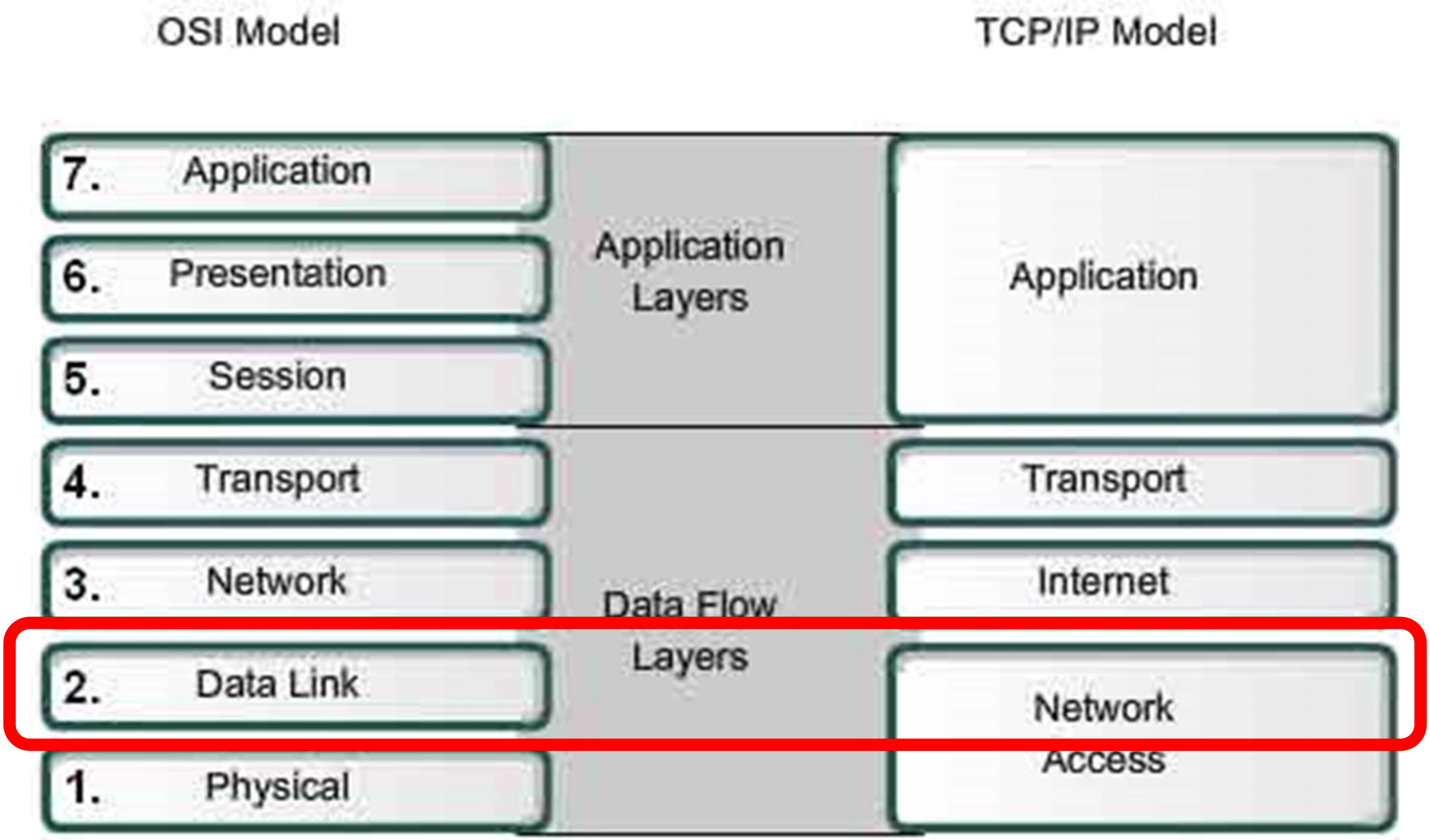
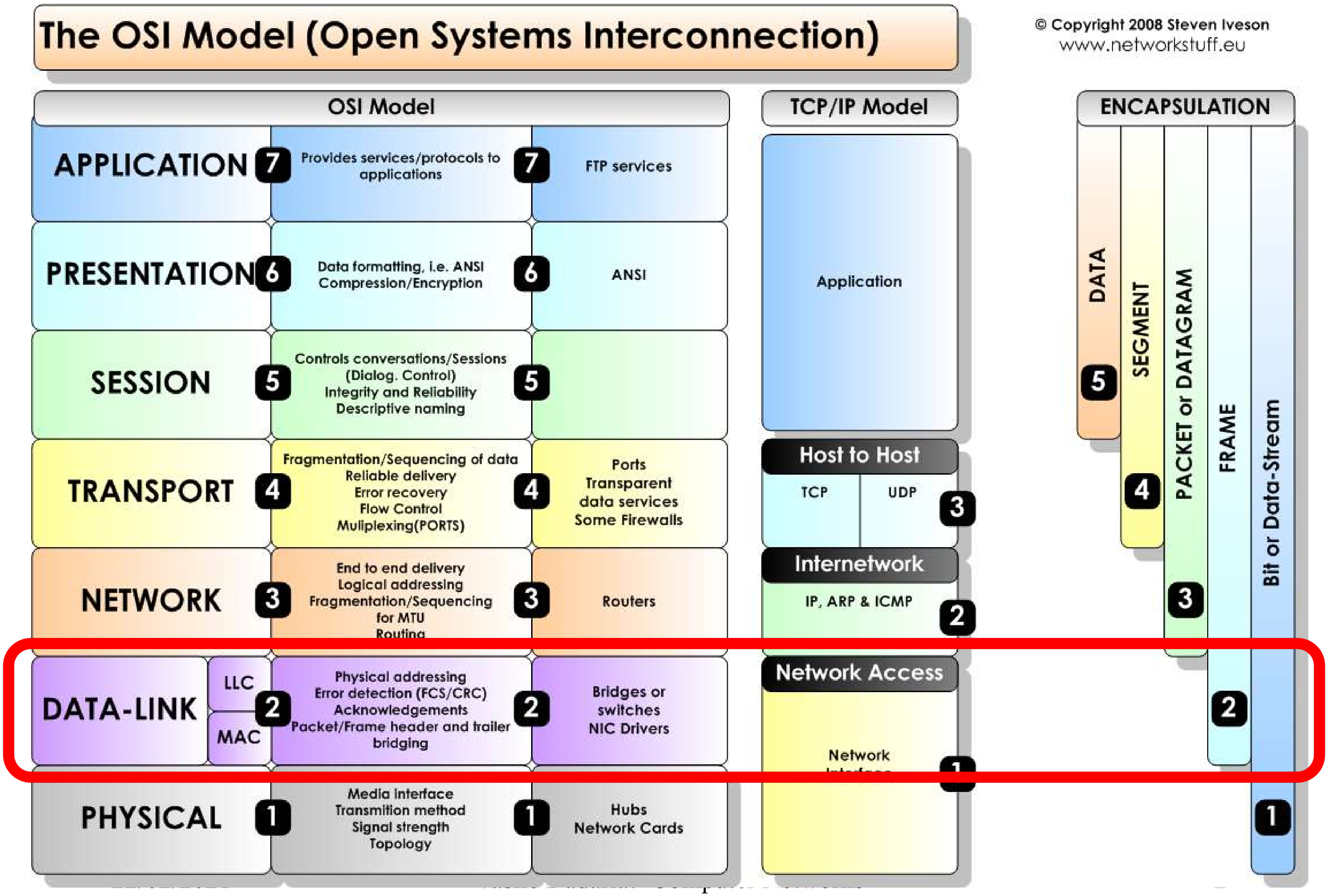

OSI Model
TCP/IP Model
7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data Link
1. Physical
Application Layers
Data Flow Layers
Application
Transport
Internet
Network Access


The OSI Model (Open Systems Interconnection)
© Copyright 2008 Steven Iveson
www.networkstuff.eu
OSI Model
TCP/IP Model
ENCAPSULATION
APPLICATION 7
Provides services/protocols to applications
7 FTP services
PRESENTATION 6
Data formatting, i.e. ANSI
Compression/Encryption
6 ANSI
SESSION 5
Controls conversations/Sessions
(Dialog. Control)
Integrity and Reliability
Descriptive naming
5
TRANSPORT 4
Fragmentation/Sequencing of data
Reliable delivery
Error recovery
Flow Control
Muliplexing(PORTS)
4
Ports
Transparent
data services
Some Firewalls
NETWORK 3
End to end delivery
Logical addressing
Fragmentation/Sequencing
for MTU
Routing
3 Routers
DATA-LINK
LLC
MAC
2
Physical addressing
Error detection (FCS/CRC)
Acknowledgements
Packet/Frame header and trailer
bridging
2
Bridges or
switches
NIC Drivers
PHYSICAL 1
Media interface
Transmition method
Signal strength
Topology
1
Hubs
Network Cards
Application
Host to Host
TCP
UDP
3
Internetwork
IP, ARP & ICMP
2
Network Access
Network
1
DATA 5
SEGMENT 4
PACKET or DATAGRAM 3
FRAME 2
Bit or Data-Stream 1

Data Link Control

On layer 1 (ISO/OSI) **physical** aspects of transmissions, or **how transmit signals on a transmission link**.

