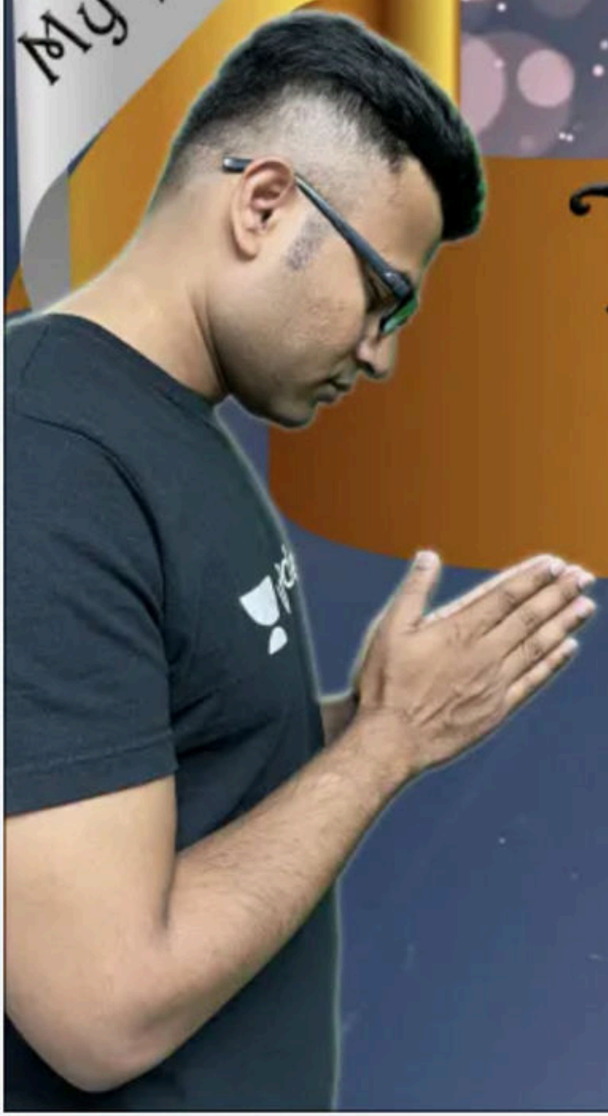
TIME OUT TIMER COMPUTATION ALGO -BASIC AND JACOBSON, Silly Window Syndrome,

Complete Course on Computer Networks - Part III

Ravindrababu Ravula • Lesson 13 • Apr 6, 2021

My philosophy



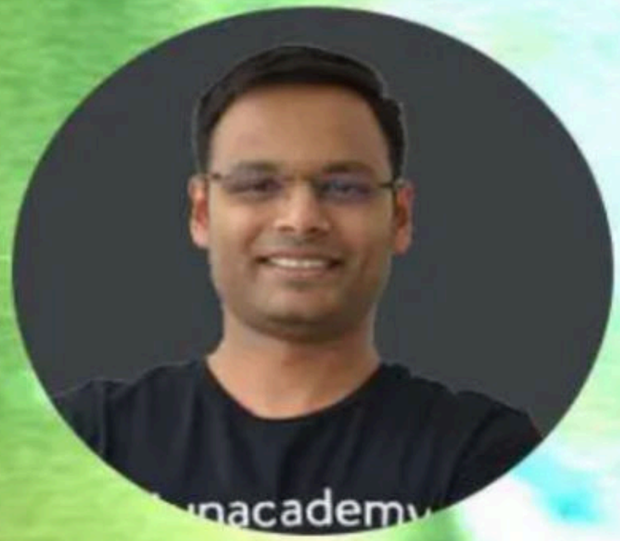
TEACHING IS WORSHIP

STUDENTS ARE GODS

6-04-2021 Full Day Schedule of RBR

Lecture Name	Time
GATE Question on Normalizations - Set-1 | DBMS	06:00 AM - 07:00 AM
TIME OUT TIMER COMPUTATION ALGO -BASIC AND JACOBSON, Silly Window Syndrome | CN	07:00 AM - 08:00 AM
Analysing time and space complexity Part-2 | L:2 | Data Structures and Algorithms	05:00 PM - 06:00 PM
How to declare, initialize and use an array in Java | 1-D array, 2-D array | L:13 | Java Course	06:00 PM-07:00PM
Introduction to shell programming and example programs | L:7 | Linux course for Engg and UGC-NET	07:00 PM - 08:00 PM
What is version control system? | Why do we need VCS? | L:1 | Git Basics	08:00 PM - 09:00 PM

unacademy



LEARN FROM TOP EDUCATORS

Destination	Distance	Next Hop
A	0	A
B	2	B
C	5	B
D	1	D

A

Destination	Distance	Next Hop
A	2	A
B	0	B
C	3	C
D	3	A

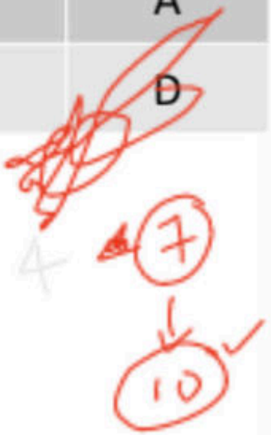
B

Destination	Distance	Next Hop
A	5	B
B	3	B
C	0	C
D	6	B

C

Destination	Distance	Next Hop
A	1	A
B	3	A
C	6	A
D	0	D

D

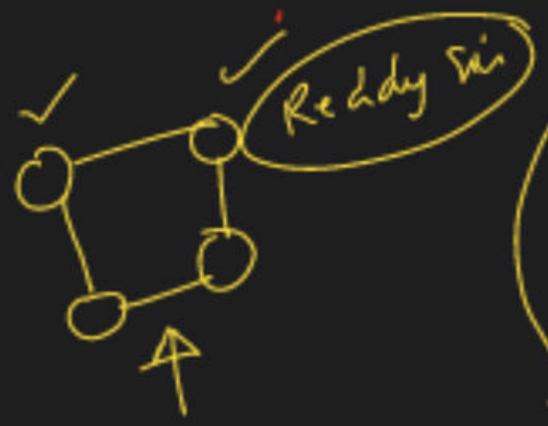


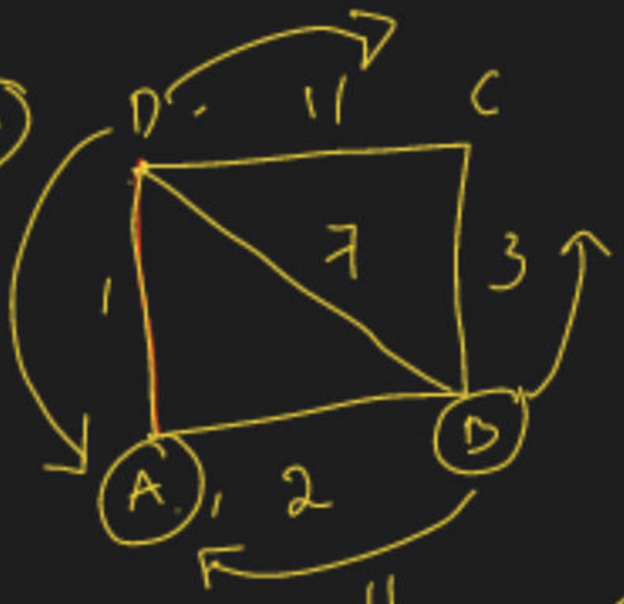
Identifying Unused Links-

After routing tables converge (becomes stable),
- Some of the links connecting the routers may never be used.
- In the above example, we can identify the unused links as-

We have-
- The value of next hop in the final routing table of router A suggests that only edges AB and AD are used.
- The value of next hop in the final routing table of router B suggests that only edges BA and BC are used.
- The value of next hop in the final routing table of router C suggests that only edge CB is used.
- The value of next hop in the final routing table of router D suggests that only edge DA is used.

Thus, edges BD and CD are never used





DV

Converge ?

3 rounds ✓

n-2

Initial → paths of length 1 are considered. ✓

next → path of length 2 are consider.

next → path of length 3 are consider.

n-1
n-2
n

n → nodes

maximum length of shortest path

algo

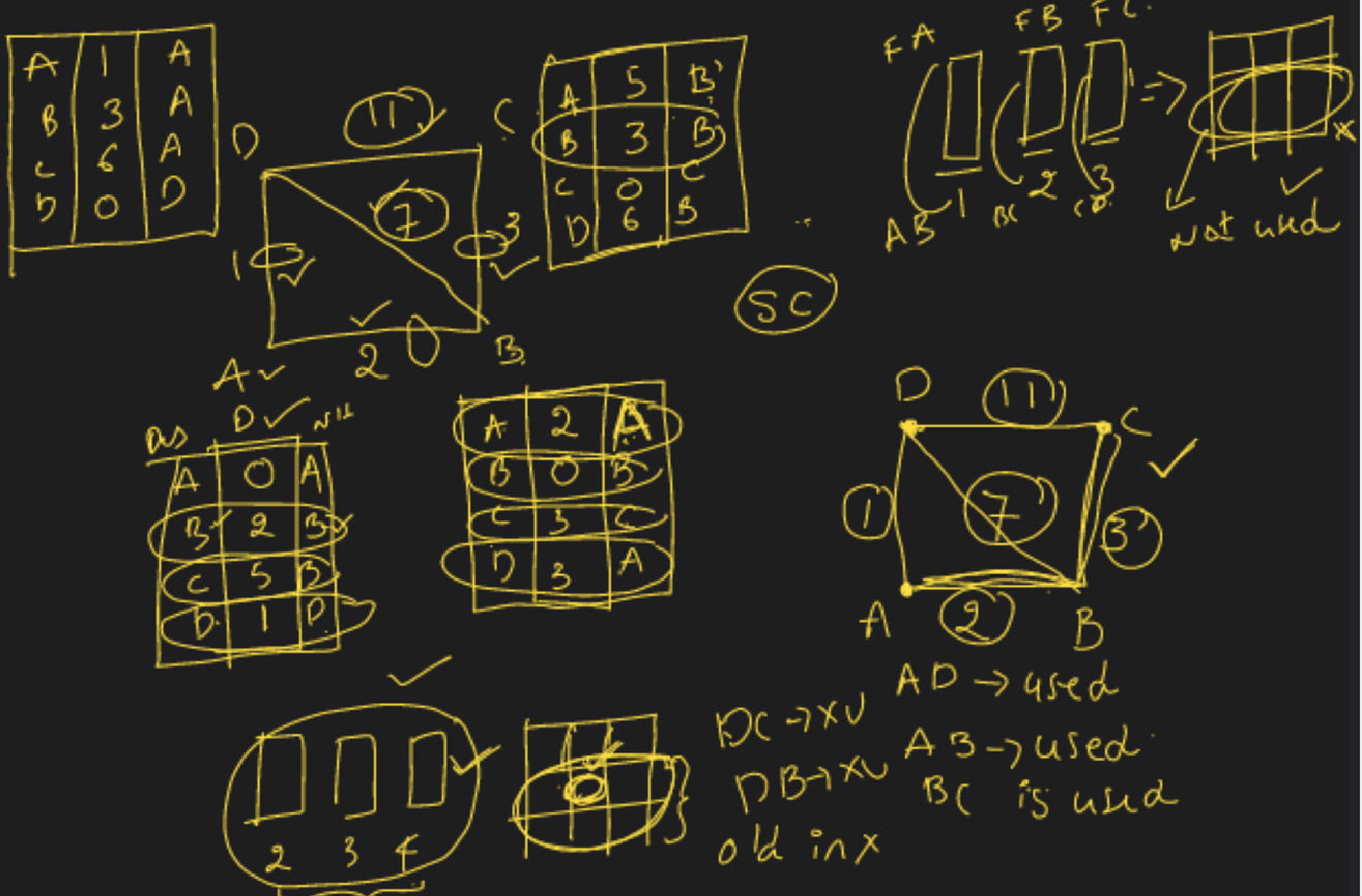
"N-1"

Periodically ?

Topo

Traffic

→ Dynamic.

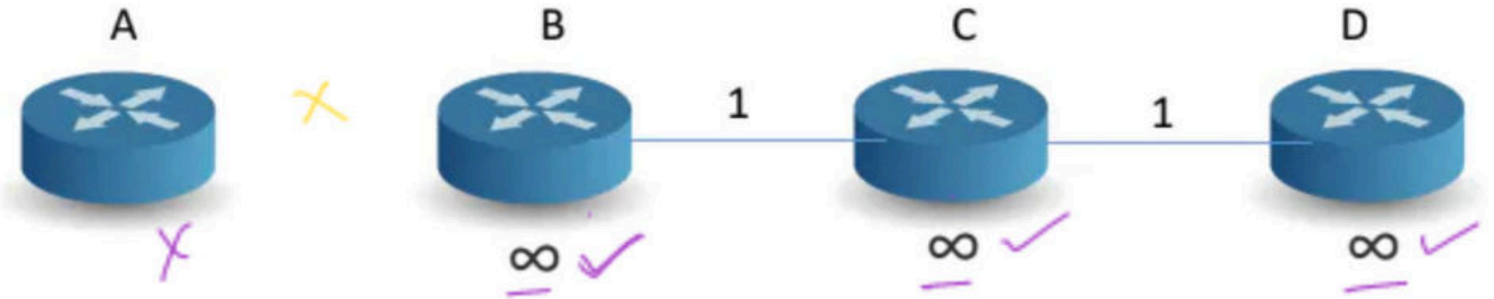

FA
FB
FC
AB
BC
CD
not used
SC
AD → used
AB → used
BC is used
DC → XU
DB → XU
old inx

