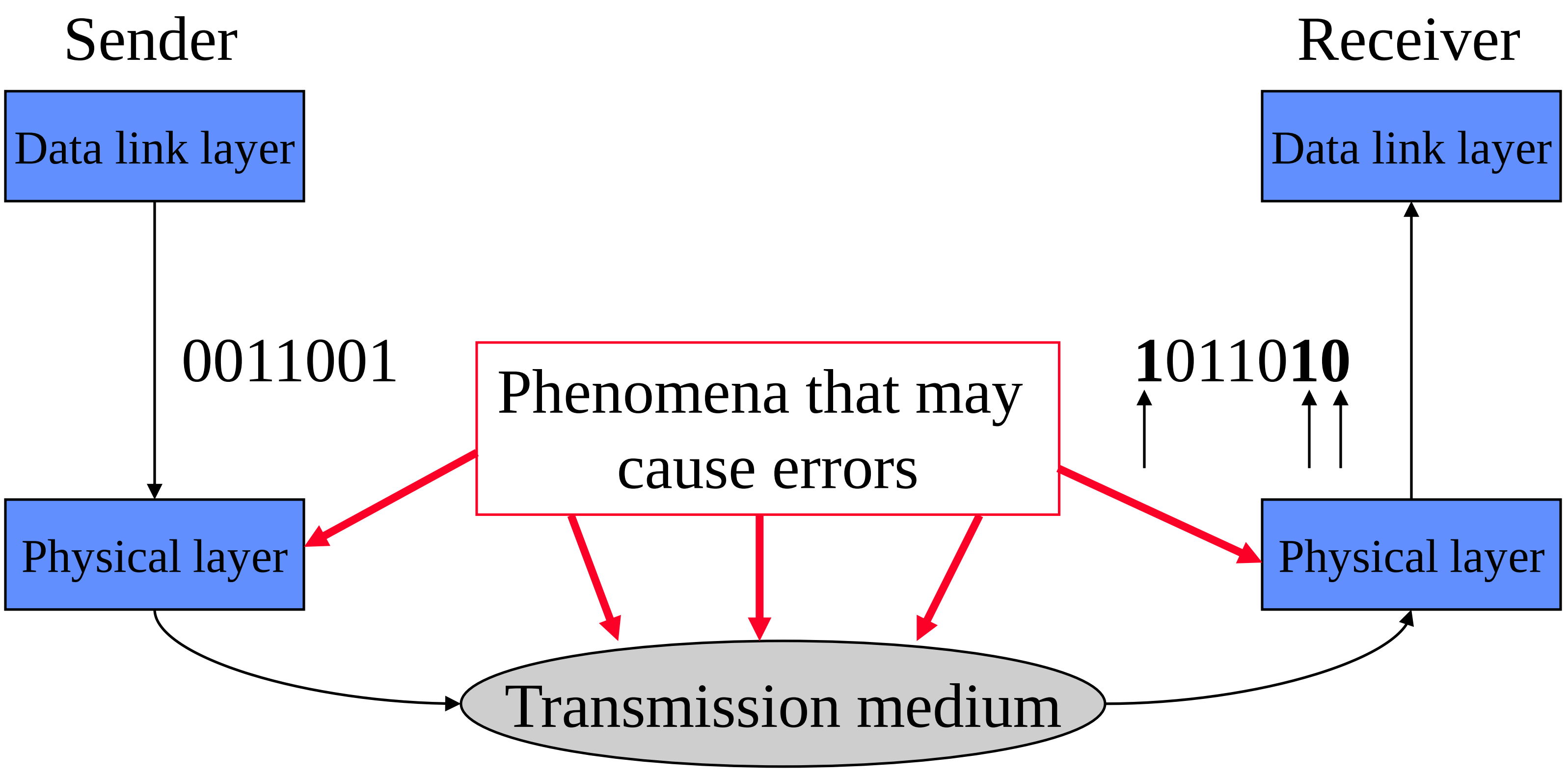
Telecommunications Networking

Errors


Sender
Receiver
Data link layer
Data link layer
0011001
Phenomena that may cause errors
1011010
Physical layer
Physical layer
Transmission medium