How control and manage the information exchange? Adding a little logic above the physical interface!

Follows the **Data Link** layer, concerned on **how to send data over a data communication link.**

Data unit at this level: the **frame**. Problems arising with frames transmission:

-**frame synchronization**

-flow control

-error control

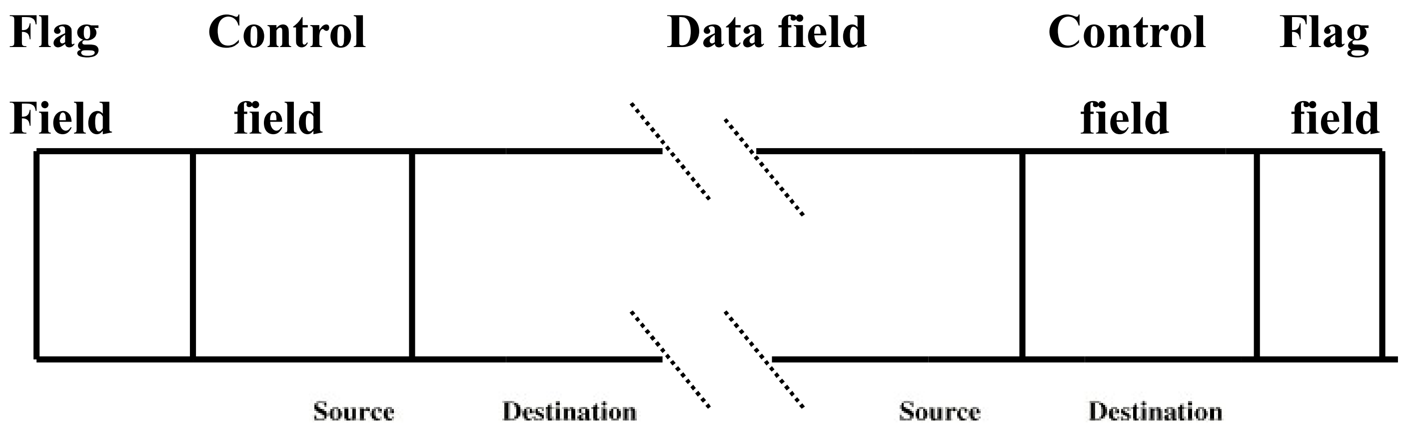
-addressing

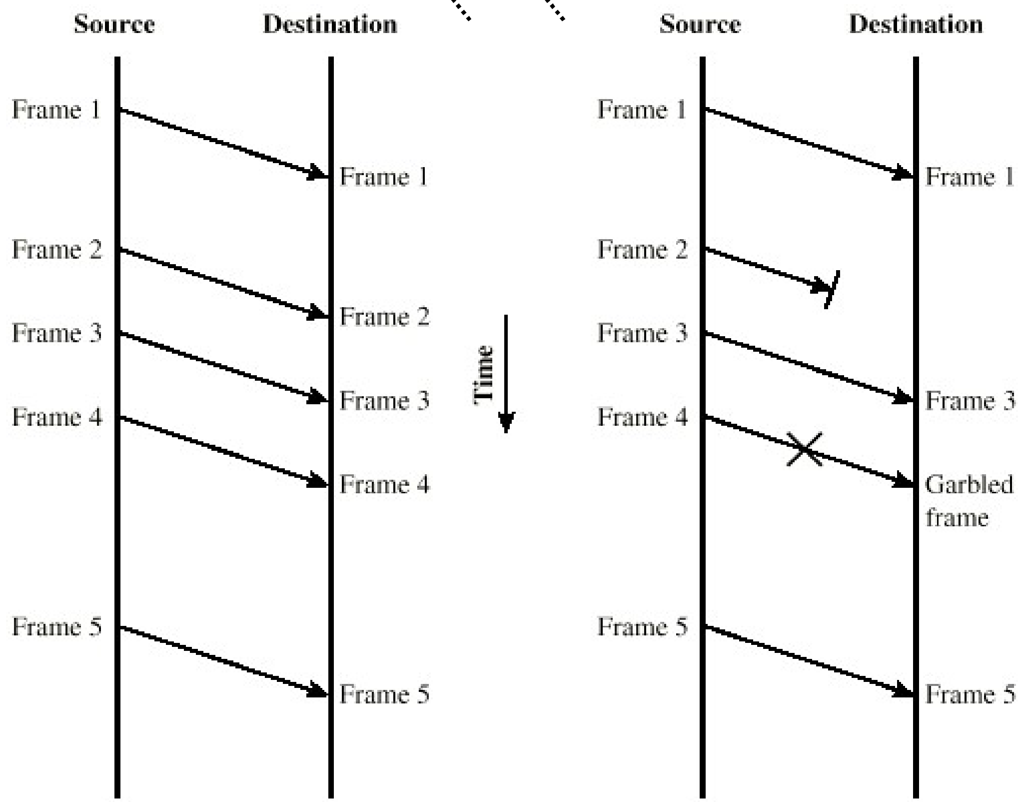
-carrying data and control on a single link

-link management

Frame synchronization

Beginning and end of frames need be detected => special synchronization fields, **flag** fields; may exist for both parts, also may use inter-frame transmission gap, instead of end flag.





(a) Error-free transmission

(b) Transmission with losses and errors

Flow Control

Technique ensuring the sending entity does not overwhelm the receiving entity, i.e. the receiver’s data buffer doesn’t fill up and overflow.

Temporal parameters used for transmission control:

Transmission time - time taken for a station to emit all bits into medium

Propagation time - time taken for a bit to traverse the link, from source to destination stations.

Flow Control Techniques

Stop and Wait

Sliding Window

Stop-and-Wait Flow Control Protocol

Algorithm steps:

-Source transmits a frame

-Destination receives frame and, if wants to continue, replies back with an acknowledgement (ACK) for that frame

-Source waits for ACK before sending next frame.

