Computer Communication Networks

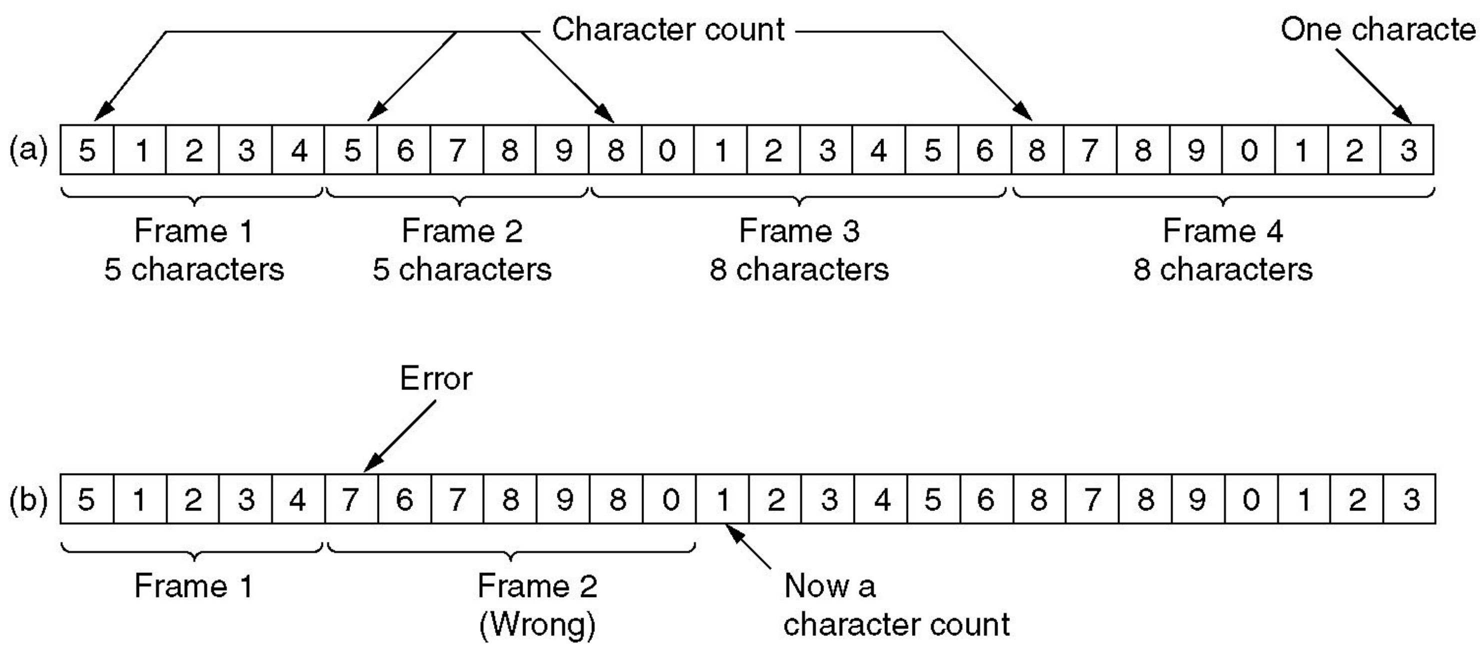
Link Layer

Link layer

- Service provided by physical layer
  - bit delivery
    - hertz, baud, symbol-per-second, bit-per-second
- Service provided to network layer
  - frame delivery
  - error control (using checksum)
  - flow control
  - medium access (with shared medium)

