10/02/2020

CN

Error Detection

Ex: S → R

101 → 100

1 to 0

Error: Received message is ~~ident~~ not identical to transmitted message due to some external noise.

Errors

- Single bit error
- Burst bit error

Single bit error: only one bit is changed from 0 to 1 or 1 to 0

Burst bit:
Two or more bit are changed from 0 to 1 or 1 to 0

S: | 1 | 0 | 1 | 1 | 0 | 1 | → R: | 1 | 1 | 0 | 1 | 1 | 1 |

0 to 1

burst length

Error Handling:

- error detection
- error correction

error detection → identification of error
error correction → identification & correction of error

Error detection techniques :-
1.) Single Parity Bit.
2.) Two dimensional Parity
3.) Cyclic Redundancy Code. (CRC) check.
4.) CheckSum

Identification & correction of error
1.) Hamming code.

1.) Single Parity Bit :- (single bit error detection techniques)
└ redundant / extra bit
Adding one extra bit or appending one extra bit
used to detect single bit error.
Two types :-
1.) odd parity.
2.) even parity.

Ex.

S

msg → 1011

10000

i) odd parity :-

S → R

$1011 \rightarrow \underbrace{10110}_{odd} \xrightarrow{transmit} \underbrace{10100}_{even}$ ← error detected

2) even parity :-

S → R

$1011 \rightarrow \underbrace{10111}_{even} \xrightarrow{transmit} \underbrace{10101}_{odd}$ ← error

Two dimensional Parity Check

Sequence of bits is represented in a tabular form (rows & columns)

Original Data :- $\underbrace{1100}_{4}, \underbrace{0111}_{4} \underbrace{0101}_{4} \underbrace{1111}_{4}$

odd parity :-

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

→ 11001 01110 01010 11111 11100

even

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

11000 01111 01010 11110 00011

Hamming distance :-

No. of bits changed from one message to another message.

M_1	M_2
101	110

$d_h(M_1, M_2) = 2.$

M_1	M_2
11001	11100

$d_h(M_1, M_2) = \underline{3}$

hamming distance ~~is less tha~~ is 1 less than message size.

CRC :- Cyclic Redundant ~~Code~~ check

- Cyclic → cyclic code
- Redundant → adding some extra bits
- check → Checking at the receiving end.

generator & ~~of~~ checker.

→ Based on binary division.

CRC generator :-

generate the code word.

CRC checker :-

```
              111101
       1101 ) 100100000
              1101
              x1000
               1101
               x1010
                1101
                x1110
                 1101
                 x0110
                  0000
                  x1100
                   1101
                   x001
```

divisor (CRC generator)

~~1101~~ perform XOR operation

CRC code word → ~~11110~~

100100001

printf ("%d", a[99]);

11/02/2020 CN

256	128	64	32	16	8	4	2	1
1	0	0	1	0	0	0	0	1

256+32+1

256
33
289

8	4	2	1
1	1	0	1

(13)

On checker side

```
         11111.01
1101 ) 100100001
       11011↓
       x1000
        1101↓
        x1010
         1101↓
         x1110
          1101↓
          x0110
           0000↓
           x1101
            1101
            x000  ← accepted
```

when Remainder is 0 then code is accepted

(100100001)

~~code word + m~~ codeword = msg + CRC

CRC generator can also be expressed with algebraic polynomial form.

$x^7 + x^6 + x^4 + x^3 + 1$

degree = 7

represent bit value (coefficient)

power represent bit location.

$\Rightarrow 1\times x^7 + 1\times x^6 + 0\times x^5 + 1\times x^4 + 1\times x^3 + 0\times x^2 + 0\times x^1 + 1\times x^0$

$\rightarrow$ | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | $\Rightarrow$ 11011001

7 6 5 4 3 2 1 0

Q) A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is $x^4 + x + 1$. What is the actual bit string transmitted?

Sol^n

generator func^n $\Rightarrow x^4 + x + 1$

$\rightarrow$ ~~10011~~

$1\times x^4 + 0\times x^3 + 0\times x^2 + 1\times x^1 + 1\times x^0$

$\Rightarrow$ 10011 $\rightarrow G(x)$

```
             1100001010
10011 ) 11010110110000
        10011
        x10011
         10011
         x00001
          00000
          x00010
           00000
           x00101
            00000
            x01011
             00000
             x10110
              10011
              x01010
               00000
               x10100
                10011
                x01110
                 00000
                 x1110
```

(Actual bit)
code word $\Rightarrow$ 11010110111110

Q. A bit stream 10011101 is transmitted using CRC method. The generator polynomial is x^3+1
100111010100
(i) what is the actual bit transmitted?
(ii) Suppose the 3rd from the left is inverted during transmission. How will receiver detect this error.

Checksum :-

→ based on binary addition.