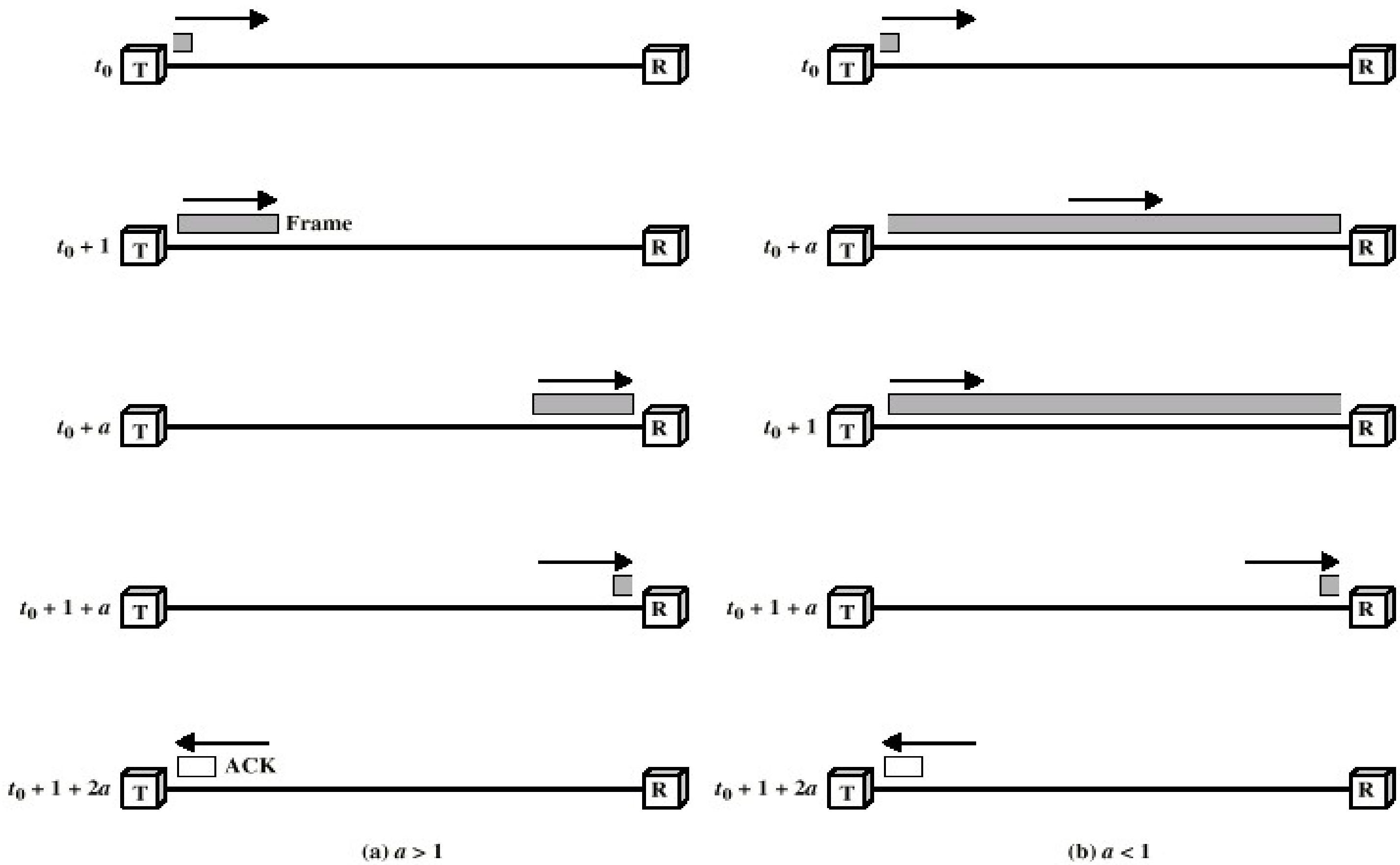
Protocol characteristics:

-Destination can stop the data flow by not sending ACK

-Works well for a few large frames; that's why use of frame fragmentation, i.e. large block of data may be split into small frames. It necessitates because:

- -Limited buffer size at destination
- -Errors detected sooner (when whole frame received)
- -On error, retransmission of smaller frames is needed
- -Prevents one station occupying medium for long periods (problem in LANs)

When use of multiple frames for same message, Stop and wait algorithm becomes inadequate, data link being not efficiently used. When propagation time >> transmission time (high data transmission speed or long distance) the line is under-utilized.



Stop-and-Wait Link Utilization (transmission time = 1; propagation time = *a*)

Sliding Windows Flow Control

Allows for multiple frames to be in transit on the link. Efficient algorithm for full duplex links and speedy transmissions.

Main hints for algorithm:

-Receiver has buffer W long

-Transmitter can send up to W frames without waiting for ACK

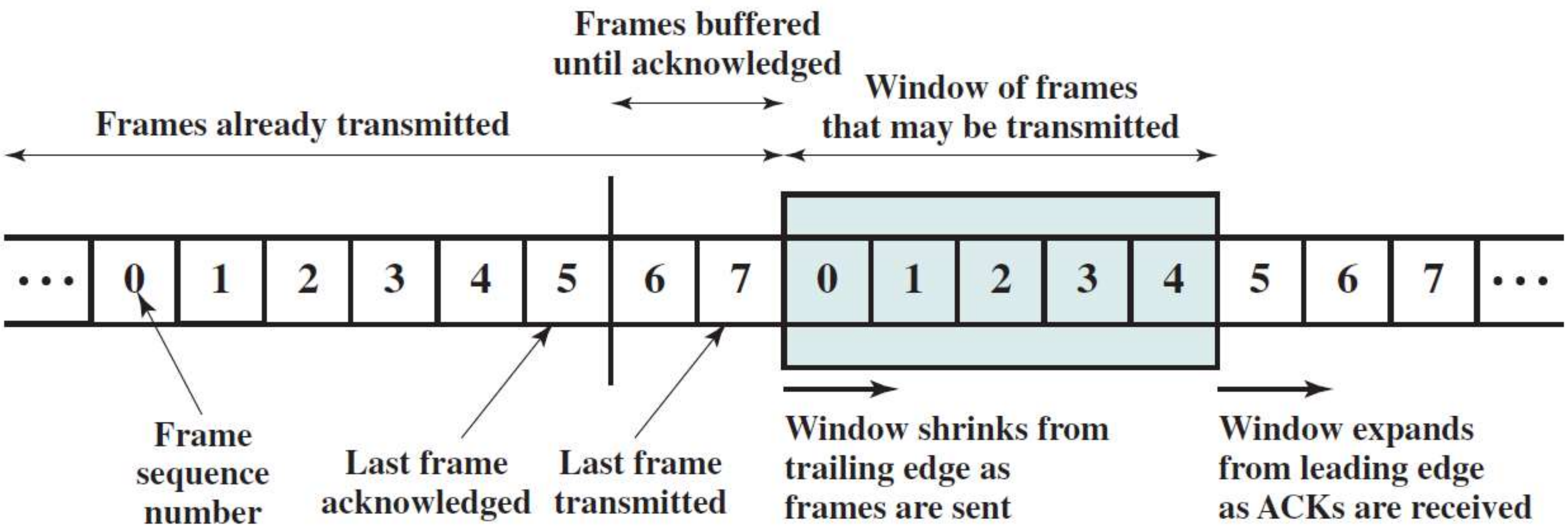
-for keeping track of acknowledged frames, each frame is numbered by the sender

-the receiver sends ACK frame, including number of *next frame expected*

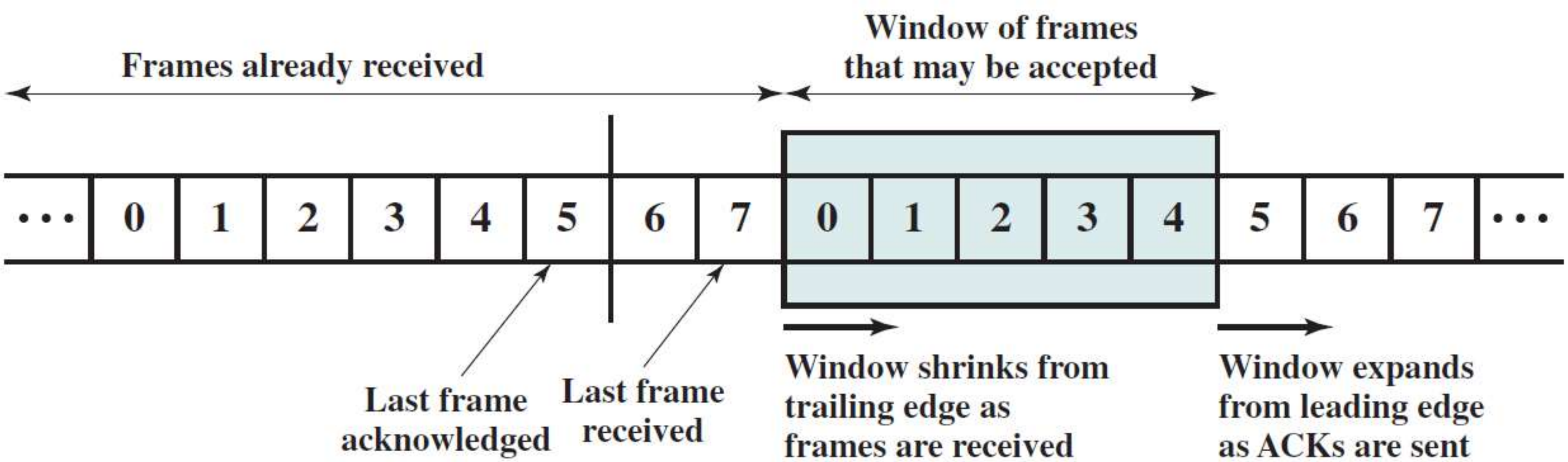
-Sequence number is bounded by the size of control field in the frame (k), so frames are numbered modulo 2^k

