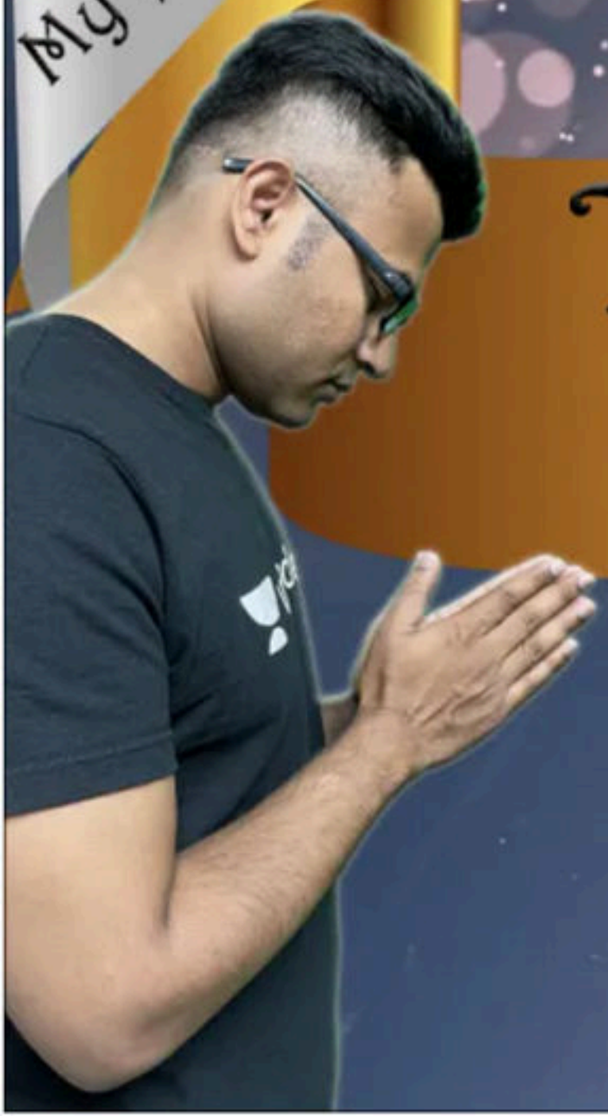
Doubt Clearing Session

Complete Course on Computer Networks - Part IV

Ravindrababu Ravula • Lesson 3 • May 6, 2021

My philosophy



TEACHING IS WORSHIP

STUDENTS ARE GODS

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

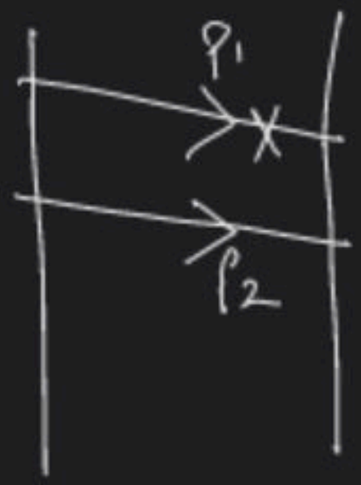
Options field is used for parameters negotiation.
Example- During connection establishment,
Both sender and receiver have to specify their maximum segment size.
To specify maximum segment size, there is no special field.
So, they specify their maximum segment size using this field and negotiates.

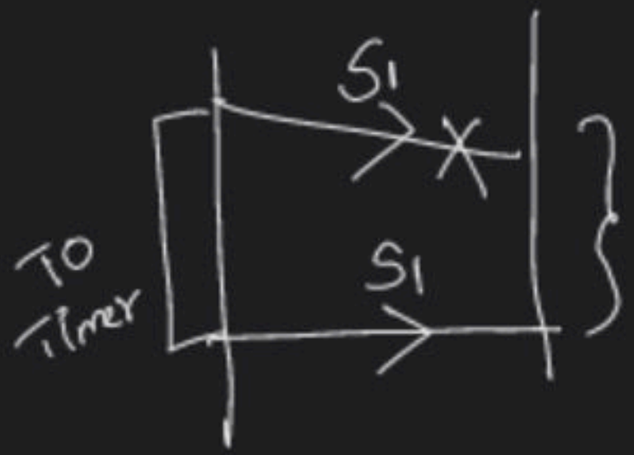
Padding

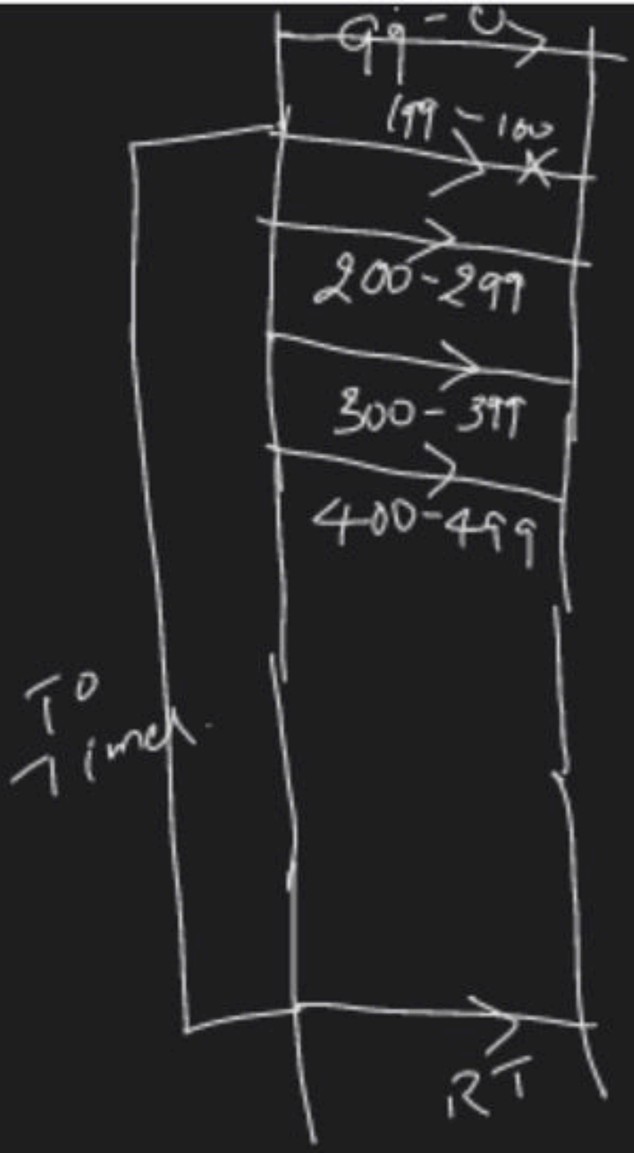
Addition of dummy data to fill up unused space in the transmission unit and make it conform to the standard size is called as padding.
Options field is used for padding.