- It first divides the message into k^{th} segment each of m bit.

original msg :-

$\underbrace{10011001}_{1}, \underbrace{11100010}_{2}, \underbrace{00100100}_{3}, \underbrace{10000100}_{4}$

m = 8 bit

```
1 →        10011001
2 →        11100010
        1) 01111011          1) 11111111
                  +1                   1
sum(1+2)   01111100          1)
3 →        00100100
sum(1+2+3) 10100000
4 →        10000100
        1) 00100100
                  +1
           00100101   sum(1+2+3+4)
```

1's compl. of sum(1+2+3+4)
→ 11011010 (checksum)

code word → msg + checksum.

10011001 11100010 00100100 10000100 11011010

$\underbrace{\hspace{6cm}}_{\text{total of 40 bits code word}}$

Now at receiver side we will again repeat the process. (breaking in segments of 8 bit & adding).

sum(1,2,3,4) 0 0 1 0 0 1 0 1
5 → 1 1 0 1 1 0 1 0
sum(1,2,3,4,5) 1 1 1 1 1 1 1 1

1's of sum → 0000 0000
receiver msg is correct.

18/02/2020 English Tutorial Mod-2

Syllabus :-
On ~~friday~~-13. 18

→ Group Presentations.
→ Group Discussion.
→ Business Letters : Claim, order and adjustment.
→ Drills on listening.

12/02/2020 CN Tut

Vulnerable bits = data rate * burst ~~length~~ duration
bps (unit) sec (unit)

Q. what is the max. effect of a 2ms burst of noise on data transmitted at the following rates
(a) 1500 bps (b) 12 kbps (c) 100 kbps
(d) 100 Mbps.

Soln a) 1500kbps

Vulnerable bits = $1500 \times 2 \times 10^{-3}$
= 3 bits

(b) data rate = 12kbps = 12×10^3 bps

Vulnerable bits = $12 \times 10^3 \times 2 \times 10^{-3}$
= 24 bits.

(c) 100kbps = $100 \times 10^{3} = 10^5$ bps
vulnerable bit = $10^5 \times 2 \times 10^{-3} = 2 \times 10^2 = 200$ bits

d) 100Mbps ⇒ $100 \times 10^6 \times 2 \times 10^{-3}$
$= 200 \times 10^3$ bits

Note :- A small duration (of burst of noise) can affect so many bits if data rate is high.

Q→ What is the hamming distance for each of the following code words :- repeater.
(a) d (10000, 00000) = 1 (b) d (10101, 10000) = 2
(c) d (11111, 11111) = 0 (d) d (000, 000) = 0

Detection of error
$d_{min} = s + 1$ ~~s : ki the bit pe error hai~~ s : no. of bits corrupted.

Correction of error
$d_{min} = 2t + 1$ total no. of bit for correction

Q→ Find the minimum hamming distance for the following cases :-
(i) detection of two errors
=
ii) detection of 3 errors or correction of 2 errors
$d_{min} = 4$ (या) $d_{min} = 2 \times 2 + 1 = 5$
$\boxed{d_{min} = 5}$

$M = x^{11} + x^9 + x^6 + x^5 + x^4$
$G = x^4 + x^2 + x + 1$

① Message = 10100111 $G(x)$ = 10111

```
            10011011
10111 ) 10100111 0000
        10111
        x00111 1
         00000
         x01111 1
          00000
          x11111
           10111
           x10000 0
            10111
            x01110 0
             00000
             x11100 0
              10111
              x10110
               10111
               x0001   <— CRC bit
```

16 8 4 2 1
1 0 1 1 1

16+4+2+1
= 23

```
      11
23 ) 2673
     23
     37
     23
     143
```

2673

Code word = 101001110001

```
            10001101
10111 ) 101101110001
        10111
        x00011 1
         00000
         x00111 1
          00000
          x01111
           00000
           x11110
            10111
            x10010
             10111
             x01010
              00000
              x10101
               10111
               x0010
```

$x^7 + x^6 + 1$ → $\frac{\text{CRC-8}}{\text{ATM}}$, $\frac{\text{CRC-10}}{\text{ATM}}$, $\frac{\text{CRC-16}}{\text{HDLC}}$

$\frac{\text{CRC-32}}{\text{LAN}}$

Scanned with CamScanner

12/02/2020

CN

Error Correction

Hamming Code

Use to detect ~~single bit~~ & correct single bit error.

→ Based on parity

Hamming Code = Data bit + Parity bit

Steps to generate hamming code :-

i) Calculate no. of parity bit.

$$2^P \geq m + p + 1$$

where $m \rightarrow$ bits in data

$p \rightarrow$ parity.

Q- Data = 1101

1	1	0	1
D_3	D_2	D_1	D_0

(1) $2^P \geq m + p + 1$

$m = 4$, put $p = 0, 1, 2, 3, 4, \ldots$

→ $2^0 \geq 4 + 0 + 1 \rightarrow 1 \geq 5$ False.

$p = 1$

$2^1 \geq 4 + 1 + 1 \rightarrow 2 \geq 6$ False

$p = 2$

$2^2 \geq 4 + 2 + 1 \rightarrow 4 \geq 7$ False

$p = 3$

$2^3 \geq 4 + 3 + 1 \rightarrow 8 \geq 8$ True

So, $\underline{P=3}$

② ~~Total no. of~~ Size of code word.

