Data Link Control

- So far we have covered transmission of signals over a *transmission link*
- For effective digital data communications much more is needed to control and manage the exchange
- A layer of *control logic* is required above the physical layer in each Source/Destination device
- This logic is referred to as *Data Link Control*. Its use transforms the *transmission link* to a *data communications link*
- The logic/rules are encapsulated within a protocol known as the *Data Link Control Protocol*

Data Link Control Requirements

- *Frame Synchronisation* :
  - Frames must be recognizable by the Receiver (already covered)
- *Flow Control* :
  - The Sender must <u>not</u> overload the Receiver
- *Error Control* :
  - Errors should be detectable by the Receiver
- *Addressing* :
  - For a multi-point configuration each station must be uniquely identifiable
- *Control and Data on same link* :
  - The Receiver must not have to wait for control information to arrive before it can process a frame
- *Link Management* :
  - Procedures for establishing and relinquishing the link must be adhered to

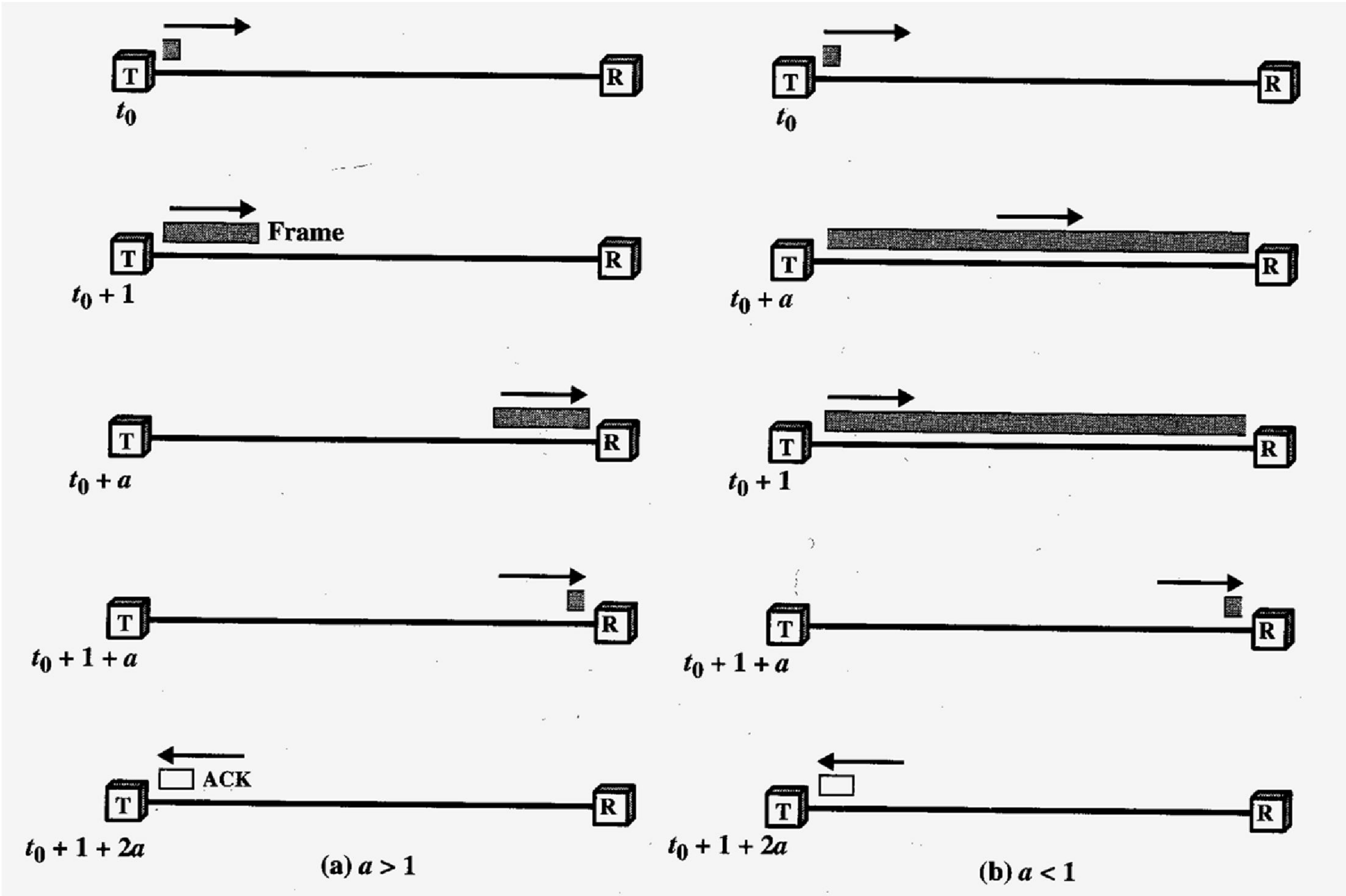
Flow Control