-the window size usually smaller

-Sender maintains a list with the sequence numbers it is allowed to transmit, the receiver maintains a list with the sequence numbers is prepared to receive; so a *window* of frames; operation referred as sliding-window flow control



(a) Sender's perspective



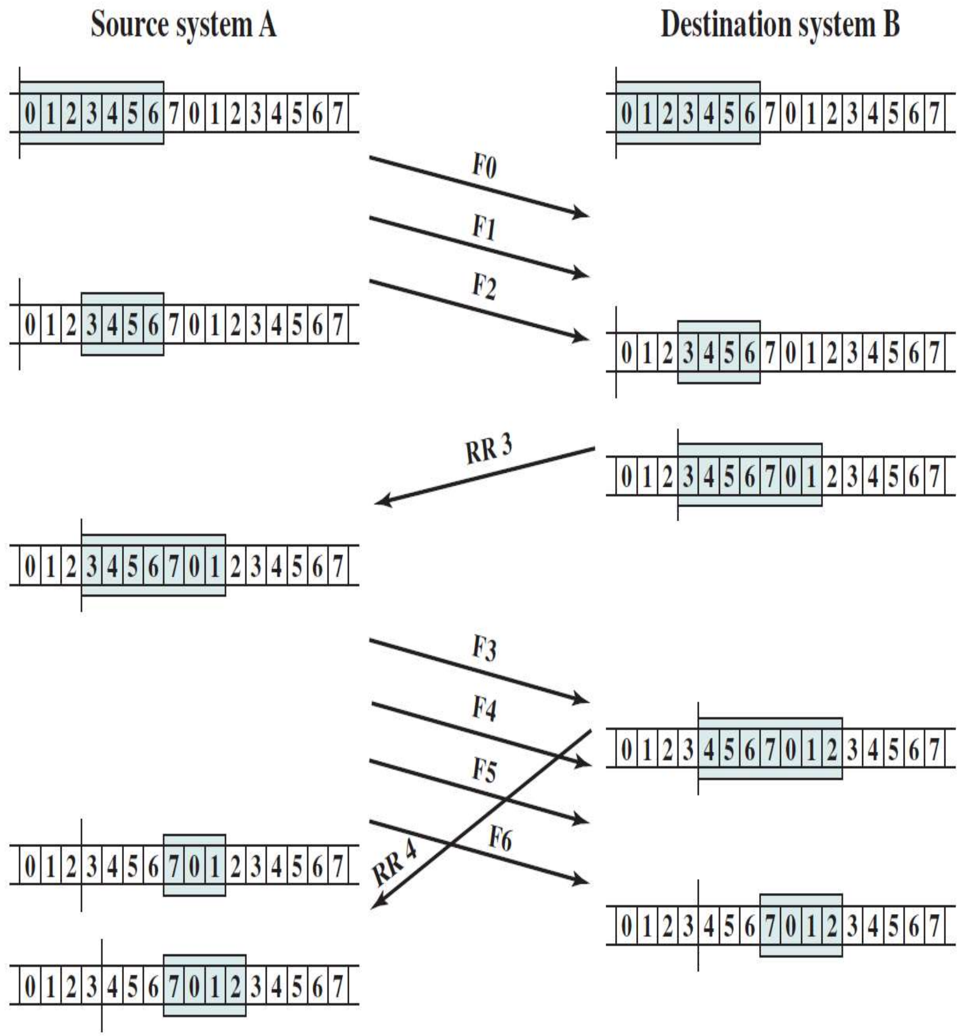
(b) Receiver's perspective

F0 – F7: normal data frames

RRn: ack frames (Receive ready), where **n** the number of the next frame expected at receiver

A form of 'negative acknowledgment', like RNRn (Receive not Ready), acts as ACK for former frames, but forbidding transfer of future frames; transmission cut-off.

Resume with a positive ACK.



When bidirectional transmissions, both stations implement 2 windows.

The ACK information may be sent as special frames (RR, RNR), or be embedded in data frames, for transmission efficiency (piggybacking) => frame has special fields carrying it's own frame sequence number and sequence number used for counterpart acknowledgement.

Error Detection

Even the media is secure, data error may occur, with different probabilities. The error may affect one bit (bit error rate), affect more bits, and error may be (or not) **detected** by the receiver. ***Detection doesn't imply correction!***

Use of *additional bits*, added by transmitter, for implementing an error detection code, calculated as a function of transmitted bits. Error-detection code functions need be known by both transmission parts. Receiver performs same calculation over received bits and compares the results (its error detection code with that arrived from the sender). If mismatch, a detected error occurs!

Parity Check

Simplest method, cheap and easy to implement: append a parity bit at the end of a block of data (ex.: a character), in such that the entire block (character) – after appending- has an even (even parity) or odd (odd parity) number of ones.

Works well for one damaged bit, or an even number of them, but can't detect an odd number of damaged bits.

Odd Parity used for asynchronous transmissions, Even Parity usually for synchronous transmissions.

Cyclic Redundancy Check (CRC)

Very powerful; acts as it follows:

For a block of *k* bits (the original message), transmitter generates a *n* bit sequence, known as *frame check sequence* (FCS).

The entire frame becomes *k+n* bits, which is exactly divisible by some number (predetermined divisor)

Receiver divides frame by that number

If no remainder, assume no error!

Predetermined divisor, represented as polynomials (variable X) is one of the CRC polynomials:

CRC-12 = $X^{12}+X^{11}+X^3+X^2+X+1$

CRC-16 = $X^{16}+X^{15}+X^2+1$

CRC-CCITT = $X^{16}+X^{12}+X^5+1$

CRC-32 = $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$

Error Control

Means not only error **detection**, but also **correction** of detected errors

In transmission of frames, there are some types of errors:

- **Lost frames**
- **Damaged frames**

Mechanisms for controlling frame errors: **Automatic repeat request** (ARQ), based on:

- Error detection
- Positive acknowledgment
- Retransmission after timeout
- Negative acknowledgement and retransmission

Role of ARQ: turn a potentially unreliable data link into a reliable one

Versions of ARQ:

- **Stop-and-wait ARQ**
- **Go-back-N ARQ**
- **Selective-reject ARQ**

Stop and Wait ARQ

Based on stop-and-wait flow control. Steps:

-Source transmits one single frame

-Stops and Wait for ACK

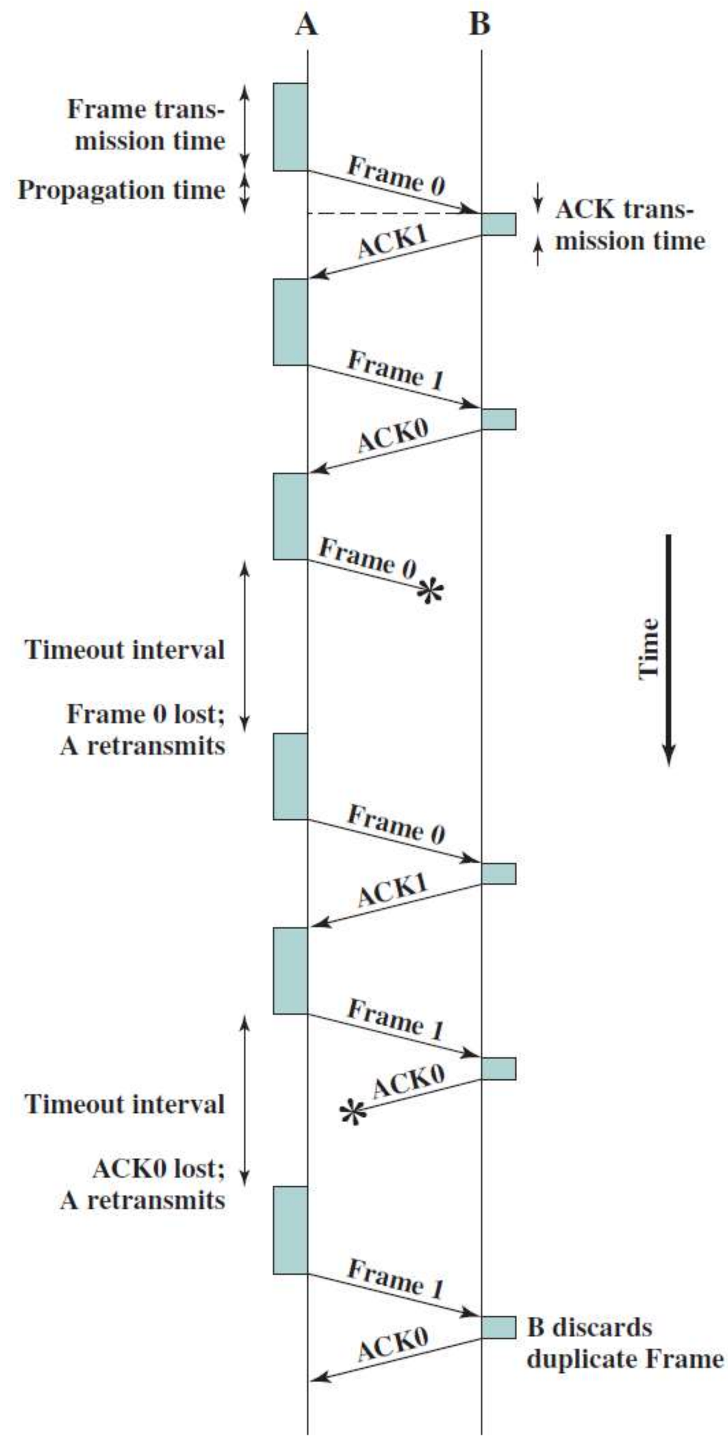
May have two kind of errors:

If received frame is damaged, receiver discards it

- -Transmitter has timeout
- -If no ACK within timeout, source retransmits frame (need for a frame copy)

If ACK damaged, transmitter will not recognize it

- -Transmitter will retransmit
- -Receiver gets two copies of same frame, as if they were separated
  - Alternative frame labelling (1 and 0), so use of ACK0 and ACK1



Stop and Wait ARQ: simple but not efficient

Go Back N ARQ

Based on sliding window flow control => frames numbered sequentially modulo the window size.

Steps are:

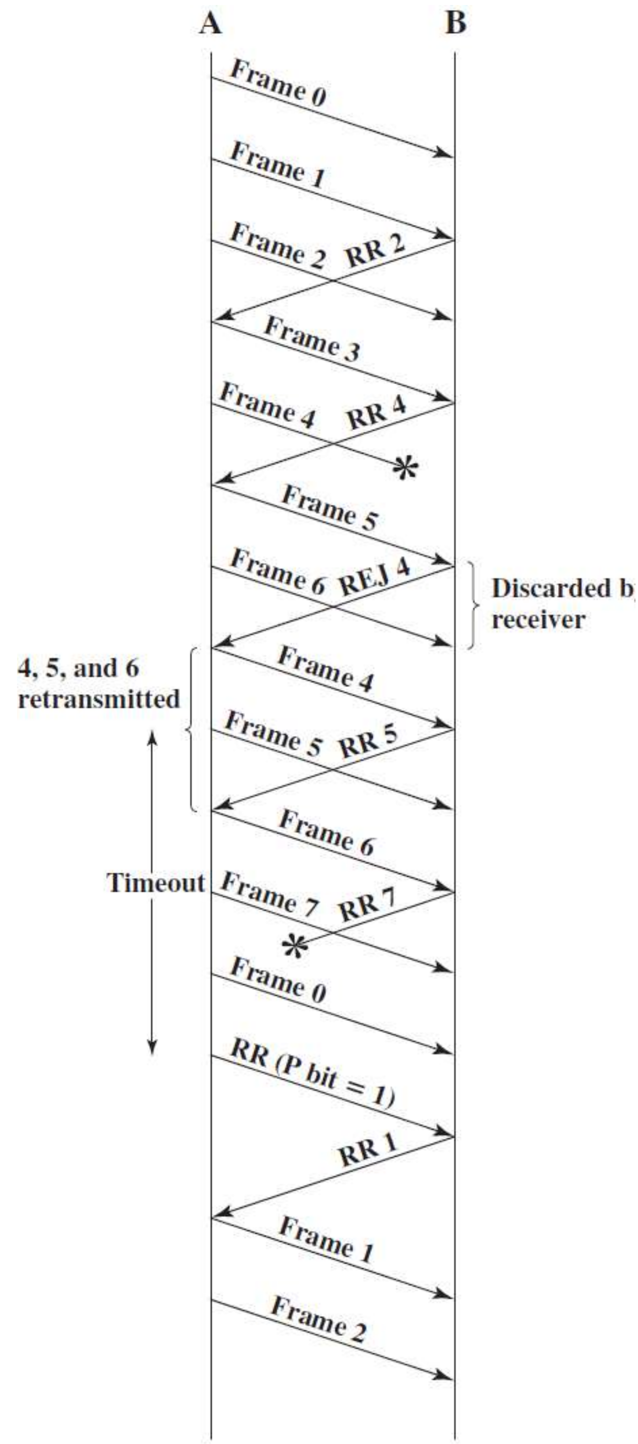
-If no error, ACK used as usual, carrying number of next frame expected (RRn, or piggybacking); use window to control number of outstanding frames

-If error, reply with rejection (RNRn or REJn, or piggybacking)

  -Receiver discards that frame and all future frames, until erroneous frame received correctly

  -Transmitter must go back and retransmit that frame and all subsequent frames

Need for buffers at the sender and receiver.



Go Back N - *Damaged Frame*

Receiver detects error in frame *4*

Receiver sends rejection-*4*

Transmitter gets rejection-*4*

Transmitter retransmits frame *4* and all subsequent

Go Back N - *Lost Frame* – case 1

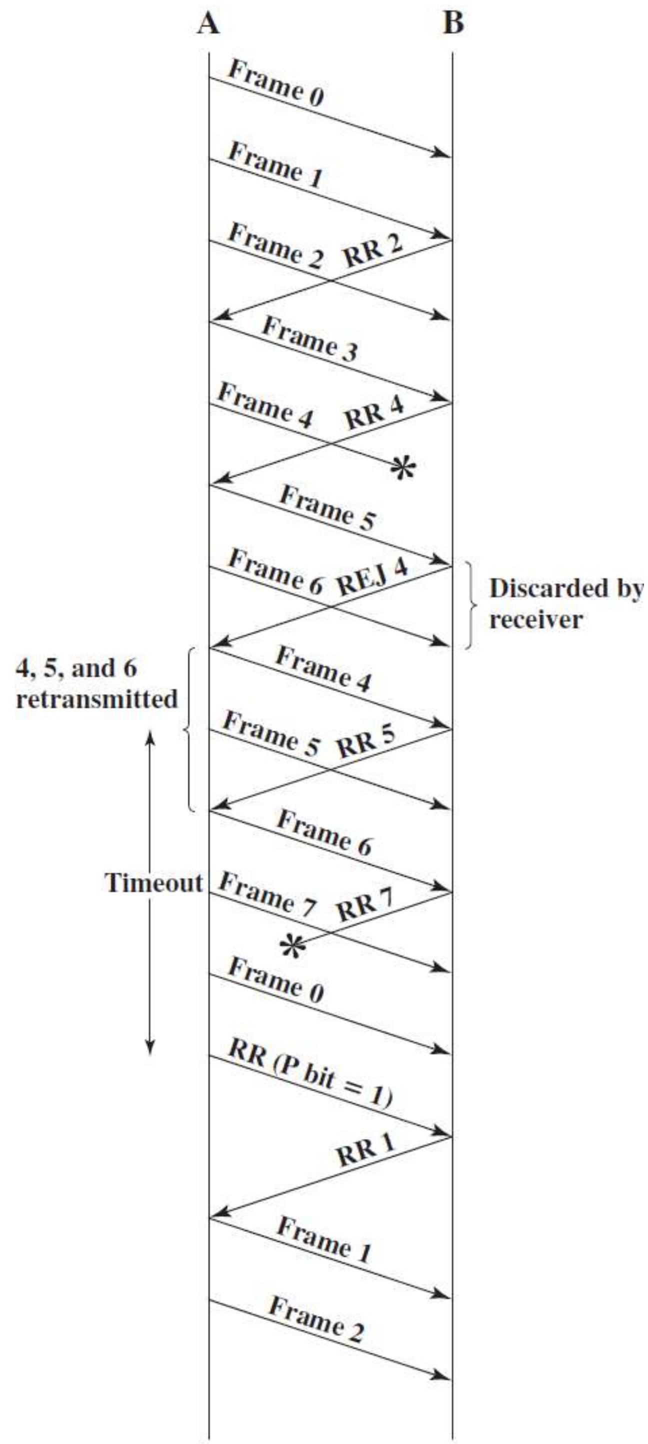
Frame *i* lost, so not feedback from receiver