→ code word = data bit + parity bit.
= 4 + 3
code word = 7 bit

③ Location of parity :-
ascending power of 2 ⇒ $2^0, 2^1, 2^2, 2^3$.

location ⇒ $2^0, 2^1, 2^2$
⇒ 1, 2, 4.

D_4	D_3	D_2	P_4	D_1	P_2	P_1
1	1	0		1		
7	6	5	4	3	2	1

④ Bit designation table.

Bit Designation	D_7	D_6	D_5	P_4	D_3	P_2	P_1
Bit location	7	6	5	4	3	2	1
Binary Value	111	110	101	100	011	010	001
Data bit (D)	1	1	0		1		
Parity bit (P)				x		x	x
				0		1	0

⑤ Calculation of Parity bit value.
Case even : P_1 → 1, 3, 5, 7
x 1 0 1
0 1 0 1

$\underline{P_1 = 0}$

Scanned with CamScanner

P_2 :- 2, 3, 6, 7
x 1 1 1
1 1 1 1 $\quad$ $P_2 = 1$

P_4 :- 4, 5, 6, 7
x 0 1 1
$P_4 = 0$

8 4 2 1
1 0 1 0

$2^0 \geq$
$1 \geq 3$
$2 \geq 4$
$4 \geq 4$
$P \in 2$
$2^0 \; 2^1$
1 2

Hamming code = 1100110
detect for 1110110 Error

	7	6	5	4	3	2	1
	111	110	101	100	011	010	001
Data	1	1	1		1		
Parity				0		1	0

P_1 = 1, 3, 5, 7
0 1 1 1 → 1

P_2 = 2, 3, 6, 7
1 1 1 1 → 0 $\quad$ = (101)
location 5.

P_4 = 4, 5, 6, 7
0 1 1 1 → 1

↳ 1100110 ← correction

Q → M = 100110 $\quad$ | M = 10001001

Calc. parity bit.
$2^P \geq m + p + 1$
m = 6
$2^0 \geq 6 + 0 + 1$
$1 \geq 7$ x
$2 \geq 6+1+1 = 8$ x

$3 \Rightarrow 2^{?}\ 4 \geq 6+2+1 = 9$ ×

$8 \geq 6+3+1 = 10$ ×

~~$16 \geq 6+4+1 = 11$~~ → ✓

P = 4

Q Size of code word = 6+4 = 10 bit.

$2^0,\ 2^1,\ 2^2,\ 2^3$

1, 2, 4, 8

Location Bit designation table

	10	9	8	7	6	5	4	3	2	1
	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data	1	0		0	1	1		0		
Parity			×				×		×	×

D_6	D_5	P_8	D_4	D_3	D_2	P_4	D_1	P_2	P_1
10	9	8	7	6	5	4	3	2	1
1	0	×	0	1	1	×	0	×	×

13/02/2020

S/w Engg

Software Design :- D_p ← SRS — Requirement.

Blueprint of ideads. (creative thinking)

Design Engineering

→ Design → Data design

Interface Architecture

Design Principles :-

A good design always have very less dependency.

Analysis → Requirement

Deliverables → O/p of every activity.

17/02/2020

CN

Framing :-

Data link layer breaks the stream into discrete frames and compute the checksum for each frame. At the destination end checksum is recomputed.

Frame :- Small manageable data units.

The breaking of bit stream by inserting spaces or time gaps is called framing.

Header	Payload	Trailer

Header : Contains source & destination address.
Payload : Data / packet received from N/w Layer.
Trailer :- contains error checksum.

Frames are of two types ~~fixed & vari~~

1.) Fixed frame
2.) Variable frame.

1) Fixed frame :- frame is of fixed size.
Drawback
if Data size < frame size
It suffers from internal fragmentation.
Solution :-
Padding.

→ There is no need to define the start and end boundaries of the frame.

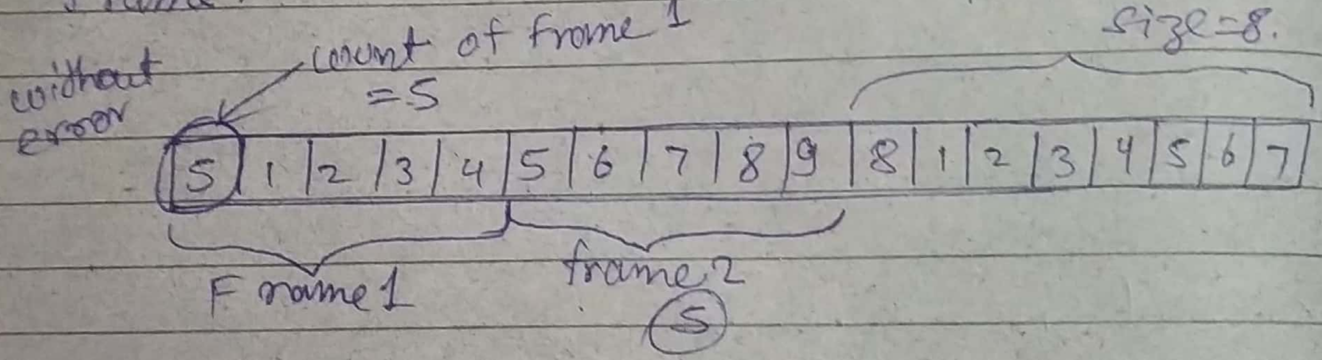
2.) Variable size :-
→ Frame is of variable size.
→ Need to define start & end boundaries of a frame.