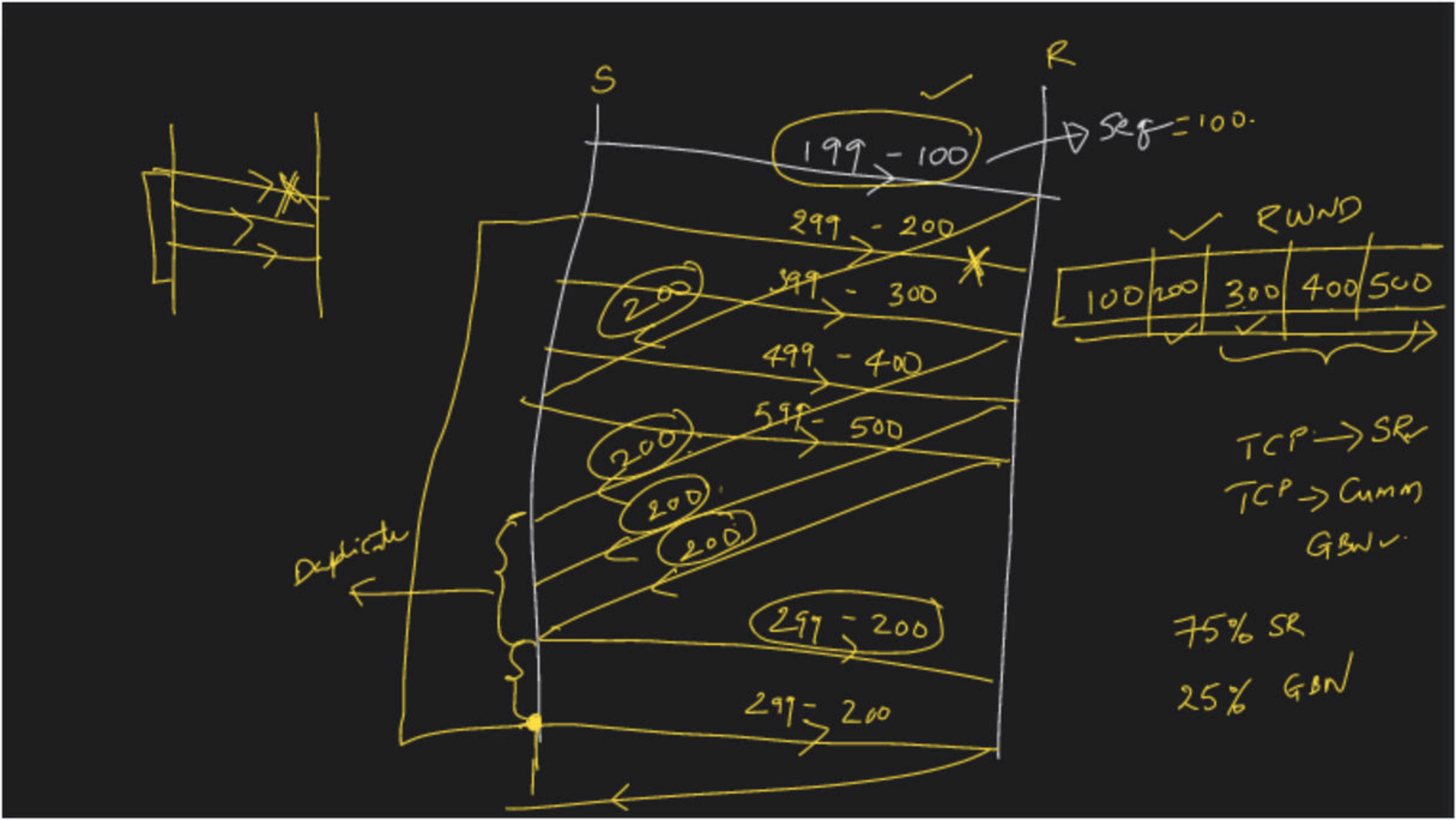
Computer Networks

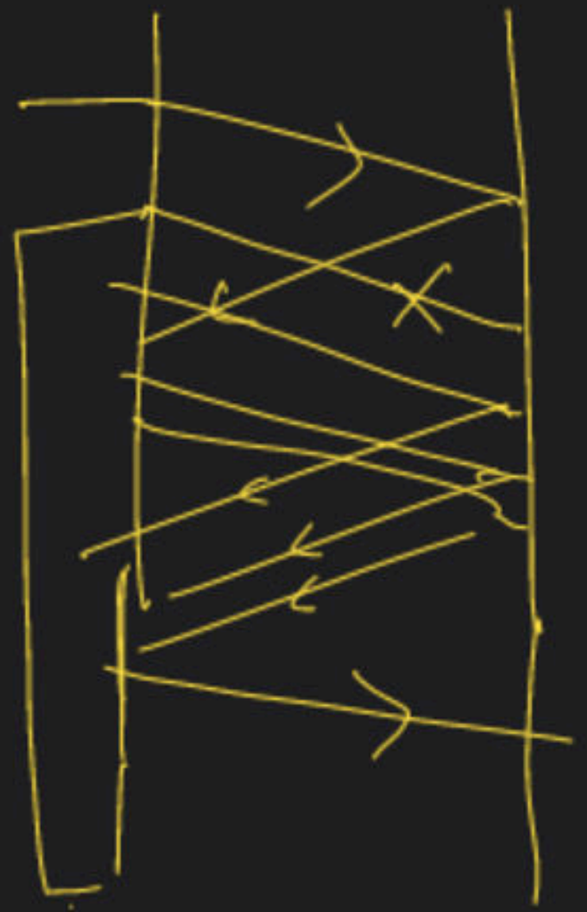
Retransmission in TCP

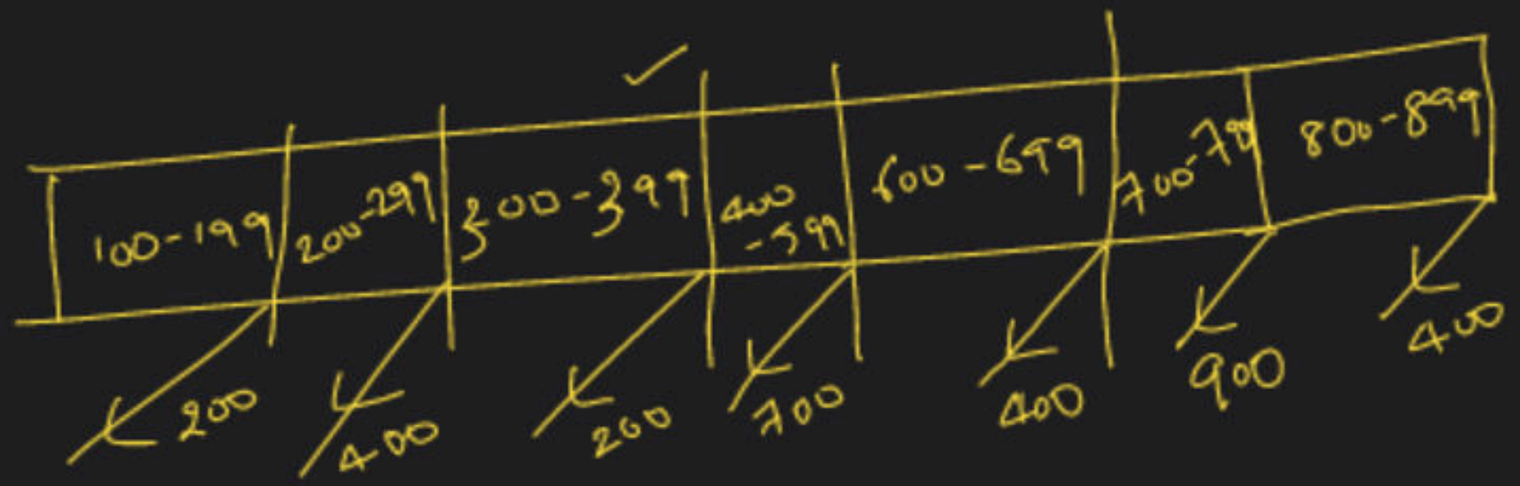

P1
P2

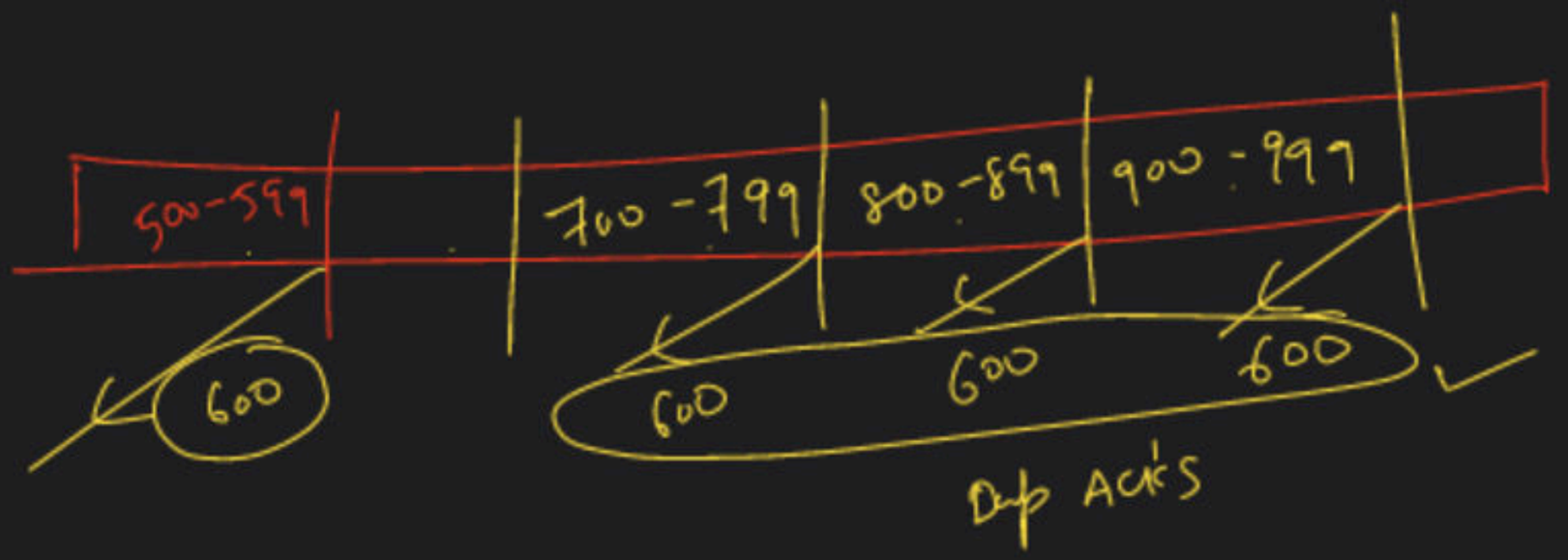

S1
S1
TO
Timer


199 - 100
200-299
300-399
400-499
TO
Timer
RT


S
R
199 - 100
Seq = 100.
299 - 200
399 - 300
499 - 400
599 - 500
200
200
200
200
Duplicate
299 - 200
299 - 200
RWND
100
200
300
400
500
TCP → SR
TCP → Cumm
GBN
75% SR
25% GBN




100-199
200-299
300-399
400-599
600-699
700-799
800-899
200
400
200
700
400
900
400


500-599
700-799
800-899
900-997
600
600
600
600
Dup Acks

Re-transmission in TCP

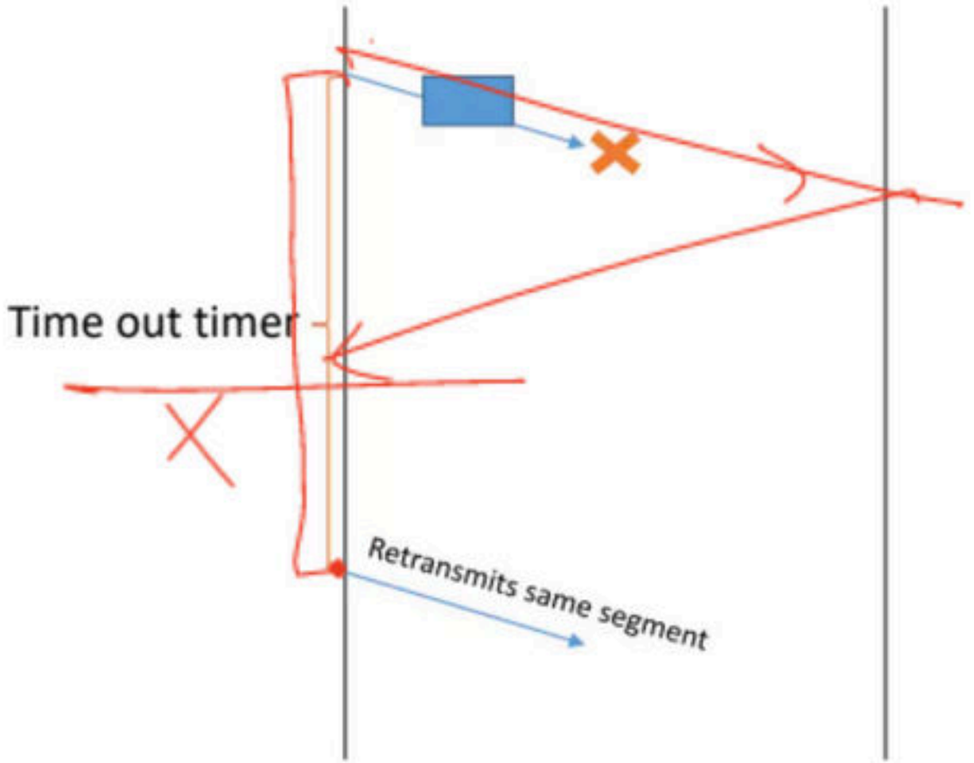
Sender discovers that the TCP segment is lost when-

1. Either Time Out Timer expires
2. Or it receives three duplicate acknowledgements

TCP uses both SR and GBN -
SR has Ws=Wr and out of order delivery
GBN has cumulative ACK

TCP is 75 % SR and 25 % GBN

Time out timer expires



Each time sender transmits a TCP segment to the receiver, it starts a Time Out Timer.
Now, following two cases are possible-