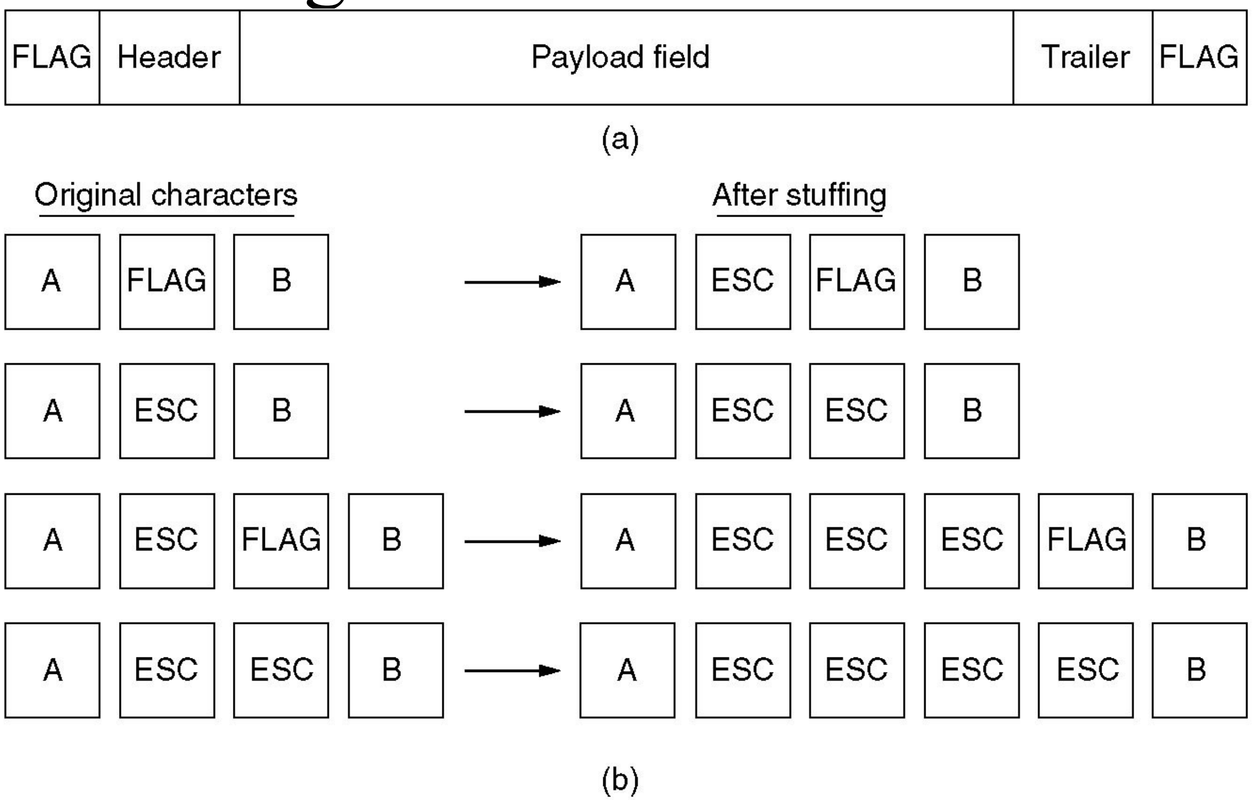
Byte-oriented framing

- Character count
  - count error, and error propagation



Byte-oriented framing: more

- Byte stuffing



Bit-oriented framing

- Flag: 01111110
  - data transparency: bit stuffing
    - sender: insert a 0 after 5 1's
    - receiver: remove a 0 after 5 1's

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Error Control

- How to ensure the frames are delivered to the other end in order
    - Acknowledgement
- How about disappeared frame?
    - Timeout
- How to let the receiver know whether the received frames are correct or not?
    - Check bits