- The Receiver and the Sender will each allocate a data *buffer* of a fixed size
- Received data remains in the buffer until it has been processed by the Receiving station i
- *Flow control* enables the Receiving station to indicate to the Sending station that its buffers are full
- There are <u>two</u> *flow control* techniques to consider:
  – *Stop-and-Wait*
  – *Sliding Windows*

Stop-and-Wait Flow Control

- Here the Sending station transmits a frame
- The Sending station must get an *acknowledgement* from the Receiving station before transmitting the next frame
- The Destination station can control the flow of data by *withholding* acknowledgements

Stop and Wait Flow Control



Sliding Window Flow Control

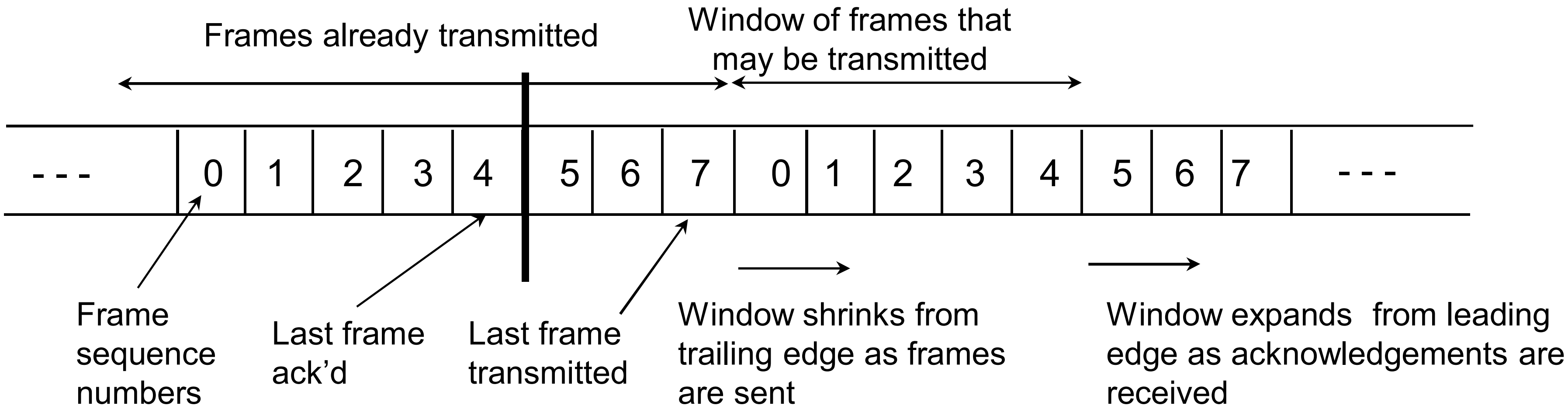
- This technique allows *multiple* frames to be in transit simultaneously
- Both stations use an extended buffer size to hold multiple frames
- The Sending/Receiving stations maintain a list of frames already sent/received
- This technique allows for much <u>more</u> *efficient link utilization*
- The transmission link is effectively treated as a *pipeline* that can be *filled* with many frames in transit *simultaneously*