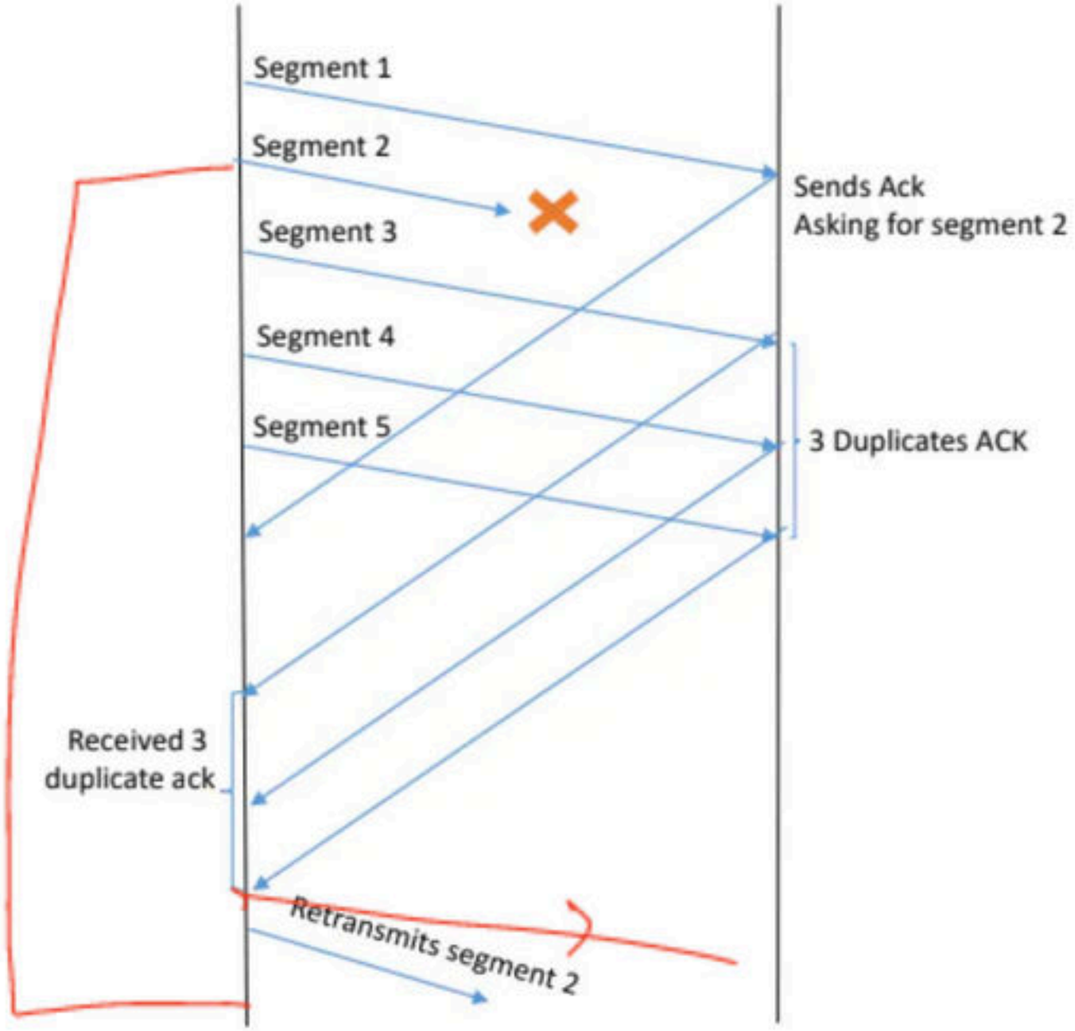
Case-1:

Sender receives an acknowledgement for the sent segment before the timer goes off.
In this case, sender stops the timer.

Case-2:

Sender does not receive any acknowledgement for the sent segment and the timer goes off.
In this case, sender assumes that the sent segment is lost.
Sender retransmits the same segment to the receiver and resets the timer.

3 Duplicate Acknowledgments



Consider sender receives three duplicate acknowledgements for a TCP segment sent by it.
Then, sender assumes that the corresponding segment is lost.
So, sender retransmits the same segment without waiting for its time out timer to expire.

After receiving the retransmitted segment-2,

- Receiver does not send the acknowledgement asking for segment-3 or 4 or 5.
- Receiver sends the acknowledgement asking for segment-6 directly from the sender.
- This is because previous segments have been already received and acknowledgements for them have been already sent (although wasted in asking for segment-2).

Computer Networks

Gate Questions

1.) Consider the following statements about the functionality of an IP based router.
I. A router does not modify the IP packets during forwarding.
II. It is not necessary for a router to implement any routing protocol.
III. A router should reassemble IP fragments if the MTU of the outgoing link is larger than the size of the incoming IP packet.
Which of the above statements is/are TRUE?
[GATE CS 2020]

A) I and II only
B) II only
C) I only
D)II and III only

1.)Consider the following statements about the functionality of an IP based router.
I. A router does not modify the IP packets during forwarding.
II. It is not necessary for a router to implement any routing protocol.
III. A router should reassemble IP fragments if the MTU of the outgoing link is larger than the size of the incoming IP packet.
Which of the above statements is/are TRUE?
[GATE CS 2020]

A) I and II only
B) II only
C) I only
D)II and III only

SOLUTION: II only
I: The packet contains Header and data. The router modifies the header details like TTL
II: is True
III: Ressemble is not necessary at the router.

2.) Consider a long-lived TCP session with an end-to-end bandwidth of 1 Gbps (= 10^9 bits/second). The session starts with a sequence number of 1234. The minimum time (in seconds, rounded to the closest integer) before this sequence number can be used again is _____.M [GATE CS 2018]

2.) Consider a long-lived TCP session with an end-to-end bandwidth of 1 Gbps (= 10^9 bits/second). The session starts with a sequence number of 1234. The minimum time (in seconds, rounded to the closest integer) before this sequence number can be used again is _____.

SOLUTION:
In TCP, Sequence number field is 32 bit, which means 2^{32} sequence number per byte are possible. Whatever be the starting sequence number the possible number will be 2^{32}bytes
The process of using all the sequence number and repeating a previously used sequence number.
The time taken to wrap around is called wrap around time:
Minimum Time = Wrap around time = Total number of bits in sequence number /
Bandwidth = $2^{32} * 8 / 10^9 = 34.35 == 34$ (closest integer)

3.)

Field	Length in bits
P. UDP Header's Port Number	I. 48
Q. Ethernet MAC Address	II. 8
R. IPv6 Next Header	III.32
S. TCP Header's Sequence Number	IV. 16

[GATE CS 2018]

3.)

Field	Length in bits
P. UDP Header's Port Number	I. 48
Q. Ethernet MAC Address	II. 8
R. IPv6 Next Header	III.32
S. TCP Header's Sequence Number	IV. 16

SOLUTION:

P. UDP Header’s Port Number - 16 bits

Q. Ethernet MAC Address - 48 bits

R. IPV6 Next Header - 8 bits

S. TCP Header’s Sequence Number - 32 bits

4.) Consider an IP packet with a length of 4,500 bytes that includes a 20-byte IPv4 header and a 40-byte TCP header. The packet is forwarded to an IPv4 router that supports a Maximum Transmission Unit (MTU) of 600 bytes. Assume that the length of the IP header in all the outgoing fragments of this packet is 20 bytes. Assume that the fragmentation offset value stored in the first fragment is 0.

The fragmentation offset value stored in the third fragment is _____. [GATE CS 2018]

A.)144
B.)145
C.)146
D.)147

4.) Consider an IP packet with a length of 4,500 bytes that includes a 20-byte IPv4 header and a 40-byte TCP header. The packet is forwarded to an IPv4 router that supports a Maximum Transmission Unit (MTU) of 600 bytes. Assume that the length of the IP header in all the outgoing fragments of this packet is 20 bytes. Assume that the fragmentation offset value stored in the first fragment is 0.

The fragmentation offset value stored in the third fragment is _____. [GATE CS 2018]

SOLUTION:
MTU = 600 bytes, IP header = 20 bytes
Therefore Payload = 600 - 20 = 580 bytes.
As we know fragment size should be multiple of 8 but 580 bytes is not a multiple of 8, therefore fragment size is 576 bytes.
Offset value of k^{th} fragment = Fragment size *(k^{th} fragment - 1) / scaling factor
Offset value of third fragment = 576 * (3-1) / 8 = 144

5.)Consider a TCP client and a TCP server running on two different machines. After completing data transfer, the TCP client calls close to terminate the connection and a FIN segment is sent to the TCP server. Server-side TCP responds by sending an ACK, which is received by the client-side TCP. As per the TCP connection state diagram (RFC 793), in which state does the client-side TCP connection wait for the FIN from the server-side TCP?[GATE CS 2017]

A.)LAST-ACK
B.)TIME-WAIT
C.)FIN-WAIT-1
D.)FIN-WAIT-2

5.)Consider a TCP client and a TCP server running on two different machines. After completing data transfer, the TCP client calls close to terminate the connection and a FIN segment is sent to the TCP server. Server-side TCP responds by sending an ACK, which is received by the client-side TCP. As per the TCP connection state diagram (RFC 793), in which state does the client-side TCP connection wait for the FIN from the server-side TCP?

SOLUTION:

Client has sent FIN segment to the server and moves to FIN-WAIT-1, i.e. waiting for the ACK for own FIN segment.

There are two possibilities here:

I. If Client receives ACK for its FIN then client will move to FIN-WAIT-2 and will wait for matching FIN from server side.

After receiving the FIN from server, client will send ACK and move to TIME-WAIT state.

II. Client has sent FIN segment but didn't get ACK till the time.

Instead of ACK, client received FIN from server side.

Client will acknowledge this FIN and move to CLOSE state.

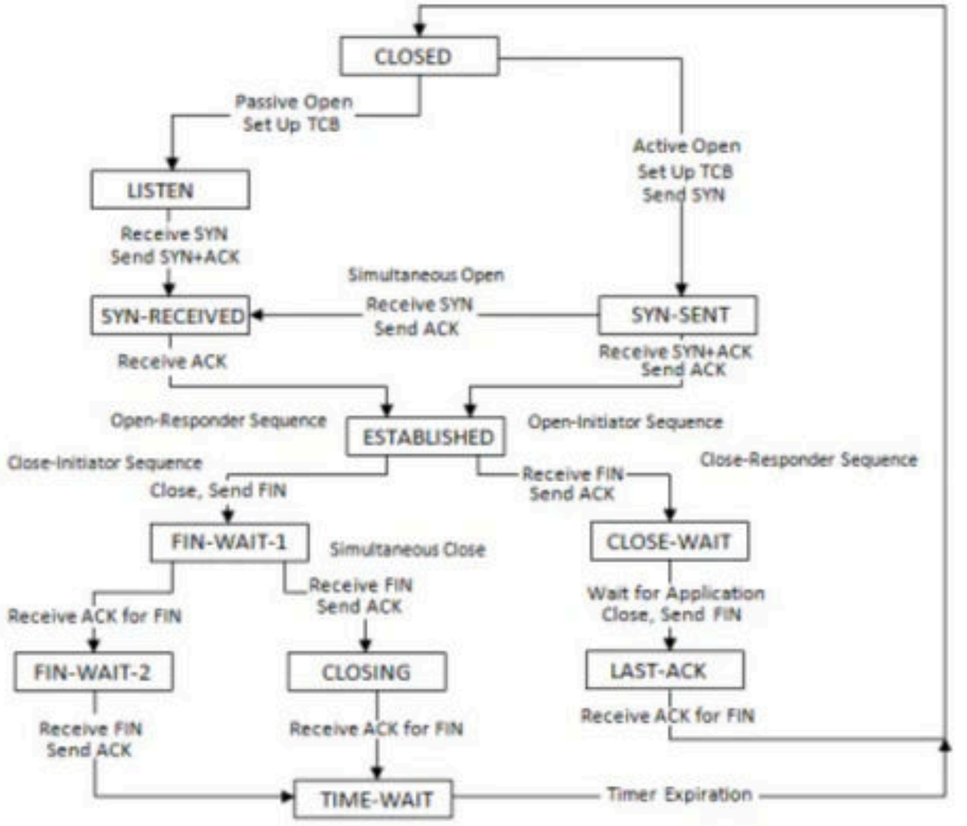
Here Client will wait for the ACK for its own FIN.

After receiving ACK, client will move to TIME-WAIT state.

Here we encounter First Case.

