Data communication I

Lecture 4 –

flow control, error control, multiplexing

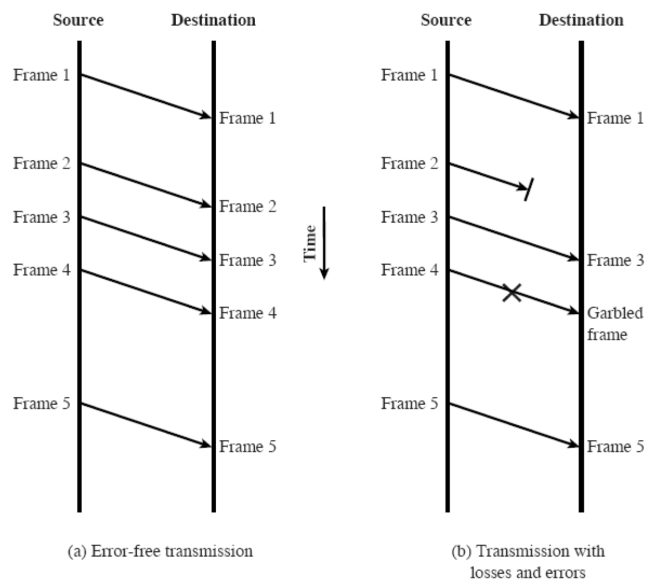
Layer 2 functionality

- Flow control
  - The receiver regulates the data rate coming from the transmitter
- Error control
  - An error was detected and could not be corrected by the receiver itself
  - The receiver asks for retransmission
- Multiplexing
  - Sharing the medium
- Medium Access Control (MAC)
  - Who gets access to the medium, when and for how long?

Flow control

- The transmitter might send data at a higher rate than the receiver is able to handle
- Receiving data
  - Each data frame needs to be processed before being sent to the upper layers (e.g. error detection/correction)
  - Meanwhile the frames are stored in input buffers
  - Too high data rate:
    - Buffers overflow (buffer = small memory)
    - Perfectly good data has to be thrown away
- Flow control = the receiver tells the transmitter to adjust the data rate

Model of frame transmission



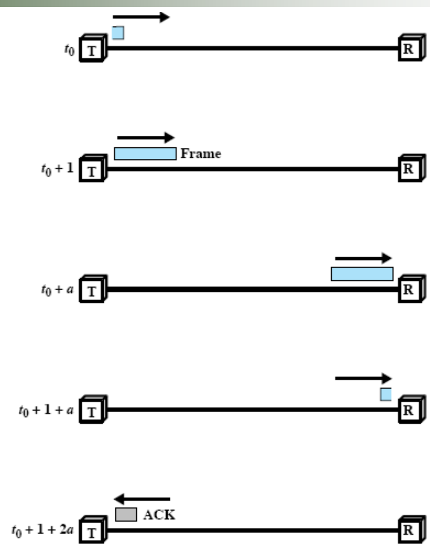
”Stop-and-wait” flow control

- Receiver sends an acknowledgement (ACK) for each received packet back to the transmitter
- Transmitter is only allowed to send the next data frame if the previous data frame is acknowledged
- Problems with this approach?

”Stop-and-wait” flow control

- Receiver sends an acknowledgement (ACK) for each received packet back to the transmitter
- Transmitter is only allowed to send the next data frame if the previous data frame is acknowledged
- Advantage
  - Receiver can regulate the flow on a per frame level
- Problems with this approach?
  - Only one frame is on the link at a time
  - Relatively short frames on a long link
    - Link is idle for a long times
    - Inefficient link utilization

Inefficient link utilization with ”stop and wait”



(b) $a > 1$

(transmission time = 1; propagation time = a)

So, why don’t we send long frames anyway?

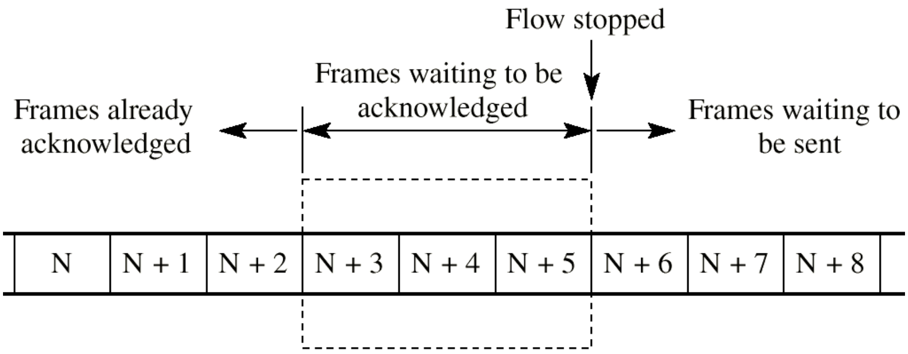
So, why don't we send long frames anyway?

- Limited buffer size at the receiver
- The more bits in a frame the higher the probability of an error
  - A retransmission or the loss of a frame is more costly
- On a shared link one station is not allowed to occupy the link for a longer time

"Sliding window" flow control

- More than one frame on the link at a time
- Mechanism
  - Receiver specifies window (number) of frames that its buffer can accept
  - E.g., window size W = 3 means 3 frames in a row can be sent without acknowledgement
  - Transmitter sends W frames, then stops the flow until ACK is received
  - ACK specifies the sequence number of the next expected frame and thereby acknowledges that all previous frames were received correctly
- Requirements
  - Packets must be numbered (sequence number)
  - Full duplex link