Computer Networks

DVR – Count To Infinity

Count to Infinity – Bad news spreads slow, Good news spreads fast

Good news



∞ in a table

32 -> TTL

Initially A is not connected to the network

B

A	∞	-

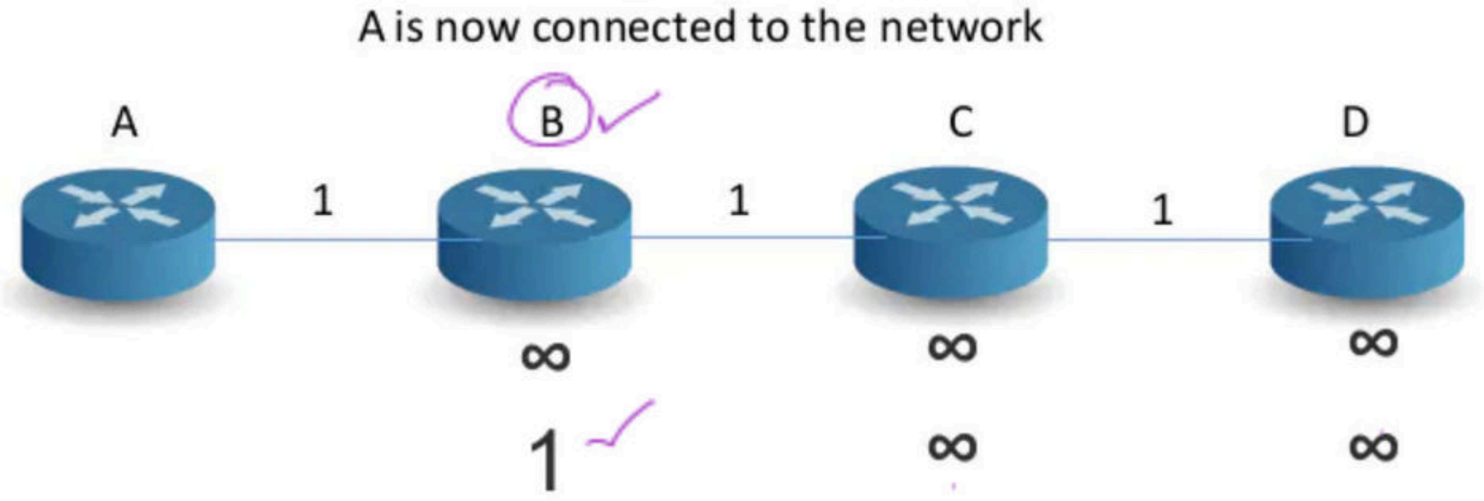
C

A	∞	-

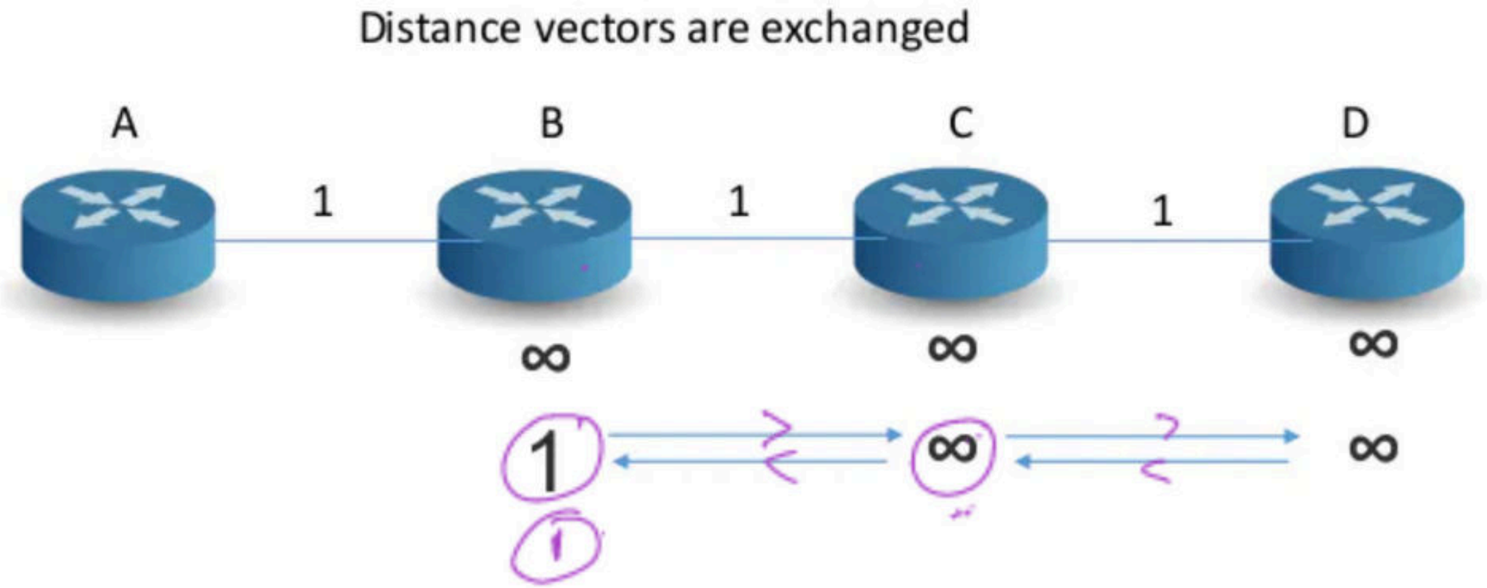
D

A	∞	-

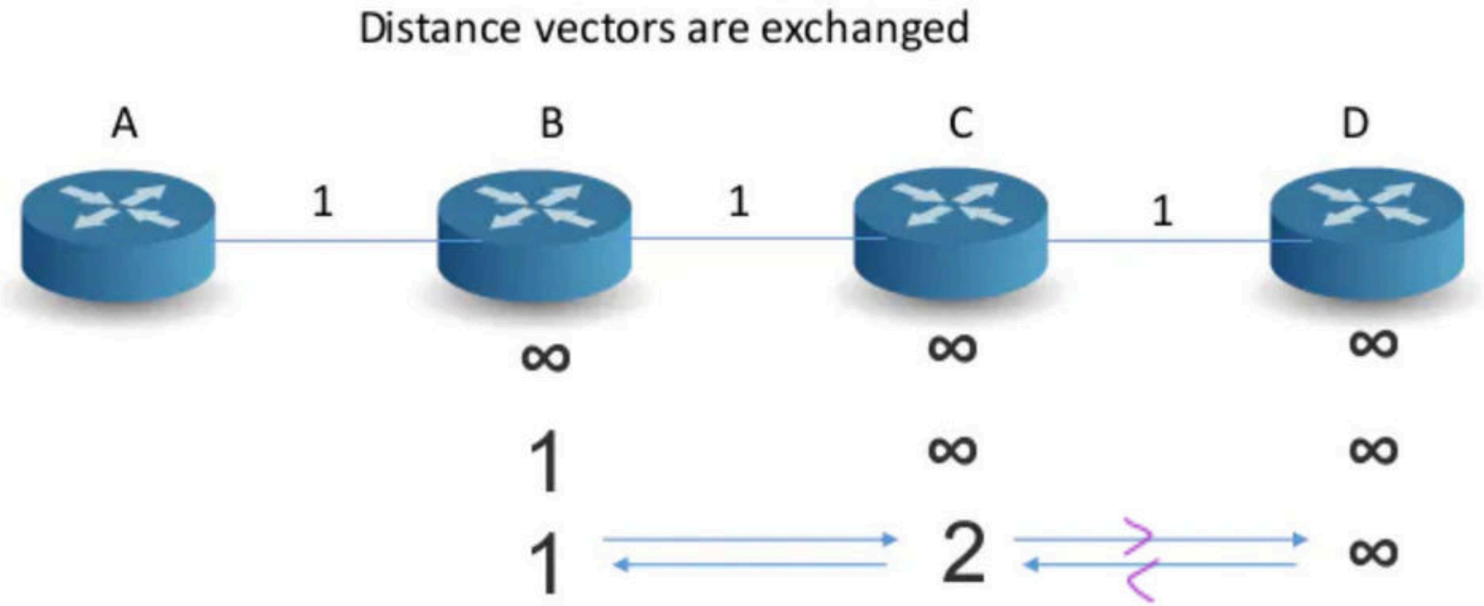
Count to Infinity – Bad news spreads slow, Good news spreads fast



Count to Infinity – Bad news spreads slow, Good news spreads fast

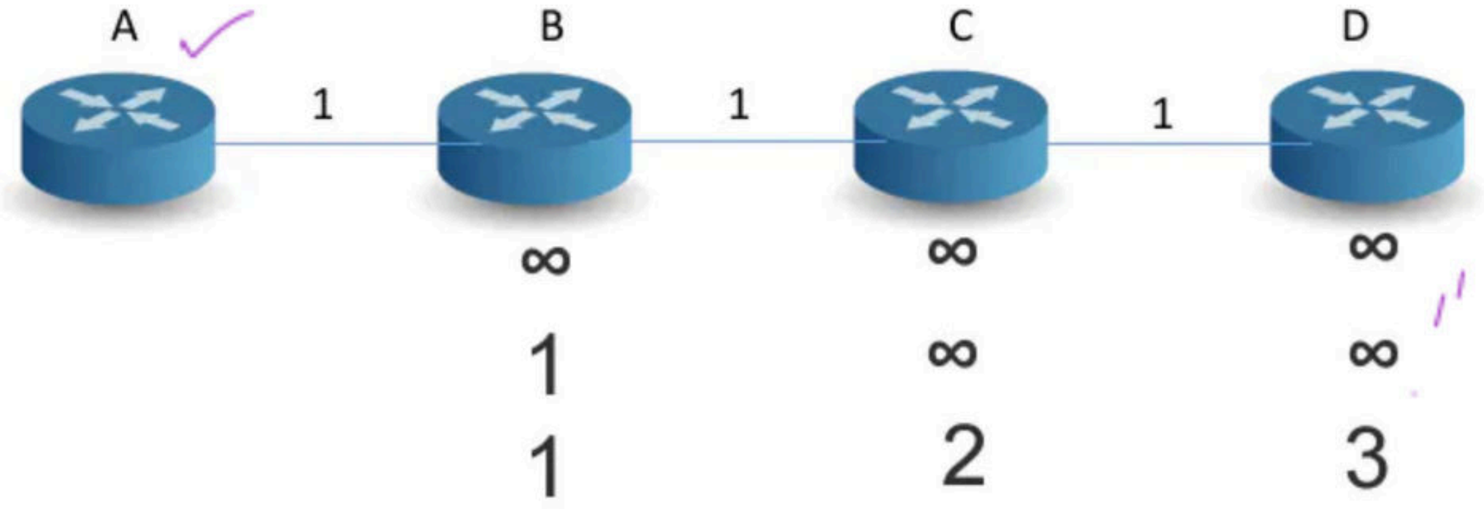


Count to Infinity – Bad news spreads slow, Good news spreads fast



Count to Infinity – Bad news spreads slow, Good news spreads fast

Distance vectors are exchanged

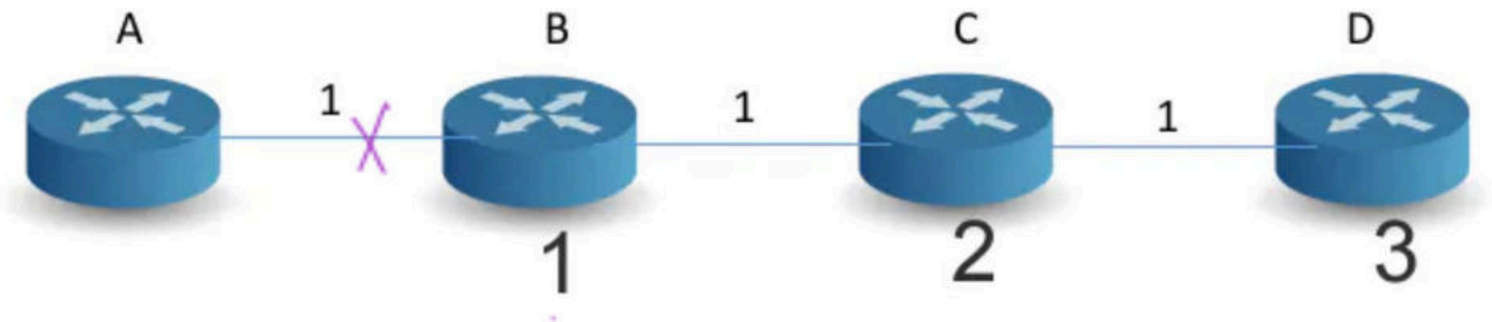


BN

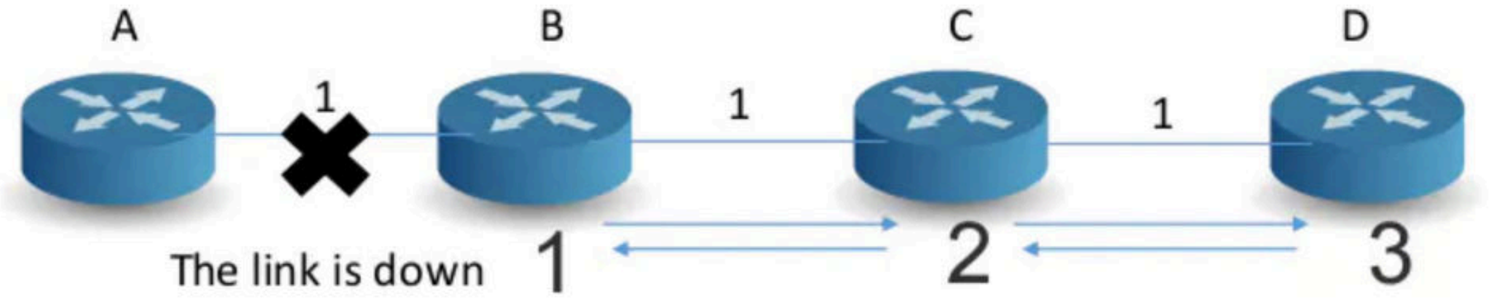
THIS IS GOOD NEWS → Spreads faster in DVR

Count to Infinity – Bad news spreads slow, Good news spreads fast

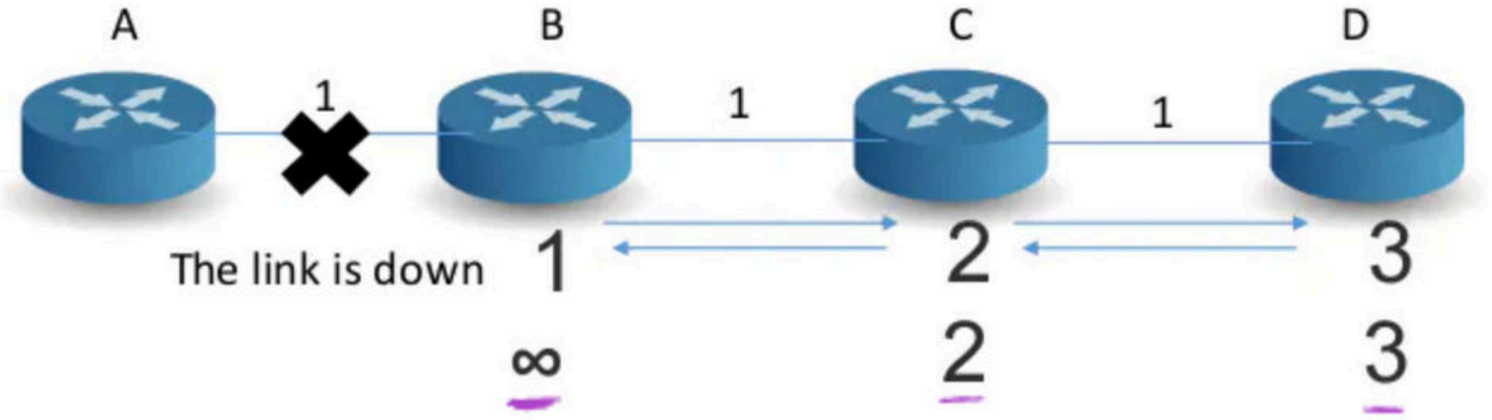
Bad news



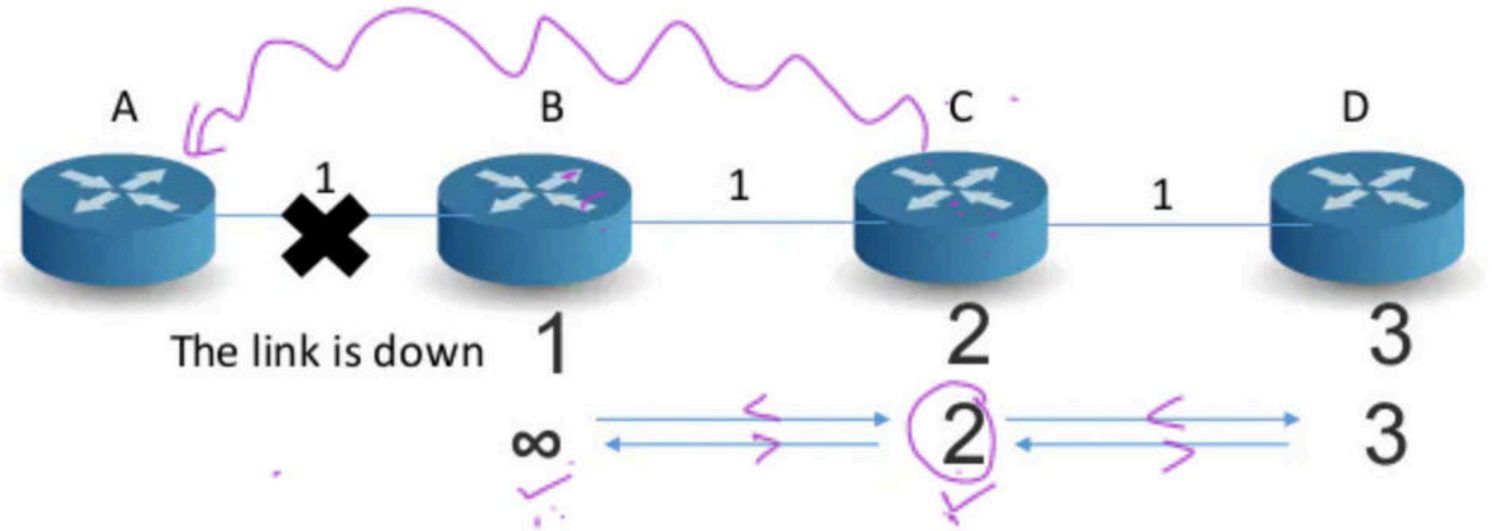
Count to Infinity – Bad news spreads slow, Good news spreads fast



Count to Infinity – Bad news spreads slow, Good news spreads fast



Count to Infinity – Bad news spreads slow, Good news spreads fast



DVR
↓
1980's

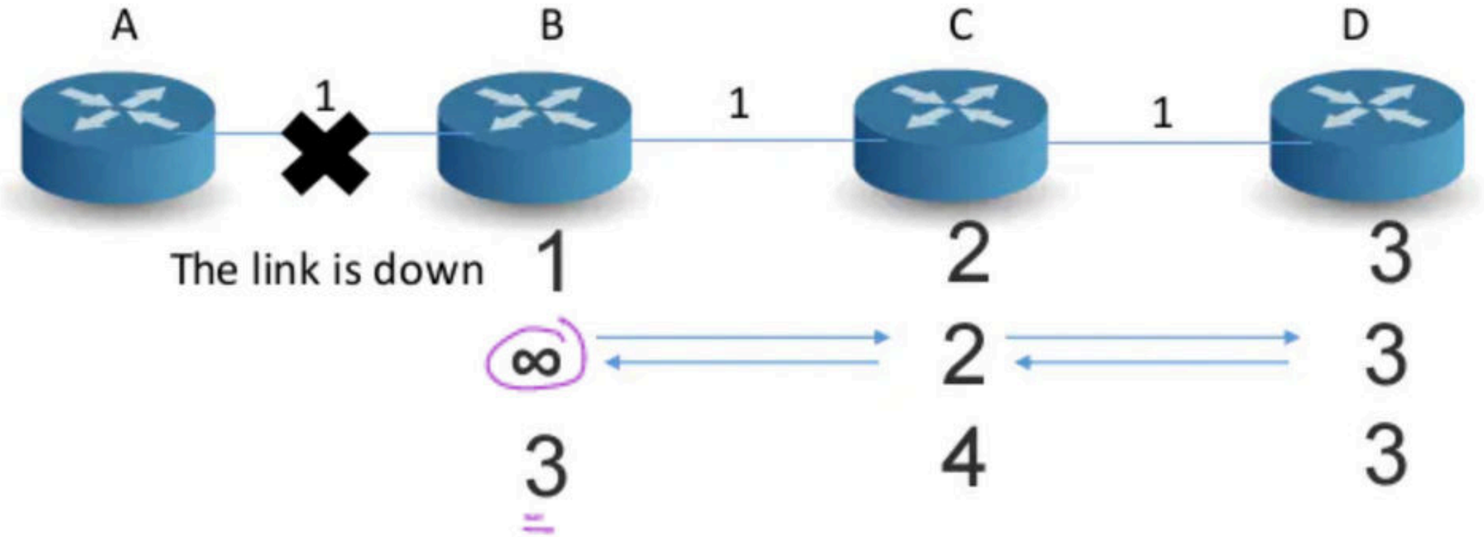
why.