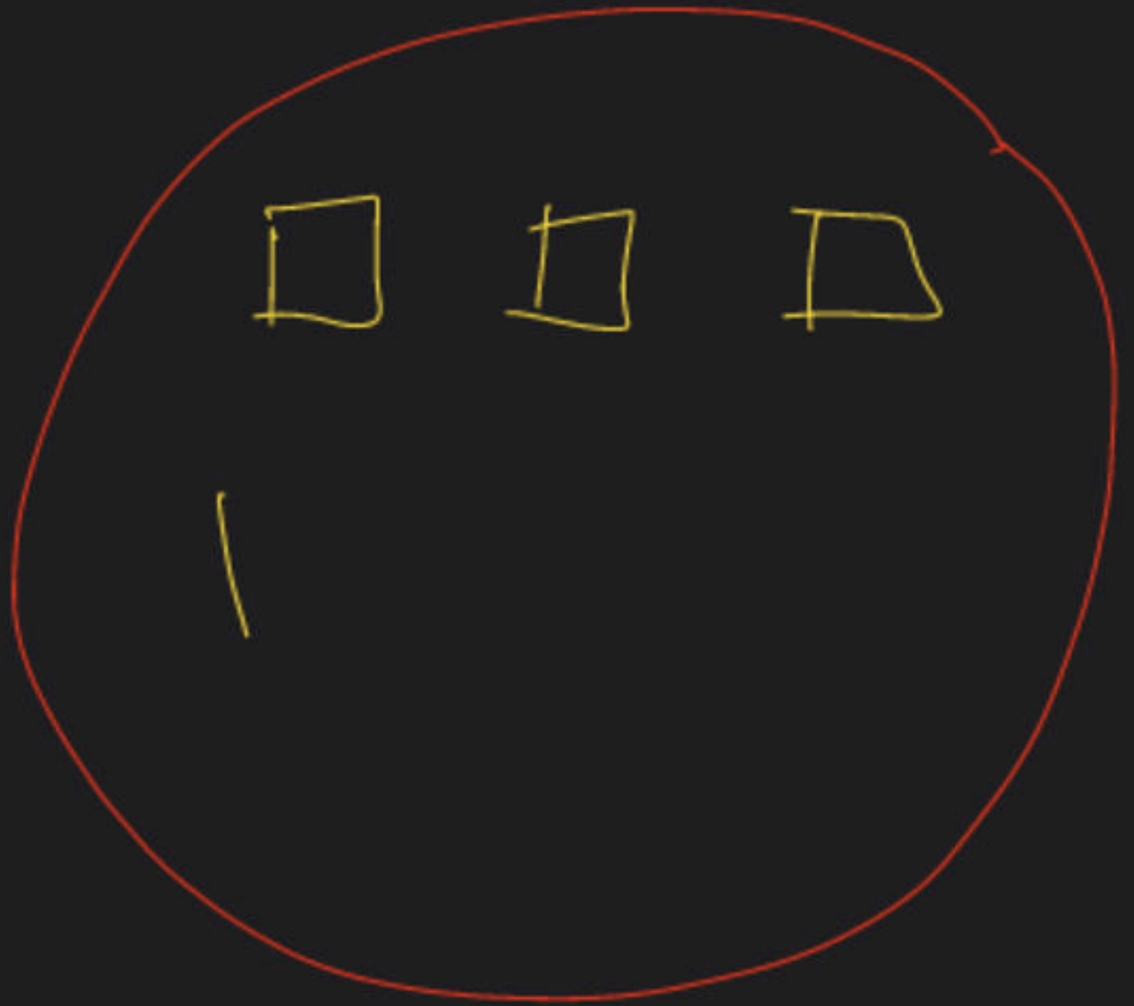
So, the solution is (D).

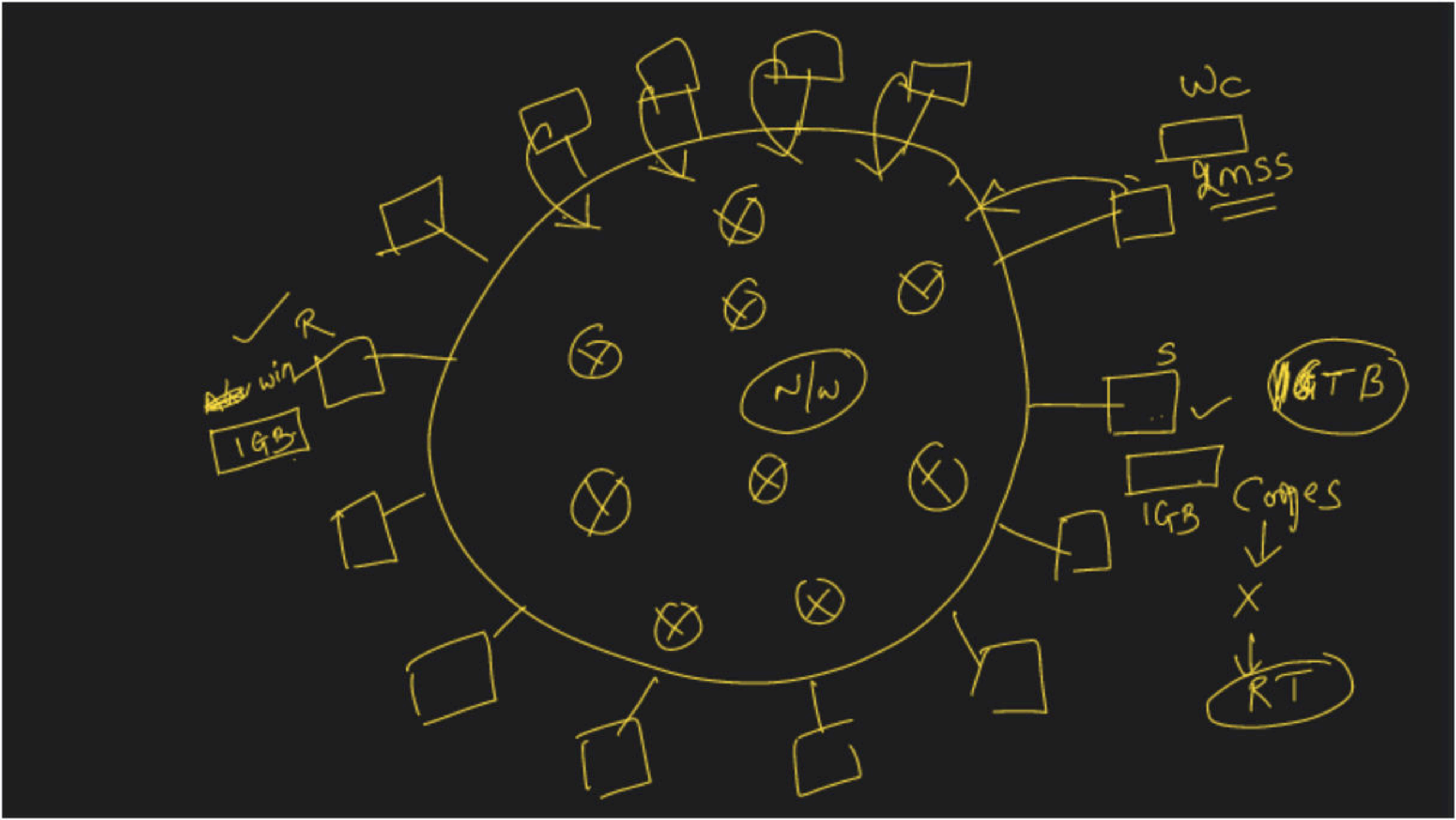
Refer this TCP state transition diagram:



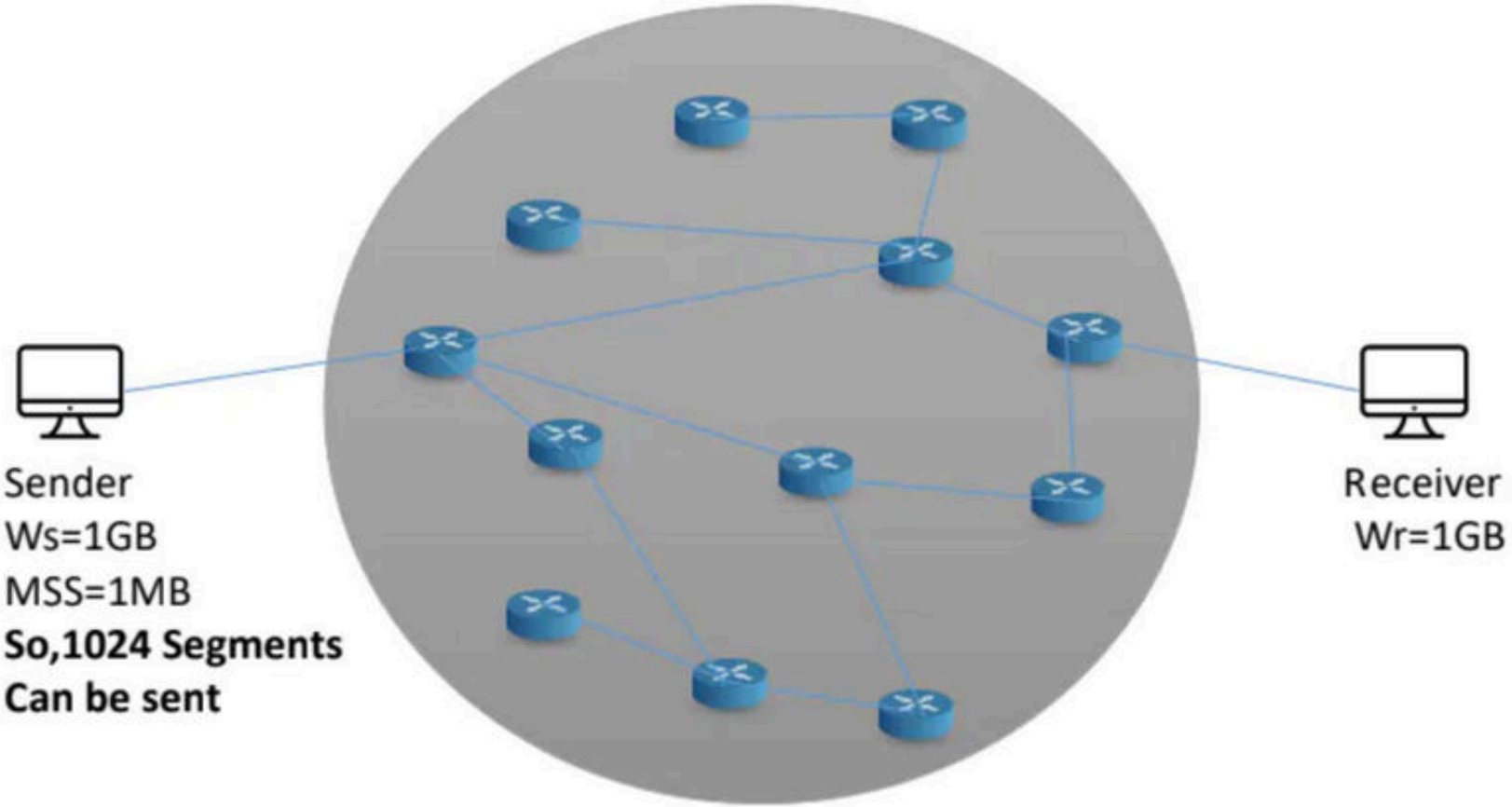
Computer Networks

Introduction to Congestion Control




Wc
2mss
R
win
1GB
N/W
S
1GB
1TB
Copies
X
RT

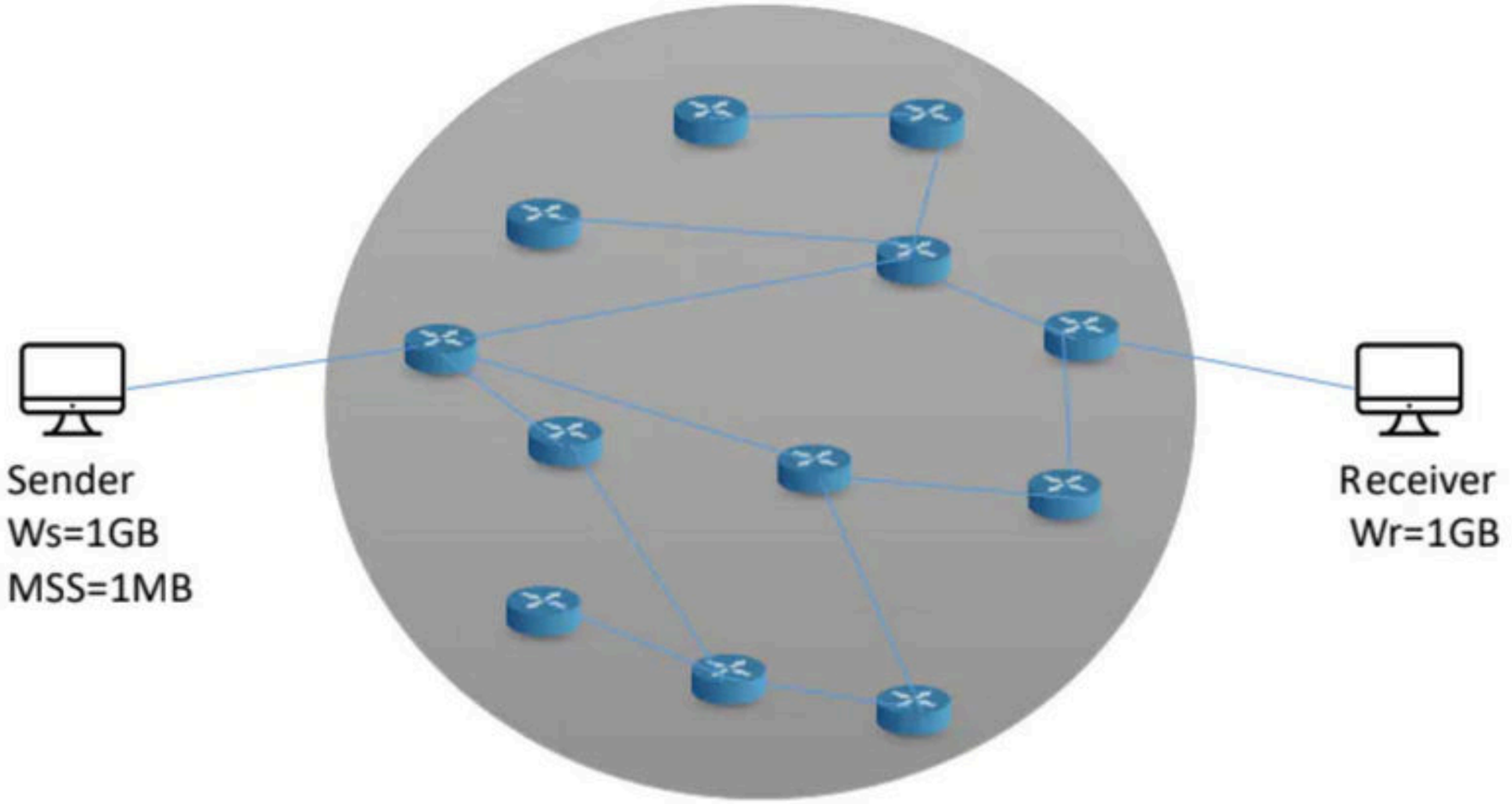
Introduction to Congestion Control



Though the receiver can receive 1024 segments but, the network may not have the same capacity !

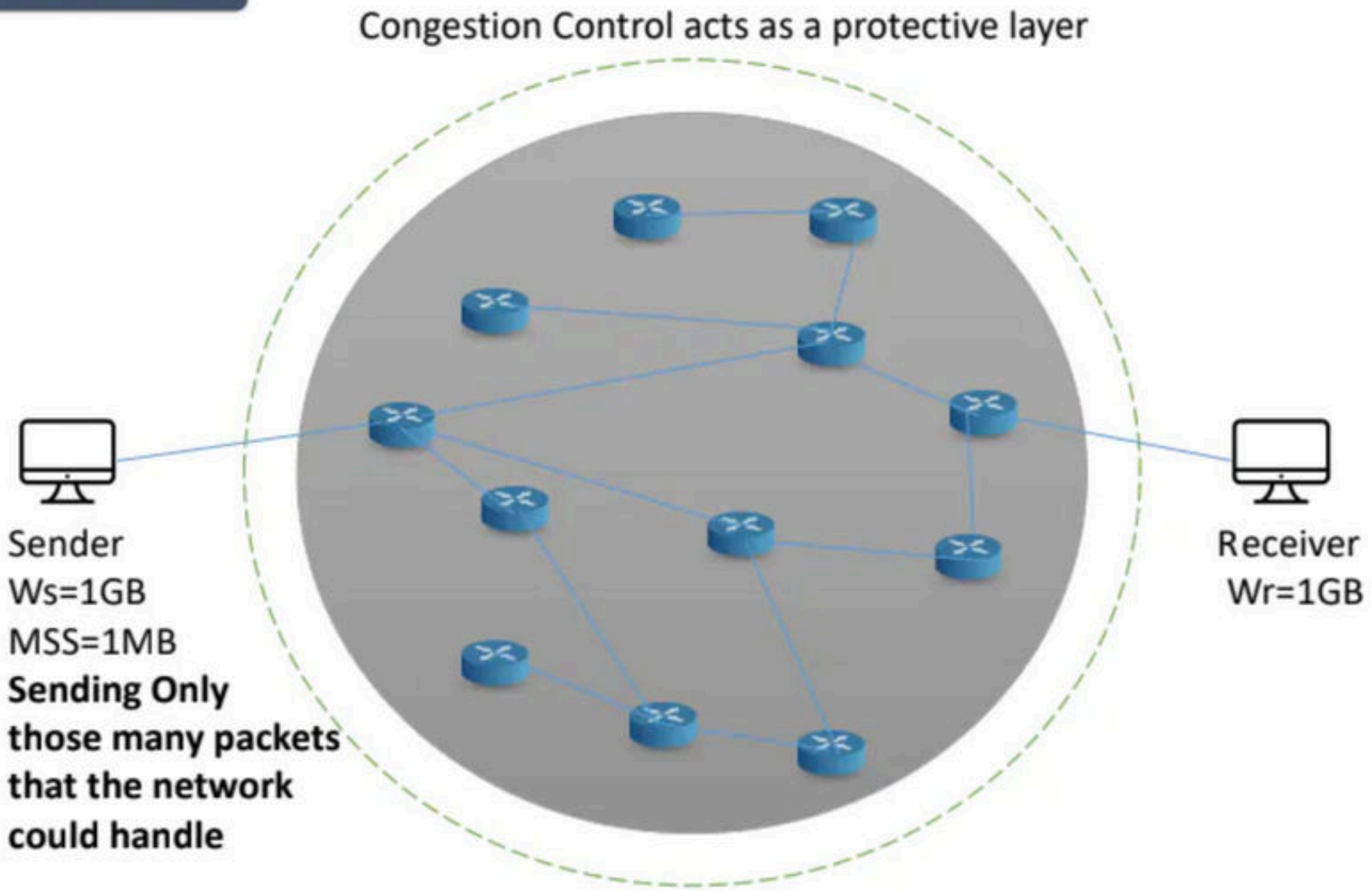
Introduction to Congestion Control

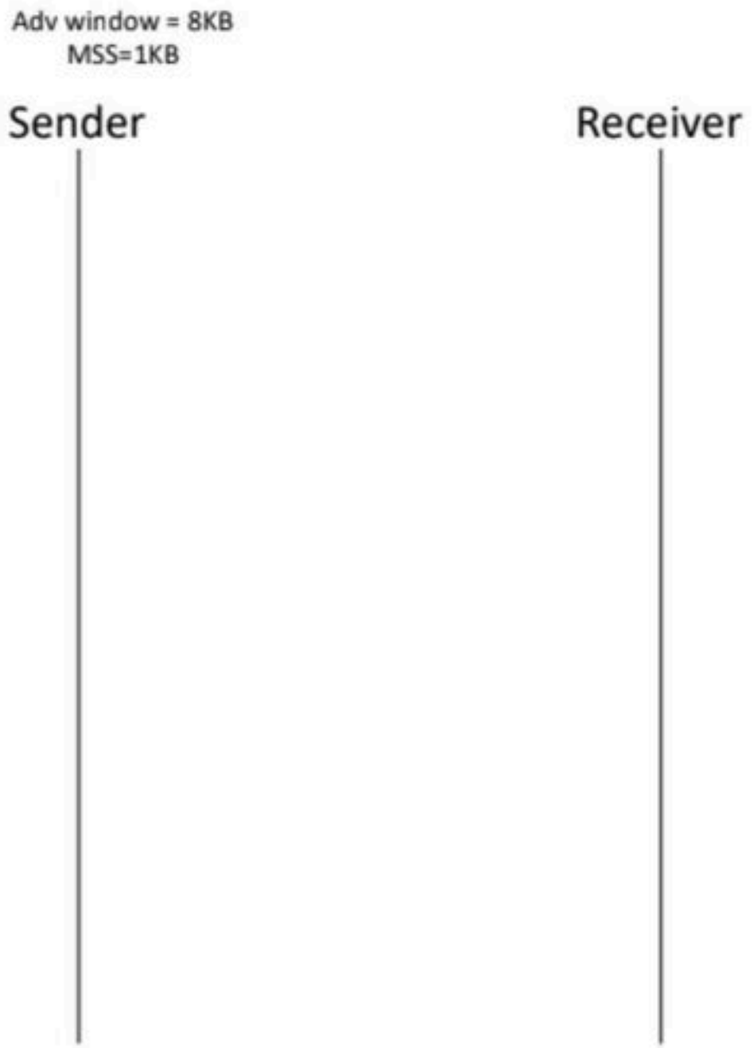
For protecting the receiver we need Flow Control and to protect the network we need Congestion Control



Let us assume that capacity of Network is Wc and capacity of the receiver is Wr then sender should send min{Wc,Wr}