Sliding Window Flow Control

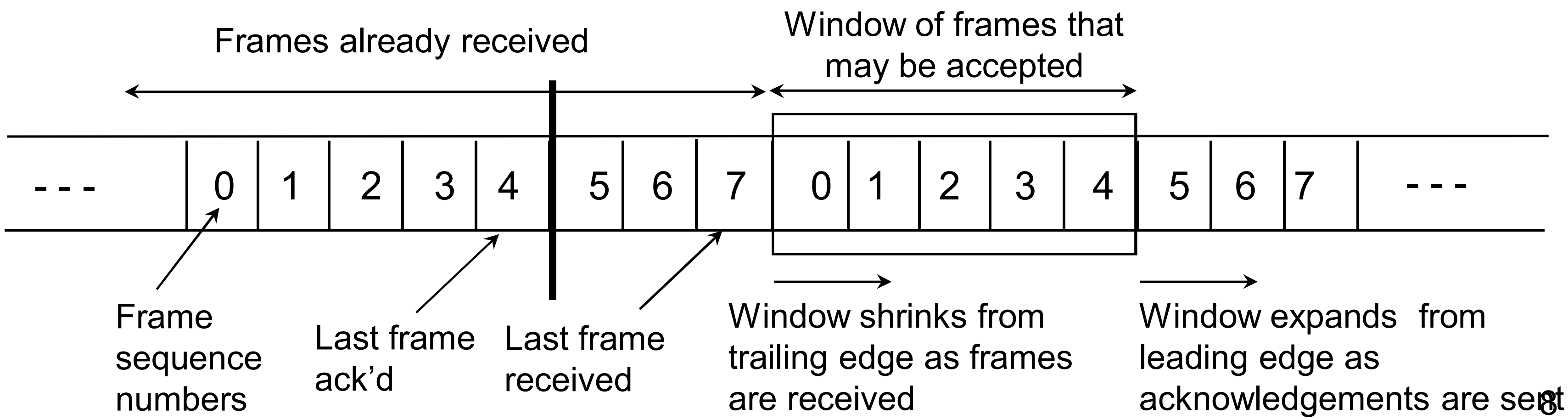
- Stations A and B each allocate buffer space for *W* frames
  - i.e. Station B can *accept W* frames and Station A can *send W* frames without any acknowledgement being sent or received
- Each frame contains a *sequence number*
- Station B sends *acknowledgements* that include the sequence number of the *next* frame expected
  - i.e. Station B is prepared to receive the next *W* frames starting at the *sequence number* indicated e.g. RR5