Error Control

Three ways to deal with errors after detection:

Retransmission — ARQ/TCP

- infrequent errors
- when time permits

Forward Error Correction — real-time services

- frequent errors
- when time does not permit retransmissions

Discard (!) — UDP

- when strict reliability is not required/too expensive

Principle of Error Detection

- message M: k bits
- an operation O on M gives result R (n-k bits)
- sender transmits codeword C (n bits) consisting of M and R: $C = (M,R)$
- receiver checks received codeword $C^* = (M^*, R^*)$ by operation O on M^* which gives R^{**}
- if ($R^* = R^{**}$) then

  receiver assumes that $M^* = M$ and $R^* = R$:

  error-free transmission (not sure!)

  else

  transmission errors (sure)

Delft University of Technology

Fast error detection

- The operation O should be fast to compute:
  - The result R should contain only a small number of bits
  - But, the check should give sufficient guarantee that, if $R^*=R$, then it is very likely that $M^*=M$

Error Detection: Single Parity Check

- Operation O: addition k message bits mod 2
- Sender:
  - append R = sum of message bits mod 2
  - e.g. [0110100] becomes [0110100 | 1]
  - transmit
- Receiver (single parity check):
  - compute sum R^* of received codeword C^* bits mod 2
  - if $R^* = 0$,
    no errors
    else error(s)

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

Single bit error is detected

Horizontal & Vertical Parity Checks

1	0	0	1	1	0	0	1
0	1	0	1	0	0	1	1
1	0	0	1	1	1	0	0
0	1	0	0	1	0	0	0
1	0	0	1	0	0	0	0
1	0	0	0	1	1	1	0

Horizontal checks

Vertical checks

Parity Check Codes

or “Linear Codes”

s_1	s_2	s_3	c_1	c_2	c_3	c_4
0	0	0	0	0	0	0
1	0	0	1	1	1	0
0	1	0	0	1	1	1
0	0	1	1	1	0	1
1	1	0	1	0	0	1
1	0	1	0	0	1	1
0	1	1	1	0	1	0
1	1	1	0	1	0	0