NH

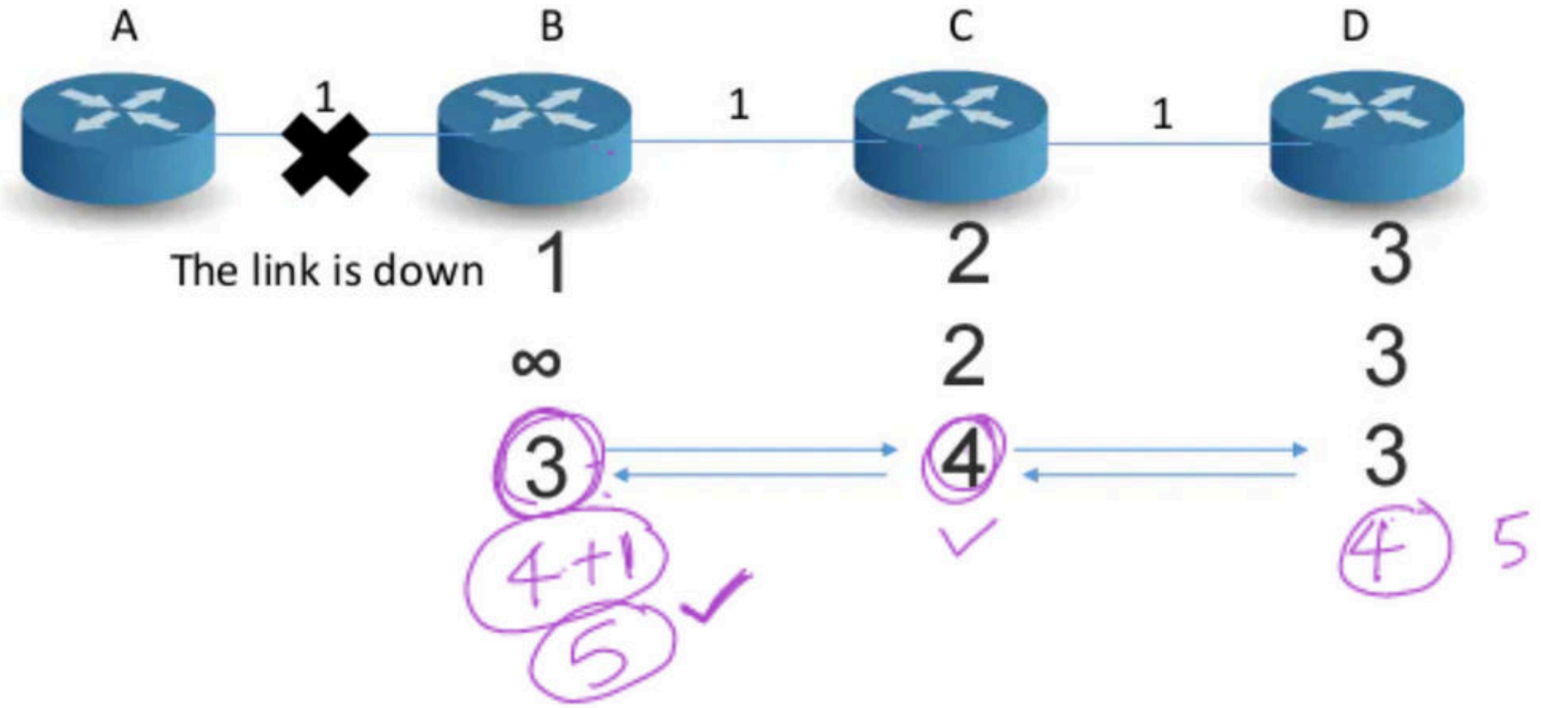
via B.

Size { why can't we send entire RT

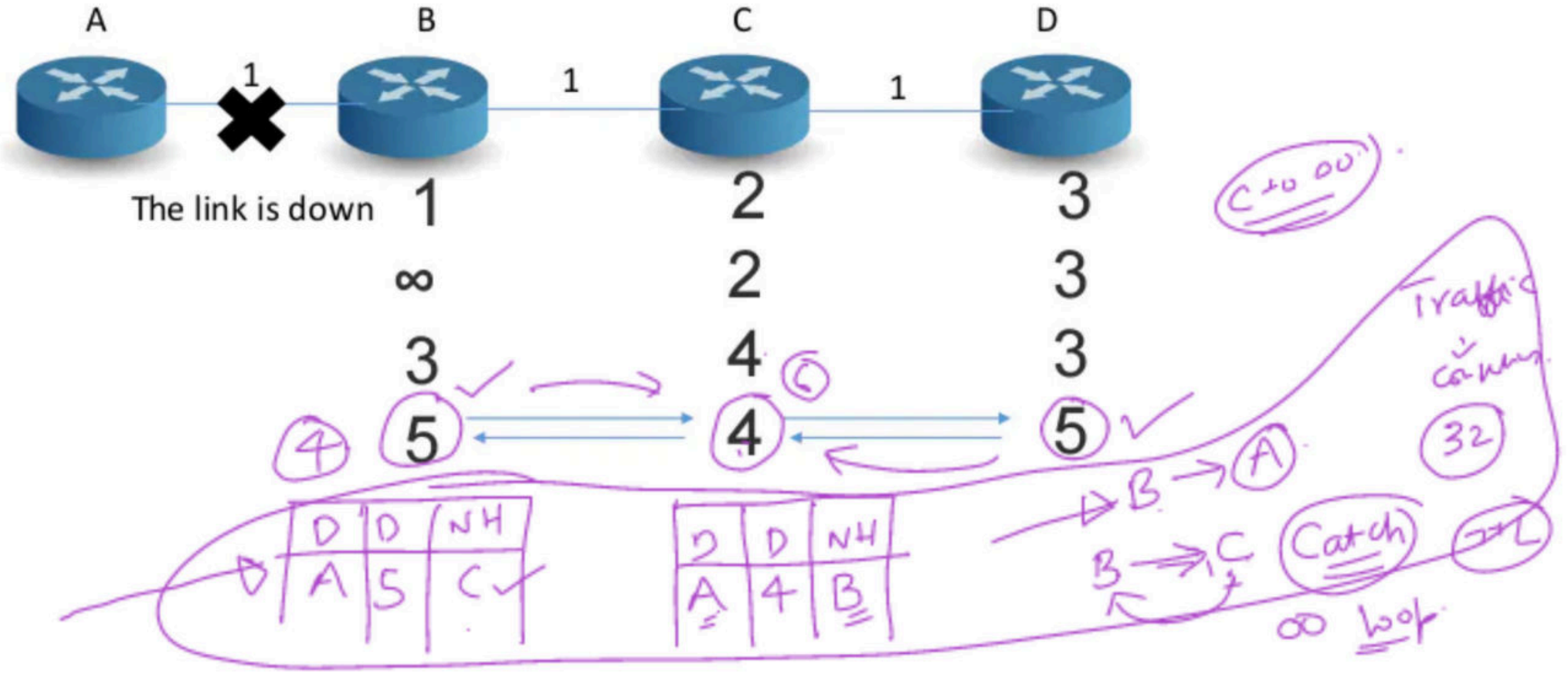
Count to Infinity – Bad news spreads slow, Good news spreads fast



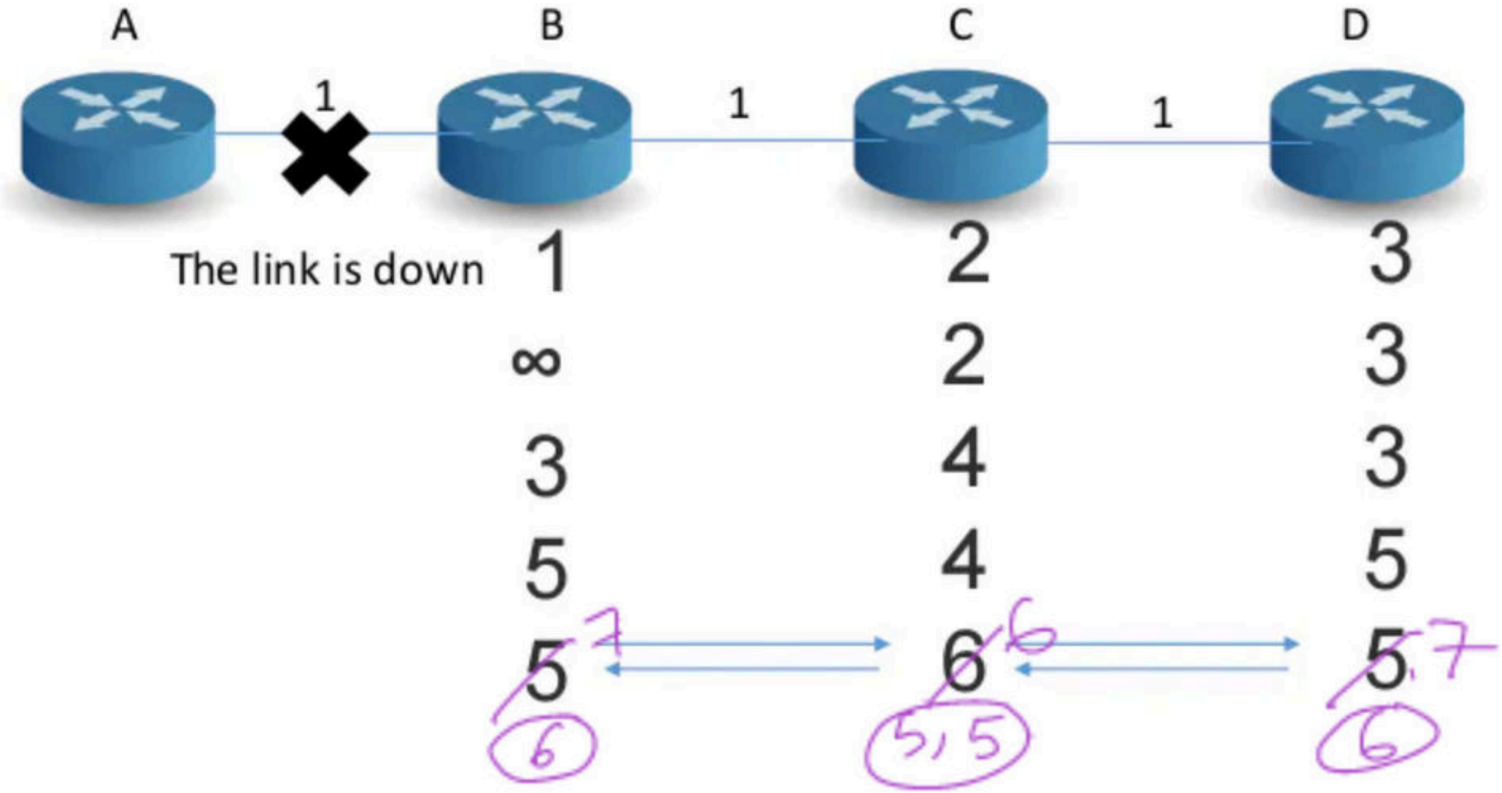
Count to Infinity – Bad news spreads slow, Good news spreads fast



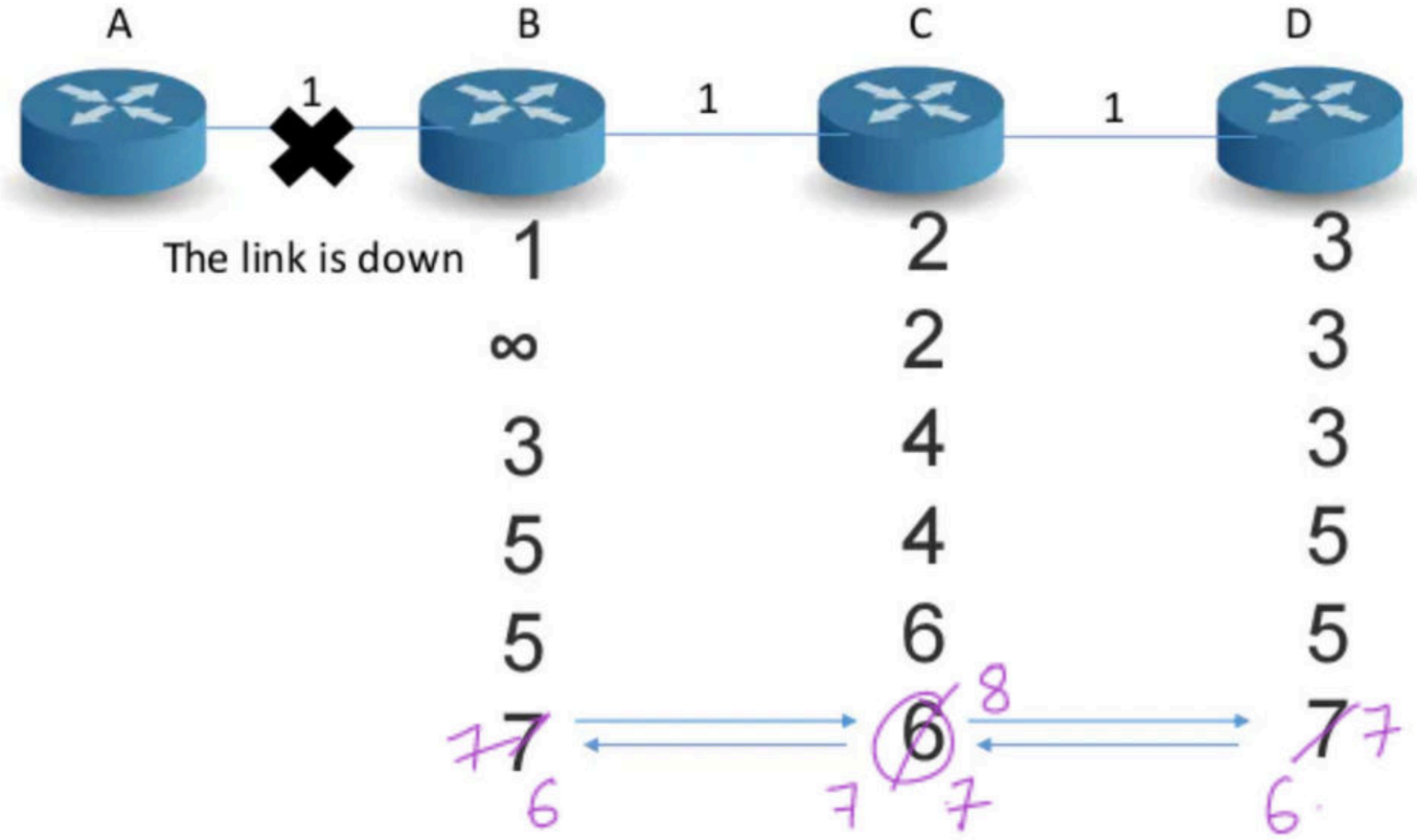
Count to Infinity – Bad news spreads slow, Good news spreads fast