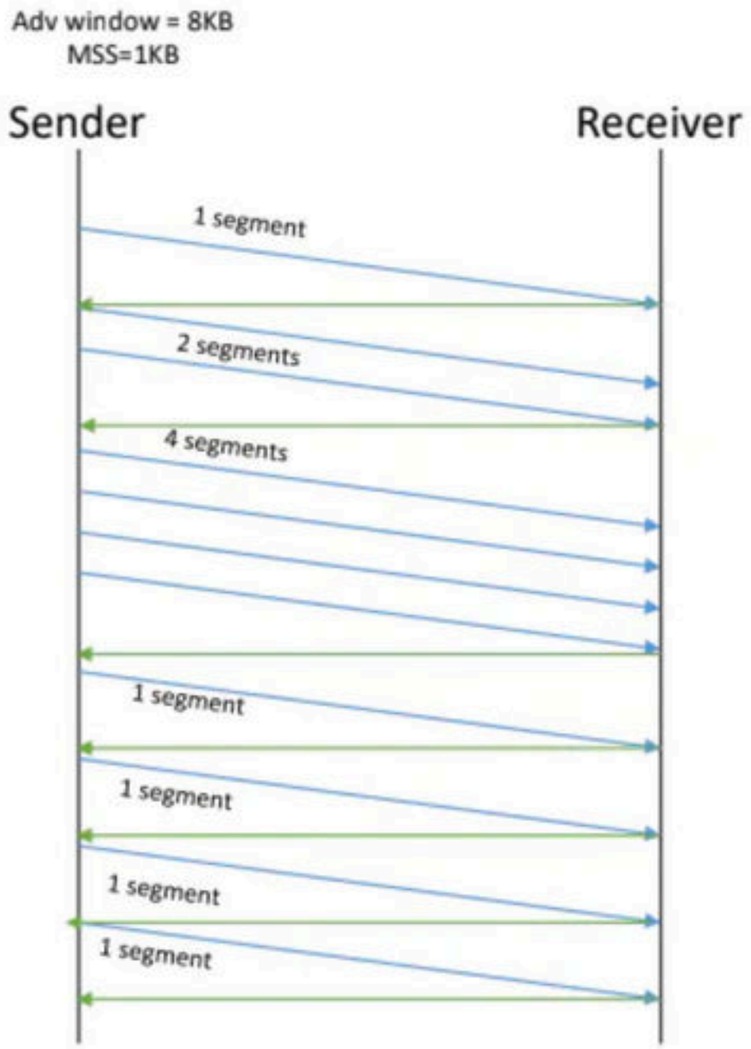
Introduction to Congestion Control





Assume a scenario,

Adv window = 8KB
MSS=1KB
Therefore, 8 segments can be sent
Wr= 8 segments
Even though we can send 8 segments
But we are not going to do that,
We will maintain a congestion window
Wc=1 segment
Therefore Ws=min{Wr,Wc}=1 segment

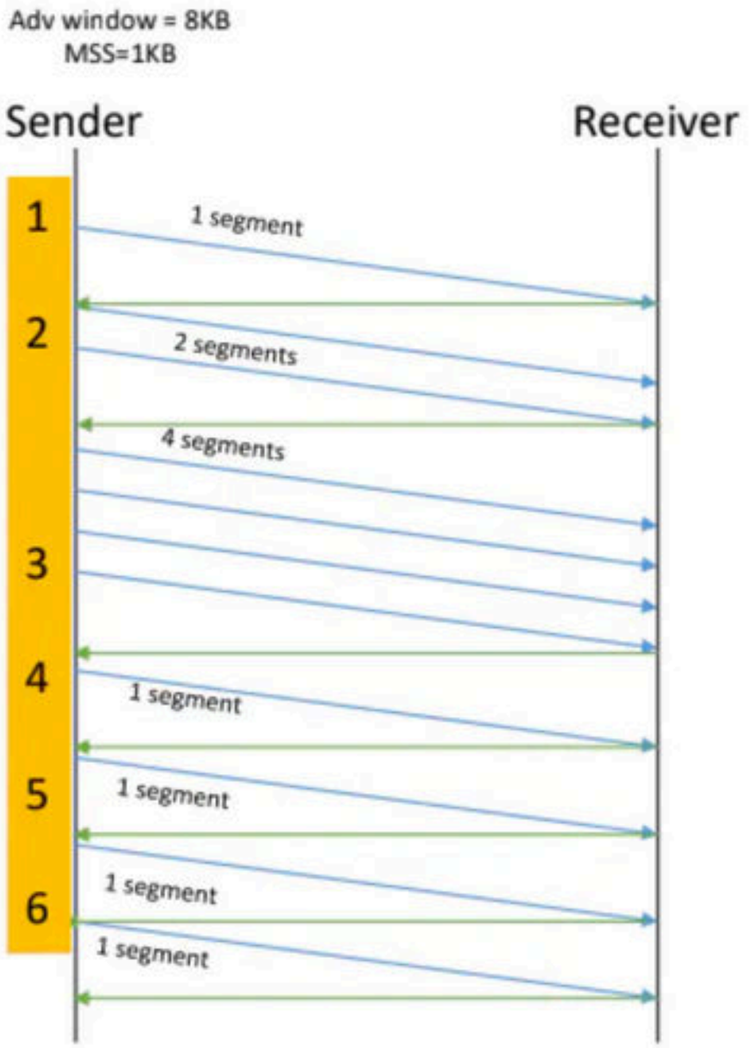


This phase continues until the congestion window size reaches the slow start threshold.

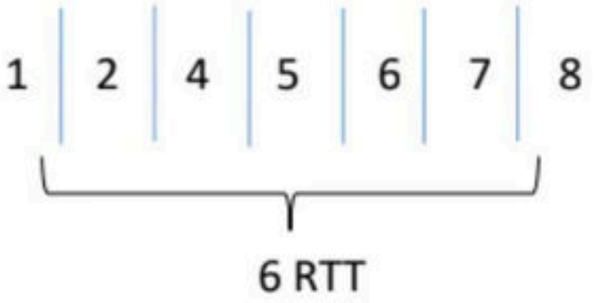
Threshold= Maximum number of TCP segments that receiver window can accommodate / 2

= (Receiver window size / Maximum Segment Size) / 2 = 8/2 =4 MSS

Wc=1
2
4 ← Threshold
5
6
7
8

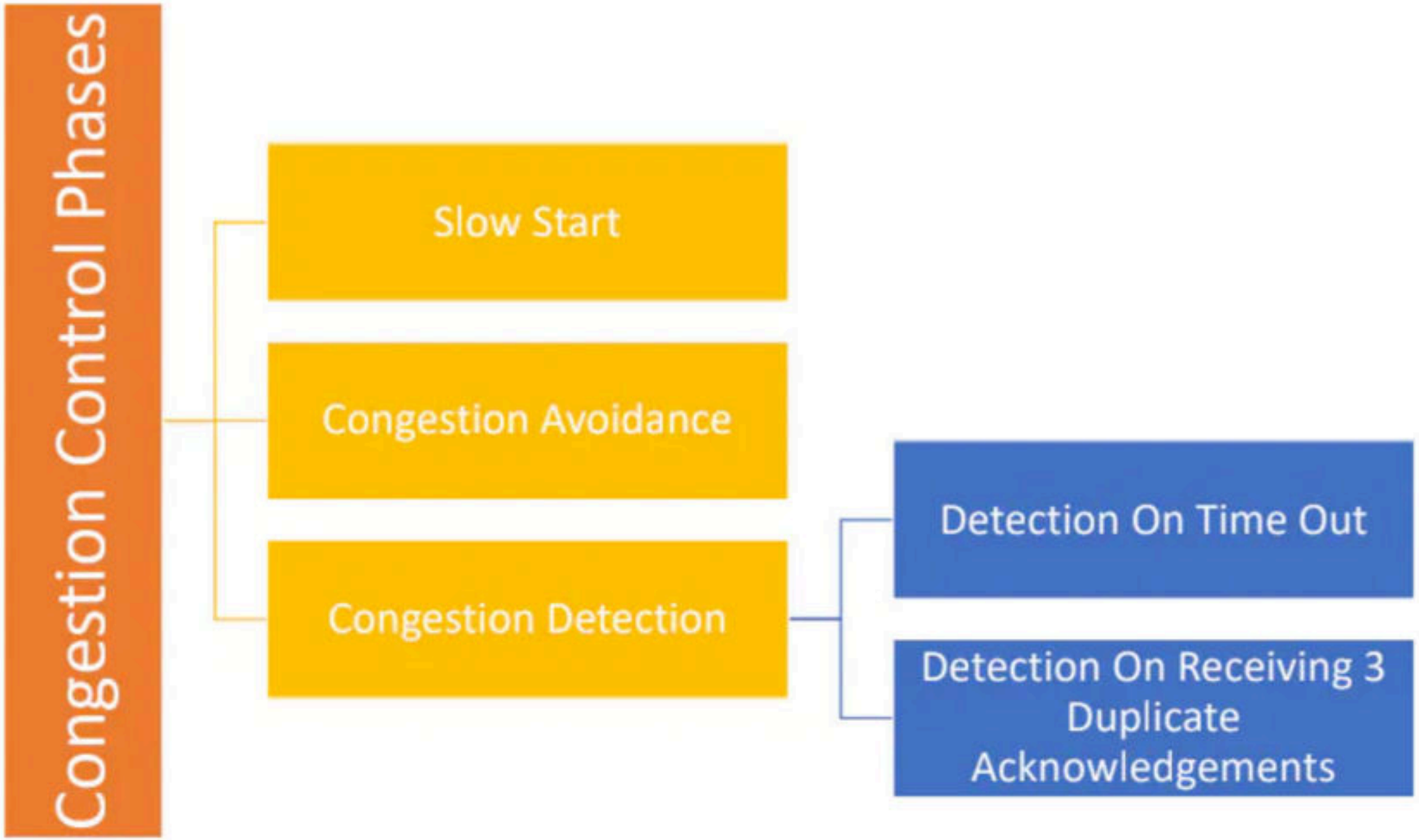


After how many Round Trip Time we will reach the Maximum sender capacity ?
RTT = 6
After how many RTT the Segments of maximum sender capacity will be sent?
After 6



Computer Networks

TCP Congestion Control Algorithm with an Example


Congestion Control Phases
Slow Start
Congestion Avoidance
Congestion Detection
Detection On Time Out
Detection On Receiving 3 Duplicate Acknowledgements

Slow Start Phase and Congestion Avoidance Phase

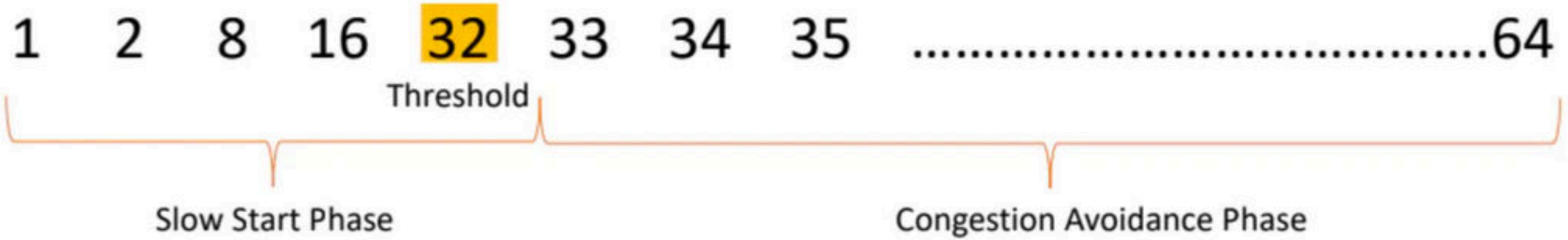
Understanding with an Example:

Wr=64 KB

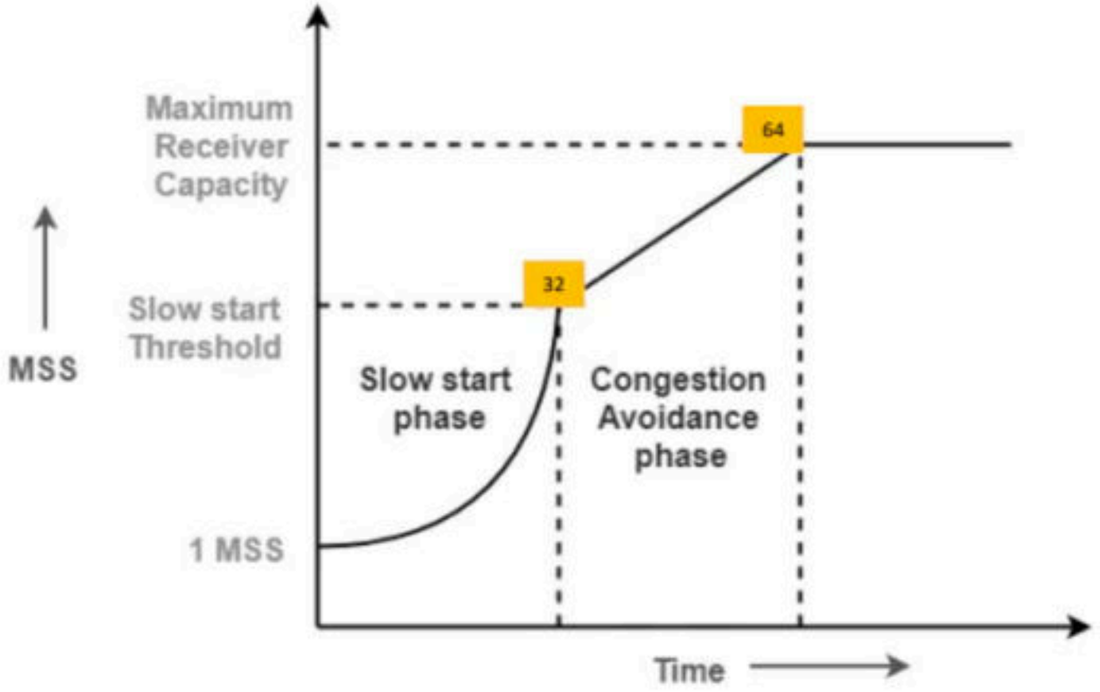
MSS=1KB

Wr=64MSS

Th=64/2 = 32MSS



Slow Start Phase and Congestion Avoidance Phase



Congestion Detection Phase – Time out

New threshold = 34/2=17

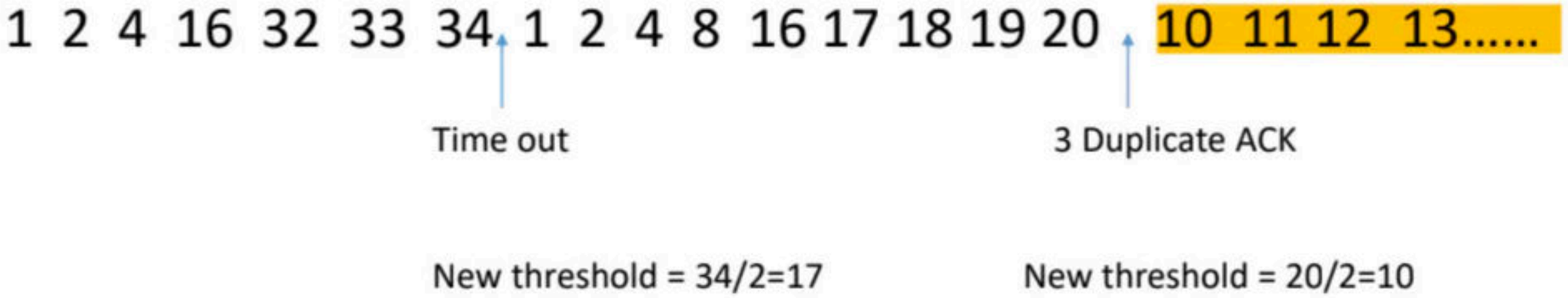
Time Out Timer expires before receiving the acknowledgement for a segment.

In this case, sender reacts by-

Setting the slow start threshold to half of the current congestion window size.
Decreasing the congestion window size to 1 MSS.
Resuming the slow start phase.

Congestion Detection Phase – 3 Duplicate Ack

Lets Continue with the same example to understand 3 Duplicate Ack case



Sender receives **3 duplicate acknowledgements** for a segment.

In this case, sender reacts by-

Setting the slow start threshold to half of the current congestion window size.
Decreasing the congestion window size to slow start threshold.
Resuming the congestion avoidance phase.

Computer Networks

Timers

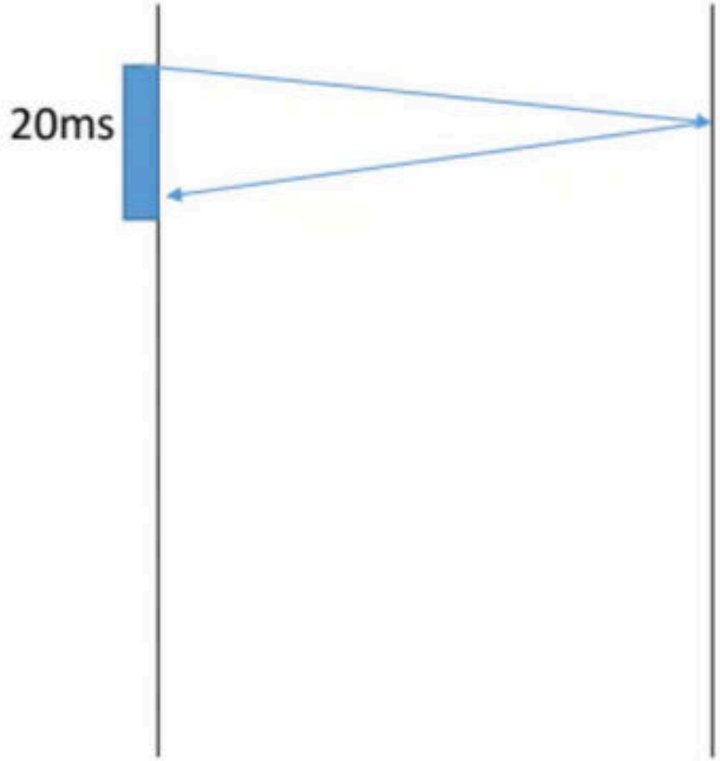

TCP timers
Time out Timer
TCP uses a time out timer for retransmission of lost segments.
Time Wait Timer
TCP uses a time wait timer during connection termination.
Persistent Timer
TCP uses a persistent timer to deal with a zero-widow-size deadlock situation.
It keeps the window size information flowing even if the other end closes its receiver window.
Keep Alive Timer
TCP uses a keep alive timer to prevent long idle TCP connections.
Acknowledgement Timer
TCP uses a Acknowledgment timer to send ack for segments received in that particular time.

Computer Networks

Algorithms for Computing time out timer

Basic Algorithm

Initially sender is not aware of the timer out timer, so the sender is going to guess,
Let that guessing time be the initial time out timer = IRTT
Let IRTT = 10 ms
TO=2RTT =20 ms



But the ACK came back in 15ms
ARTT=15ms
So we have 2 options,
Either go with the old values of TO which is said to be static
or we can set the TO according to ARTT
Then it is said to be Dynamic
Both are dangerous

Also if we set small TO then Unnecessary retransmissions
And if the TO is large then it leads to waste of time

NRTT = α IRTT + (1 – α)ARTT

Here, α is called smoothing factor where 0 <= α <= 1

IRTT = 10 ms
TO=2RTT =20 ms
A=0.5
NRTT=12.5

IRTT=12.5
TO=2RTT=2*12.5 =25ms
ARTT=20ms
NRTT=16.25

.
.
.
.

Jacobson's Algorithm

Jacobson's Algorithm is a modified version of the basic algorithm.
It gives better performance than Basic Algorithm.
The only difference between the two is TO formula

1. IRTT=10ms
   ID=5ms
   TO=4*D*RTT
   =4*5+10
   =30ms

ARTT=20ms
AD=10ms(IRTT-ARTT)

NRTT=α(IRTT)+(1- α)(ARTT)=15ms.. A=0.5
ND= α(ID)+(1- α)(AD)=7.5ms

2. IRTT=15ms
   ID=7.5ms
   TO=4*D*RTT
   =45ms

ARTT=30ms
AD=15ms(IRTT-ARTT)

NRTT=α(IRTT)+(1- α)(ARTT)=22.5ms.. α =0.5
ND= α(ID)+(1- α)(AD)=11.25

3. IRTT=22.5ms
   ID=11.5ms
   TO=4*D*RTT
   =67.5ms

ARTT=10ms
AD=12.5ms(IRTT-ARTT)

NRTT=α(IRTT)+(1- α)(ARTT)=16.25ms.. A=0.5
ND= α(ID)+(1- α)(AD)=11.875

Karn's modification states-

Whenever a segment has to be re transmitted, do not apply either of Basic or Jacobson's algorithm since actual RTT is not available.
Instead, double the time out timer (TOT) whenever the timer times out and make a retransmission.

Computer Networks

Silly Window Syndrome

Silly window Syndrome problem arises due to following causes

1. Sender transmitting data in small segments repeatedly
2. Receiver accepting only few bytes at a time repeatedly

Sender Transmitting Data In Small Segments Repeatedly

This problem is solved using **Nagle's Algorithm.**

Nagle's Algorithm tries to solve the problem caused by the sender delivering 1 data byte at a time.

Sender should send only the first byte on receiving one byte data from the application.
Sender should buffer all the rest bytes until the outstanding byte gets acknowledged.
In other words, sender should wait for 1 RTT.

Receiver Accepting Only Few Bytes Repeatedly

This problem is solved using **Clark's Solution**.

Clark's Solution tries to solve the problem caused by the receiver consuming one data byte at a time.

Receiver should not send a window update for 1 byte.
Receiver should wait until it has a decent amount of space available.