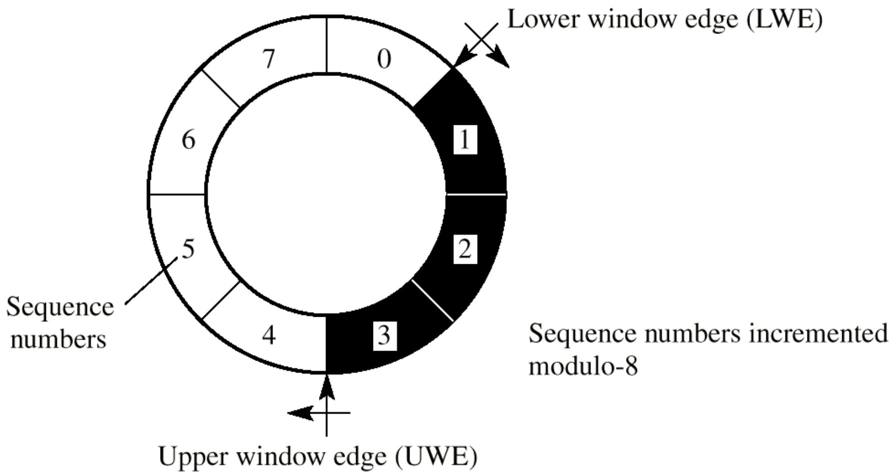
Transmitter perspective



- Window size W = 3
- No more than W frames are sent unacknowledged
- Transmitter stops data flow until ACK(N+6) is received

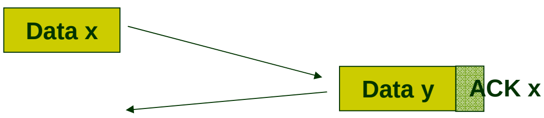
Sequence numbers

- Sequence number in the layer 2 frame header
- Limited number of bits for sequence number
  - E.g. 3 bits → numbers between 0 and 7 possible
  - Modulo numbering system
    - 3 bits sequence numbers given by modulo 8
  - Number of sequence numbers limits window size!
    - k bits for sequence number
    - 2^k sequence numbers
    - Maximum window size W = $2^k - 1$

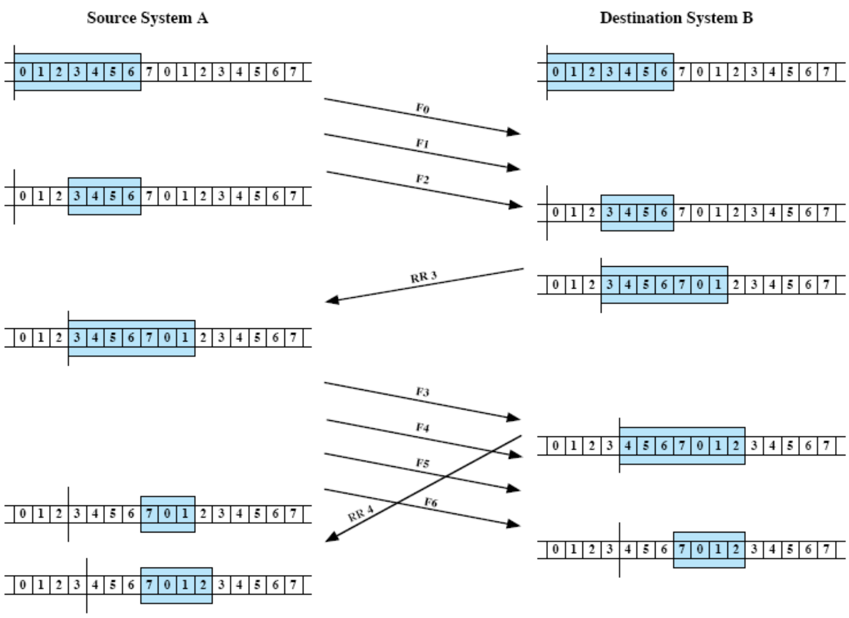


Acknowledgements

- Receive Ready (RR)
  - Receiver states that it is ready for the next W frames
  - At the same time all previously unacknowledged frames are acknowledged
- Receive Not Ready (RNR)
  - Acknowledges all previously unacknowledged frames
  - Tells the transmitter to stop the data flow (possibly due to limited receiver buffer resources)
  - Normal RR ACK must be sent to restart the transmission
- Pickyback
  - Data frames are exchanged between 2 stations
  - ACKs for received frames are added on to data frames (piggybacked)



Sliding window example



Flow control performance comparison

- T_{1frame} = time to send one frame over the link and receive an ACK for it
- $T_{propData}$ = time to send one data frame over the link
- $T_{transData}$ = time to send one data frame out on the link
- $T_{procData}$ = time to process the incoming data frame
- $T_{propACK}$ = time to send one ACK frame over the link
- $T_{transACK}$ = time to send one ACK frame out on the link
- $T_{procData}$ = time to process the incoming ACK
- T_{all} = time to send n data frames over the link and receive ACKs for them

$T_{1frame} = T_{propData} + T_{transData} + T_{procData} + T_{propACK} + T_{transACK} + T_{procACK}$

Approximately: $T_{1frame} = 2T_{propData} + T_{transData}$

$T_{all} = n(2T_{propData} + T_{transData})$

Error control

- Types of error
  - Lost frames
  - Damaged frames
- Receiver needs mechanisms to let the transmitter know
  - that an error was detected
  - or that everything is fine
- Automatic Repeat Request (ARQ)
  - Stop-and-wait ARQ (also known as ”Idle RQ”)
  - Continuous RQ
    - Go-back-N ARQ
    - Selective-reject ARQ (also known als ”selective repeat ARQ”)

Stop-and-wait ARQ

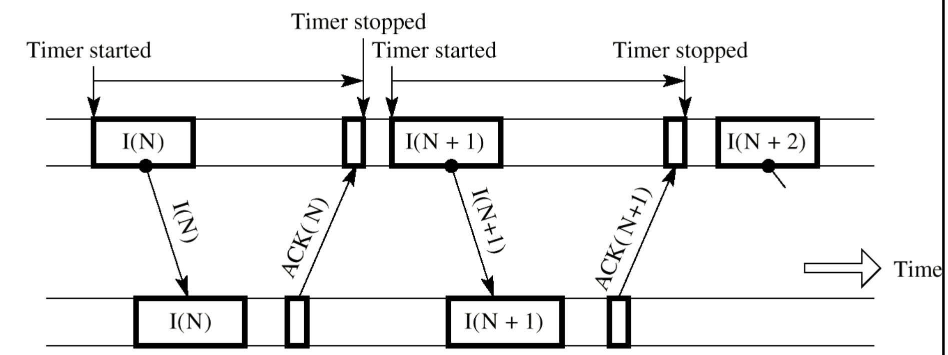
- Based on stop-and-wait flow control
  - Sending one frame at a time, awaiting ACK
- Mechanism