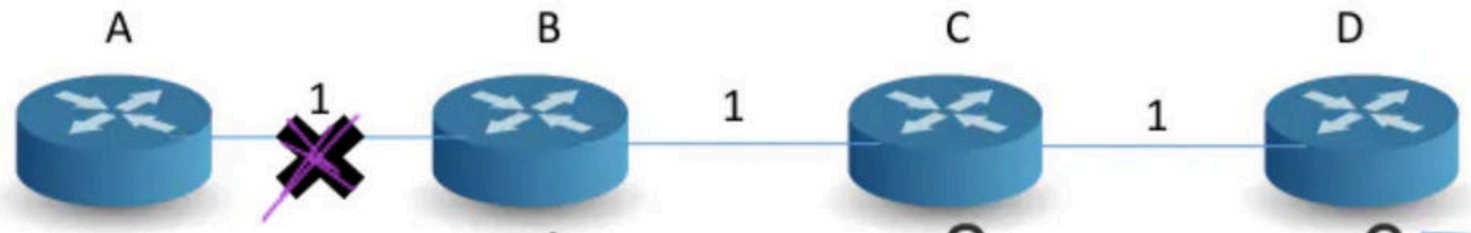
Count to Infinity – Bad news spreads slow, Good news spreads fast



Count to Infinity – Bad news spreads slow, Good news spreads fast



Count to Infinity – Bad news spreads slow, Good news spreads fast



The link is down

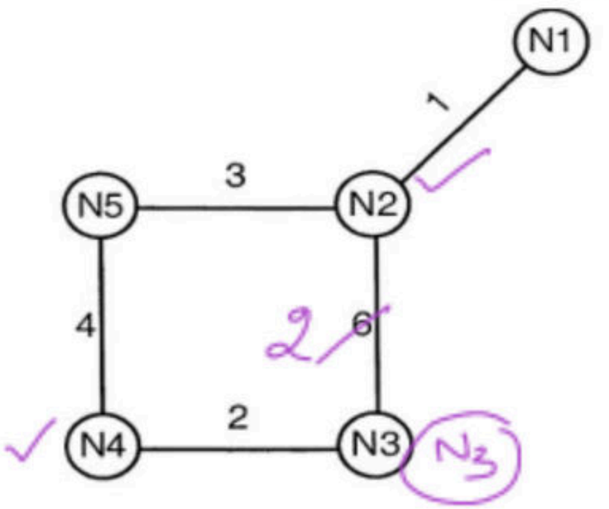
B	C	D
1	2	3
∞	2	3
3	4	3
5	4	5
5	6	5
7	6	7
.	.	.
.	.	.
∞	∞	∞

Count to Infinity

Fix ∞ → 32

GATE CS 2011

1.) Consider a network with five nodes, N1 to N5, as shown below.



The network uses a Distance Vector Routing protocol.
Once the routes have stabilized, the distance vectors at different nodes are as following.

N1: (0, 1, 7, 8, 4)
N2: (1, 0, 6, 7, 3)
N3: (7, 6, 0, 2, 6)
N4: (8, 7, 2, 0, 4)
N5: (4, 3, 6, 4, 0)

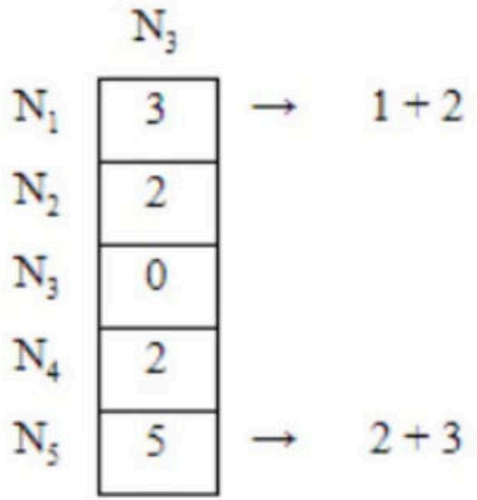
Each distance vector is the distance of the best known path at the instance to nodes, N1 to N5, where the distance to itself is 0. Also, all links are symmetric and the cost is identical in both directions. In each round, all nodes exchange their distance vectors with their respective neighbors. Then all nodes update their distance vectors. In between two rounds, any change in cost of a link will cause the two incident nodes to change only that entry in their distance vectors. 52. The cost of link N2-N3 reduces to 2(in both directions). After the next round of updates, what will be the new distance vector at node, N3.

(A) (3, 2, 0, 2, 5)
(B) (3, 2, 0, 2, 6)
(C) (7, 2, 0, 2, 5)
(D) (7, 2, 0, 2, 6)

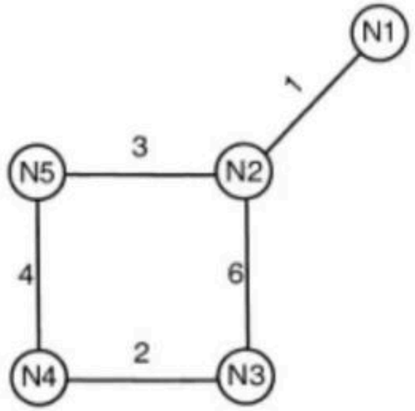
at N3

7
6
0
2
6

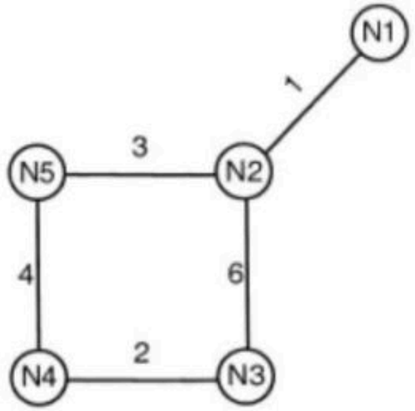
Solution:
In the next round, every node will send and receive distance vectors to and from neighbors, and update its dostance vector.
N3 will receive (1, 0, 2, 7, 3) from N2 and it will update distances to N1 and N5 as 3 and 5 respectively.



2.)Consider a network with five nodes, N1 to N5m as show below. The network a distance vector protocol. What is the routes Have stabilized, the distance vector a different notes are as follows N1:(0,1,7,8,4) N2:(1,0,6,7,3) N3:(7,6,0,2,6) N4:(8,7,2,0,4) N5:(4,3,6,4,0)



2.)Consider a network with five nodes, N1 to N5m as show below. The network a distance vector protocol. What is the routes Have stabilized, the distance vector a different notes are as follows N1:(0,1,7,8,4) N2:(1,0,6,7,3) N3:(7,6,0,2,6) N4:(8,7,2,0,4) N5:(4,3,6,4,0)



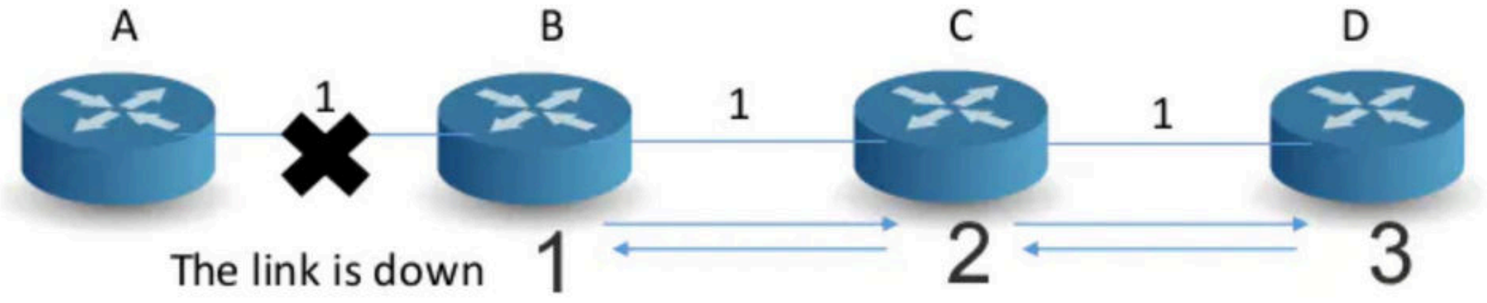
Solution:
N3 has neighbors N2 and N4
N2 has made entry ∞
N4 has the distance of 8 to N1
N3 has the distance of 2 to N4
So 2 + 8 = 10