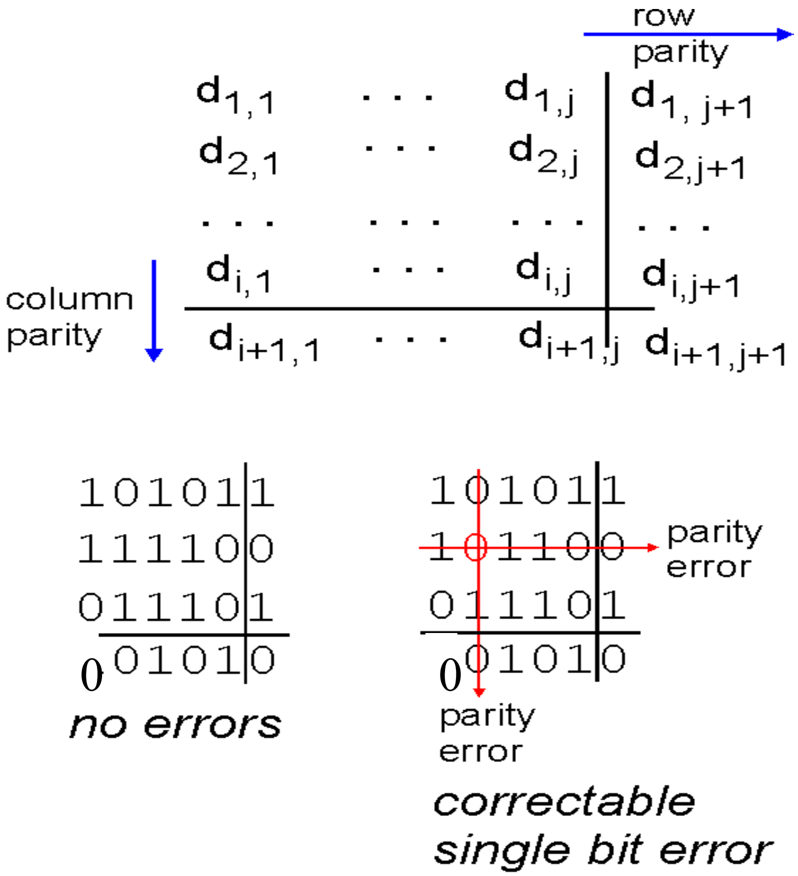
Error control

m bits data + r check bits => n bits codeword

- Hamming distance of codeword *a* and *b*
  - number of *pairwisely* different bits
    - number of bit flips needed to turn *a* to *b*
- Hamming distance of codeword set $\{a_i\}$
  - minimal distance btw a_i and a_j, when i != j
- A codeword set of Hamming distance *d*
  - detect up to *d*-1 bit error
  - correct up to floor((*d-1)*/2) bit error

Error correcting

- 2-d parity: Hamming distance 4, correct 1-bit error



Hamming code

- Minimum number of check bits for correcting 1-bit error, given m-bit data?

$(n+1)2^m \le 2^n$, i.e., $m+r+1 \le 2^r$

- Hamming code:

Most efficient code

to correct 1-bit error

Char.	ASCII	Check bits
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	01111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	10101011111
d	1100100	11111001100
e	1100101	00111000101

Order of bit transmission

Hamming Code

- Label the $m+r$ bits position from 1 to $m+r$
- The r check bit positions with indices that are powers of 2: indices 1, 2, 4, 8, ...
- The m data bits go in the other positions
- Choose values for the check bit j: XOR of all k-th bits for which (j AND k) = j
- Sum up the location of check bits in errors to identify the error bit

Hamming code

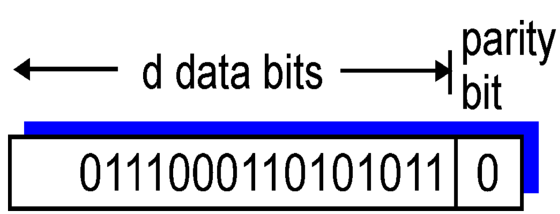
- How to deal with bursty errors?

  Arrange data in an interleaving manner.