$c_1=s_1+s_3$

$c_2=s_1+s_2+s_3$

$c_3=s_1+s_2$

$c_4=s_2+s_3$

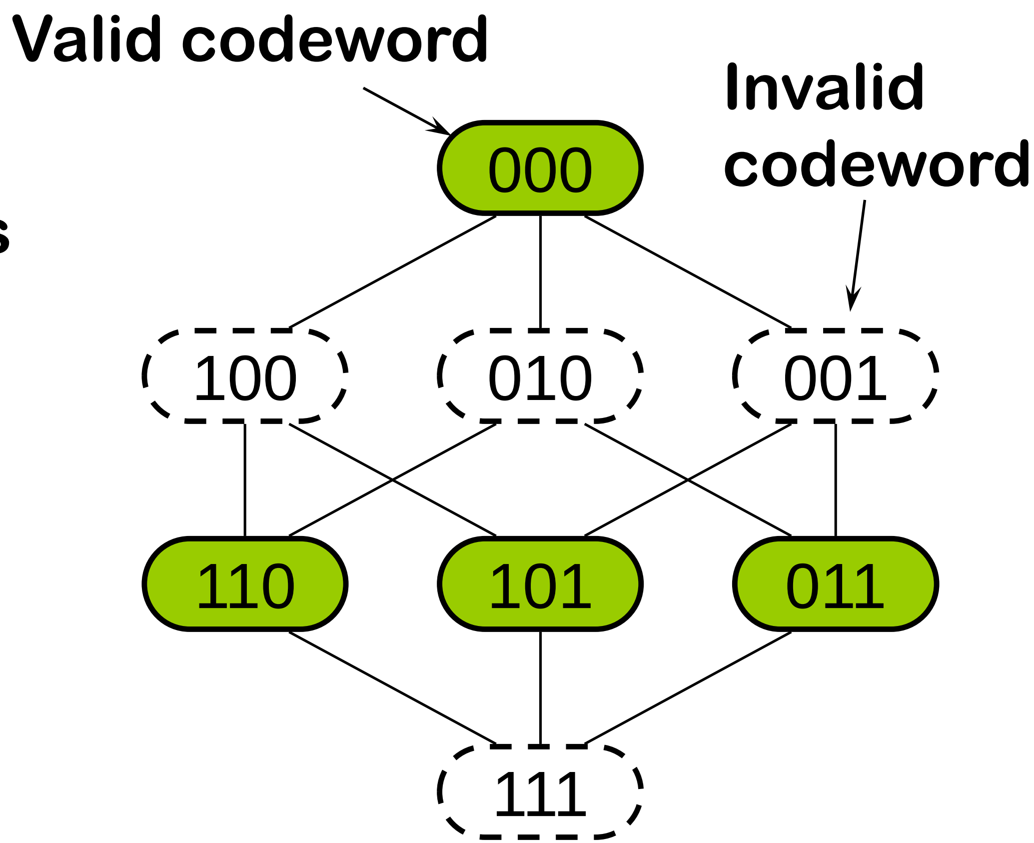
$d=4$

Hamming distance

Parity bits

A	0	0	0
B	0	1	1
C	1	0	1
D	1	1	0

Codeword

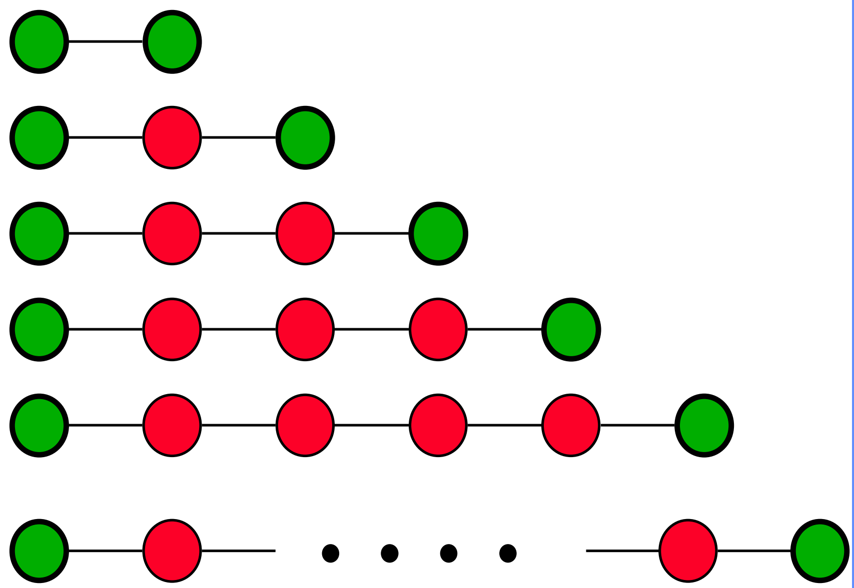


Hamming distance = 2

Error detection and correction

Minimum distance for error-detecting and error-correcting codes

🟢 valid codeword
🔴 invalid codeword



Hamming distance	error detection	error correction
1	0	0
2	1	0
3	2	1
4	3	1
5	4	2
d	d - 1	$\left\lfloor \frac{d-1}{2} \right\rfloor$

To detect d errors : distance $\geq d + 1$

To correct d errors: distance $\geq 2d + 1$

Modulo 2 Arithmetic

Modulo 2 arithmetic… **…without carry**

[same as "*exclusive or*" *(XOR)*] [digit by digit]

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

Examples:

$$\begin{array}{r} 10001011 \\ \oplus \; 10100010 \\ \hline 00101001 \end{array} \quad \text{and} \quad \begin{array}{r} 10001011 \\ \oplus \; 10001011 \\ \hline 00000000 \end{array}$$

Is not addition in base 2!
Base 2 : 10 + 11 = 101
Modulo 2 without carry: 10 + 11 = 01

Addition modulo 2 = subtraction modulo 2

Cyclic Redundancy Check (CRC)