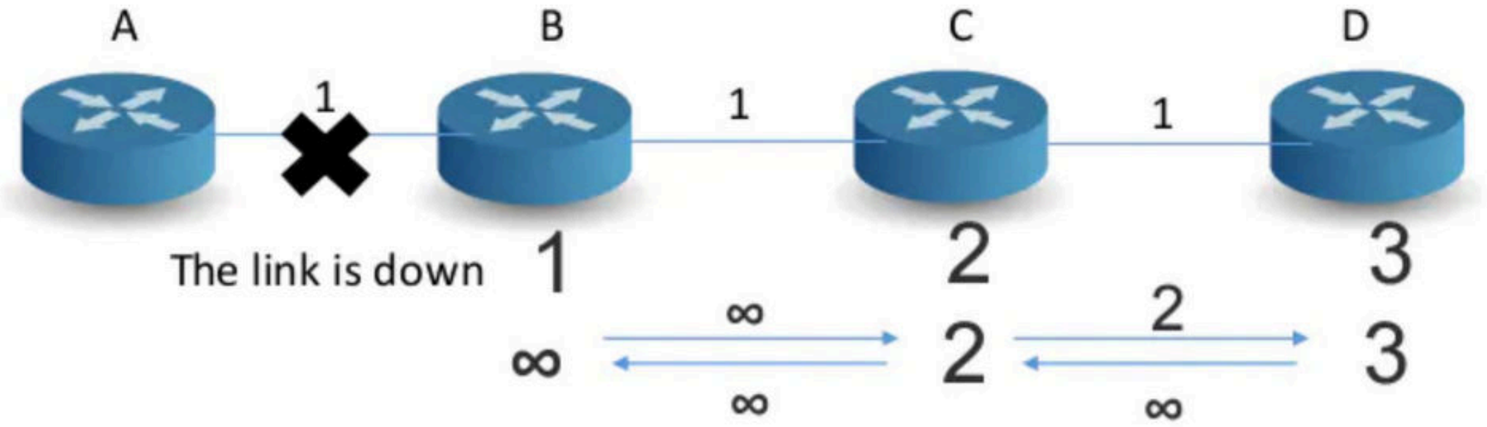
Computer Networks

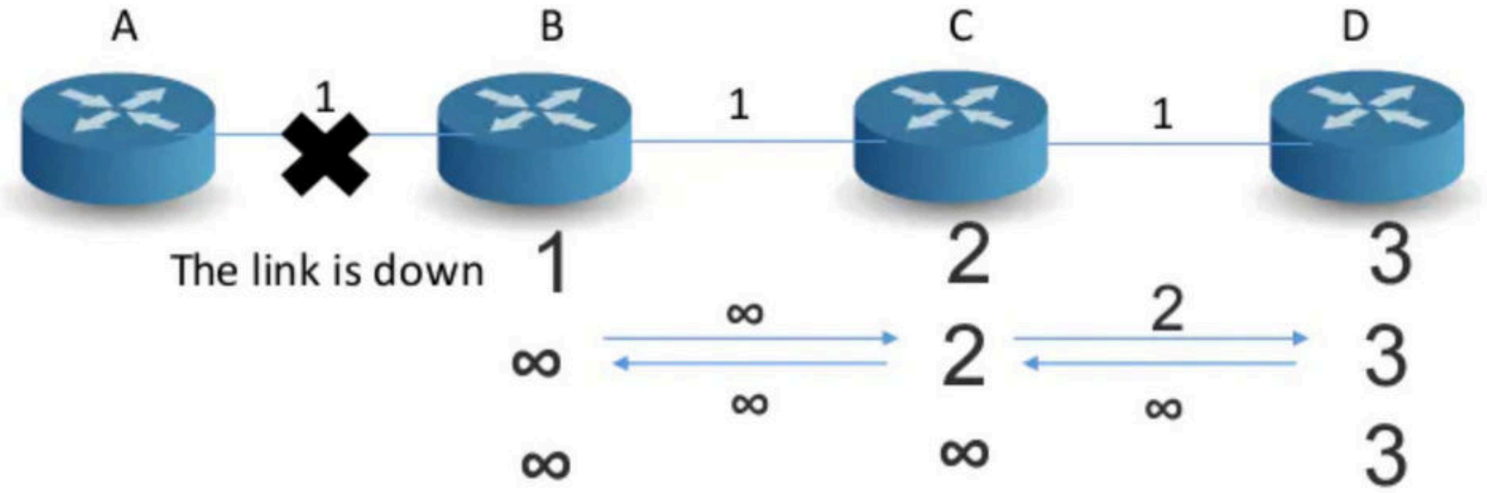
Split Horizon

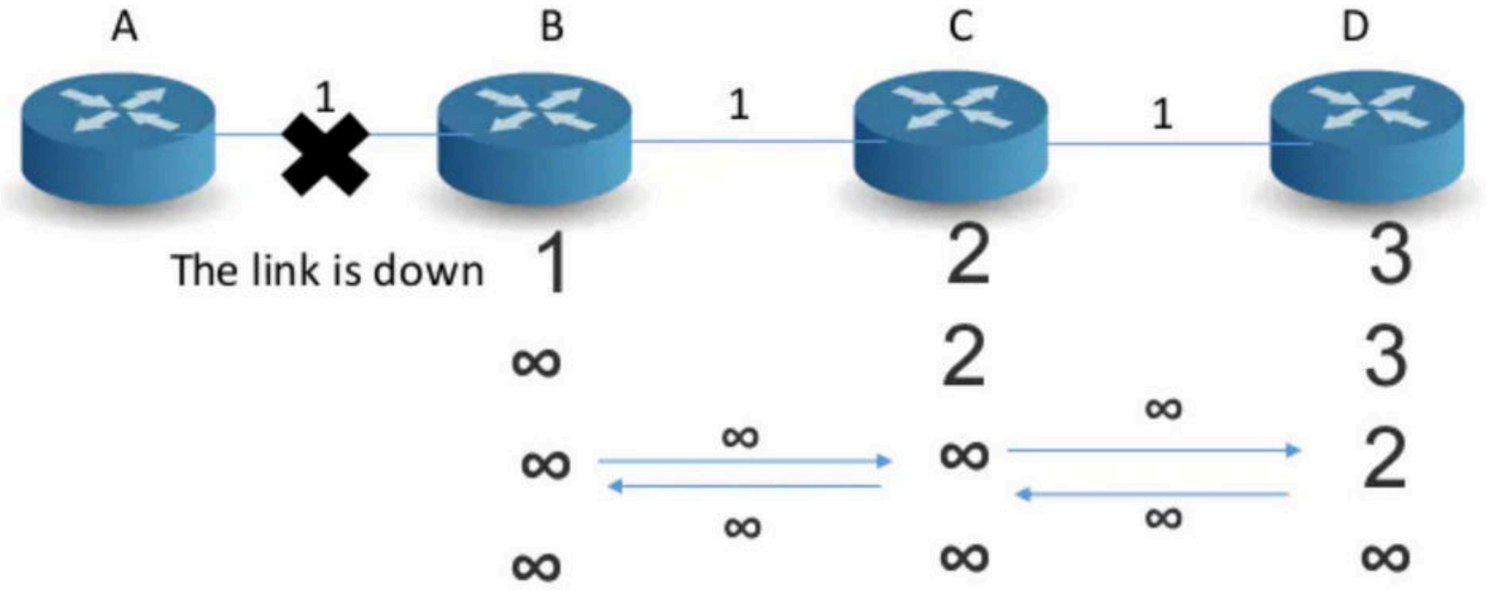
Split Horizon

Solution to Count to Infinity Problem




A
B
C
D
1
1
1
The link is down
1
2
3
∞
2
3
∞
2
∞
∞


A
B
C
D
1
1
1
The link is down
1
∞
∞
∞
2
2
∞
2
3
3
3
2
∞



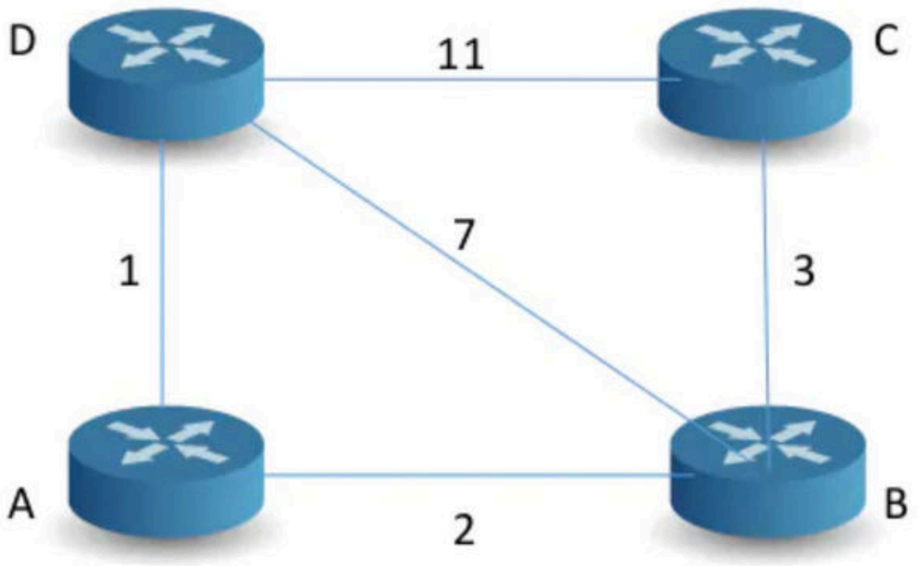
Problem solved

Computer Networks

Link State Routing

D	
C	11
B	7
A	1

C	
D	11
B	3



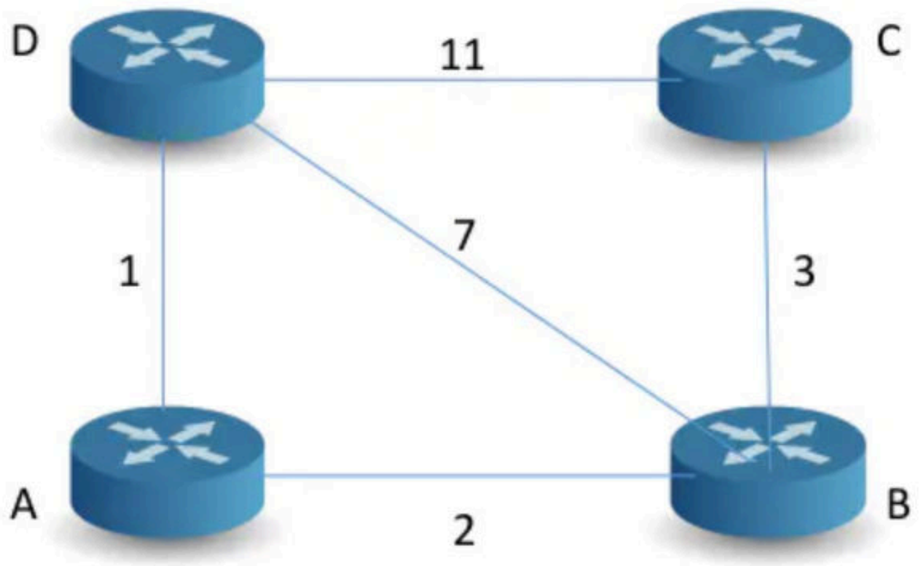
A	
B	2
D	1

B	
A	2
D	7
C	3

Because of the Global Knowledge A has come to know all the distances and
Now it will compute the Single source shortest path which will be done Similarly by the other routers

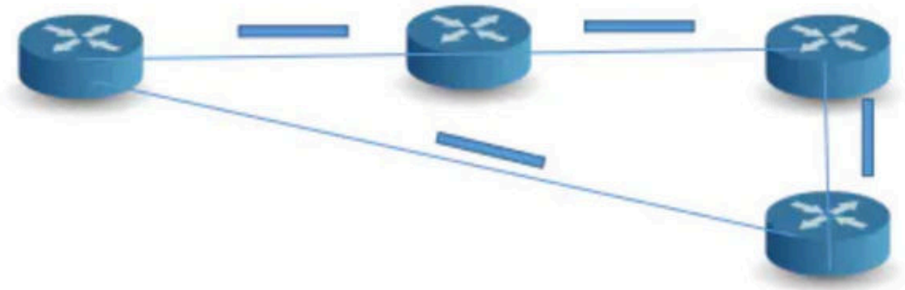
At A

dest	dist	NH
A	0	A
B	2	B
C	5	B
D	1	D



Problems in Link State Routing

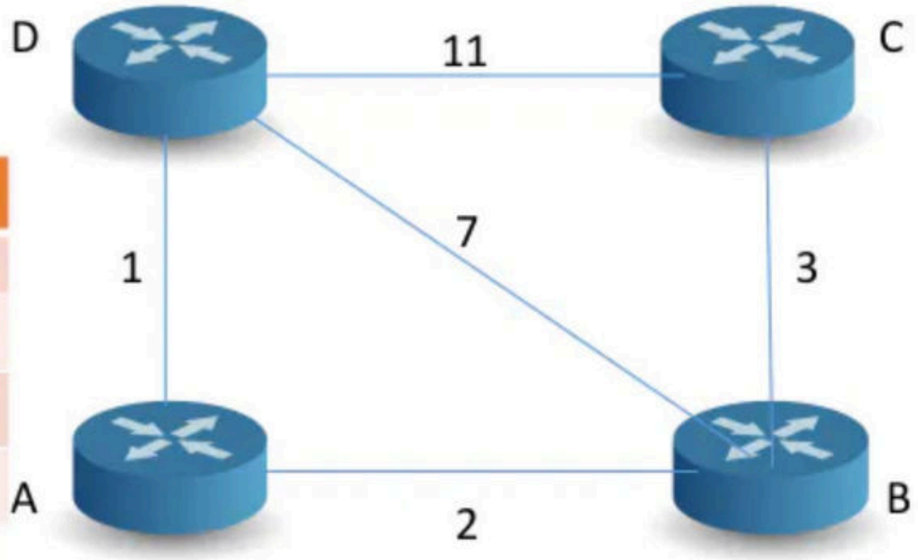
1.) Heavy Traffic



D	
Sequence No	
C	11
B	7
A	1

A	
Sequence No	
B	2
D	1

D	
Node	Seq No
B	10
C	20
D	30

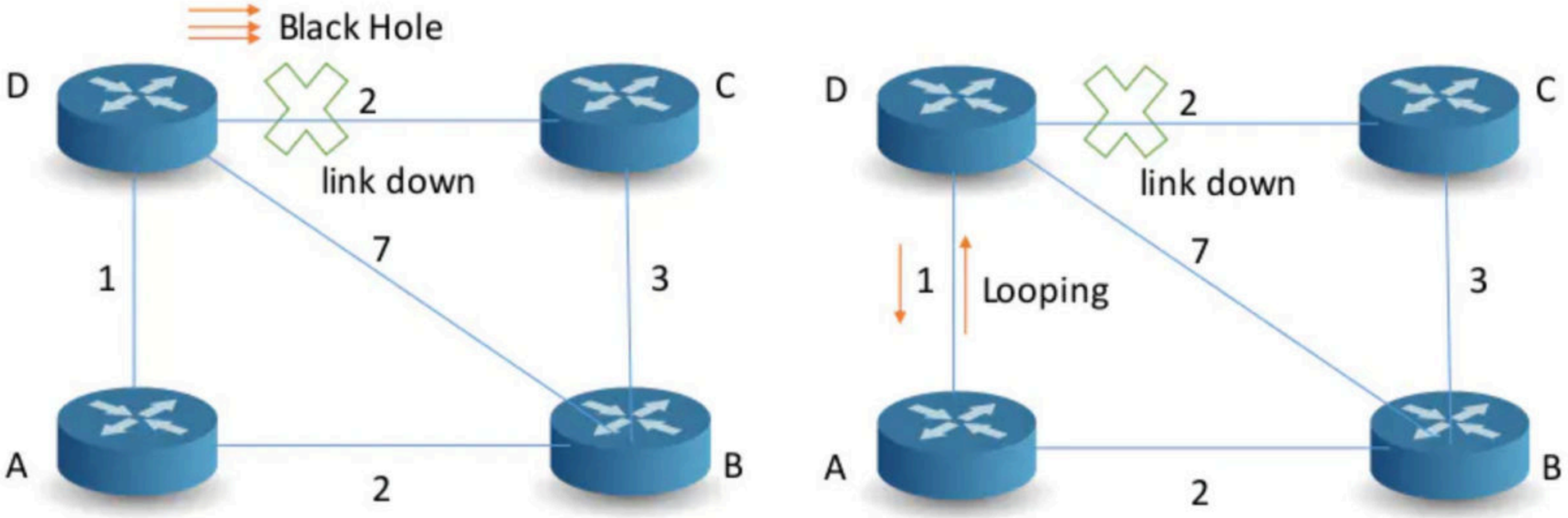


C	
Sequence No	
D	11
B	3

B	
Sequence No	
A	2
D	7
C	3

Every router will maintain a record of the incoming packets and their sequence nos.

2.) Transient Problem



BASIS FOR COMPARISON	DISTANCE VECTOR ROUTING	LINK STATE ROUTING
Algorithm	Bellman ford	Dijsktra
Network view	Topology information from the neighbour point of view	Complete information on the network topology
Best path calculation	Based on the least number of hops	Based on the cost
Updates	Full routing table	Link state updates
Updates frequency	Periodic updates	Triggered updates
CPU and memory	Low utilisation	Intensive
Simplicity	High simplicity	Requires a trained network administrator
Convergence time	Moderate	Fast

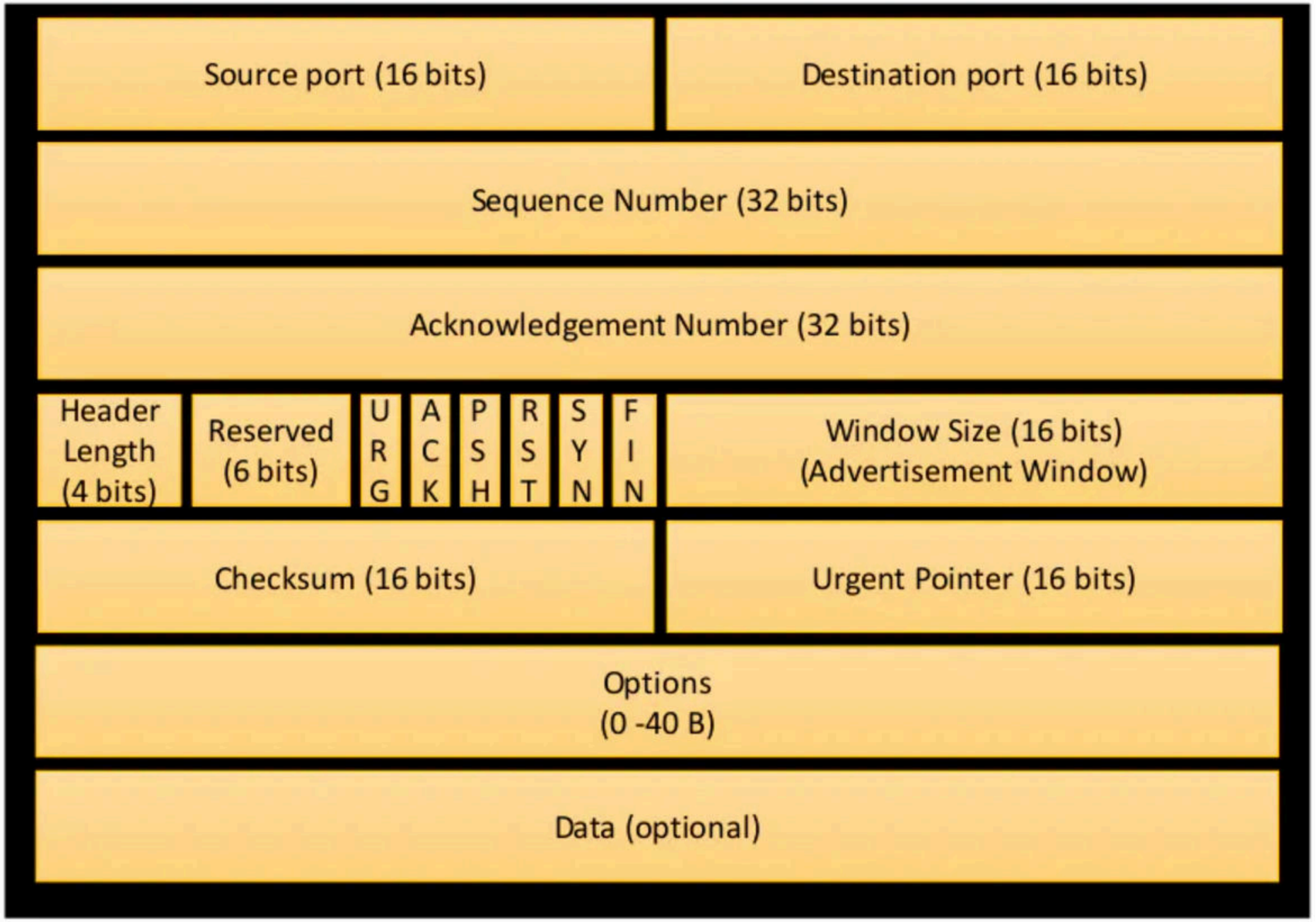
Computer Networks

TCP Header

Source Port, Destination Port,

Sequence Number, Acknowledgment Number

TCP HEADER

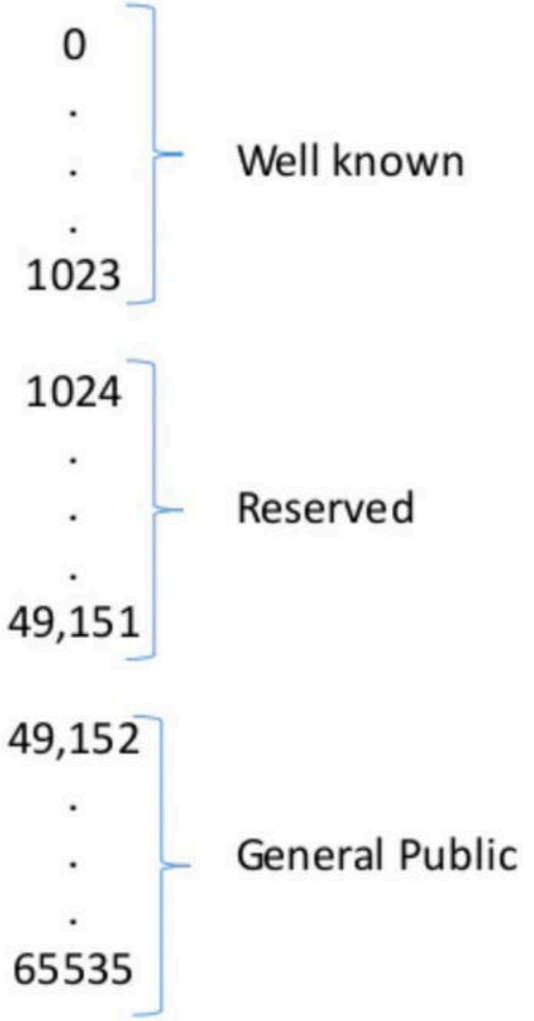

Source port (16 bits)
Destination port (16 bits)
Sequence Number (32 bits)
Acknowledgement Number (32 bits)
Header Length (4 bits)
Reserved (6 bits)
URG
ACK
PSH
RST
SYN
FIN
Window Size (16 bits) (Advertisement Window)
Checksum (16 bits)
Urgent Pointer (16 bits)
Options (0 -40 B)
Data (optional)

TCP and Port Numbers

- End to End Protocol. It is byte stream protocol
- Multiplexing and Demultiplexing
- TCP is a connection oriented protocol.