Sliding Window Flow Control

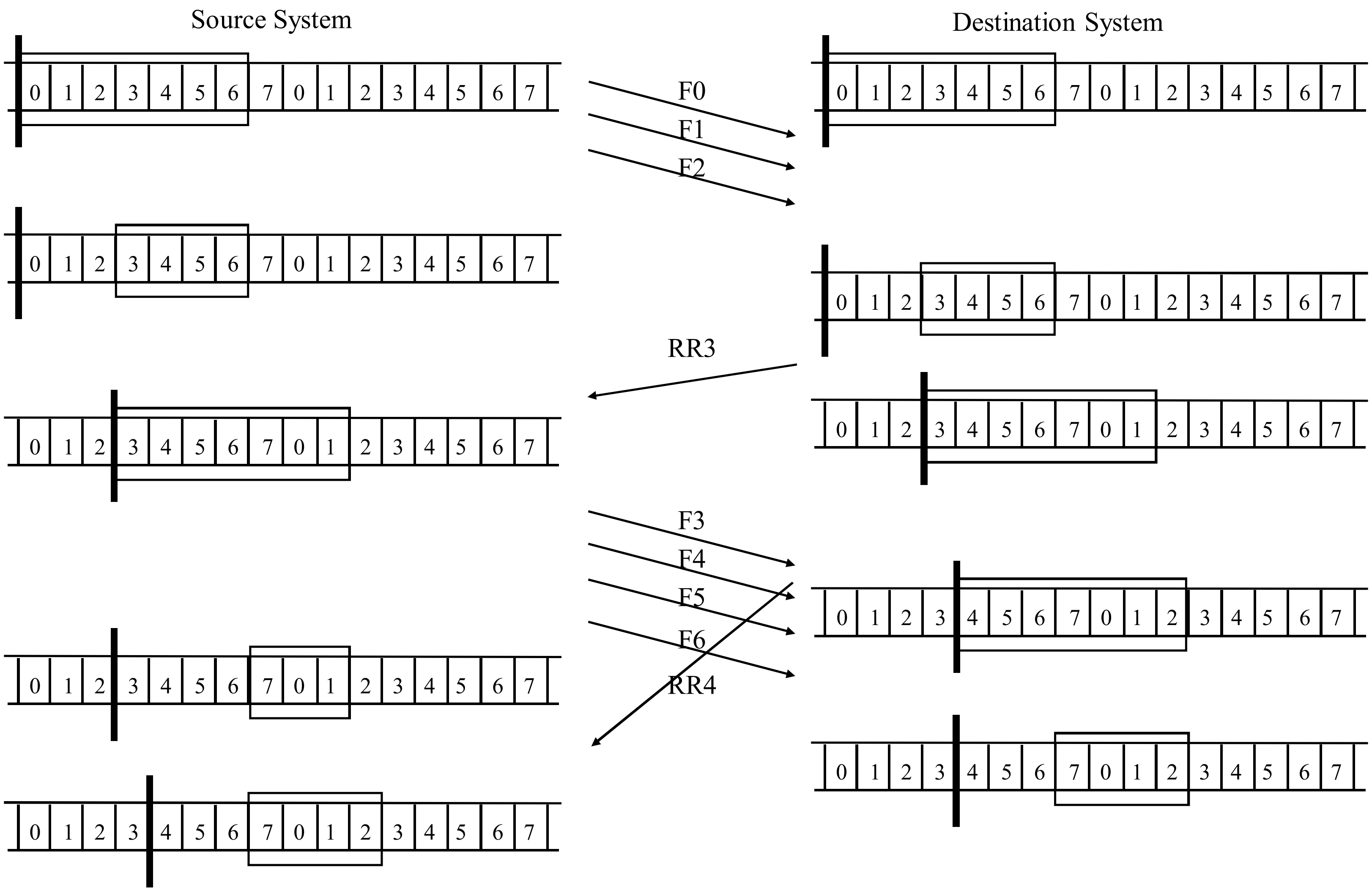
Transmitters Perspective



Receiver's Perspective



Example Sliding Window



Sliding Window Flow Control

- *Multiple* frames can be *acknowledged* using a single control message (*implicit acknowledgement*)
  - e.g. Receipt of ACK for frame **2** (RR3) followed later by ACK for frame **5** (RR6) *implies* acknowledgement of frames **3** and **4**
- Station A maintains a list of frame numbers it is allowed to *send*
- Station B maintains a list of frame numbers it is prepared to receive
- These lists can be considered as *windows*

Sliding Window Flow Control

- To impose flow control the Receiver can send a different control message, *Receive Not Ready* (RNR)
  - This acknowledges previous frames but stems the flow from the Sender e.g. RNR5 acknowledges all frames up to frame 4
- To resume flow the Receiver sends a *Receiver Ready* (RR) message
  - e.g. RR5 – recommence at frame 5

Sliding Window Flow Control

- When two stations exchange data in *Full Duplex* mode each must maintain two sliding windows, one for *transmit* and one for *receive*
- When this is done acknowledgements can be *piggybacked* with data in a single frame
- *Sliding Window* is potentially much more efficient than *Stop-and-Wait* in terms of *transmission link utilization*
  - With Stop-and-Wait only one frame can be in transit
  - With *Sliding Window* the link is treated like a *pipeline* that can be filled with *multiple* frames