Transmitter sends *i+1*

Receiver gets frame *i+1* out of sequence

Receiver send reject *i*

Transmitter goes back to frame *i* and retransmits all subsequent frames



Go Back N - *Lost Frame* case 2

Frame *i* lost and no additional frame sent

Receiver gets nothing and returns neither acknowledgement nor rejection

Transmitter times out and sends a special acknowledgement frame with P bit set to 1

Receiver interprets this as command which it acknowledges with the number of the next frame it expects (frame *i*)

Transmitter then retransmits frame *i*

Go Back N - *Damaged Acknowledgement*

Receiver gets frame *i* and send acknowledgement (*i+1*) which is lost

Acknowledgements are cumulative, so next acknowledgement (*i+n*) **may arrive**

before transmitter times out on frame *i*

If transmitter times out, it sends acknowledgement with P bit set, as before

This can be repeated a number of times before, and if successive failures, a reset procedure is initiated

Go Back N - *Damaged Rejection*

As for lost frame (2)

Selective Reject ARQ

Also called selective retransmission; Steps are:

-Only rejected frames are retransmitted

-Subsequent frames are accepted by the receiver and buffered, till valid frame received

-Reordering of frames

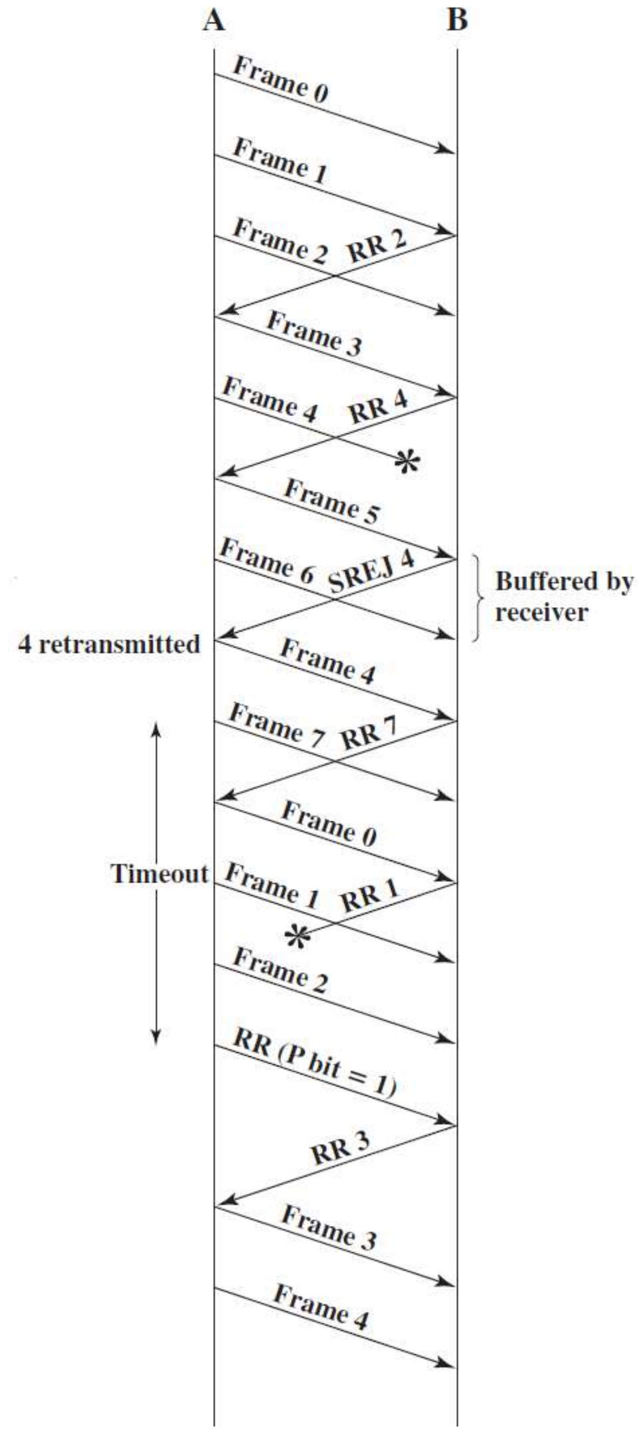
Advantages:

-Minimizes retransmissions

Complexity:

-Receiver must maintain large enough buffer

-More complex logic in transmitter (selective choose and retransmission of erroneous frame)



Data Link Control Protocols

Early DL control protocols – character oriented: IBM's BISYNC, ARPA's IMP-IMP
Now bit-oriented protocols: IBM's **SDLC** (Synchronous Data Link Control), modified by ISO 4335 standard and becoming **HDLC** (High-level Data Link Control).

HDLC (High-level Data Link Control)
Basic characteristics:

-Station Types

- -Primary station
  - Controls operation of link
  - Frames issued are called commands
  - Maintains separate logical link to each secondary station
- -Secondary station
  - Under control of primary station
  - Frames issued called responses
- -Combined station
  - May issue commands and responses

- **Link Configurations**

-Unbalanced

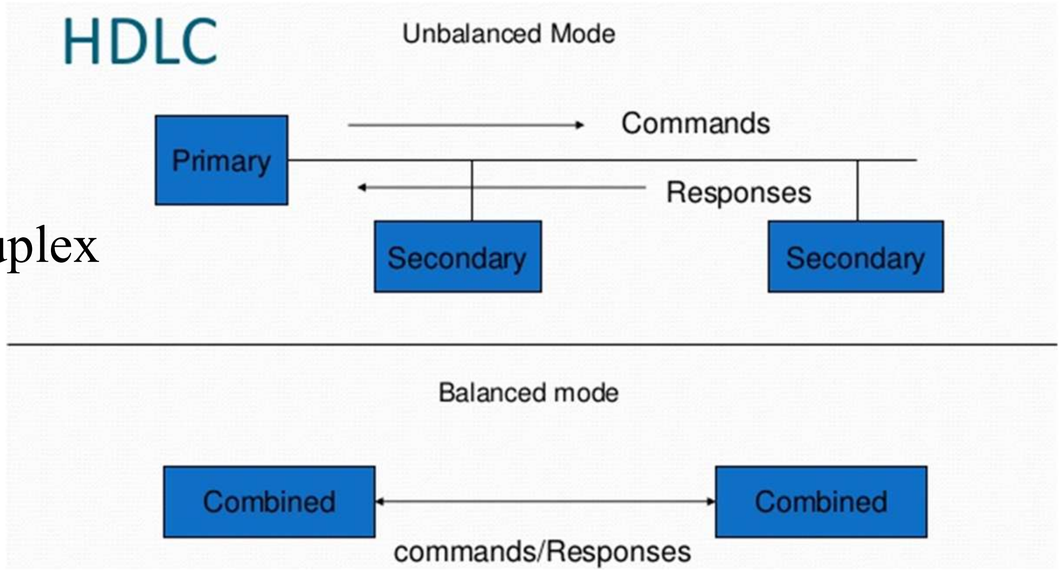
One primary and one or more secondary stations