  Transmitter sends a frame and starts a timer

  1. Error-free frame arrives at the receiver
     - ACK is sent back to the transmitter
  2. Damaged frame arrives at receiver
     - No ACK is sent
     - The transmitter’s timer expires and it automatically retransmits the frame
  3. Frame is lost on the way
     - Same as in 2)
  4. Data frame is error-free, but the ACK is lost or damaged
     - Time-out occurs at the transmitter, who retransmits the packet
     - Receiver simply discards the second packet

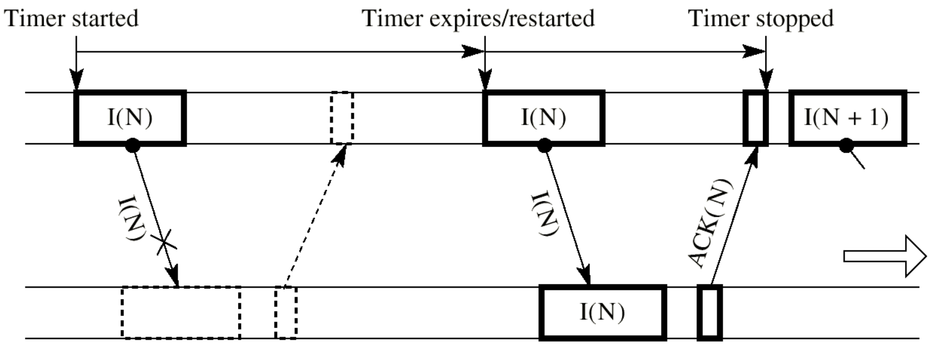
Stop-and-wait ARQ

- No damaged or lost frames
- Positive acknowledgement (ACK) used, (usually of RR type)



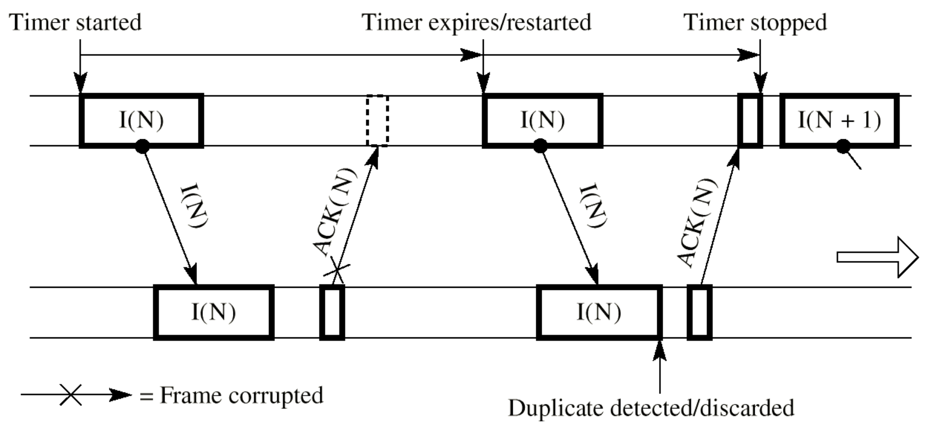
Stop-and-wait ARQ

- Lost or damaged frame
- Positive acknowledgement (ACK)



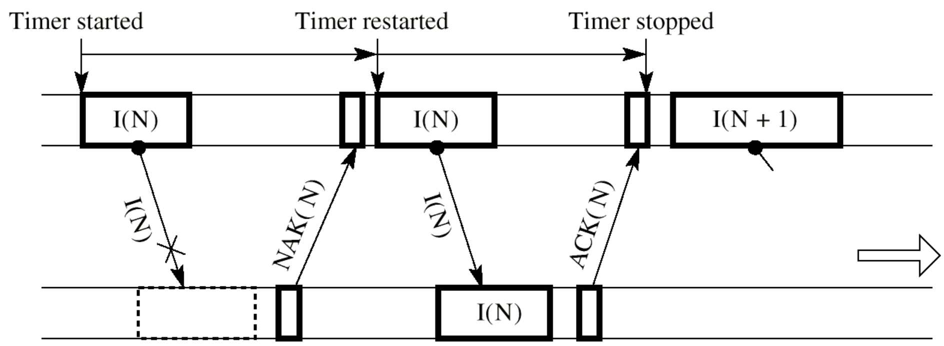
Stop-and-wait ARQ

- Lost or damaged ACK
- Positive acknowledgement (ACK)



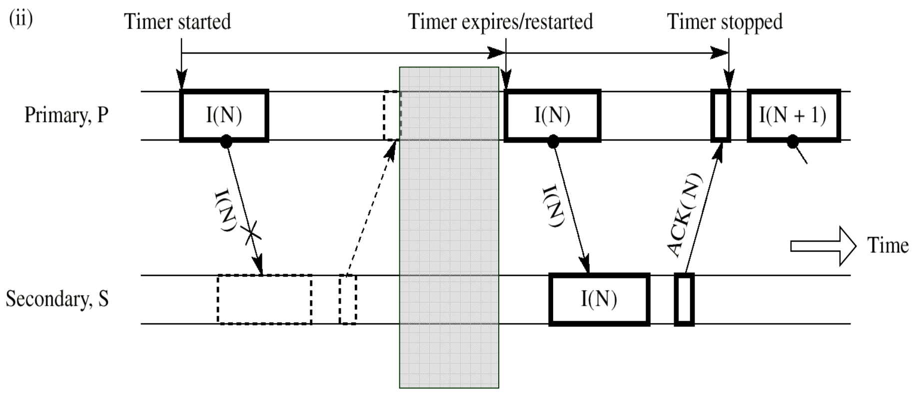
Stop-and-wait ARQ

- Damaged frame
- Negative acknowledgement (NACK) (or REJ for ”reject”)