TU Delft
Delft University of Technology

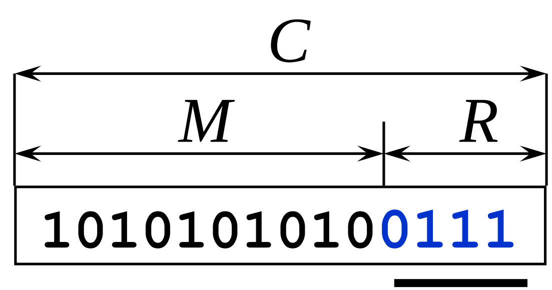
Given: *M:* 1010101010

G: 10111

Solution: $r + 1 = 5 \Rightarrow r = 4$

$M.2^r =$ 10101010100000

```
10101010100000 | 10111
10111          |----------
  10010          1001010001 = Q
  10111
    10110
    10111
        10000
        10111
          111 = R  ⇒ CRC = 0111
```

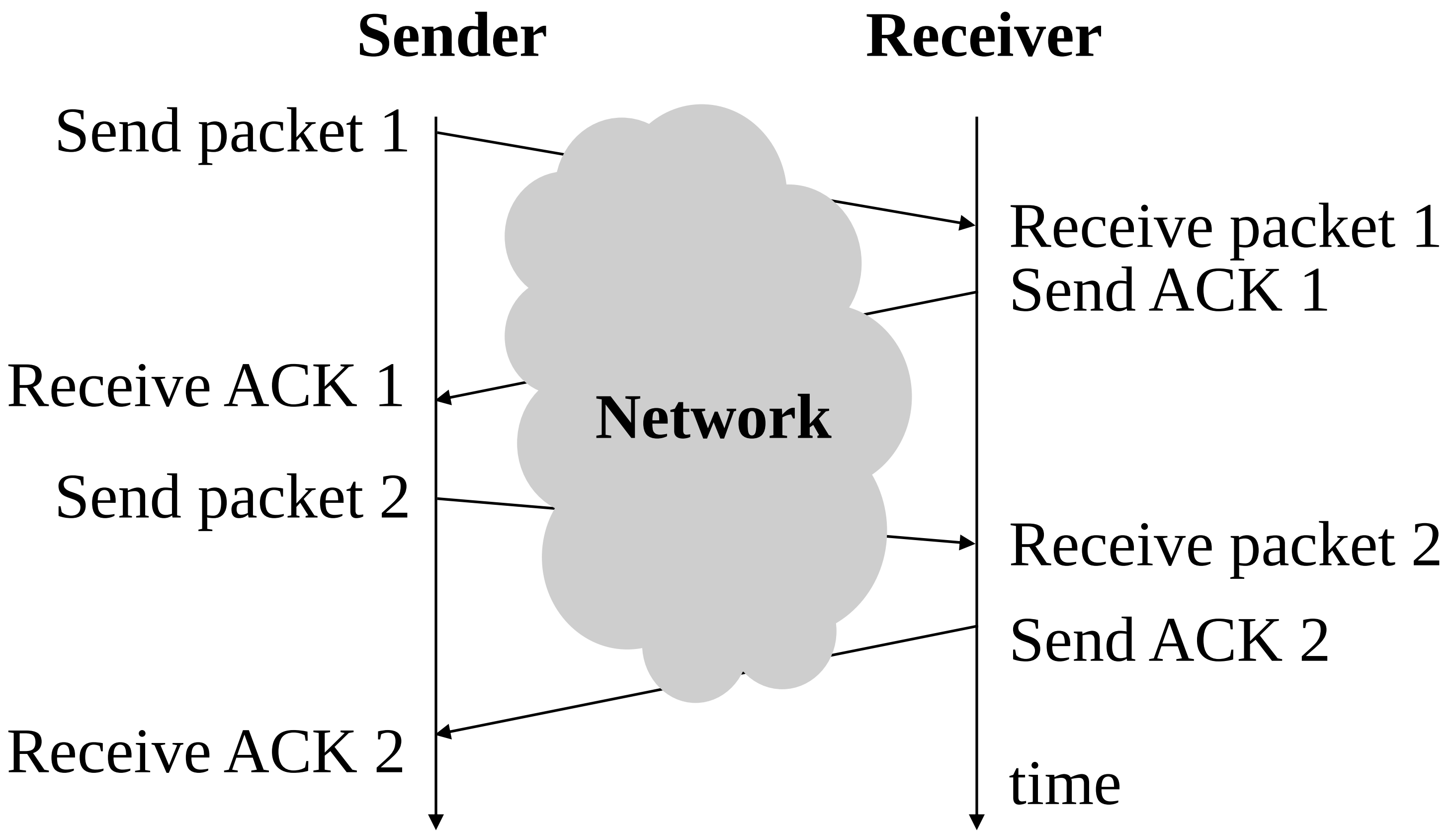


Retransmission Protocols

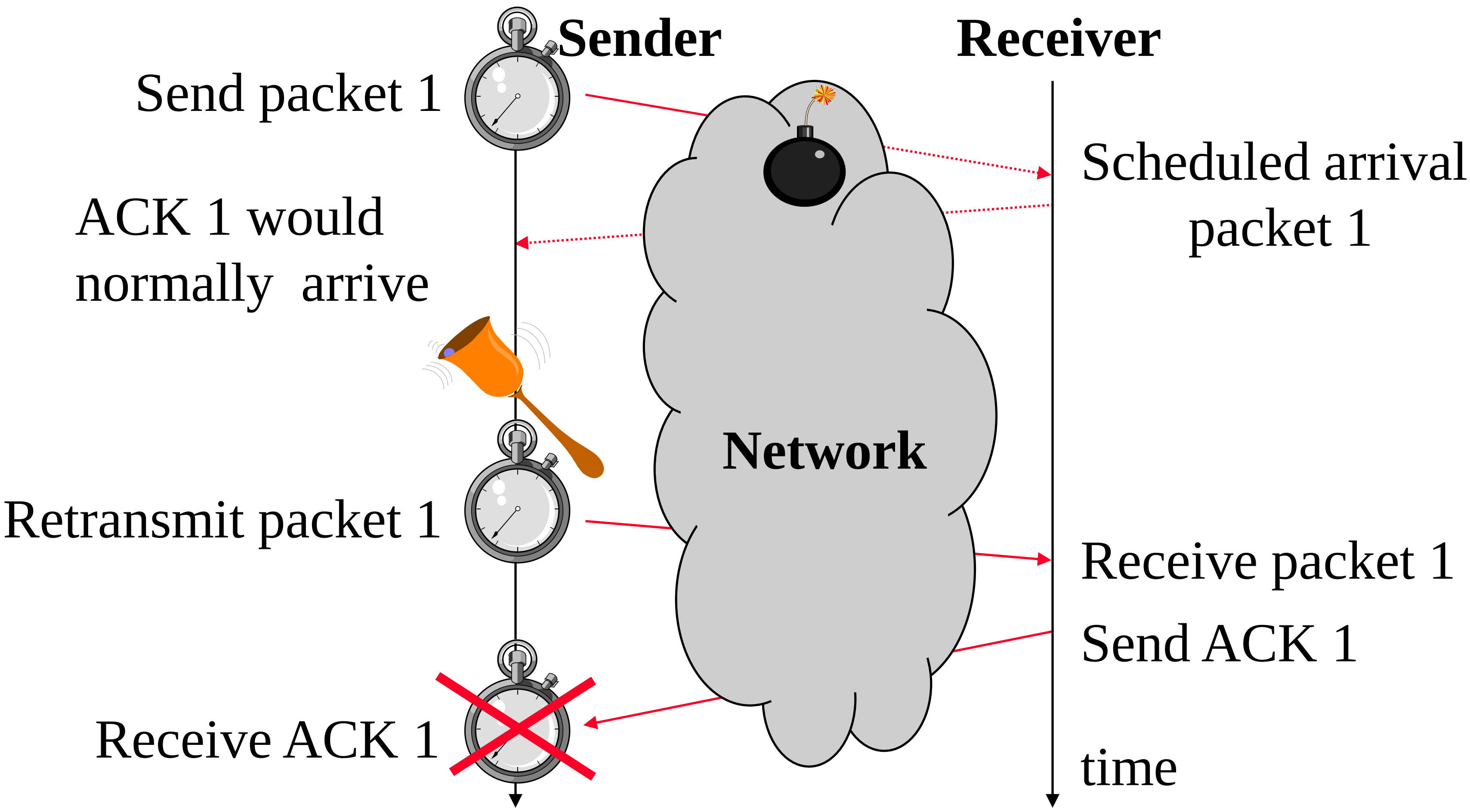
Goal: Implement reliable packet channel over an unreliable one