Supports full duplex and half duplex

-Balanced

Two combined stations

Supports full duplex and half duplex



-Transfer Modes

- Normal Response Mode (NRM)
- Asynchronous Balanced Mode (ABM)
- Asynchronous Response Mode (ARM)

Normal Response Mode (NRM)

-For unbalanced configuration

-Primary station initiates transfer to secondary station(s)

-Secondary may only transmit data in response to command from primary

-Primary ***polls*** secondary for transmitting

Used on multi-drop lines and daisy-chain polling; host computer as primary, terminals as secondary

Asynchronous Balanced Mode (ABM) - *most widely used*, no polling overhead

-Balanced configuration (combined stations)

-Either station may initiate transmission without receiving permission

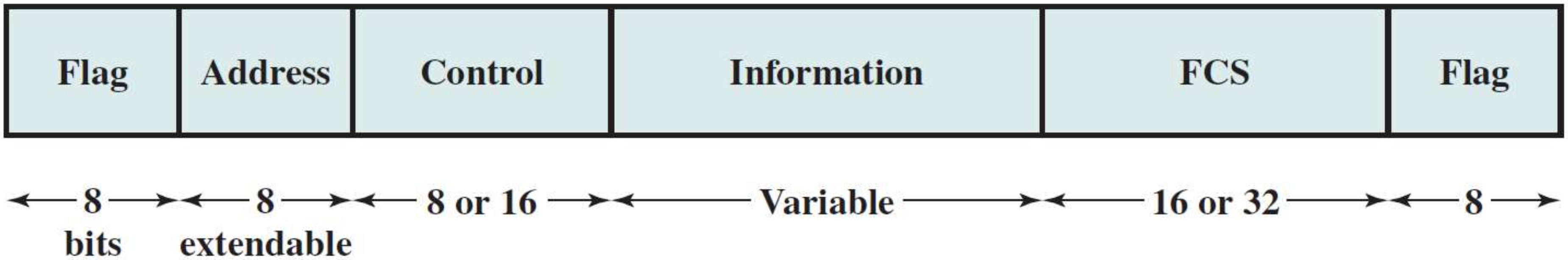
Asynchronous Response Mode (ARM)

-Unbalanced configuration

-Secondary may initiate transmission without permission from primary

Primary responsible for line management, ***rarely used*** method.

Frame Structure Diagram



HDLC uses synchronous transmission => synchronization fields

All transmissions at the DL level are done using frames

Single frame format for all data and control exchanges

Frame contains a *header* and a *trailer*, information embedded between them.

Header = **Flag + Address + Control** fields

Trailer = **FCS + Flag** fields

FCS (Frame Control Sequence), using CRC error control

Flag Fields

Delimit frame at both ends, used for frame sequence synchronization; receiver hunts for flag sequence to synchronize

Normal pattern: 01111110 (six ones between zeros)

What to do when data contains this pattern?

Bit stuffing used to avoid confusion with data containing 01111110, and to assure data transparency.

Rule:

- **0 inserted after every sequence of five 1s** at the sender part.
  - If receiver detects five 1s it checks next bit:
    - if 0, it is deleted;
    - if 1 and seventh bit is 0, accept as flag
    - if sixth and seventh bits 1, sender is indicating abort (severe error)

Original pattern:

111111111111011111101111110

After bit-stuffing:

1111101111101101111101011111010

(b) An inverted bit splits a frame in two

Address Field

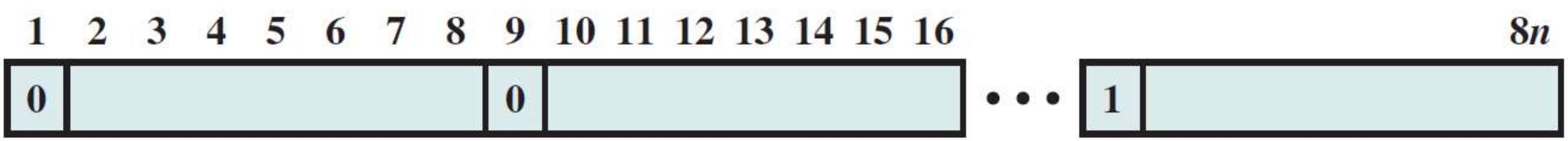
Usually 8 bits long, indicating the secondary station having transmission or to receive frame. Not useful for point-to-point links.

Usually 8 bits enough for addressing 255 secondary stations, even 127 when 7 bits used.

Address field may be extended to multiples of 7 bits, using an 'a priori' rule:

LSB of each octet indicates that it is the last octet (if 1) in the string, or not (if 0)

Special address for all stations: all ones (11111111) is **broadcast** address.



Control Field

HDLC defines three types of frames, with different control formats

Information (I-frame): data to be transmitted to user (next layer up)

Flow and error control piggybacked on information frames

Supervisory (S-frame): control for ARQ when piggyback not used

Unnumbered (U-frame): supplementary link control, if needed

Bit significance:

First one or two bits of control field identify **frame type** (bit 1, or 1 and 2)

Poll/Final Bit, used on the context:

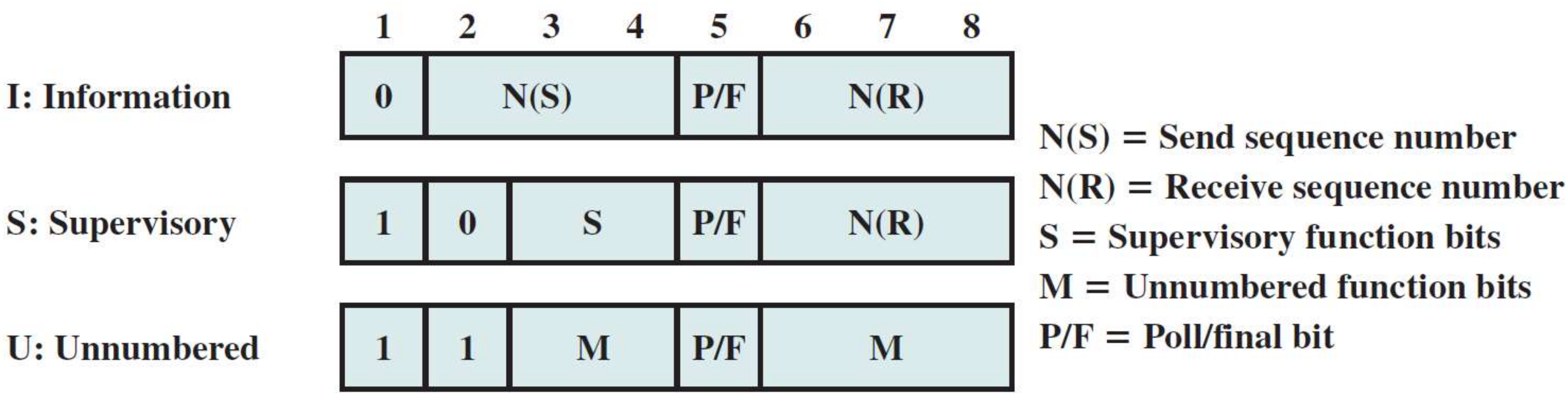
-if Command frame, means P (Polling bit), value 1 to solicit (poll) response from peer

-if Response frame, means F (Final bit), value 1 indicates the end of the response sequence to soliciting command