Port Number	Usage
20	File Transfer Protocol (FTP) Data Transfer
21	File Transfer Protocol (FTP) Command Control
22	Secure Shell (SSH)
23	Telnet - Remote login service, unencrypted text messages
25	Simple Mail Transfer Protocol (SMTP) E-mail Routing
53	Domain Name System (DNS) service
80	Hypertext Transfer Protocol (HTTP) used in World Wide Web
110	Post Office Protocol (POP3) used by e-mail clients to retrieve e-mail from a server
119	Network News Transfer Protocol (NNTP)
123	Network Time Protocol (NTP)
143	Internet Message Access Protocol (IMAP) Management of Digital Mail
161	Simple Network Management Protocol (SNMP)
194	Internet Relay Chat (IRC)
443	HTTP Secure (HTTPS) HTTP over TLS/SSL

Range $(0, 2^{16-1})$



1. Source Port-

- Source Port is a 16 bit field.
- It identifies the port of the sending application.

2. Destination Port-

- Destination Port is a 16 bit field.
- It identifies the port of the receiving application.

It is important to note-

- A TCP connection is uniquely identified by using-
  Combination of port numbers and IP Addresses of sender and receiver
- IP Addresses indicate which systems are communicating.
- Port numbers indicate which end to end sockets are communicating.

3. Sequence Number-

•Sequence number is a 32 bit field.
•TCP assigns a unique sequence number to each byte of data contained in the TCP segment.
•This field contains the sequence number of the first data byte.

4. Acknowledgement Number-

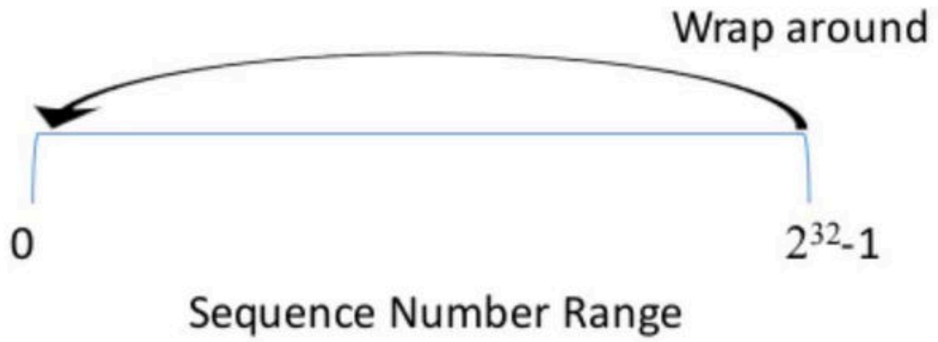
•Acknowledgment number is a 32 bit field.
•It contains sequence number of the data byte that receiver expects to receive next from the sender.
•It is always sequence number of the last received data byte incremented by 1.

Computer Networks

TCP Header Part 2

Wrap around time

Wrap Around Time



After all the 2^{32} sequence numbers are used up and more data is to be sent, the sequence numbers can be wrapped around and used again from the starting.

In general,

- If the initial sequence number chosen is X.
- Then sequence numbers are used from X to $2^{32} - 1$ and then from to 0 to X-1.
- Then, sequence numbers are wrapped around to send more data.

Example-

- Consider the initial sequence number used is 0.
- Then after sending 4 GB data, all the sequence numbers would get used up.
- To send more data, sequence numbers are reused from the starting.
- Wrapping around can be done again and again to send more and more data.

Wrap Around Time-

- Time taken to use up all the 2^{32} sequence numbers is called as **wrap around time**.
- It depends on the bandwidth of the network i.e. the rate at which the bytes go out.
- More the bandwidth, lesser the wrap around time and vice versa.

$$\text{Wrap Around Time} \propto 1 / \text{Bandwidth}$$

Formula-

If bandwidth of the network = x bytes/sec, then

$$\text{Wrap Around Time} = \frac{2^{32}}{x} \text{ sec}$$

Life Time Of TCP Segment-

In modern computers,

- Life time of a TCP segment is 180 seconds or 3 minutes.
- It means after sending a TCP segment, it might reach the receiver taking 3 minutes in the worst case.

How Wrap Around Is Possible?

It is possible to wrap around the sequence numbers because-

- The life time of a TCP segment is just 180 seconds.
- Wrap around time is much greater than life time of a TCP segment.
- So, by the time the sequence numbers wrap around, there is no probability of existing any segment having the same sequence number.
- Thus, even after wrapping around, the sequence number of all the bytes will be unique at any given time.

Reducing Wrap Around Time-

Wrap around time can be reduced to the life time of a TCP segment.

To reduce the wrap around time to the life time of segment,
•There must exist as many sequence numbers as there are number of data bytes sent in time equal to life time of segment.

Number of bits required in the sequence number field so that wrap around time becomes equal to lifetime of TCP segment
$= \log_2$ (lifetime of TCP segment x Bandwidth)

•The number of bits will be greater than 32 bits.
•The extra bits are appended in the Options field of TCP header.

Computer Networks

TCP header Part 3

Header Length and TCP Connection establishment

5. Header Length-

- Header length is a 4 bit field.
- It contains the length of TCP header.
- It helps in knowing from where the actual data begins.

Minimum and Maximum Header length-

The length of TCP header always lies in the range-

[20 bytes , 60 bytes]

- The initial 5 rows of the TCP header are always used.
- So, minimum length of TCP header = 5 x 4 bytes = 20 bytes.
- The size of the 6th row representing the Options field vary.
- The size of Options field can go up to 40 bytes.
- So, maximum length of TCP header = 20 bytes + 40 bytes = 60 bytes.

Concept of Scaling Factor-

- Header length is a 4 bit field.
- So, the range of decimal values that can be represented is [0, 15].
- But the range of header length is [20, 60].
- So, to represent the header length, we use a scaling factor of 4.

In general,

Header length = Header length field value x 4 bytes

Examples-

- If header length field contains decimal value 5 (represented as 0101), then-

Header length = 5 x 4 = 20 bytes

- If header length field contains decimal value 10 (represented as 1010), then-

Header length = 10 x 4 = 40 bytes

- If header length field contains decimal value 15 (represented as 1111), then-

Header length = 15 x 4 = 60 bytes

NOTES

It is important to note-

- Header length and Header length field value are two different things.
- The range of header length field value is always [5, 15].
- The range of header length is always [20, 60].

While solving questions-

- If the given value lies in the range [5, 15] then it must be the header length field value.
- This is because the range of header length is always [20, 60].

We will check out the other fields and flags of the TCP header along with TCP phases

TCP PHASES

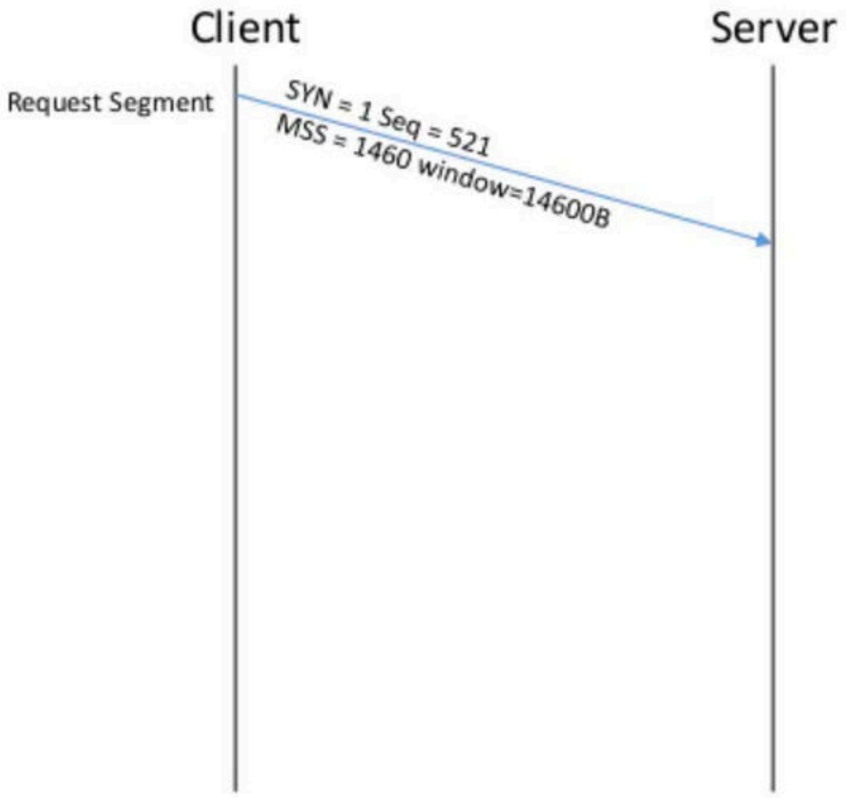
CONNECTION ESTABLISHMENT

DATA TRANSMISSION

CONNECTION TERMINATION

CONNECTION ESTABLISHMENT

Three Way Handshake is a process used for establishing a TCP connection.



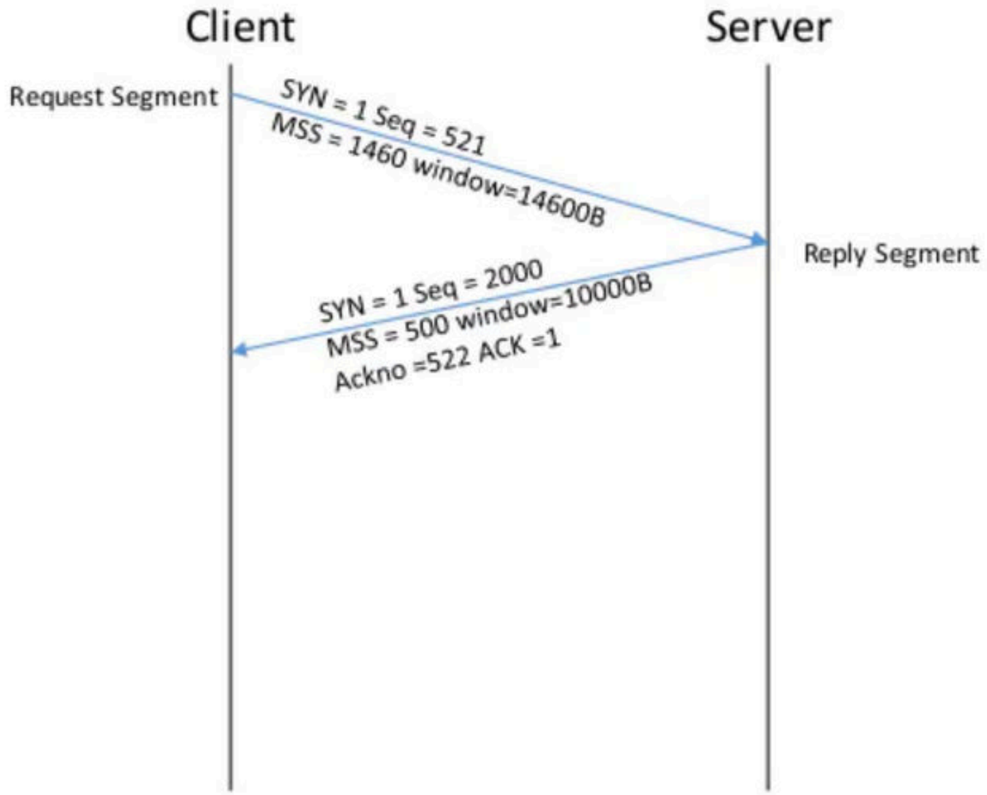
Imagine this scenario for connection Establishment

Step-01: SYN-

For establishing a connection,
•Client sends a request segment to the server.
•Request segment consists only of TCP Header with an empty payload.
•Then, it waits for a reply segment from the server.

Request segment contains the following information in TCP header-
1.Initial sequence number
2.SYN bit set to 1
3.Maximum segment size
4.Receiving window size

Three Way Handshake is a process used for establishing a TCP connection.



Imagine this scenario for connection Establishment

Step-02: SYN + ACK-

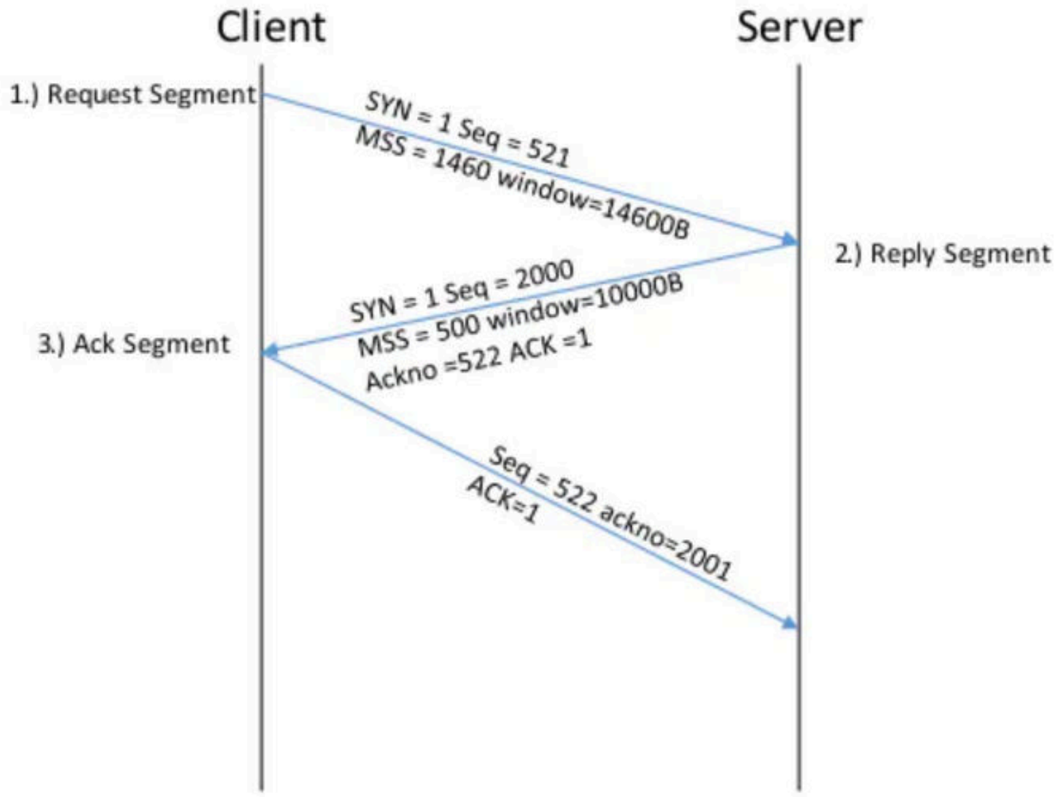
After receiving the request segment,

- Server responds to the client by sending the reply segment.
- It informs the client of the parameters at the server side.

Reply segment contains the following information in TCP header-

1. Initial sequence number
2. SYN bit set to 1
3. Maximum segment size
4. Receiving window size
5. Acknowledgment number
6. ACK bit set to 1

Three Way Handshake is a process used for establishing a TCP connection.