2.1) Methods used for variable size framing :-

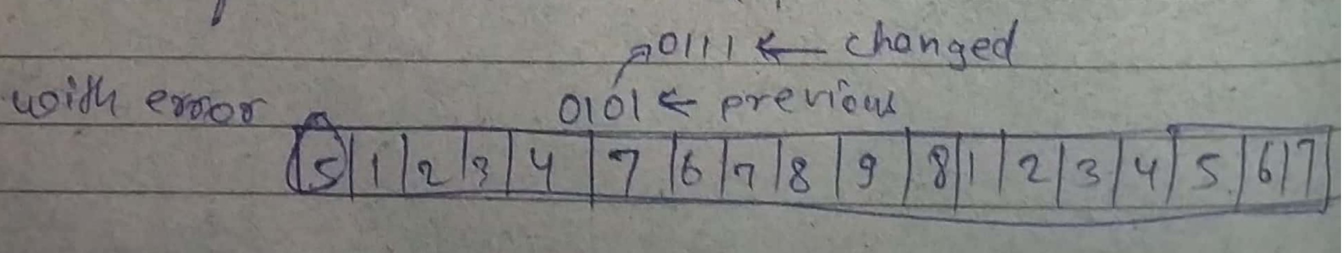
1.) Character count
2.) Start & End flags with character stuffing (Byte stuffing)
~~(Byte stuff~~
3.) Start & End flags with bit stuffing.

1.) Character count

This method uses a field in the header to describe / specify the no. of characters in the frame.



If header field is corrupted due to ~~the~~ some transmission error the destination will lose synchronization and will be unable to locate the start of the next frame.



2.) Start & End flags with character stuffing / Byte stuffing.

or header — or trailer

DLE	STX	A	B	C	ETX	DLE

DLE → Data Link escape
STX → Start of the text
ETX → end of the text

~~whenever~~

In character stuffing a special character (data link escape (DLE)) is stuffed in front of the control character when it appears as a part of data.

Flag	Header	A	flag	B	Trailer	Flag

flag → as a data value

↓ stuffing

Flag	Header	A	ESC	flag	B	Trailer	flag

ESC → to read as a data value

19/02/2020

CN

Q→ calculate the checksum

466F, 726F, 757A, 616E.

1)

```
  466F
  726F
  757A
+ 616E
------
  8FC6
    +1
------
  8FC7  ← checksum value
```

Actual message:

466F, 726F, 757A, 616E, ~~8FC7~~ 7038

→ 15's compliment

```
  FFFF
- 8FC7
------
  7038
```

```
 F F
 F F
 ---
   E
```

Receiver side

```
  466F
  726F
  757A
  616E
  7038
------
1) FFFE
     +1
------
  FFFF
```

→ 15's compliment is

0000 ← message is error free (data is correct)

A − B ⇒ +A +(−B).

```
  7E
- 8E
```

```
  0111 1110
- 1000 1110
```

128 64 32 16 8 4 2 1

```
  96
  16
  14
 ---
 126
```

−16

```
 128
 +14
 ---
 142
```

Scanned with CamScanner

0X3456, 0XABCC, 0X82BC, 0XEEEE.

3	4	5	6
A	B	C	C
8	2	B	C
E	E	E	E
1) 5	1	C	C
+			2
5	1	C	E

$$\begin{array}{r} F\ F\ F\ F \\ 15's \rightarrow -\ 5\ 1\ C\ E \\ \hline A\ E\ 3\ 1 \end{array}$$

AE31 ← checksum

6+12+12+14

16) 44 (2
32
12

16) 23
16
7

Calculate the checksum value of ~~1001001110 01~~
1001001110010011 and
1001100000100110 1 of 16 bit segment is

```
  1001001110010011
+ 1001100000100110 1
1)0010101111100000
                 1
  0010101111100001
```

1101010000011110 ← checksum.

100 11

1001001110010011, 1001100000100110 1,
1101010000011110.

Received code with burst error.

Scanned with CamScanner

Ex. I got my hair dressed/cut.
The teacher made the class laugh.
I had my car repaired.

24/02/2020

CN

Bit Stuffing:-

In bit stuffing a specific bit is stuffed into the data bits.

- A bit ~~zero~~ 0 is stuffed after 5 consecutive 1's
- Start & End delimeter = 01111110

It tells about start of the frame & end of the frame

| 01111111011011111000 | ← data.