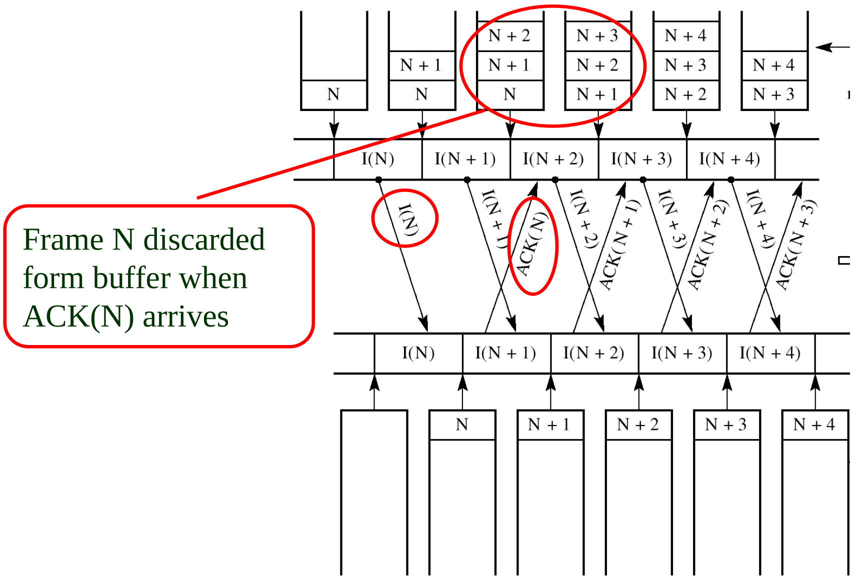
Advantage of negative acknowledgement (NACK)

- Transmitter does not have to wait until the timer expires
- Better use of resources



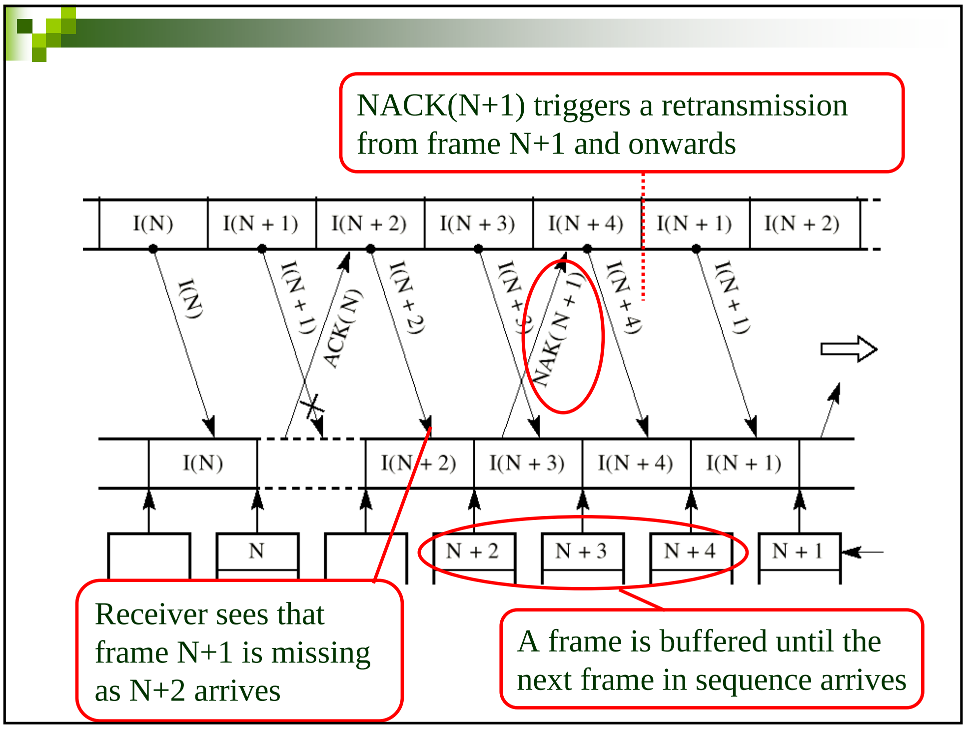
Continuous RQ

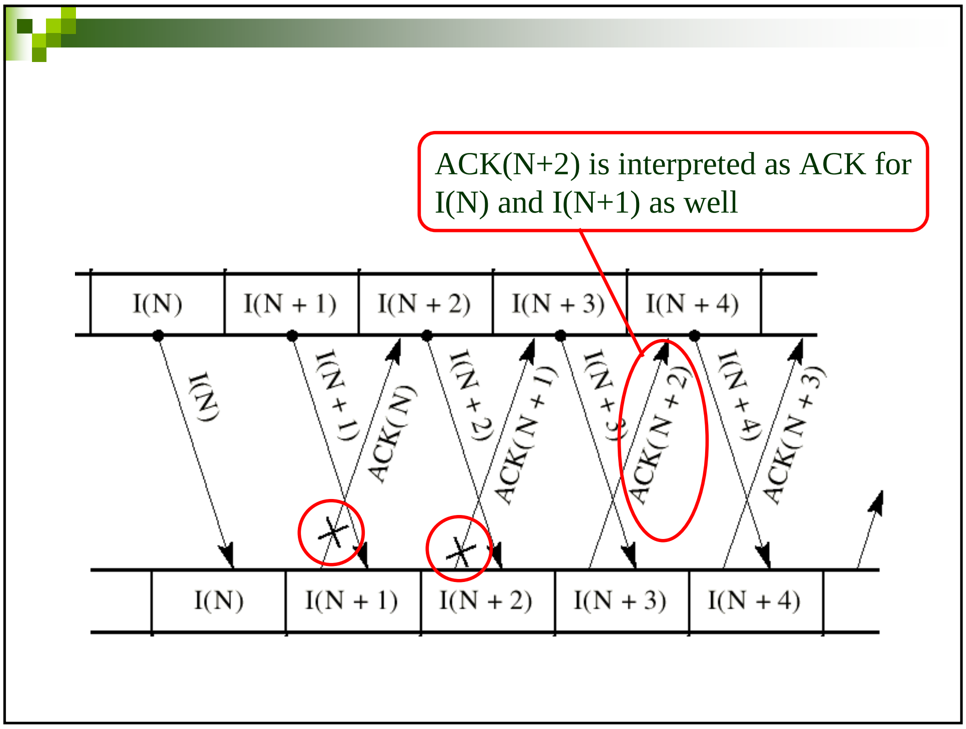
- Based on sliding window flow control
  - A series of sequentially numbered frames is sent
  - Transmitter side
    - All frames that were not acknowledged yet are buffered
  - Receiver side
- Two types
  - Go-back-N ARQ
  - Selective-repeat ARQ