  K column => correct bursty of k bits

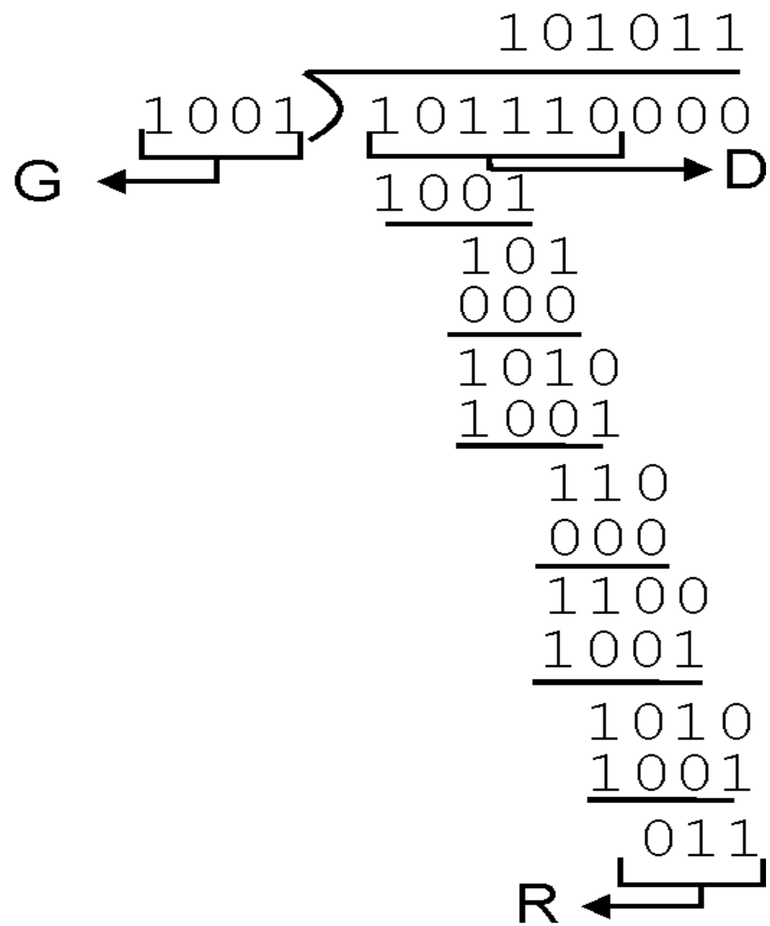
Error detecting

- Retransmitting the packet may be more efficient than adding sufficient check-bits to correct single bit. E.g.,
  - 1000-bit data needs 10 check bits to correct single bit error, leads to utilization of 1000/1010;
  - Add one bit to detect error, assuming BER 10^-6, utilization (100000)/(100100+1001) ~1000/1002
- Parity



CRC

- no carry
- $2^r D+R$ divisible by G
- detect bit error
  not divisible by G
- encoder/decoder: using
  shift-register



CRC

Message polynomial:
$D(x) = d_{m-1}x^{m-1} + d_{m-2}x^{m-2} + ... + d_1 x + d_0$, **where** $d_j \in \{0, 1\}$
Generator polynomial (with degree r, $r + 1$ **bits)**
$G(x) = x^r + g_{r-1}x^{r-1} + ... + g_1 x + 1$, **where** $g_i \in \{0, 1\}$
Codeword polynomial:
$T(x) = x^r D(x) + R(x)$
($R(x)$ **is the remainder of** $x^r D(x)/G(x)$ **modulo 2)**
We have $T(x) = G(x)Q(x)$

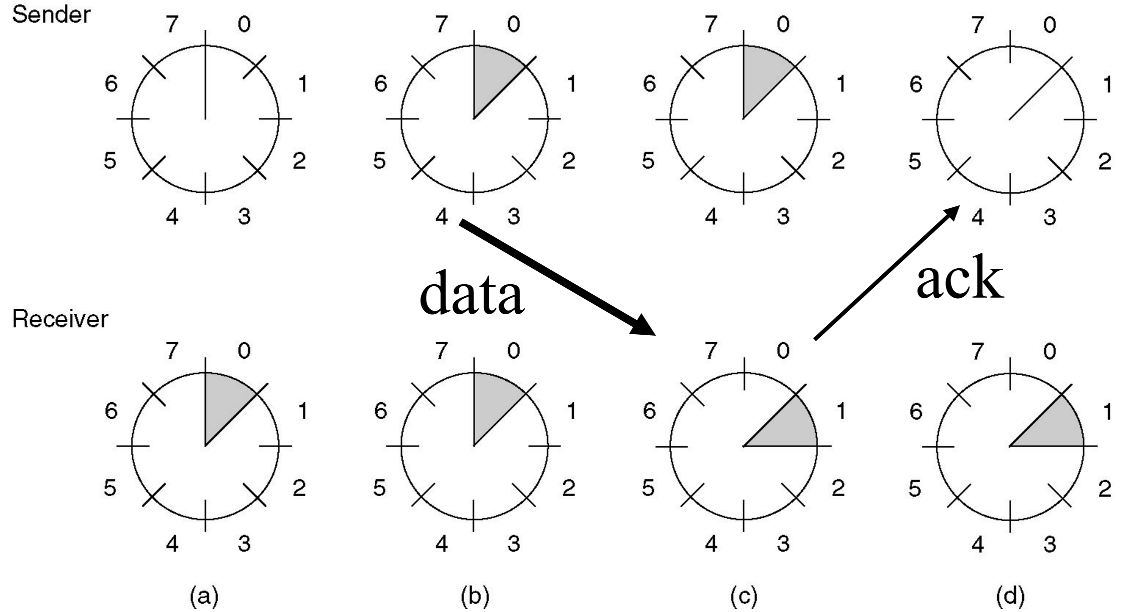
Example:
Sender side: 101110: $D(x) = x^5 + x^3 + X^2 + x$
1001: $G(x) = x^3 + 1$
We have $R(x) = x + 1$
and $T(x) = x^8 + x^6 + x^5 + x^4 + x + 1 = G(x)Q(x)$