⇓

| 01111110 | 0111110110110111110000 | 01111110 |

start, stuff, stuff, end

At receiver end

Destuffing is done.

→ 01111111011 011111000 ← data

Q → ED (End Delimeter) = 0111
Data : 011110110110

→ 0110110110010100

Extra bit.

Q. ED = 1000
Data: 0100001100'00001
1 → 010010Φ10011100100 1
S S S

Flow Control

Flow control is a mechanism to restrict the amount of flow that sender can send

Flow control Protocol :-

(Perfect channel) → For Noiseless channel
- Stop & wait

(imperfect channel) → For noisy channel
- Stop & wait ARQ
- GO-Back N-ARQ
- Selective Repeat ARQ

Automatic Repeat Request (ARQ)

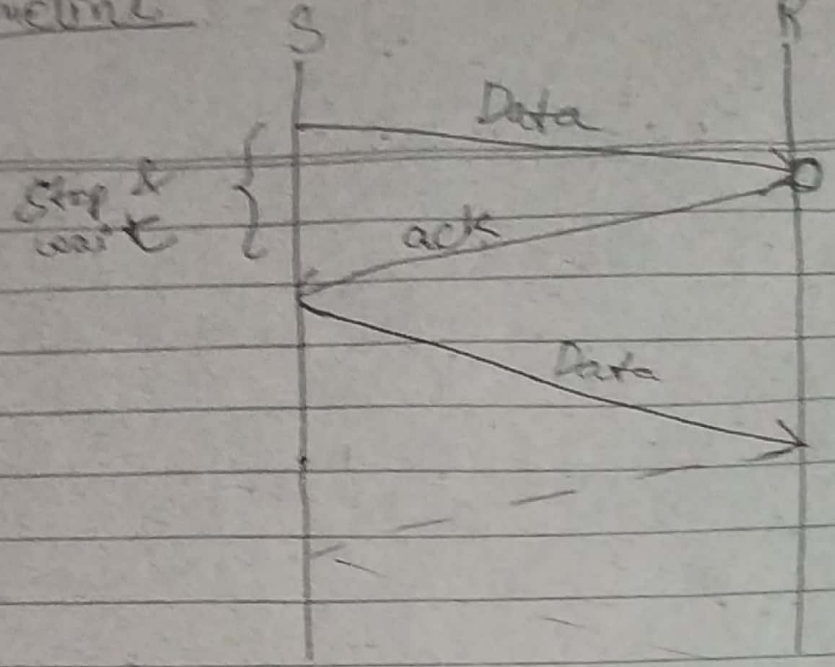
Stop & Wait Protocol
→ Assumption
→ Working
→ Analysis
→ Advantage and Limitation

- Simplest Protocol.

It works under following assumptions
- Communication channel is perfect.
- No error occurs during transmission.