Go-back-N ARQ

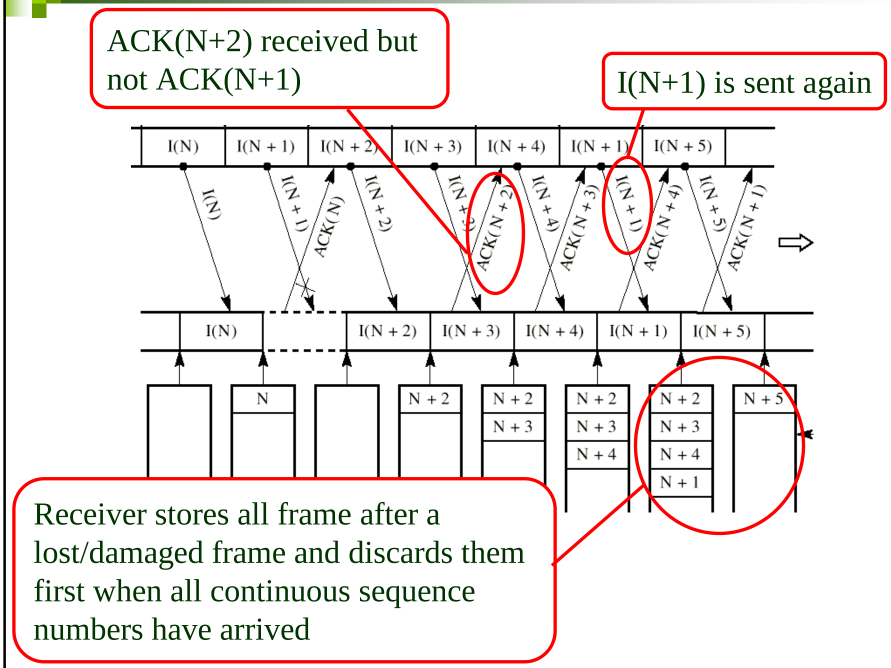
- Mechanism
  1. Damaged or lost frame
     - Receiver sends NACK
     - Discards this frame and all the following frames until the erroneous frame is received correctly
     - Transmitter receives NACK and must retransmit the frame in question and all subsequent frames it already has sent out in the meantime
  2. Damaged or lost RR (positive ACK)
     - Subsequent RR acknowledges everything up to this point
- Maximum allowed window size W = $2^k - 1$

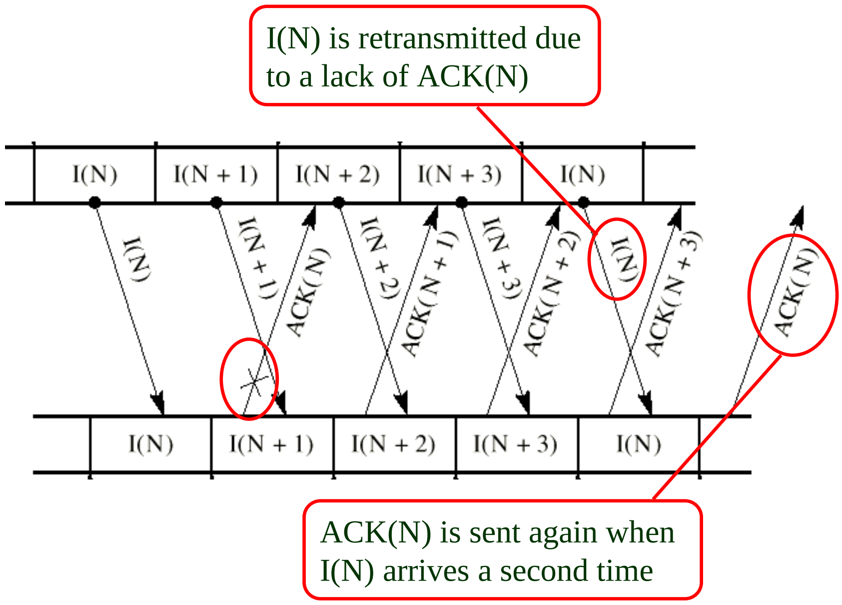

NACK(N+1) triggers a retransmission from frame N+1 and onwards
I(N)
I(N + 1)
I(N + 2)
I(N + 3)
I(N + 4)
I(N + 1)
I(N + 2)
ACK(N)
NAK(N + 1)
I(N)
I(N + 2)
I(N + 3)
I(N + 4)
I(N + 1)
N
N + 2
N + 3
N + 4
N + 1
Receiver sees that frame N+1 is missing as N+2 arrives
A frame is buffered until the next frame in sequence arrives


ACK(N+2) is interpreted as ACK for I(N) and I(N+1) as well
I(N)
I(N + 1)
I(N + 2)
I(N + 3)
I(N + 4)
ACK(N)
ACK(N + 1)
ACK(N + 2)
ACK(N + 3)
I(N)
I(N + 1)
I(N + 2)
I(N + 3)
I(N + 4)

Selective-repeat ARQ

- Retransmission only after NACK or timeout
- Maximum allowed window size W = 2^{k-1}





Comparison Continuous RQ

- Go-back-*N*
  - Needs less buffer space at the receiver
  - When an error occurs, several (potentially non-erroneous) frames must be retransmitted
  - Most frequently used retransmission scheme
- Selective-repeat
  - Makes better use of the bandwidth especially when errors occur frequently
  - The transmitter need more complicated logic to handle ”out of sequence” transmissions of frames
  - Needs more buffer space

