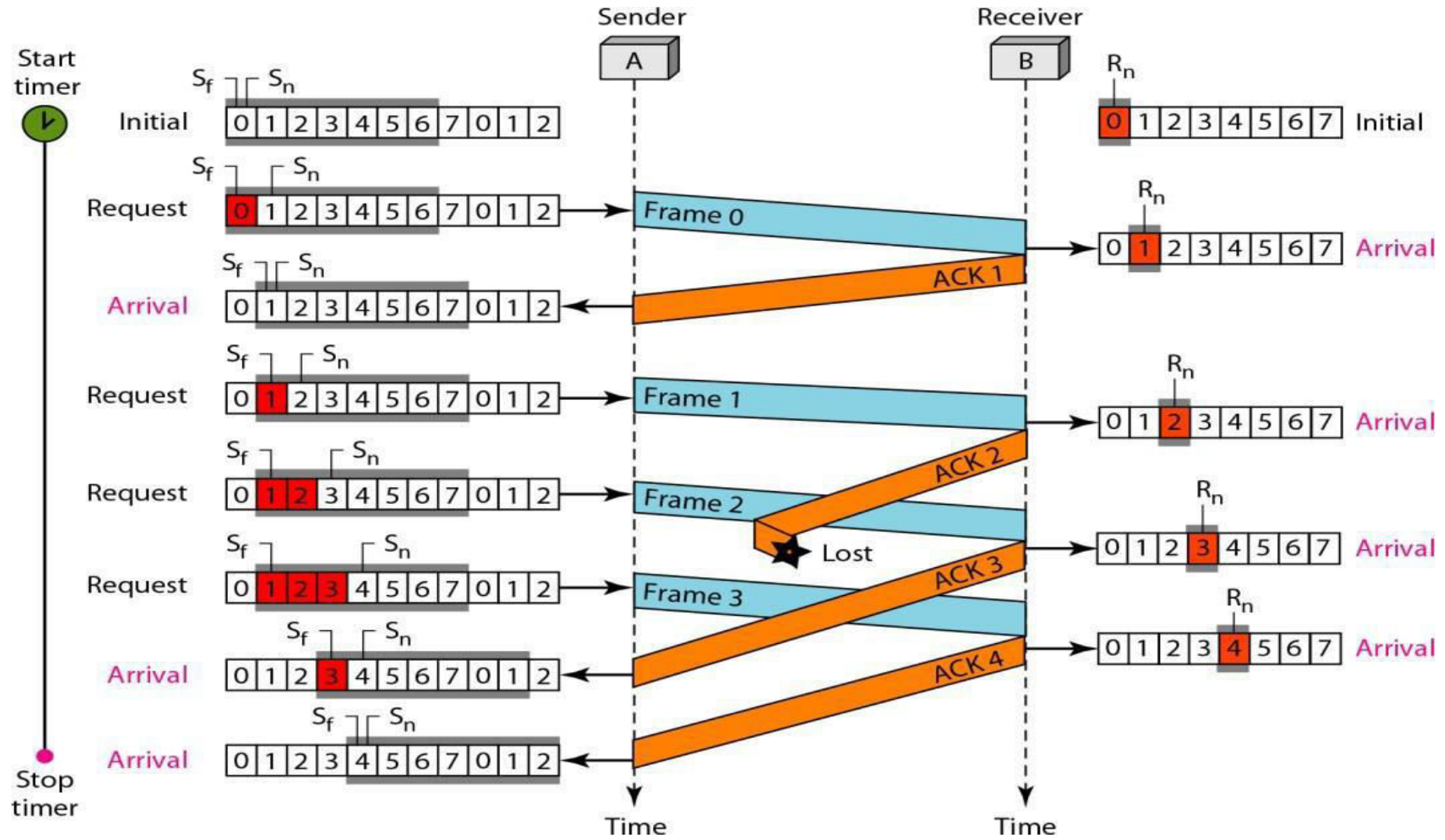
a. Receive window



b. Window after sliding

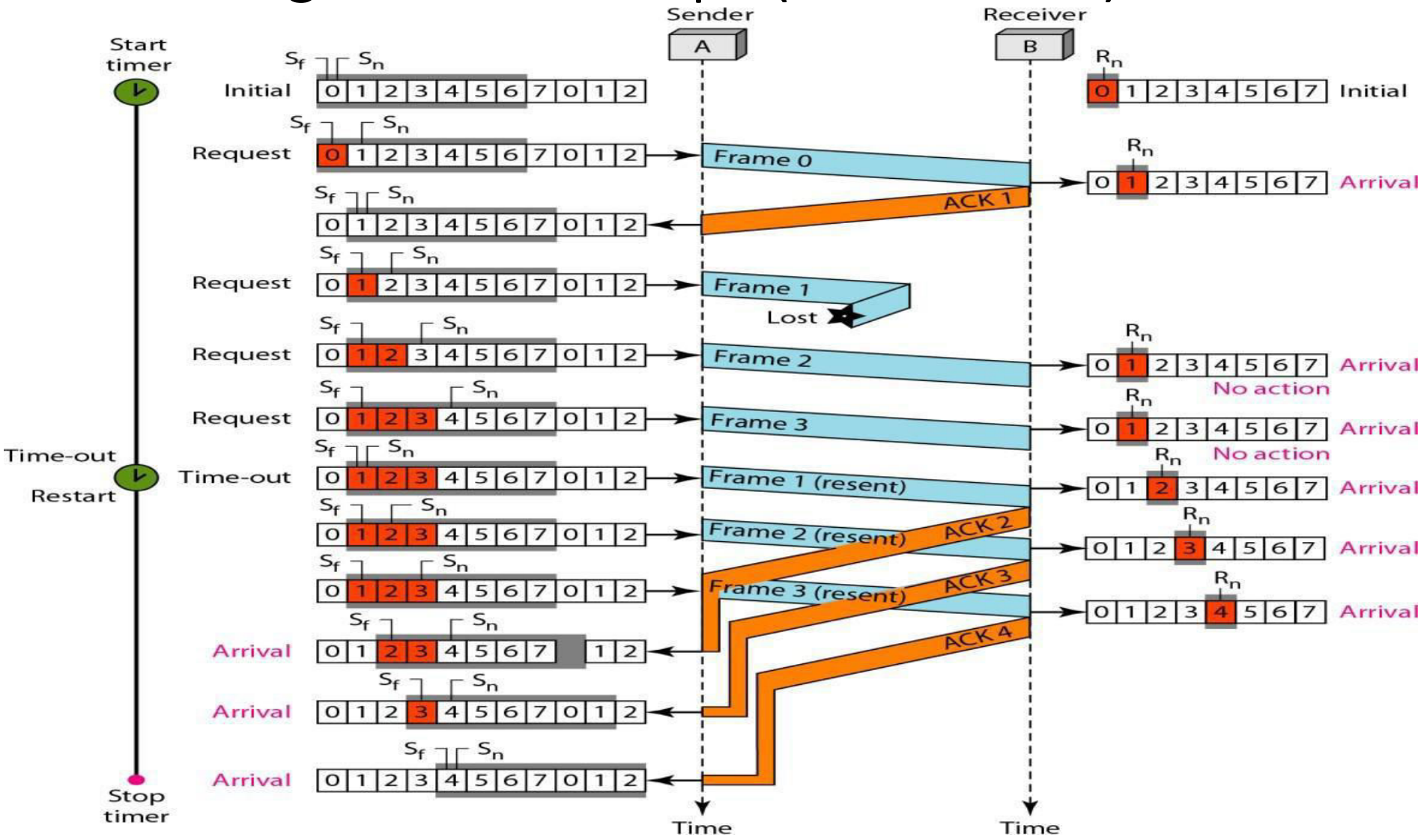
Go-Back-N Automatic Repeat Request

❖ **Below figure is an example(if ack lost)**



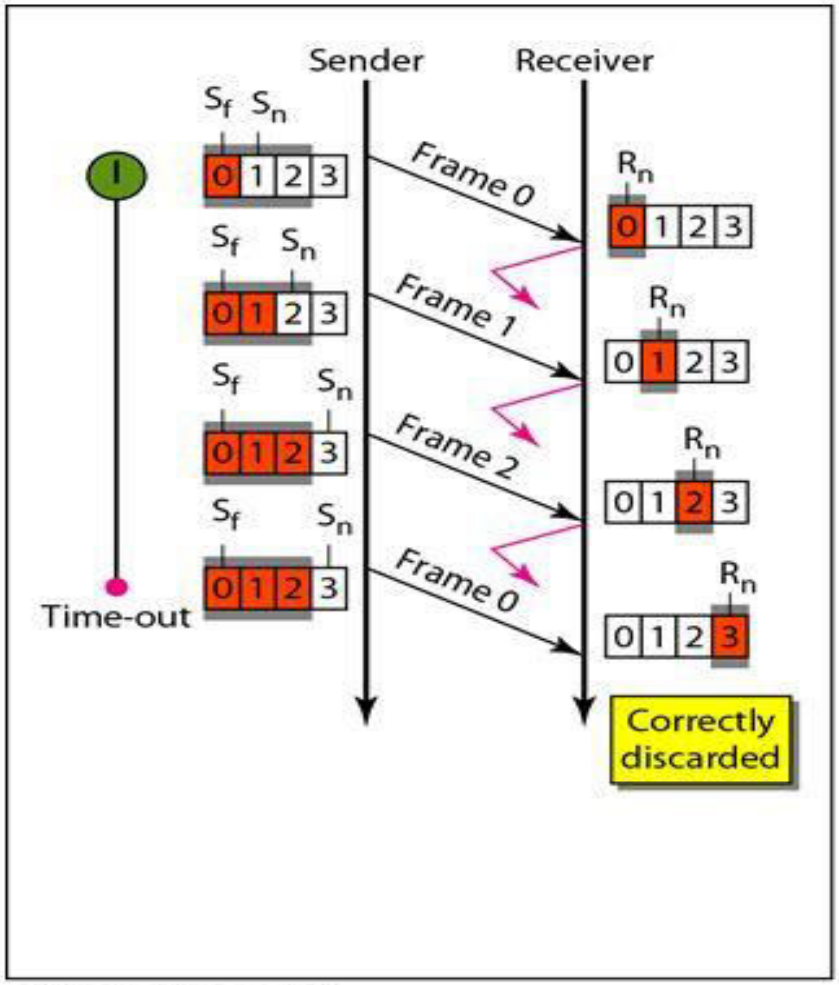
Go-Back-N Automatic Repeat Request

❖ Below figure is an example(if frame lost)

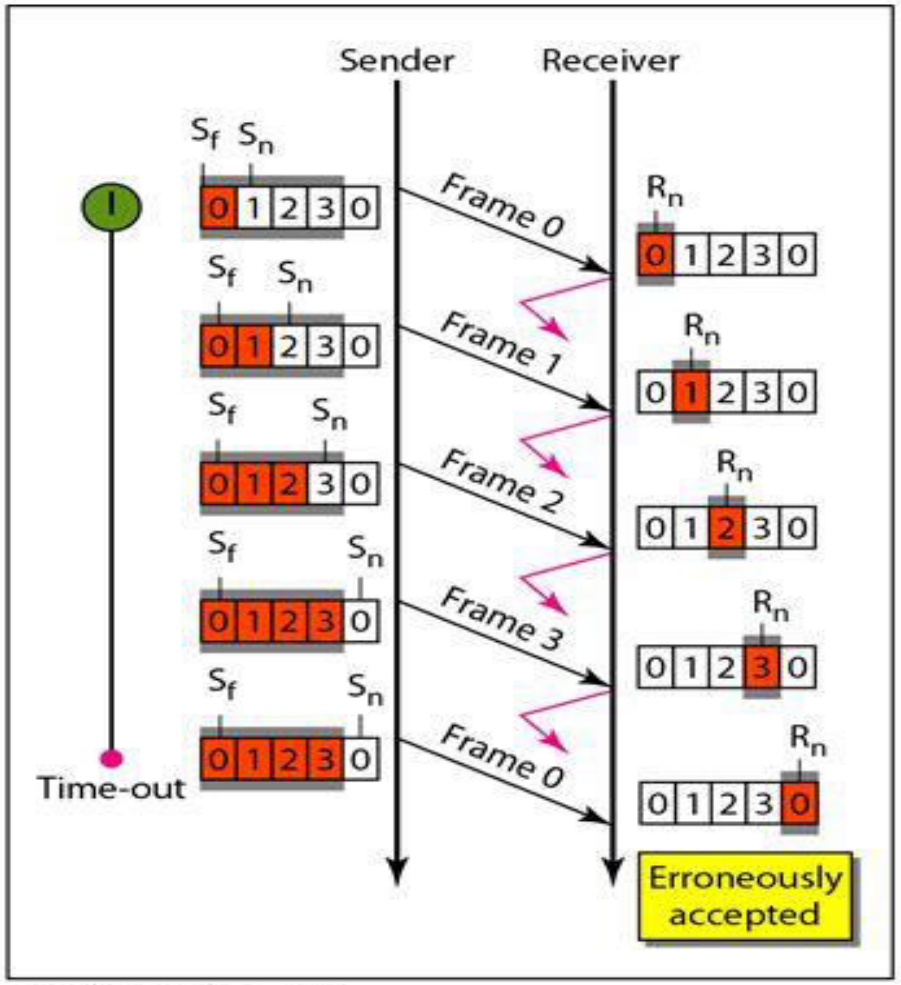


Go-Back-N Automatic Repeat Request

Note: In Go-Back-N ARQ, the size of the sender window must be less than 2^m; the size of the receiver window is always 1.



a. Window size $< 2^m$



b. Window size $= 2^m$

Go-Back-N Automatic Repeat Request

Note: Stop-and-Wait ARQ is a special case of Go-Back-N ARQ in which the size of the sender window is 1.

Efficiency:

The efficiency of Go-Back-N ARQ ,

Efficiency = N/(1+2a),

Where N is the size of sender window and a = T_p/T_t.

Where T_p is propagation delay and T_t is the transmission delay

Also, $T_t = D/B$;

and here D = data size and B = bandwidth

And $T_p = d/v$,

here d = distance and v = propagation speed.

Go-Back-N Automatic Repeat Request

Now to find the effective bandwidth (or throughput),

Effective bandwidth = efficiency * bandwidth,

which means,

Effective bandwidth = (N/(1+2a)) *B