Approach: Use timers, acknowledgments & retransmits

Acknowledgment Scheme



Time-out and Retransmission

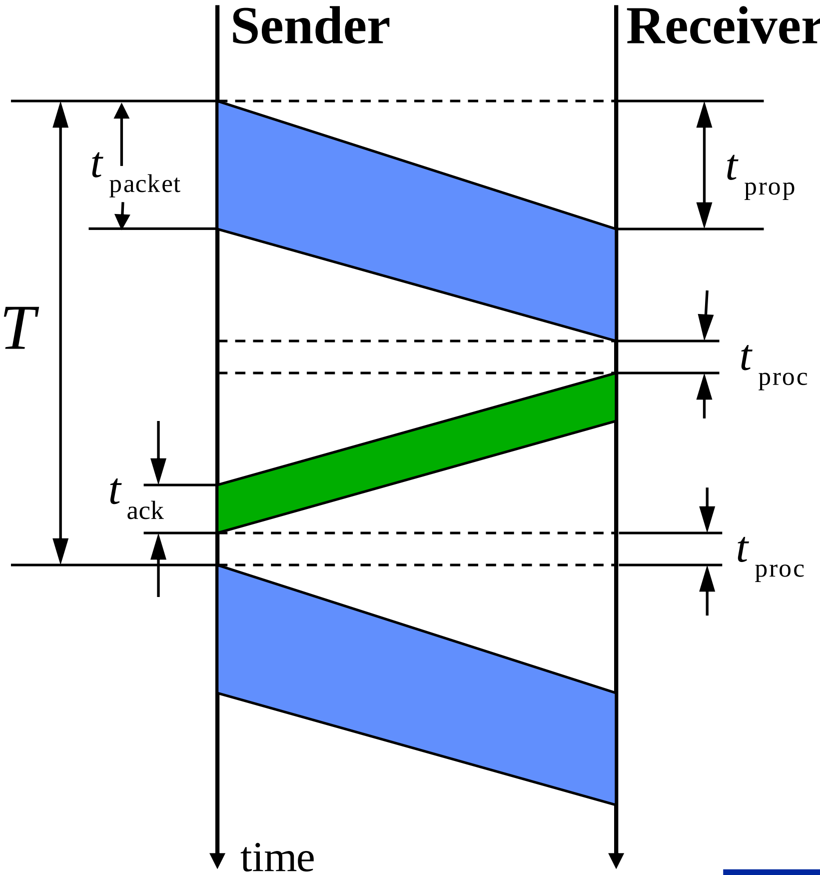


ARQ protocols

- Automatic Repeat Request (ARQ) protocols: Error detection and retransmission
  - Stop-and-Wait protocol
  - Go Back *n* protocol
  - Selective Repeat protocol

- Efficiency: $$\frac{\text{Maximum average rate (throughput)}}{\text{Link rate}}$$

Stop-and-Wait Protocol: Efficiency



No errors: $$\eta_{S\&W} = \frac{R_{S\&W}}{C}$$

Maximum effective information rate:

$$R_{S\&W} = \frac{l_{packet} - l_{header}}{T}$$

$$t_{packet} = \frac{l_{packet}\ \text{(bits)}}{C\ \text{(bits/s)}}, \quad t_{ack} = \frac{l_{ack}\ \text{(bits)}}{C\ \text{(bits/s)}}$$