Flow control performance comparison

U = link utilization (= time spent on transmitting actual data compared transmitting overhead)

n_r = mean number of transmission needed to get one frame correctly over the link

$$U = \frac{T_{transData}}{n_r\left(T_{transData} + 2T_{propData}\right)}$$

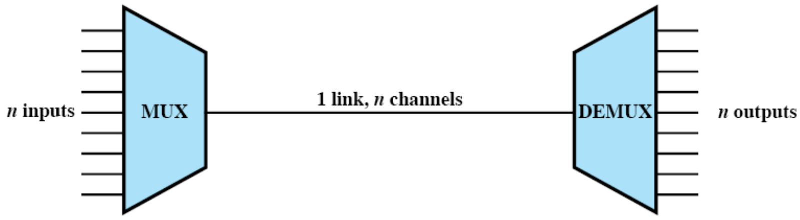
BER (Bit Error Rate) etc.

$$P_f = 1 - (1 - P)^n \qquad n_r = \frac{1}{1 - P_f}$$

- P_f = probability for an erroneous frame
- P = probability for a bit error (BER)
- n = number fo bits in a frame

Multiplexing techniques

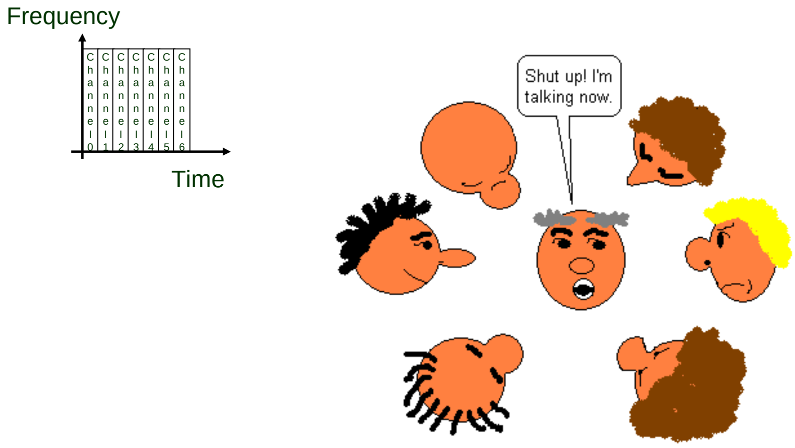
- Several transmission sources share a common medium
  - Several individual stations
  - Several applications within one station
  - Several threads within an application



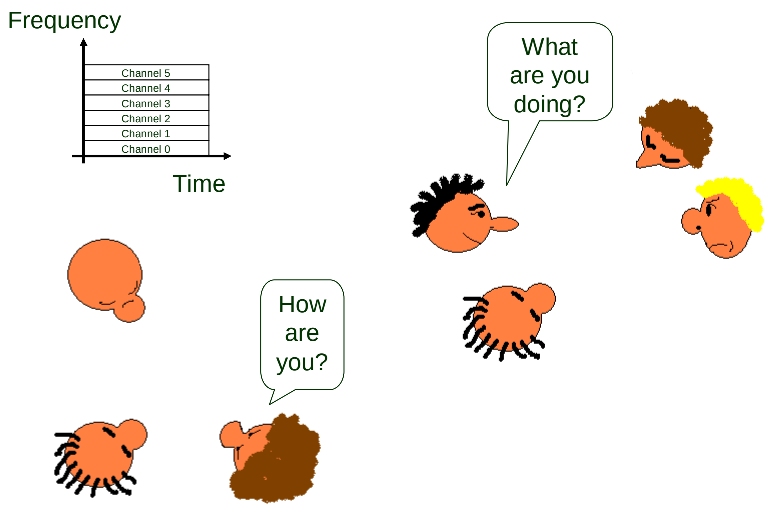
Multiplexing techniques

- 3 techniques to share the medium
  - Time Division Multiplexing (TDM)
    - Also called Time Division Multiple Access (TDMA)
  - Frequency Division Multiplexing (FDM)
    - Also called Frequency Division Multiple Access (FDMA)
  - Code Division Multiplexing (CDM)
    - Also called Code Division Multiple Access (CDMA)