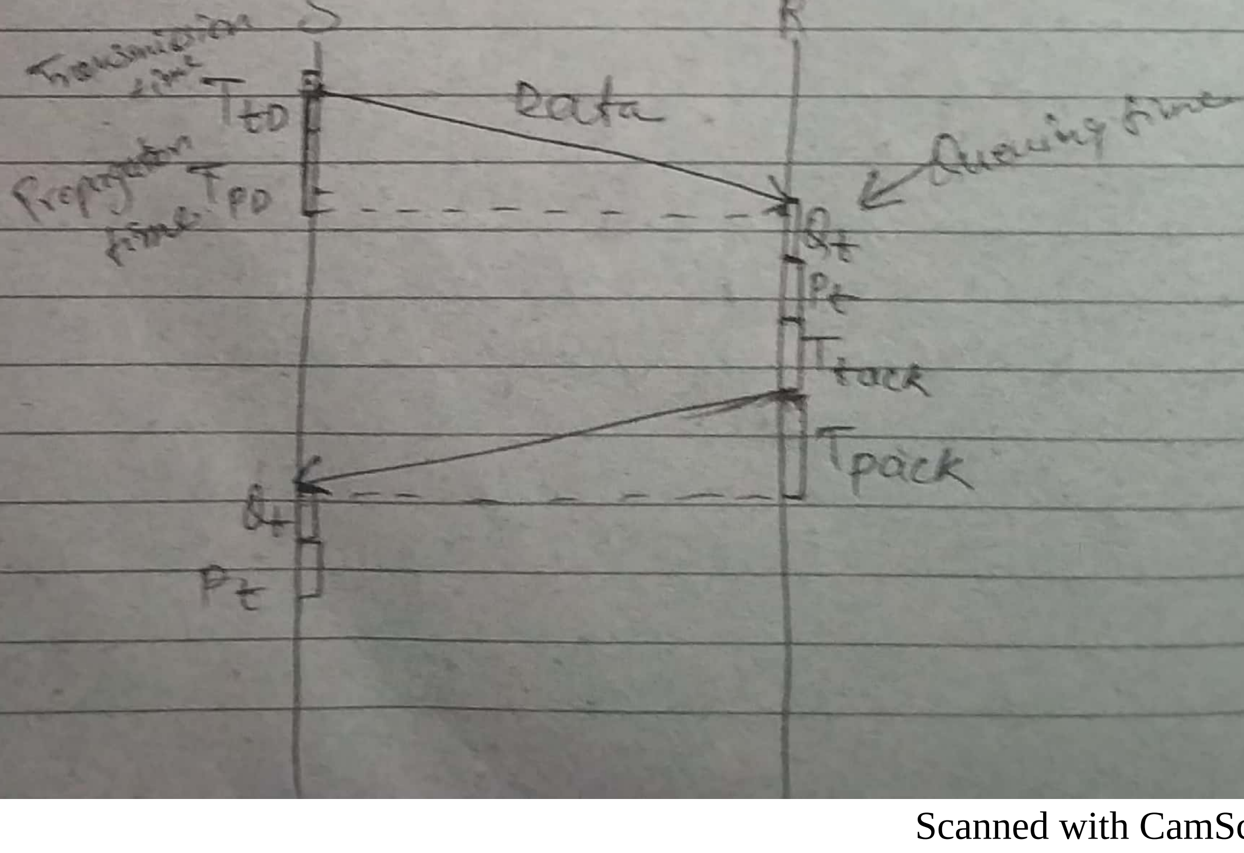
Scanned with CamScanner

Timeline

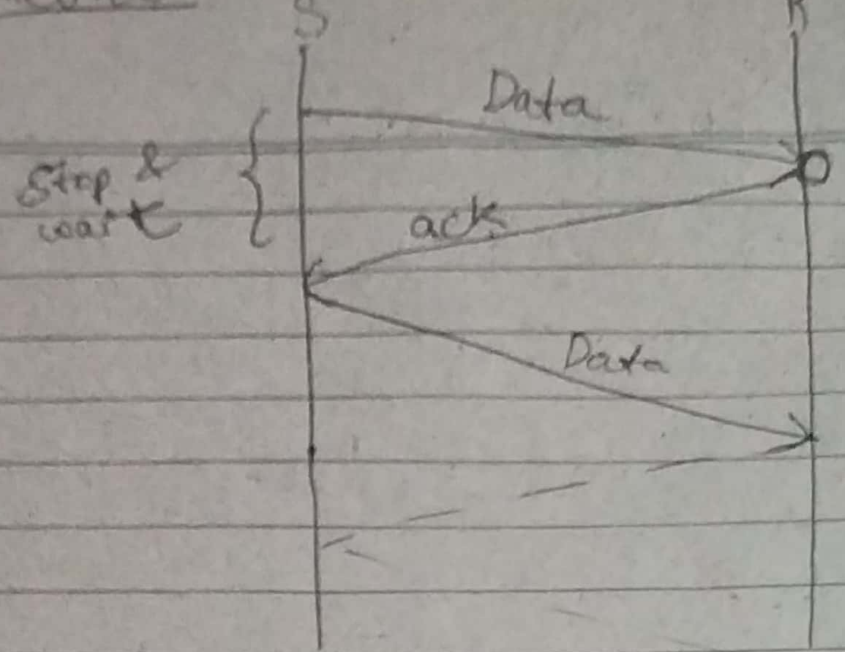


Working

1) Sender sends a data packet to the receiver
2) Sender stop & wait's for the acknowledgement for the data packet already sent
3) Receiver receives and process the data packet.
4) Receiver sends an acknowledgement (ack) to the sender.
5) Sender sends the next packet to the receiver after receiving the acknowledgement (ack).