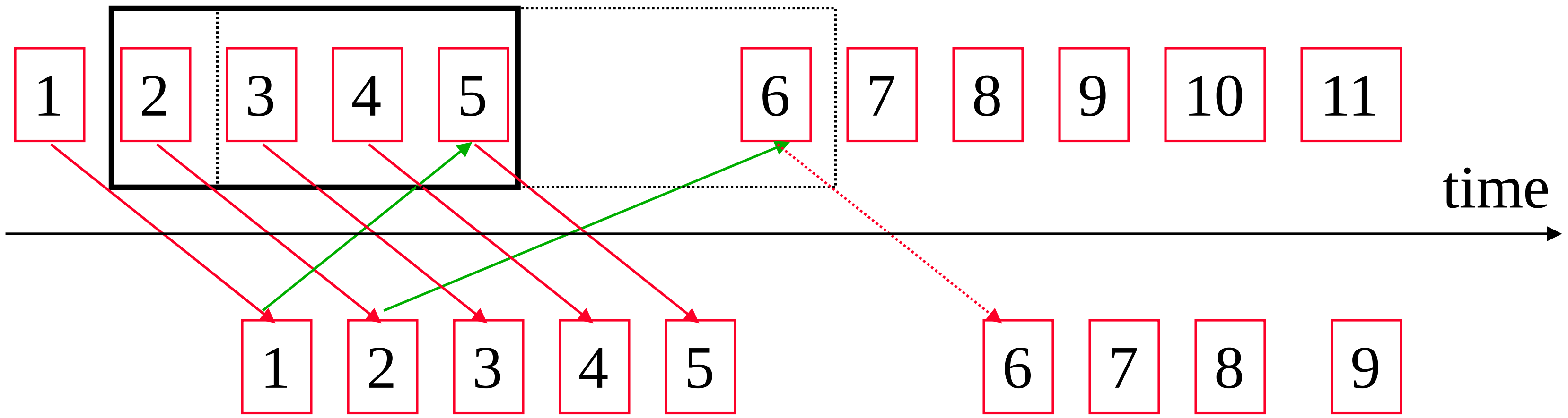
Round-Trip Time T

$$T = 2 \times t_{prop} + 2 \times t_{proc} + t_{packet} + t_{ack}$$

Combined:

$$\boxed{\eta_{S\&W} = \frac{1 - \dfrac{l_{header}}{l_{packet}}}{1 + \dfrac{l_{ack}}{l_{packet}} + \dfrac{2C\,(t_{prop} + t_{proc})}{l_{packet}}}}$$