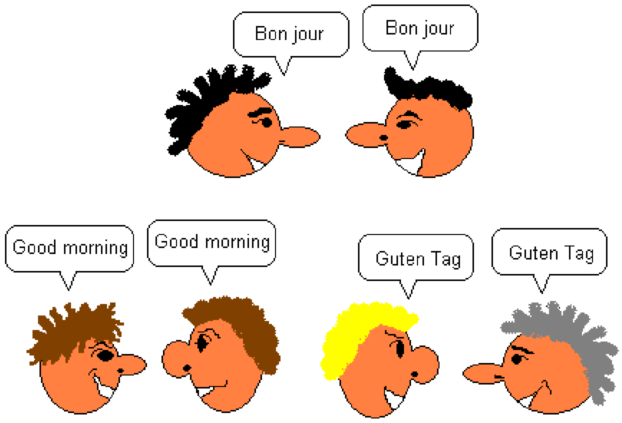
Time Division Multiple Access



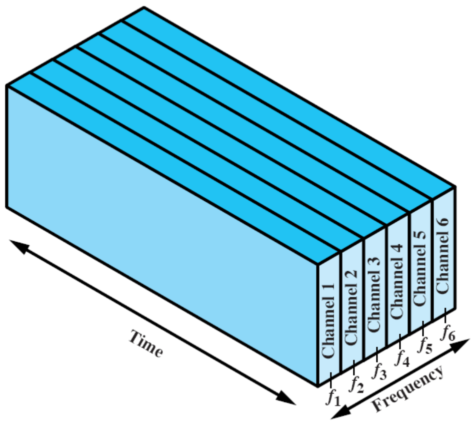
Frequency Division Multiple Access



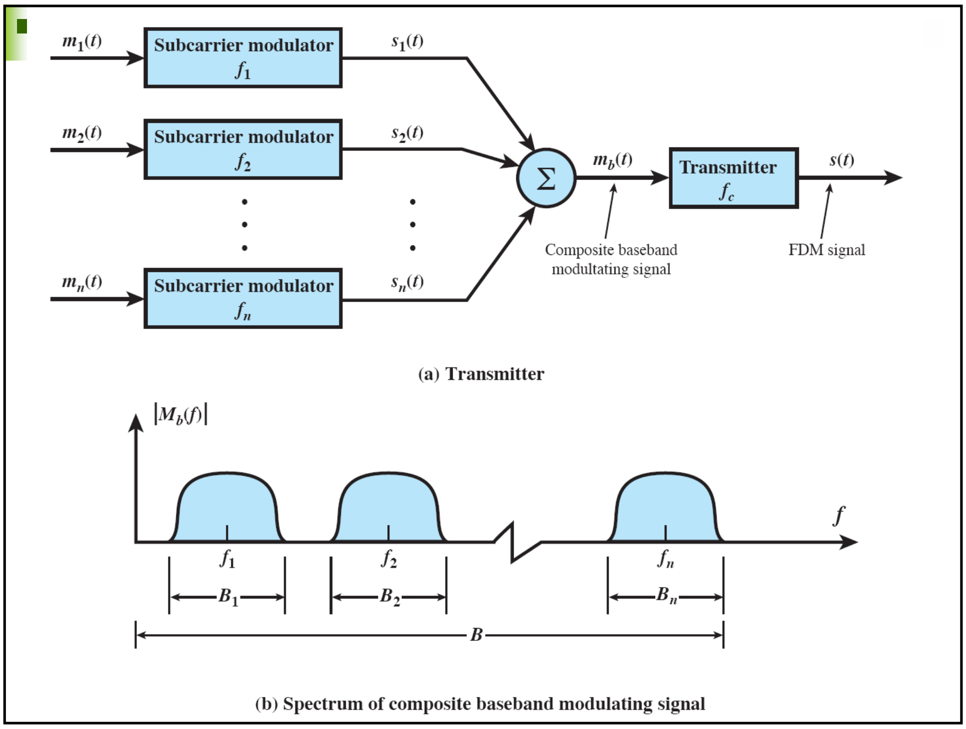
Code Division Multiple Access

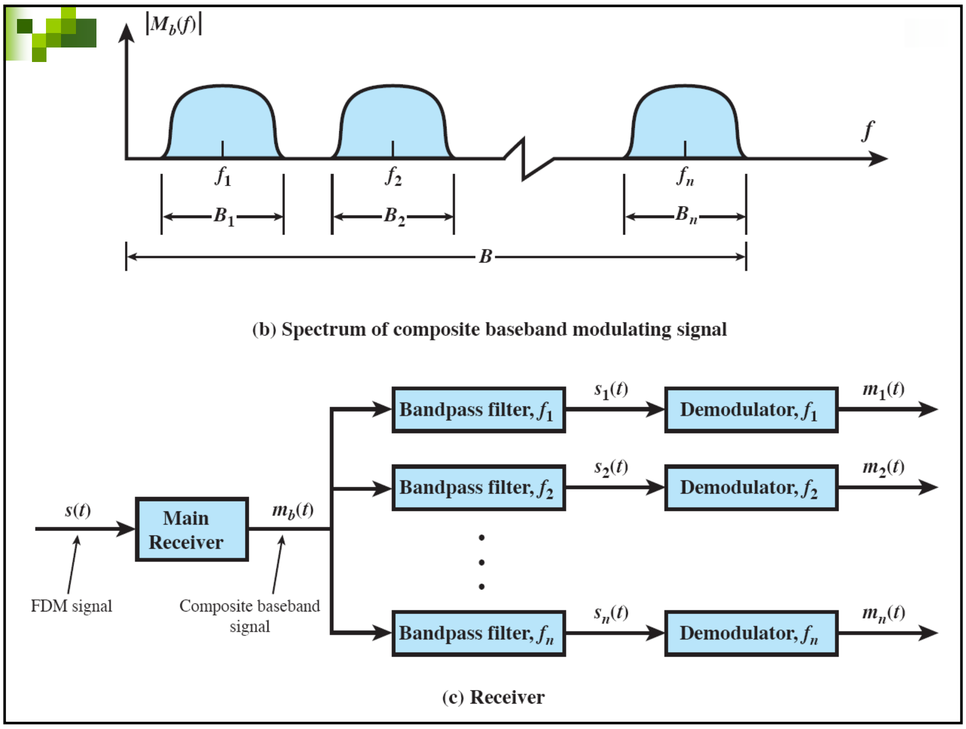


Frequency Division Multiplexing



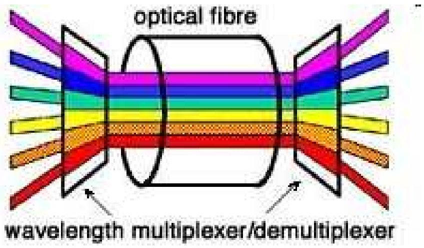
- All flows send simultaneously but on different frequencies
- The frequency channels are separated to not cause interference


m_1(t)
Subcarrier modulator f_1
s_1(t)
m_2(t)
Subcarrier modulator f_2
s_2(t)
m_n(t)
Subcarrier modulator f_n
s_n(t)
Σ
m_b(t)
Transmitter f_c
s(t)
Composite baseband modultating signal
FDM signal
(a) Transmitter
|M_b(f)|
f
f_1
f_2
f_n
B_1
B_2
B_n
B
(b) Spectrum of composite baseband modulating signal

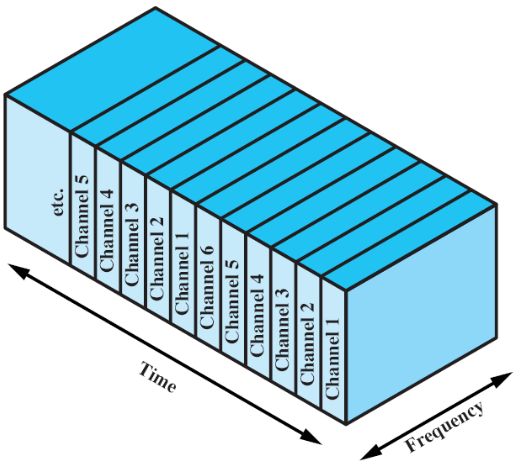

|M_b(f)|
f
f_1
f_2
f_n
B_1
B_2
B_n
B
(b) Spectrum of composite baseband modulating signal
s(t)
Main Receiver
m_b(t)
FDM signal
Composite baseband signal
Bandpass filter, f_1
s_1(t)
Demodulator, f_1
m_1(t)
Bandpass filter, f_2
s_2(t)
Demodulator, f_2
m_2(t)
Bandpass filter, f_n
s_n(t)
Demodulator, f_n
m_n(t)
(c) Receiver