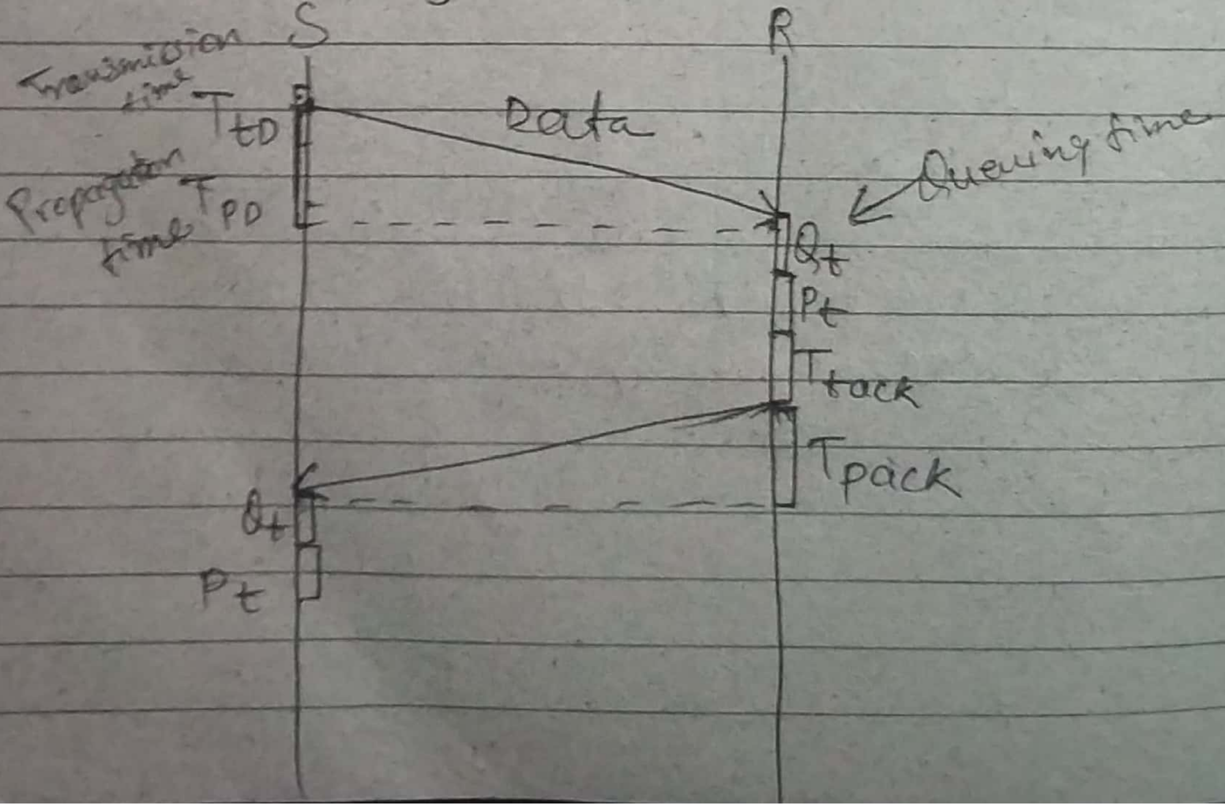


Timeline



Working

1) Sender sends a data packet to the receiver
2) Sender stop & waits for the acknowledgement for the data packet already sent.
3) Receiver receives and process the data packet.
4) Receiver sends an acknowledgement (ack) to the sender.
5) Sender sends the next packet to the receiver after receiving the acknowledgement (ack).



Analysis

Total time for sending one packet

$$= (T_t + T_p + Q_D + P_D)_{Data} + (T_t + T_p + Q_D + P_D)_{ACK}.$$

Assumption :-

1) Q_t & P_t are to be 0 for both sender & receiver.
2) T_{tACK} is to be zero since its size is very small.

$$\Rightarrow T_t + T_p + T_p \Rightarrow \underline{T_t + 2T_p}$$

~~Useful~~ Useless time :-

Time for which sender is forced to wait and do knocking.

$2 \times T_p$.

$$\boxed{RTT = 2 * T_p}$$

RTT ← Round Trip time.

Useful time :-

Transmission time :- T_t.

Total time = Useful + Useless time.

Efficiency, $\eta = \dfrac{\text{Useful time}}{\text{Total time}} = \dfrac{T_t}{T_t + 2T_p}$

↳ Link Utilisation

sender Utilisation. $= \dfrac{T_t / T_t}{\dfrac{T_t}{T_t} + 2\dfrac{T_p}{T_t}}$

$$\boxed{\eta = \frac{1}{1 + 2a}}$$

where $a = \dfrac{T_p}{T_t}$

Throughput :- / Bandwidth utilization / maximum data rate
No. of bits transferred in 1 sec. or effective Bandwidth

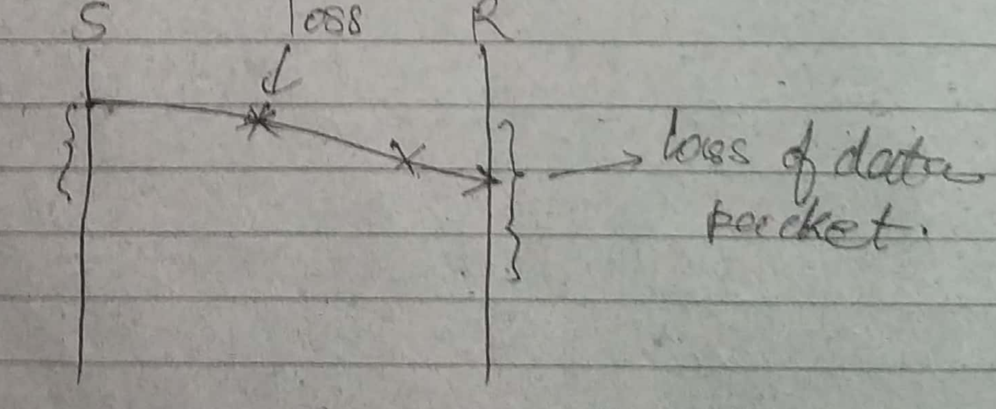
$$\Rightarrow \frac{L}{T_t + 2T_p} \Rightarrow \frac{\frac{L}{B} \times B}{T_t + 2T_p}$$