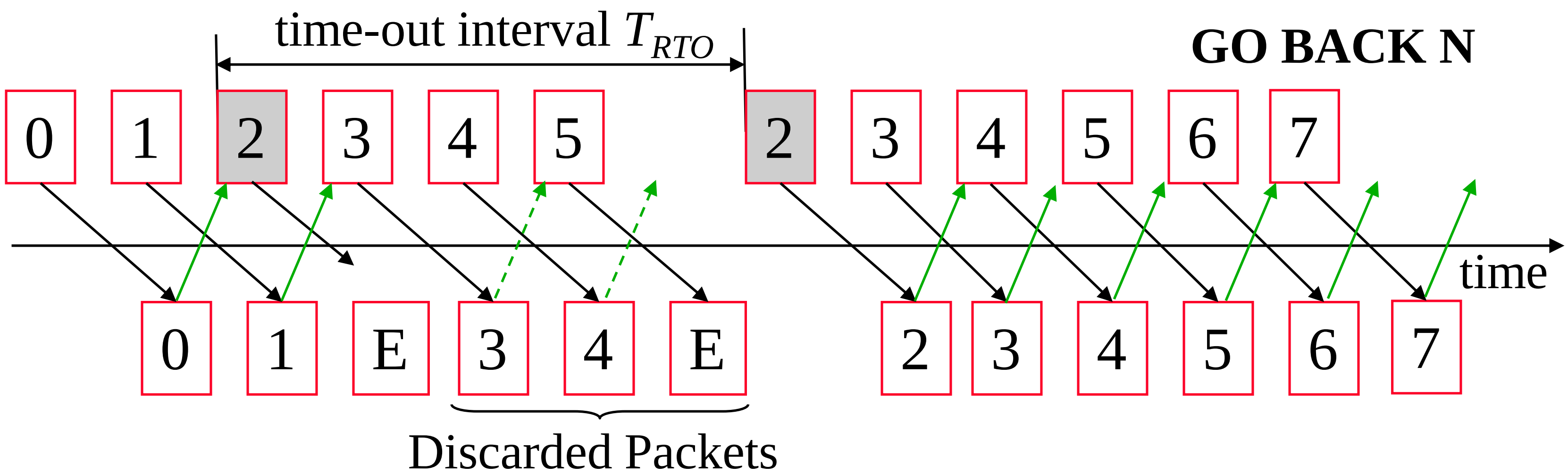
Sliding Window Size

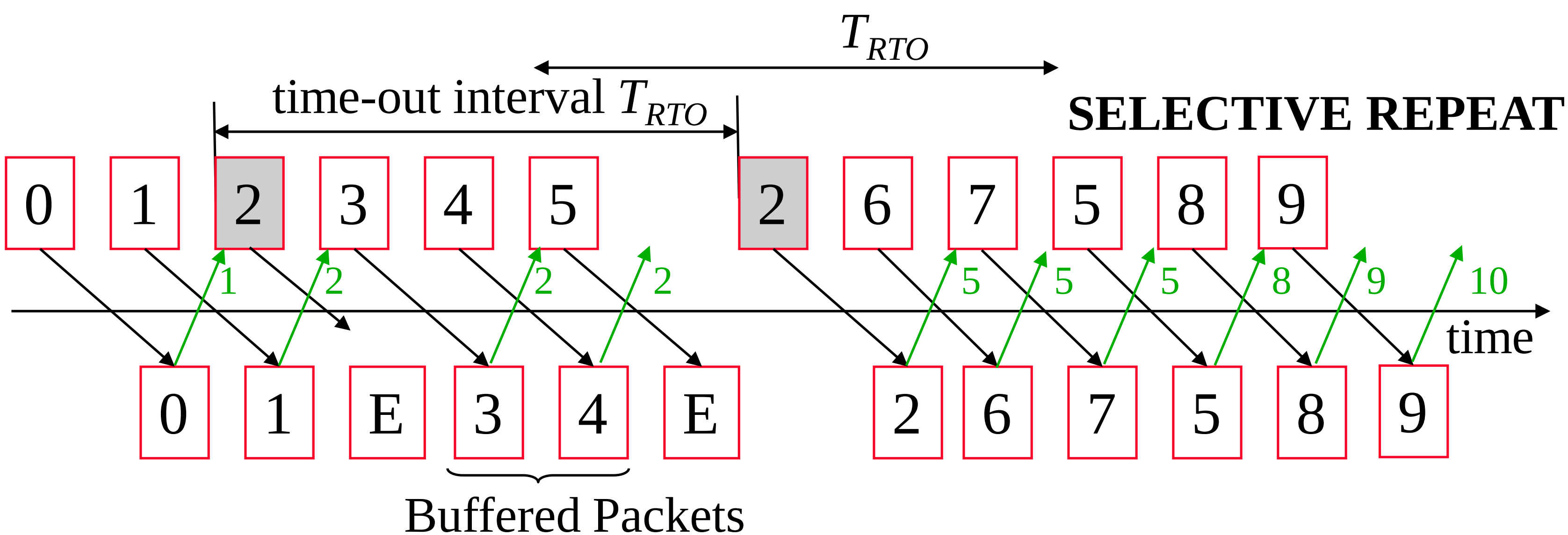


Tuning sliding window impacts number of packets in network

Flow control mechanism: optimal throughput

Sliding Window Strategies





Questions Ch. 3

- Explain how to compute a cyclic redundancy check.
- Explain the Selective Repeat ARQ protocol and the purpose of ARQ.