Receiver side: $\hat{T}(x) = T(x) + E(x)$,
where $E(x)$ due to noise, interference, etc.
Divide $\hat{T}(x)$ by $G(x)$, we have

- If the remainder is non-zero, then $E(x) \neq 0$ i.e., transmission error is detected;
- if the remainder is zero. Then
  - $E(x) = 0$, no error;
  - $E(x) = 0$, but $G(x)|E(x)$, undetected error

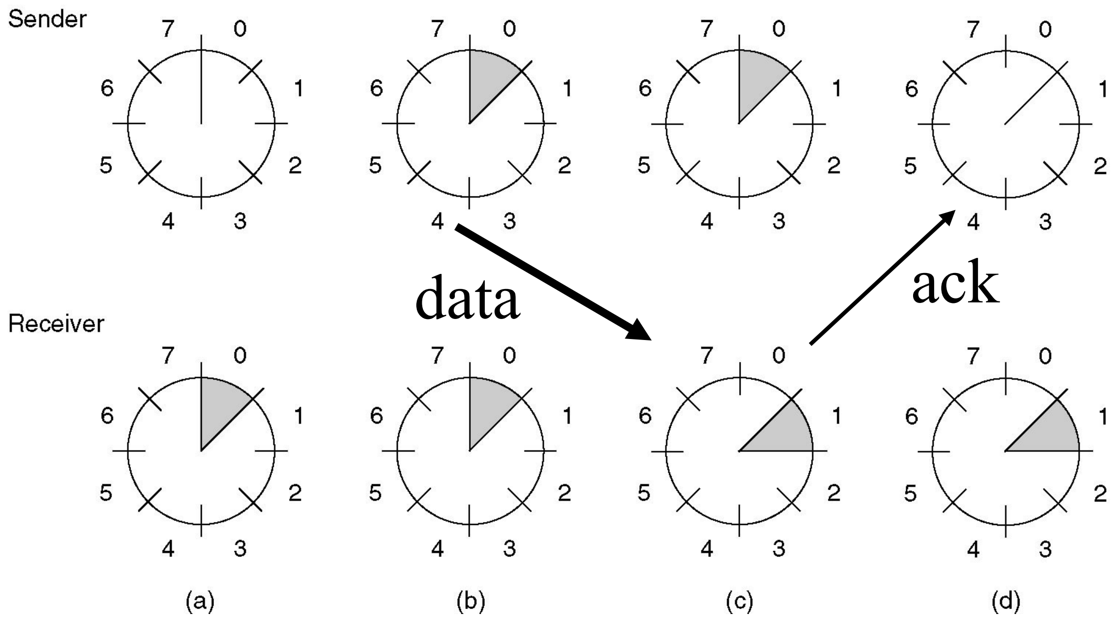
Flow Control

- Sliding window
  - e.g., window size = 1, sequence space = 8
  - maximal window size <= 1/2 sequence space (order)



Flow Control

- Sliding window
  - e.g., window size = 1, sequence space = 8
  - maximal window size <= 1/2 sequence space (order)



Error recovery

- Positive acknowledgment
  - cumulative acknowledgment
    - acknowledge packet x: acknowledge packets 1..x
  - selective acknowledgment
    - acknowledge packet x: packet x is received OK
- Negative acknowledgment
  - report: x is corrupted or *missing*

Link layer retransmissions

- Automatic Repeat reQuest (ARQ)
  - Based on timers and acknowledgements
- Stop and wait
  - One bit sequence and ACK
- Go-back-N
  - Window size N, cumulative ACK
  - If one packet (seq. n) is lost, all packets from n will be retransmitted
  - Receiver buffer size: one frame

Link layer retransmissions

- Selective Repeat (SR) ARQ
  - Window size N
  - Selective ACK
  - Only retransmit the one being corrupted
  - Receiver buffer size: N frames