Wavelength Division Multiplexing (WDM)

- Special case of FDM
- Used in optical communication
- Each wavelength is able to carry data
- Several wavelengths can travel through an optical fiber simultaneously without interference



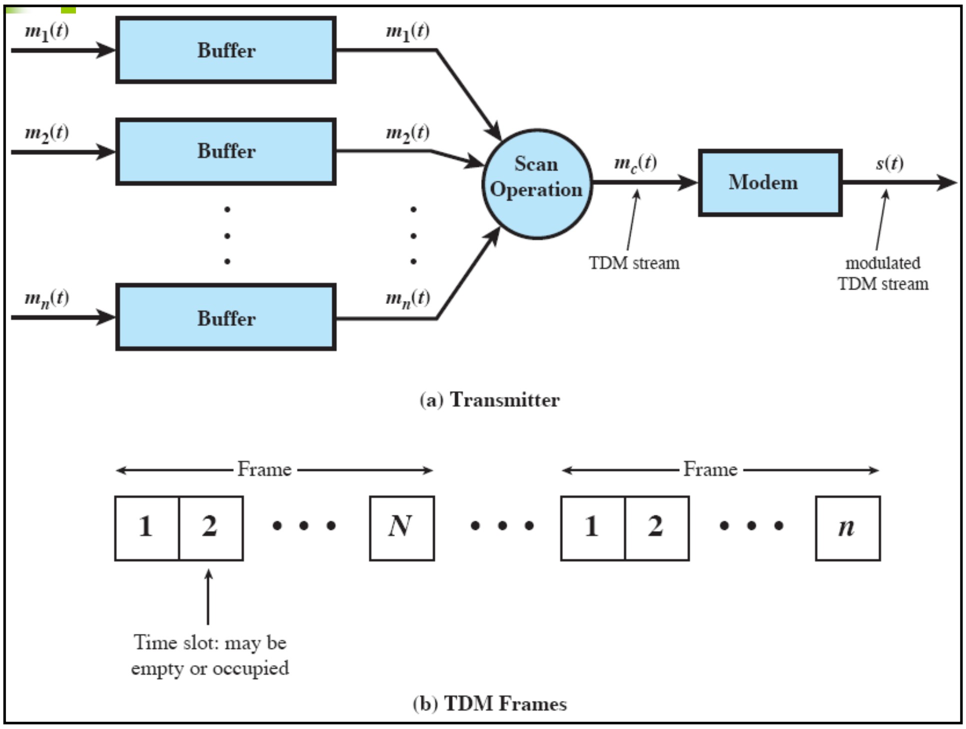
Time Division Multiplexing

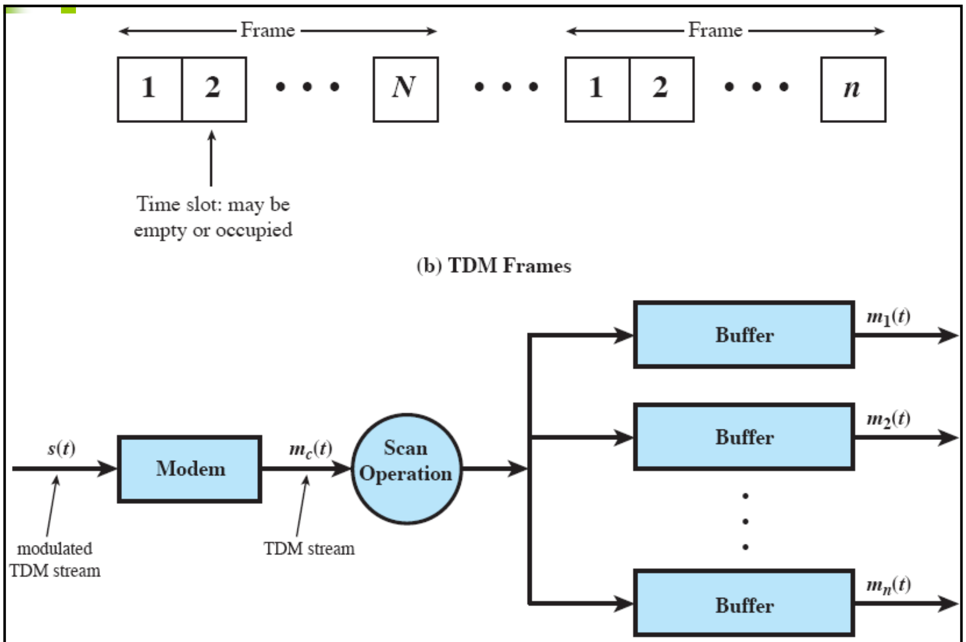


- Same frequency (or carrier in general)
- Each data flow gets its own time slot
- The assigned time slot is cyclically repeated

TDM

- TDM might be used when achievable data rate of the medium exceeds the data rate of the signals that travel over the medium
- Signals from different sources are interleaved and sent in their individual time slot without without competition
- Problem
  - If time slots are assigned to a data source and it currently has no data to send, this time slot remains empty and resources are wasted
  - Various data sources need to be synchronized





Key terms

- Flow control
- Error control
- Retransmission
- Buffer
- Acknowledgement (ACK)
- Negative Acknowledgement (NACK)
- Link utilization
- Transmission time vs. propagation time
- Sliding window flow control
- Window size
- Sequence number
- Receive-Ready acknowledgement (RR)
- Receive-Not-Ready acknowledgement (RNR)
- Piggybacked acknowledgements
- Automatic Repeat Request (ARQ) error control
- Idle RQ vs. Continuous RQ
- Stop-and wait ARQ
- Go-back-N ARQ
- Selective-repeat ARQ
- Multiplexing
- Multiplexer, demultiplexer
- Time Division Multiple Access (TDMA)
- Frequency Division Multiple Access (FDMA)
- Wavelength Division Multiple Access (WDMA)
- Code Division Multiple Access