Imagine this scenario for connection Establishment

Step-03: ACK-

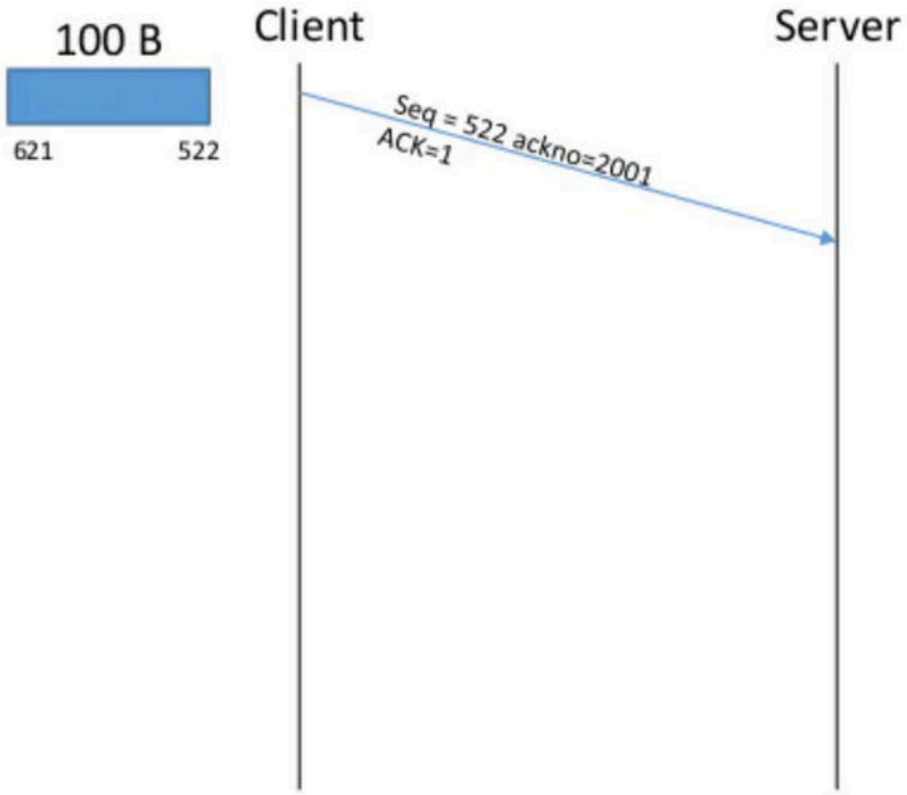
After receiving the reply segment,

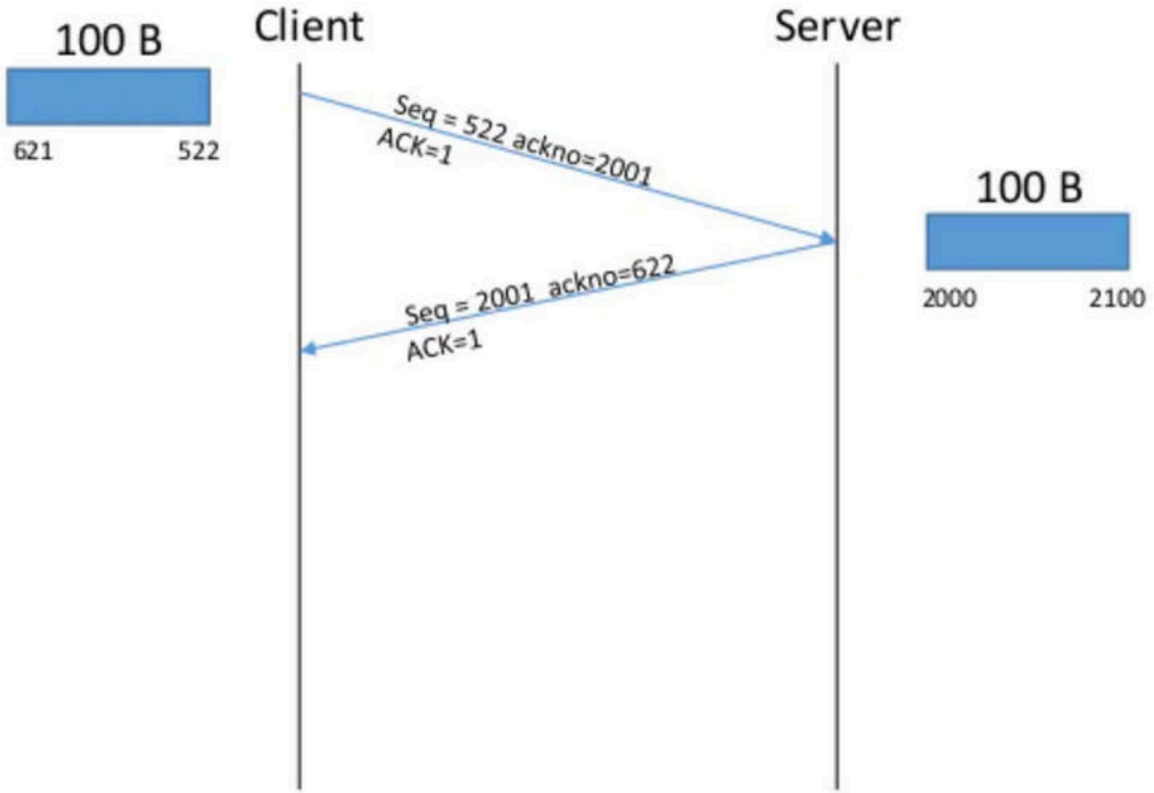
- Client acknowledges the response of server.
- It acknowledges the server by sending a pure acknowledgement.

In any TCP segment,

1. If SYN bit = 1 and ACK bit = 0, then it must be the request segment.
2. If SYN bit = 1 and ACK bit = 1, then it must be the reply segment.
3. If SYN bit = 0 and ACK bit = 1, then it can be the pure ACK or segment meant for data transfer.
4. If SYN bit = 0 and ACK bit = 0, then this combination is not possible.

DATA TRANSMISSION



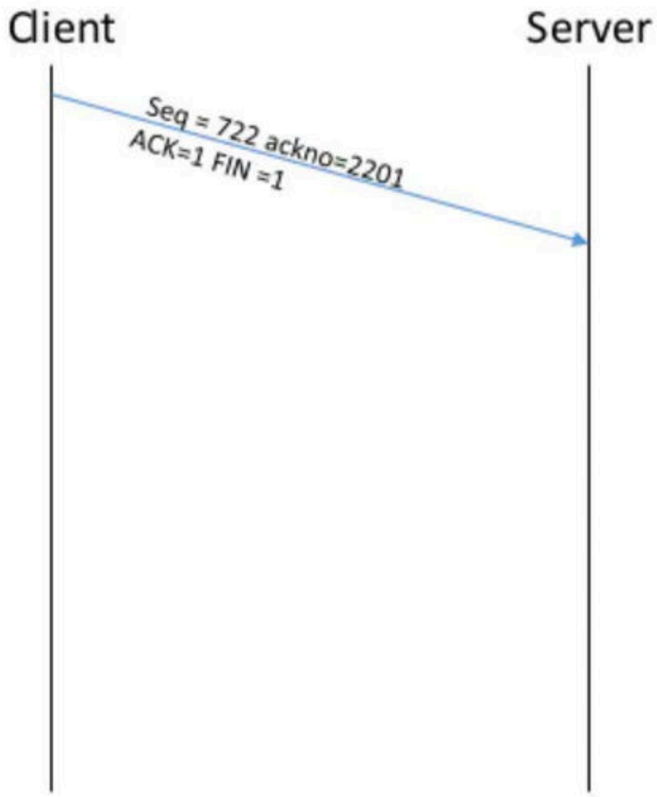

100 B
621
522
Client
Server
Seq = 522 ackno=2001
ACK=1
Seq = 2001 ackno=622
ACK=1
100 B
2000
2100


100 B
621
522
Client
Server
Seq = 522 ackno=2001
ACK=1
100 B
2000
2100
Seq = 2001 ackno=622
ACK=1
Seq = 622 ackno=2101
ACK=1


100 B
621
522
Client
Server
Seq = 522 ackno=2001
ACK=1
100 B
2000
2100
Seq = 2001 ackno=622
ACK=1
Seq = 622 ackno=2101
ACK=1
100 B
721
622
Seq = 622 ackno=2101
ACK=1
100 B
2101
2200
Seq = 2101 ackno=722
ACK=1
Seq = 722 ackno=2201
ACK=1

CONNECTION TERMINATION

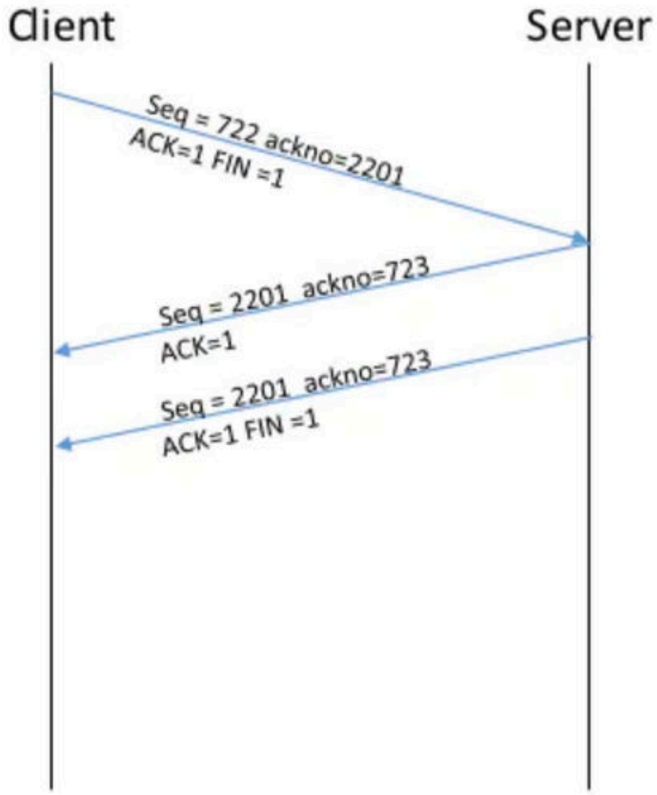
Consider-
There is a well established TCP connection between the client and server.
Client wants to terminate the connection.



Step-01:

For terminating the connection,
•Client sends a FIN segment to the server with FIN bit set to 1.
•Client enters the **FIN_WAIT_1 state**.
•Client waits for an acknowledgement from the server.

CONNECTION TERMINATION



Step-02:

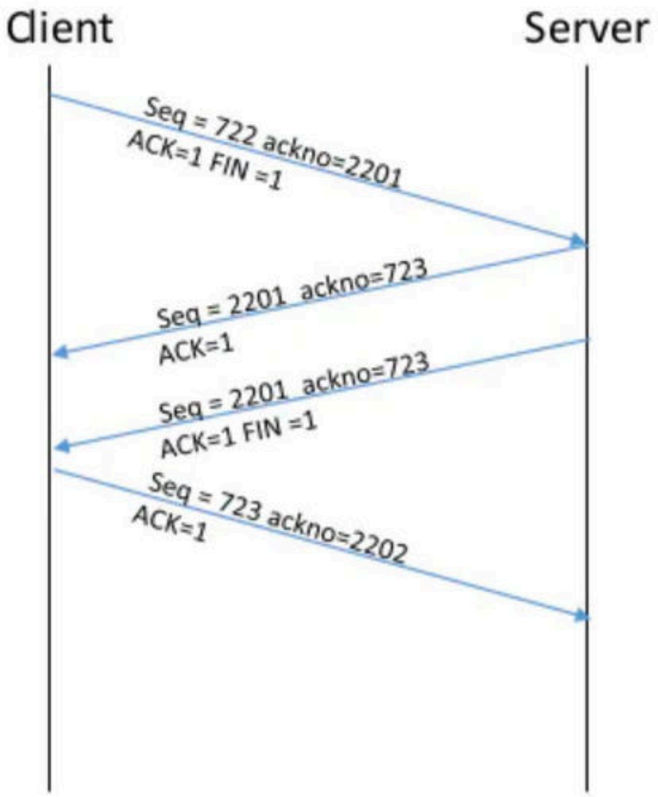
After receiving the FIN segment,
- Server frees up its buffers.
- Server sends an acknowledgement to the client.
- Server enters the CLOSE_WAIT state.

Step-03:

After receiving the acknowledgement, client enters the FIN_WAIT_2 state.

For terminating the connection,
- Server sends a FIN segment to the client with FIN bit set to 1.
- Server waits for an acknowledgement from the client.

CONNECTION TERMINATION



Step-04:

After receiving the FIN segment,

- Client frees up its buffers.
- Client sends an acknowledgement to the server (not mandatory).
- Client enters the TIME_WAIT state.

Computer Networks

TCP header Part 4

PSH Bit-

PSH bit is used to push the entire buffer immediately to the receiving application.

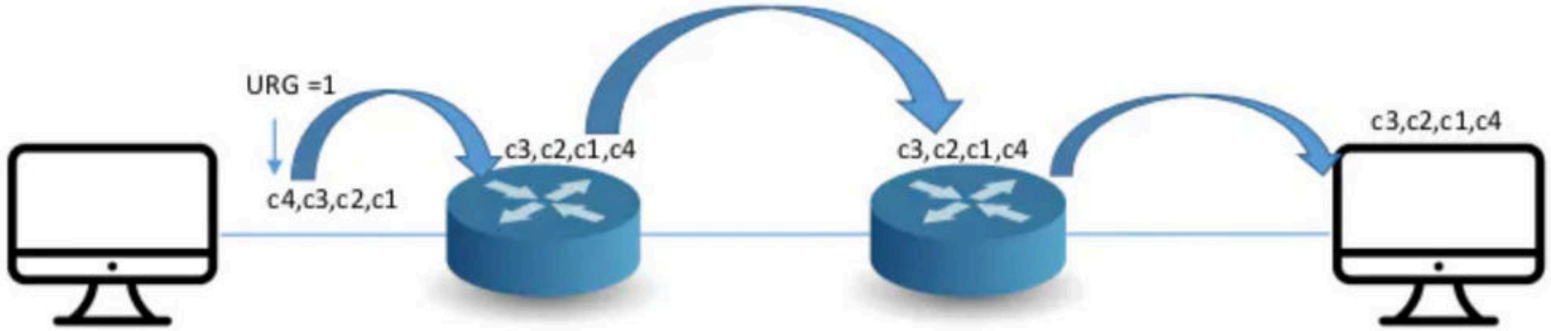
When PSH bit is set to 1,
- All the segments in the buffer are immediately pushed to the receiving application.
- No wait is done for filling the entire buffer.
- This makes the entire buffer to free up immediately.

Useful in case of chat applications

URG Bit-

URG bit is used to treat certain data on an urgent basis.

When URG bit is set to 1,
It indicates the receiver that certain amount of data within the current segment is urgent.



Urgent Pointer-

- Urgent pointer is a 16 bit field.
- It indicates how much data in the current segment counting from the first data byte is urgent.
- Urgent pointer added to the sequence number indicates the end of urgent data byte.
- This field is considered valid and evaluated only if the URG bit is set to 1.

Number of urgent bytes = Urgent pointer + 1

End of urgent byte
= Sequence number of the first byte in the segment + Urgent pointer

RST Bit-

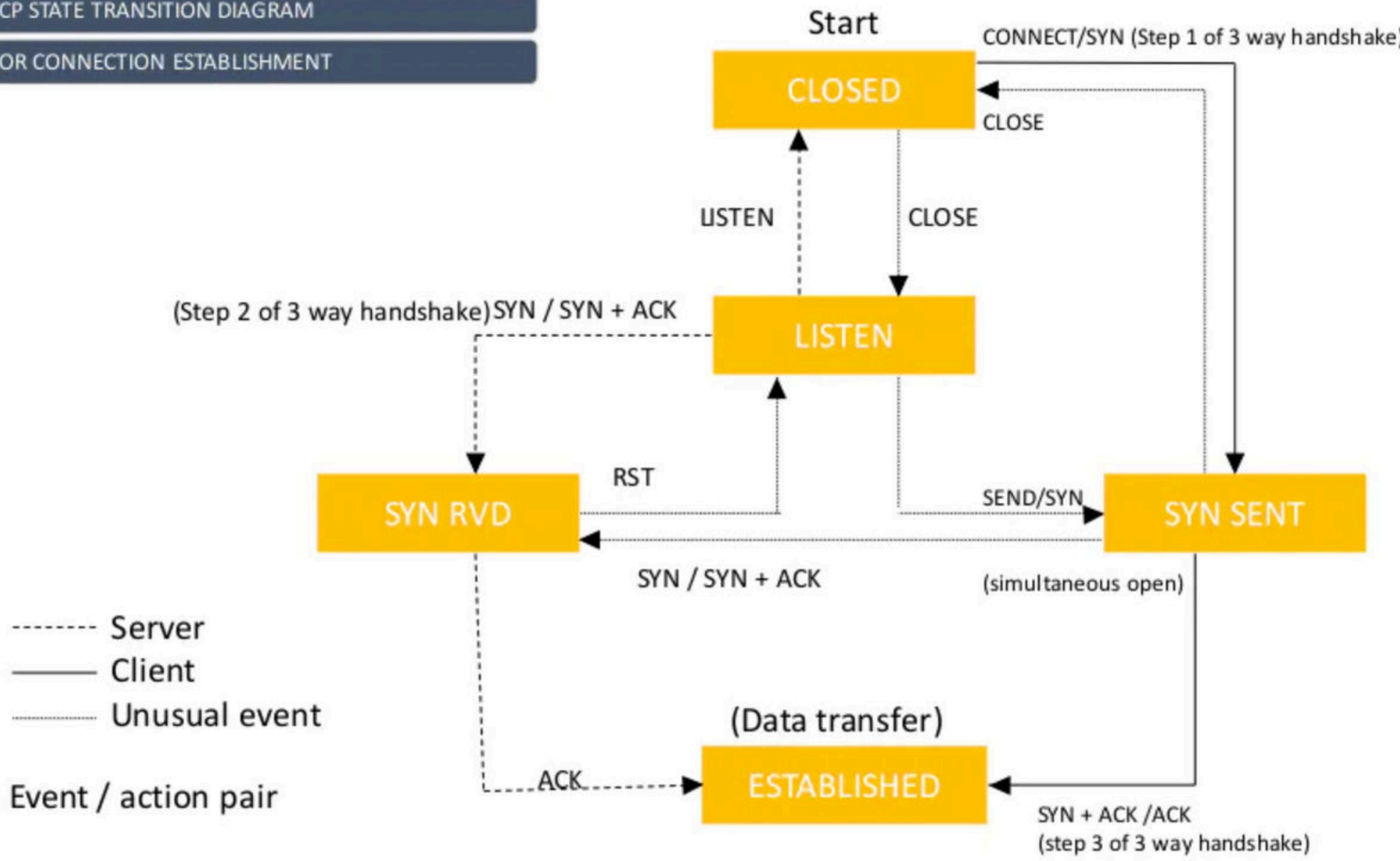
RST bit is used to reset the TCP connection.