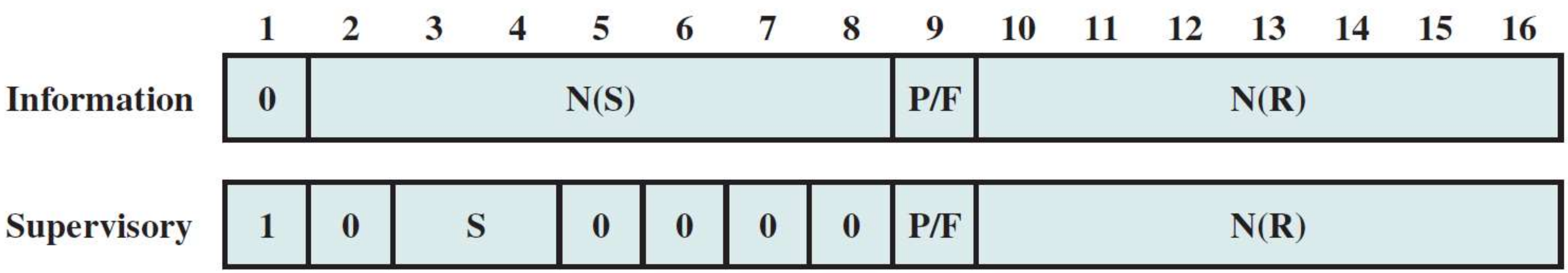
Sequence bits:

N(S) – frame’s sequence number at transmitter

N(R) – frame’s sequence number at receiver, means piggybacked ACK



8-bit control field format



16-bit control field format

Supervisory bits S, coding following commands and responses:
-00: Receive Ready – station ready for receiving or acknowledgement for N(R)
-01: Reject – retransmission request for messages starting with sequence number N(R)
-10: Receive Not Ready – acknowledgement for messages till N(R-1), but no more able to receive
-11: Selective Reject – retransmission request for message N(R)

Unnumbered function bits – M fields (2+3 bits): for further developments, allowing 32 control or response functions; may suffer further extensions.
Examples of coded commands/responses:
-11000: set ARM (Asynchronous Response Mode)
-00001: set NRM (Normal Response Mode)
-11100: set ABM (Asynchronous Balanced Mode)
-11011: set NRME (Normal Response Mode Extended)

For more commands/responses see references (ex. Stallings, 6th ed., pp.218)

Information Field

Carries user data; only in information (I-frame) and some unnumbered (U) frames

Must contain an integral number of octets

Variable length

Frame Check Sequence Field (FCS)

Error detection field; applies over Control and Information field

16 bit CRC (usually CRC-CCITT)

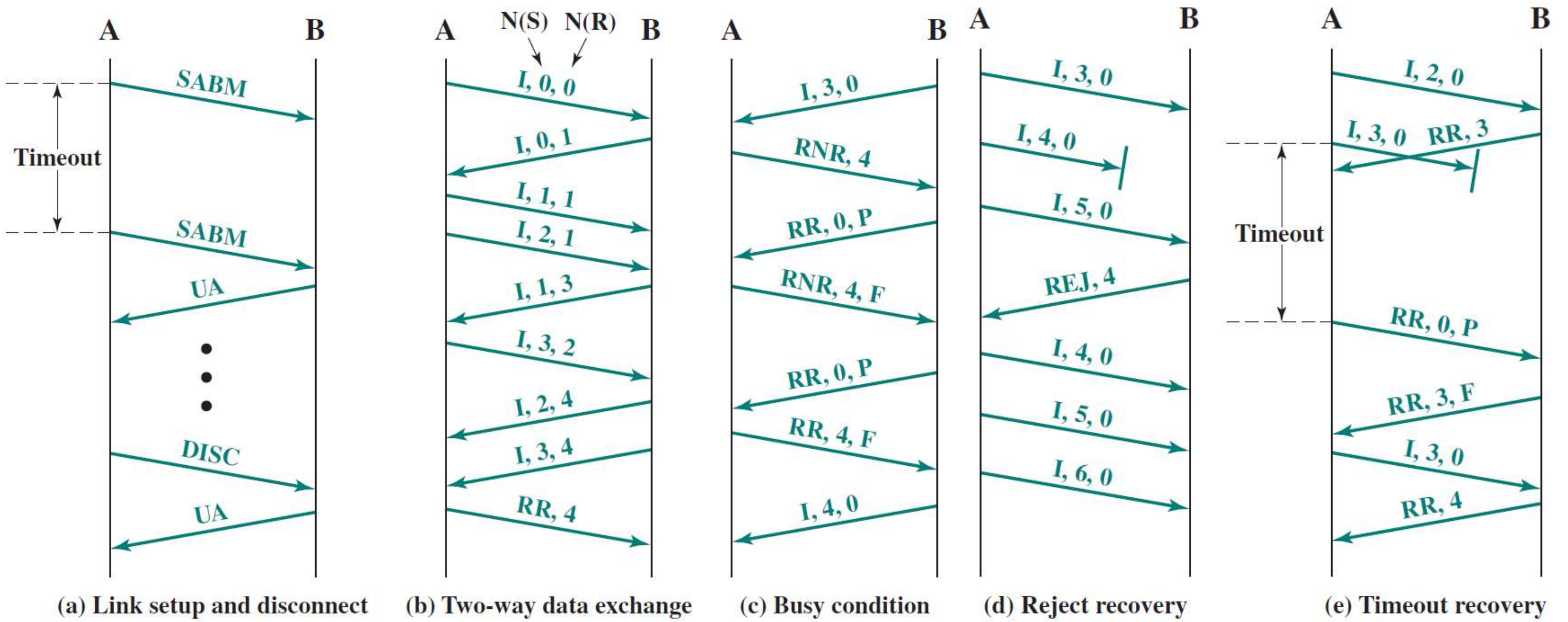
Optional 32 bit CRC (CRC-32), if info field length and link reliability calling for.

HDLC Operation

Consist in exchange of information, supervisory and unnumbered frames between the data link stations.

Any HDLC operation has three phases:

- **Initialization**, requested by any side, using a set-mode command; after this, if operation accepted by other side, a logical connection is established
- **Data transfer**, using data carrying I-frames, but also for flow and error control purposes, use of S-frames
- **Disconnect**, requested by any side, on its own initiative (if some sort of fault), or at the request of the higher level protocol (user request).



Examples of various operations (analysed by yourself)

Other DLC Protocols

Link Access Procedure, Balanced (LAPB)

Part of X.25 (ITU-T), as level 2 base protocol

Subset of HDLC – provides ABM

Point to point link between system and packet switching network node

Link Access Procedure, D-Channel (LAPD)

Part of the ITU-T recommendation for ISDN; data link control over D channels

Implements only HDLC- ABM, but there are differences:

Always 7-bit sequence numbers (no 3-bit)

16 bit address field, contains two sub-addresses: one for device and one for user (next layer up), allowing multiple devices for same user, or multiple logical users of LAP-D

16-bit CRC

Logical Link Control (LLC) - IEEE 802.2

- Component, with MAC (Medium Access Control) of the Data Link level of LANs
- Manages only the logic control of a data link
- LLC frame embedded in MAC frame
- Different frame format as a HDLC frame
- No primary and secondary stations - all stations are peers (specific to LANs)
- Two station addresses needed (appear in MAC frame):
  - Sender and receiver addresses
- LLC deals with interaction points with upper level:
  - Destination and source access points (DSAP, SSAP)
- Error detection done at MAC layer
  - Use of 32 bit CRC

LLC offers three forms of services:

- connection-oriented, similar with HDLC – ABM
- acknowledged connectionless
- unacknowledged connectionless (pure datagram service)

Frame Relay

Streamlined capability over high-speed packet-switched networks

Used in place of X.25, imposed by US

Frame Relay uses as Data Link Control the **Link Access Procedure for Frame-Mode Bearer Services** (LAPF)

In fact there are two protocols in LAPF:

Control protocol- similar to HDLC

Core protocol- subset of control protocol

Differences of control protocol vs. HDLC:

7-bit sequence numbers

16 bit CRC

2, 3 or 4 octet address field; from those bits 10, 16 and 23 respectively, implement the Data link connection identifier (DLCI); rest are flow control bits

DLCI identifies logical connection between stations.

Frame Relay **LAPF core** protocol similar with LAPF control, but allows streamlined operations, being not concerned with flow and error control

ATM (Asynchronous Transfer Mode)

Streamlined capability across high speed networks

Not HDLC based

Frame format called **cell**, providing minimum processing overhead

Fixed 53 octet (424 bit): 5 control and 48 payload.