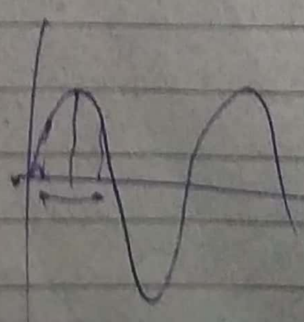
$$\Rightarrow \left(\frac{T_t}{T_t + 2T_p} \right) \times B$$

$$\boxed{\text{Throughput} = \eta \times B}$$

Advantage :-
1) Simplest protocol to implement.
2) Each packet received

Limitation :- ~~timer~~
1) Loss of data packet.
2) Loss of ack
3) Delayed Ack
4) Damage frame (data packet)





Cyber Security (25/02/2020)

Trojan Horse.
Zombie used to launch
DDoS attack
Distributed Denial of Service attacks.

Virus requires a host program.
logic bomb.
26/02/2020.

Q Consider 2 host x & y connected by a single direct link of rate 10^6 bit per second. The distance b/w two host is 10000 km and the propagation speed along the link is 2×10^8 m/s. Host x sends a file of 50000 bytes as one large message to host by continuously. Let the transmission & propagation delays be p ms & q ms resp. Then the value of p & q are

Sol^n Given :-

$B = 10^6$ bps
$d = 10000$ km
$V = 2\times10^8$ m/s
$L = 50000$ bytes.

$$\text{Transmission time} = \frac{L}{B} = \frac{50000\times8}{10^6} = \frac{40}{100}$$

$$= 0.4s = 400\,ms.$$

$$\text{Propagation time} = \frac{D}{V} = \frac{10000\times10^3}{2\times10^8} = \frac{1}{20}$$

$$= 0.05.$$

Propagation time = 50ms.

Q → If the bandwidth of the line is 1.5 Mbps, RTT is 45 ms & packet size is 1 KB then find the link utilisation in stop & wait protocol.

Given: B = 1.5 Mbps, RTT = 45 ms
packet size = 1 KB.

$\eta = \frac{1}{1+2a}$ $\quad a = \frac{T_p}{T_t}$ $\quad \frac{16}{\frac{80}{15}}3$

$T_t = \frac{1\times10^3 \times 8}{1.5\times10^6} \Rightarrow \frac{8}{1.5\times10^3} = 5.33\ ms$

RTT = $2 \times T_p$

$\frac{45}{2} = T_p \Rightarrow T_p = 22.5\ ms.$

$a = \frac{22.5}{5.33} = 4.22$

$\eta = \frac{1}{1+2\times4.22} = \frac{1}{1+8.44}$

$= \frac{1}{9.44} = 0.1059$

$= 0.106$

Q→ A channel has a bit rate of 4 Kbps & one way propagation delay of 20 ms. The channel uses stop & wait protocol. The transmission time of the acknowledgement frame is negligible. To get a channel efficiency of atleast 50% the minimum frame size should be

Sol^n $B = 4\ Kbps$

$T_p = 20\ ms \quad \eta = 50\% \quad L = ?$

Let L bit,

$T_t = \frac{L}{4\times10^3}$

$a = \frac{T_p}{T_t} = \frac{20\times10^{-3}}{\frac{L}{4\times10^3}} \Rightarrow \frac{20\times10^{-3}\times4\times10^3}{L}$

$a = \frac{80}{L}$

$\eta \geq 50\% \Rightarrow \frac{1}{1+2a} \geq \frac{1}{2}$

$\Rightarrow 2 = 1+2a \Rightarrow 2a = 1$

$a = \frac{1}{2}$

$\Rightarrow a = \frac{80}{L} \Rightarrow \frac{80}{L} = \frac{1}{2} \Rightarrow L = 160$ bit

Scanned with CamScanner

Q) If the packet size is 1KB & propagation time is 15ms the channel capacity is 10^9 bps find the transmission time & utilisation of sender in stop & wait protocol.

Sol^n $L = 1KB$ $T_p = 15ms$ $B = 10^9 bps$

$$T_t = \frac{L}{B} = \frac{1 \times 10^3 \times 8}{10^9} = \frac{8}{10^6} \rightarrow 8 \times 10^{-6} sec.$$

$$T_t = 0.008 ms.$$

$T_p = 15ms.$

$$a = \frac{T_p}{T_t} = \frac{15}{0.008} = \frac{15 \times 10^3}{8} = 1.875 \times 10^3$$

$$\eta = \frac{1}{1+2a} = \frac{1}{1+2 \times 1.875 \times 10^3} = \frac{1}{1+3.75 \times 10^3}$$

$$\eta = \frac{1}{1+3750} = 0.000267$$

$$\eta = 0.0267\%$$

Q) what is the throughput in stop & wait protocol by a maximum packet size of 1000 bytes & network span of 10Km Assume the speed of light in cable is 70% of the speed of light in

Sol^n $L = 1000$ bytes $d = 10 Km$

$V = 2.1 \times 10^8$ m/s.

Scanned with CamScanner