High-level data link control

- HDLC

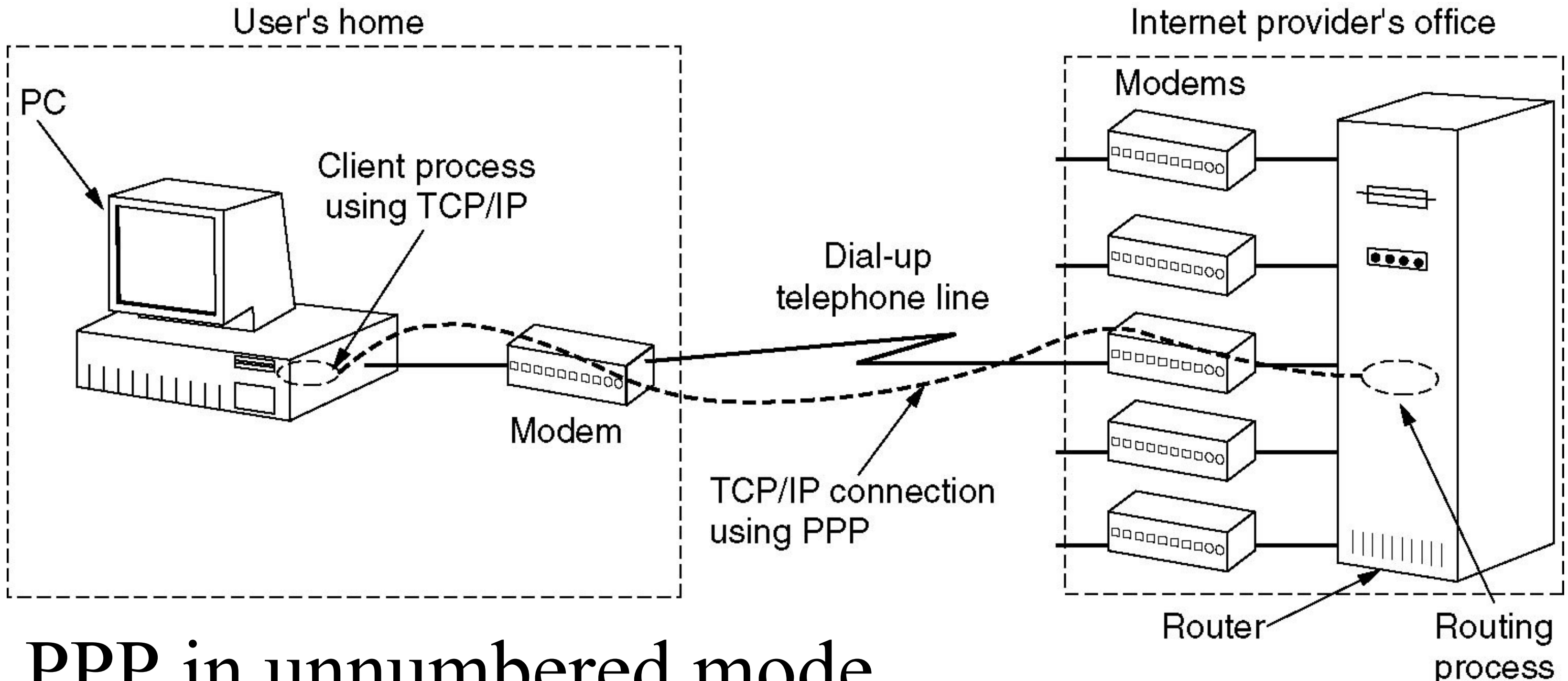
Bits	8	8	8	≥ 0	16	8
	0 1 1 1 1 1 1 0	Address	Control	Data	Checksum CRC	0 1 1 1 1 1 1 0

- Info
- Supervisory
- Unnumbered

Bits	1		3	1	3
(a)	0	Seq		P/F	Next
(b)	1	0	Type	P/F	Next
(c)	1	1	Type	P/F	Modifier

Point-to-point protocol

PPP, PPPoE



PPP in unnumbered mode

Bytes	1	1	1	1 or 2	Variable	2 or 4	1
	Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum CRC	Flag 01111110

Summary

- Link layer
  - framing
  - error control
    - error correcting, error detecting, error recovery
  - flow control
    - sliding window
  - HDLC, PPP

Next

- Medium access control