When RST bit is set to 1,
- It indicates the receiver to terminate the connection immediately.
- It causes both the sides to release the connection and all its resources abnormally.
- The transfer of data ceases in both the directions.
- It may result in the loss of data that is in transit.

This is used only when-
- There are unrecoverable errors.
- There is no chance of terminating the TCP connection normally.

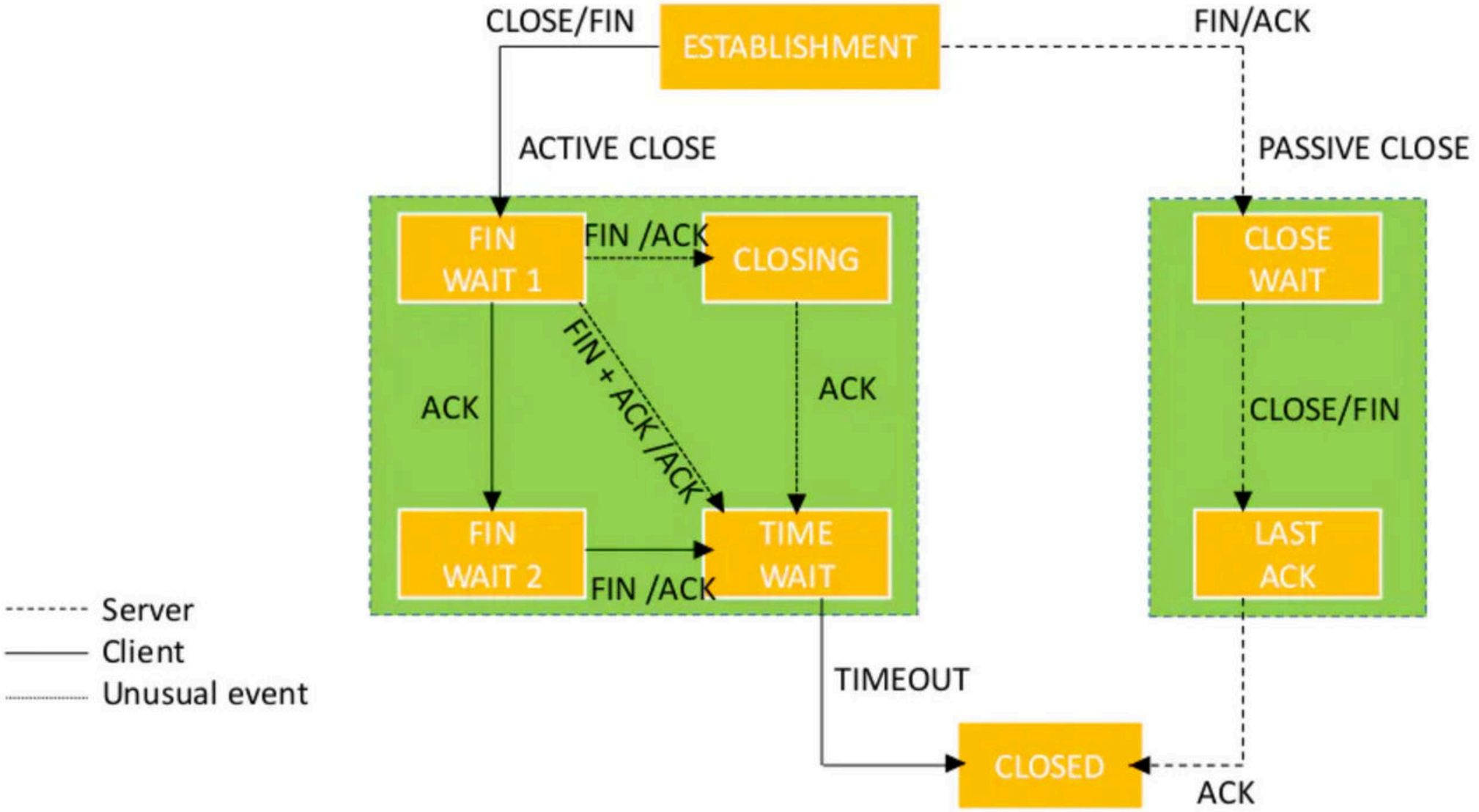
TCP STATE TRANSITION DIAGRAM

FOR CONNECTION ESTABLISHMENT



TCP STATE TRANSITION DIAGRAM

FOR CONNECTION TERMINATION

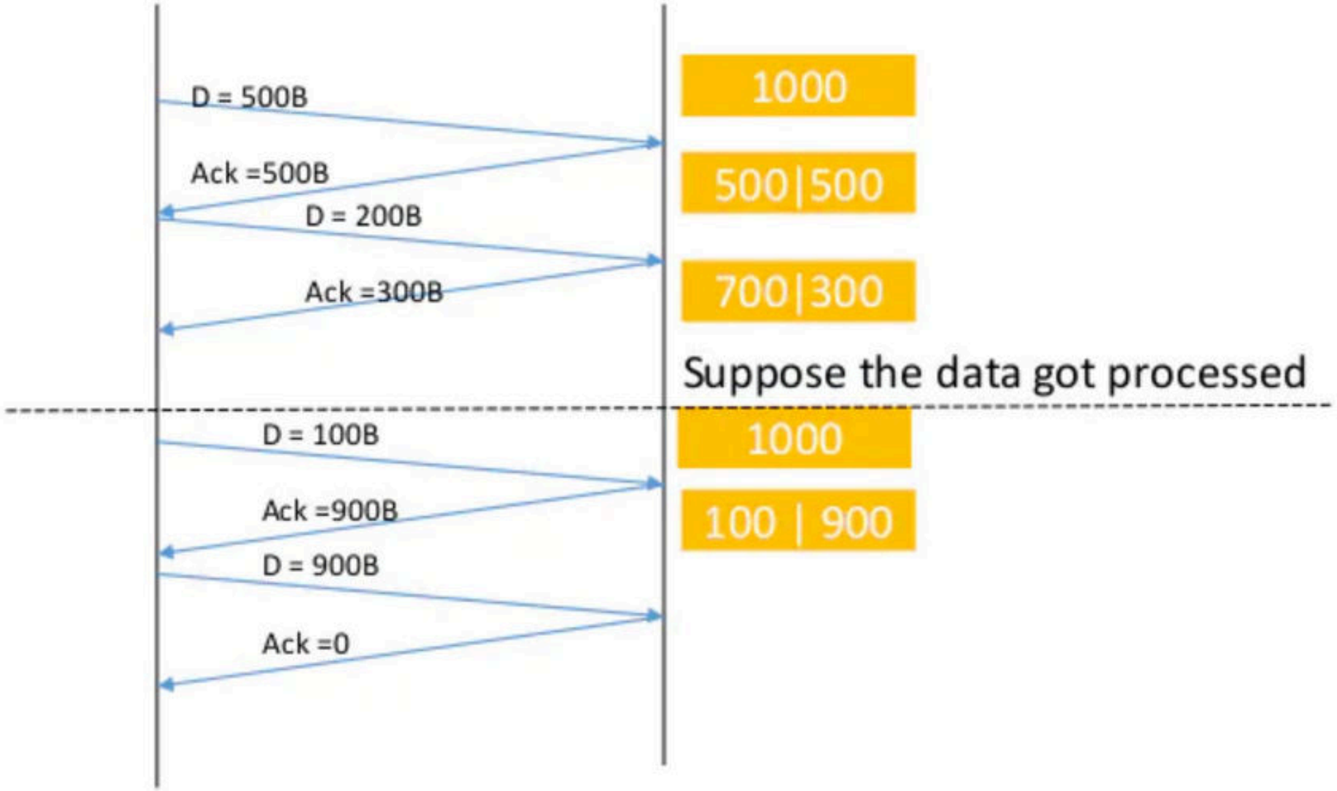


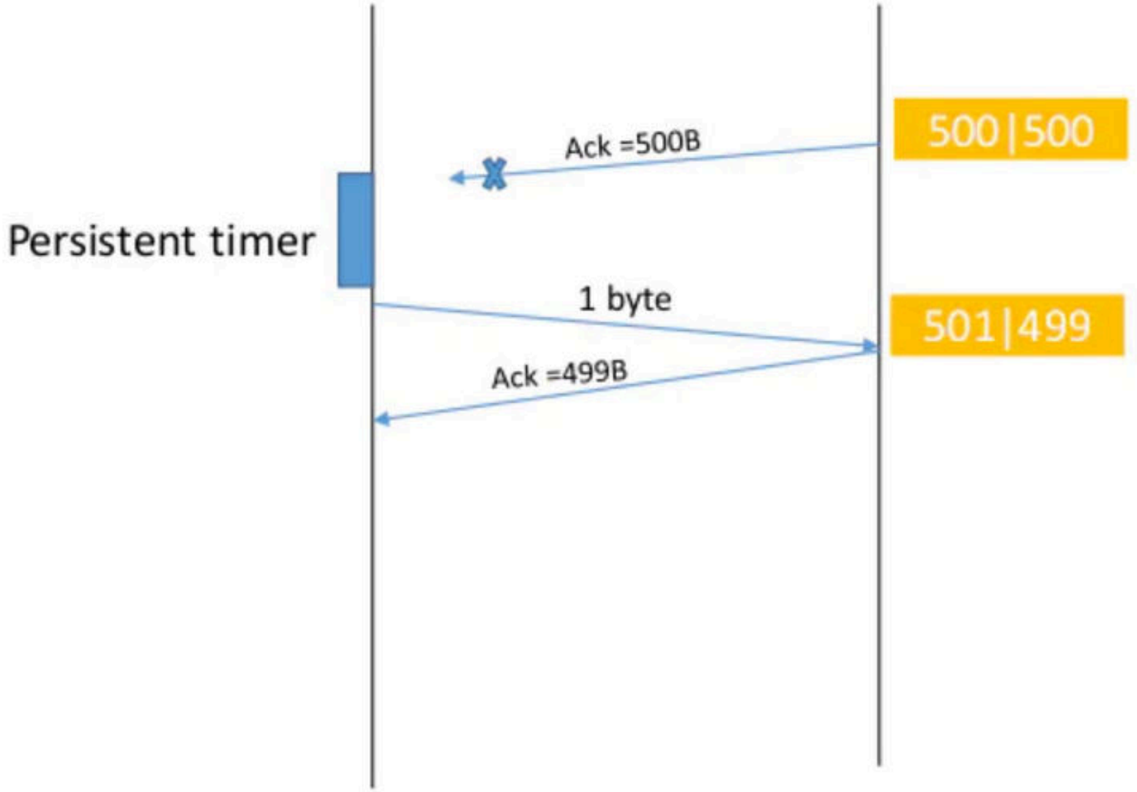
Computer Networks

TCP header part 5

Window Size

- Window size is a 16 bit field.
- It contains the size of the receiving window of the sender.
- It advertises how much data (in bytes) the sender can receive without acknowledgement.
- Thus, window size is used for **Flow Control**.





Consider the following situation-
Sender receives an acknowledgment from the receiver with zero window size.
This indicates the sender to wait.
Later, receiver updates the window size to 500B and sends the segment with the update to the sender.
This segment gets lost.
Now, both sender and receiver keeps waiting for each other to do something.
To deal with such a situation, TCP uses a persistent timer.

Checksum

Checksum is a 16 bit field used for error control.
It verifies the integrity of data in the TCP payload.
Sender adds CRC checksum to the checksum field before sending the data.
Receiver rejects the data that fails the CRC check.

Understanding with an example

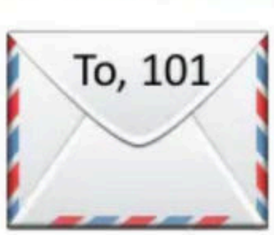
Got damaged
While transporting

Letter delivered to wrong house

Checksum

Checksum is a 16 bit field used for error control.
It verifies the integrity of data in the TCP payload.
Sender adds CRC checksum to the checksum field before sending the data.
Receiver rejects the data that fails the CRC check.

Writing the delivery address
Inside the letter too



Options

Options field is used for several purposes.
The size of options field vary from 0 bytes to 40 bytes.

Options field is generally used for the following purposes-

1. Time stamp
2. Window size extension
3. Parameter negotiation
4. Padding

Time Stamp

When wrap around time is less than life time of a segment,
Multiple segments having the same sequence number may appear at the receiver side.
This makes it difficult for the receiver to identify the correct segment.
If time stamp is used, it marks the age of TCP segments.
Based on the time stamp, receiver can identify the correct segment

Window Size Extension

Options field may be used to represent a window size greater than 16 bits.
Using window size field of TCP header, window size of only 16 bits can be represented.
If the receiver wants to receive more data, it can advertise its greater window size using this field.
The extra bits are then appended in Options field.

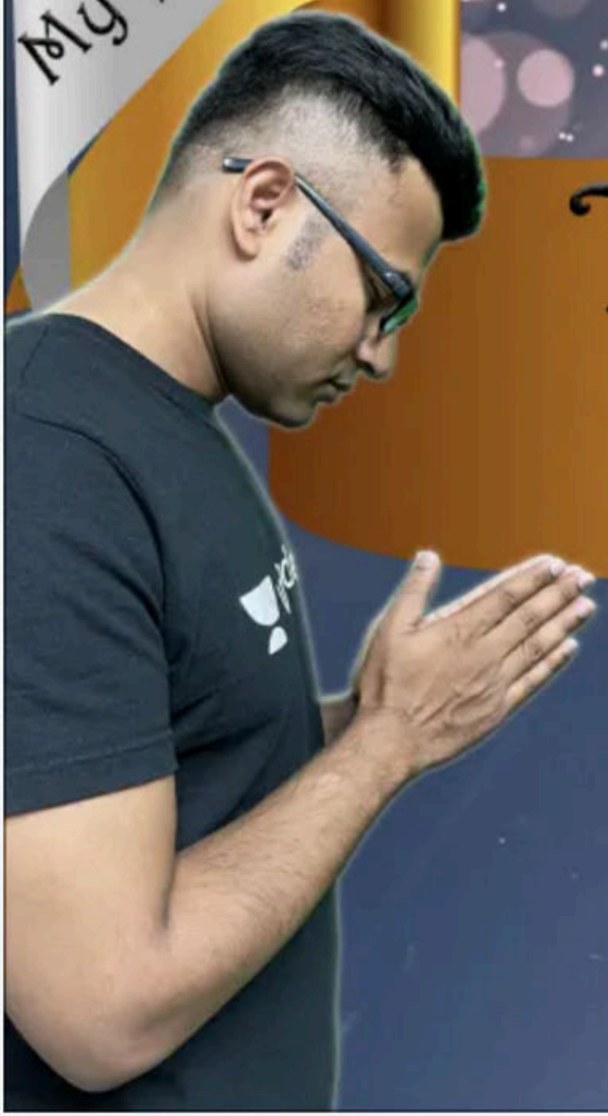
Parameter Negotiation

Options field is used for parameters negotiation.
Example- During connection establishment,
Both sender and receiver have to specify their maximum segment size.
To specify maximum segment size, there is no special field.
So, they specify their maximum segment size using this field and negotiates.

Padding

Addition of dummy data to fill up unused space in the transmission unit and make it conform to the standard size is called as padding.

My philosophy



TEACHING IS WORSHIP

STUDENTS ARE GODS