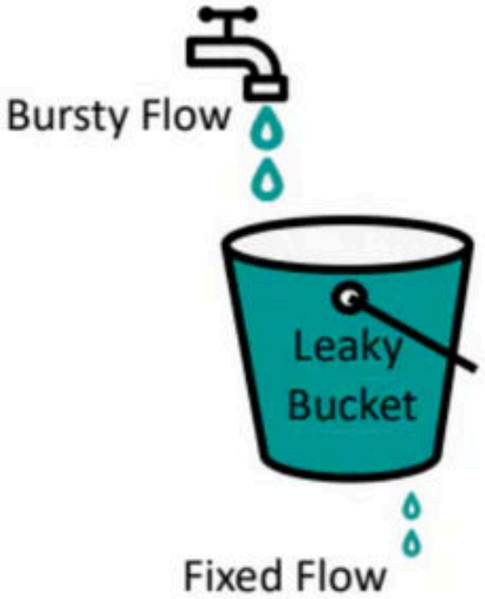
Computer Network

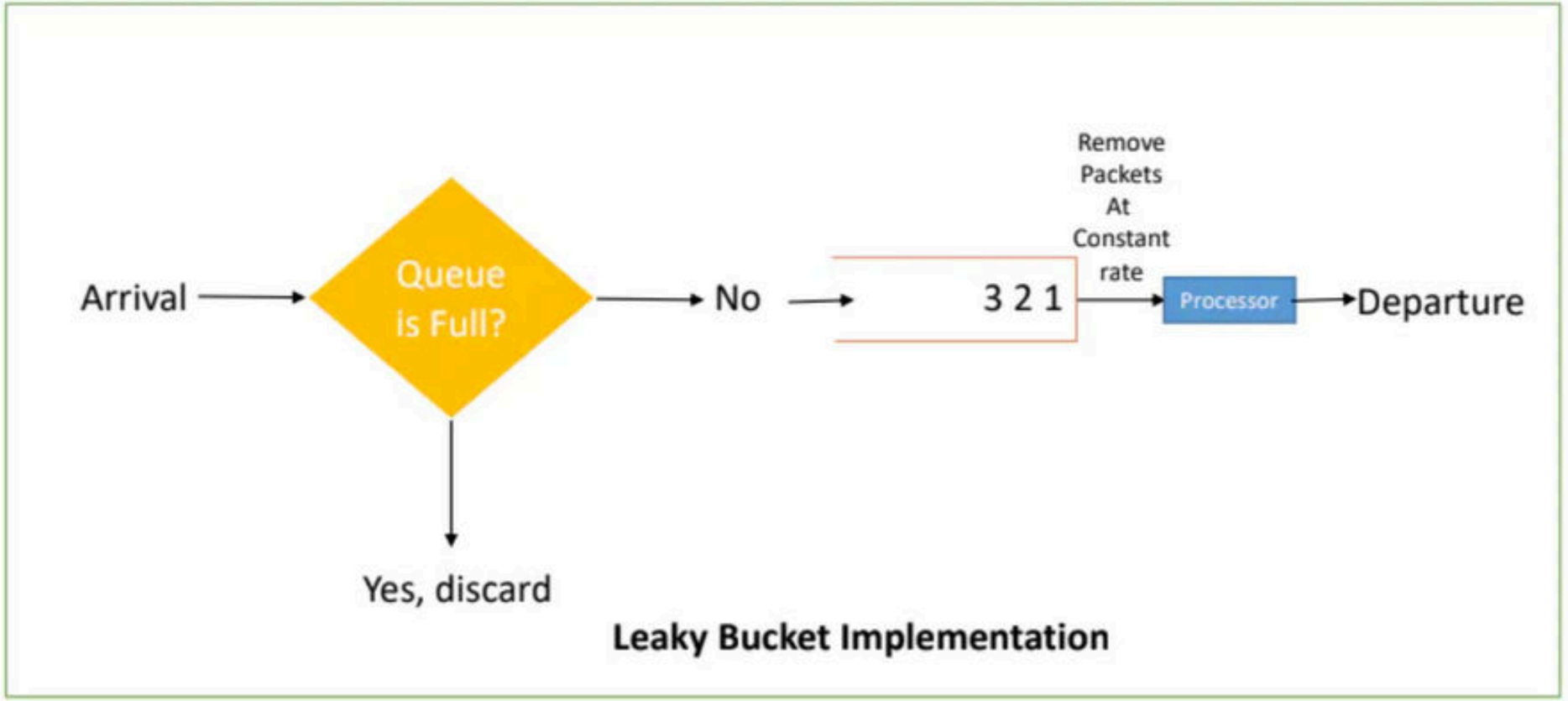
Traffic Shaping – Leaky Bucket and Token Bucket

Another way for Congestion Control is **Traffic Shaping**

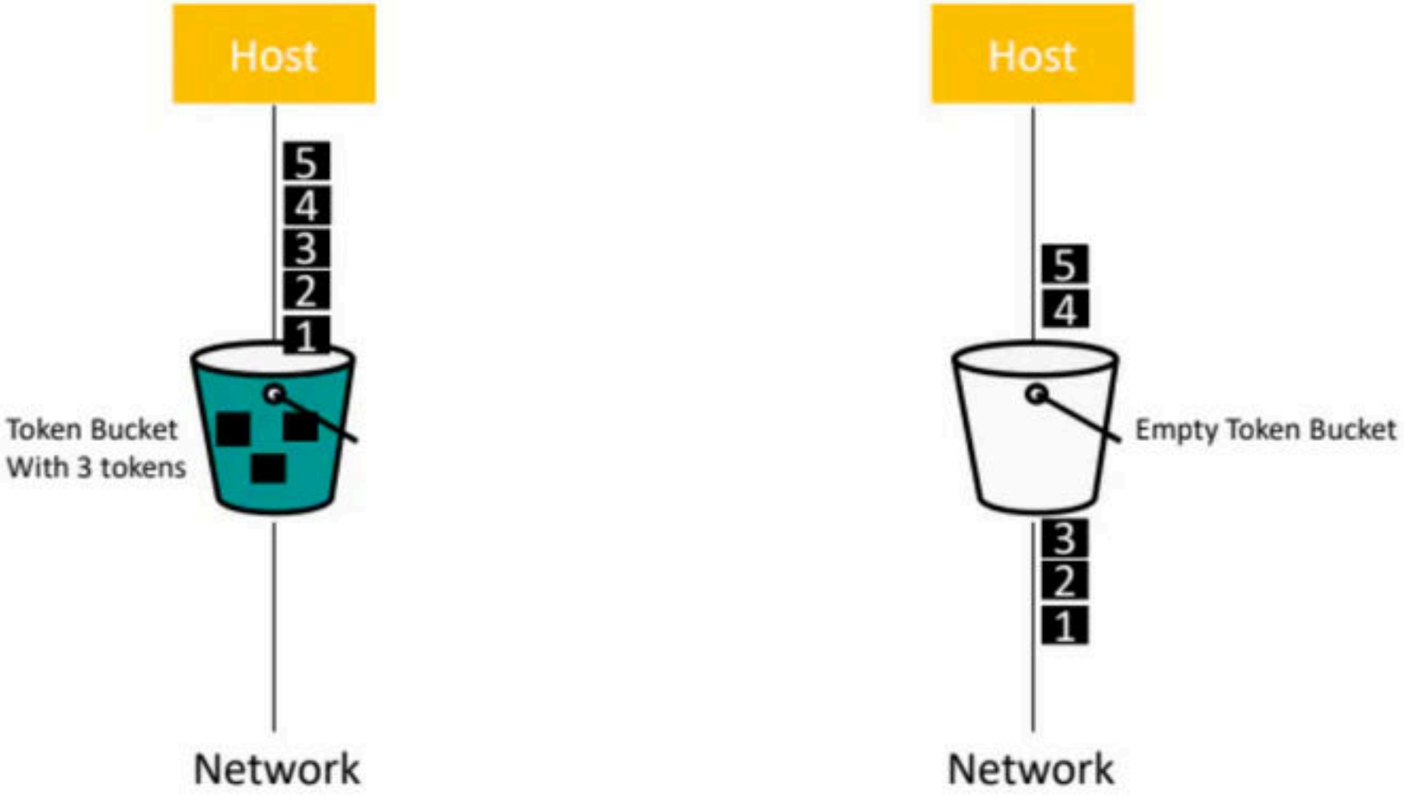
To shape the Traffic before it enters the network.
Traffic shaping controls the rate at which the packets are sent.
During connection establishment, the sender and carrier negotiate a traffic.
We can do it by 2 algorithms – Leaky Bucket and Token Bucket

Leaky Bucket Algorithm





Token Bucket Algorithm



Let the capacity of the token bucket be ‘c’ tokens and the tokens enter the bucket at the rate of ‘r’ tokens per sec,
The maximum number of packets that can enter the network during the time interval t is,

Maximum number of packets = $c+rt$

Maximum average rate = $\frac{(c+rt)}{t}$ packets per sec

Computer Networks

UDP

Need of UDP

1.) If application needs 1 request / 1 reply
DNS
BOOTP / DHCP
NTP
NNP
Quote of the day

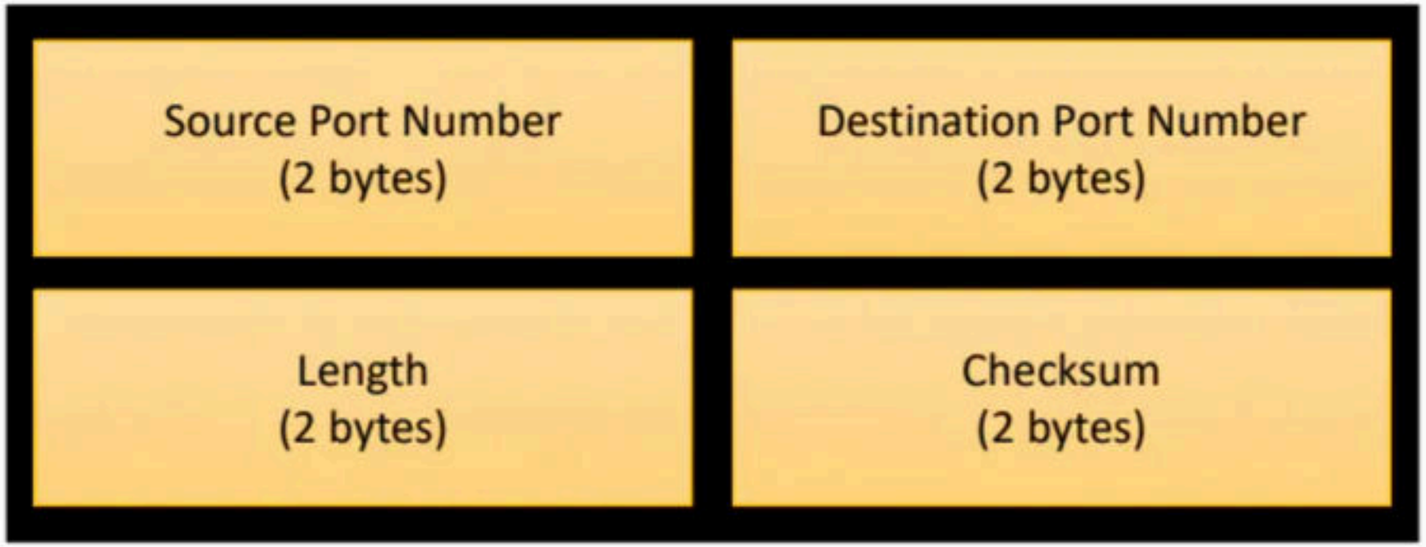
2.)Broadcasting / Multicasting

3.)In applications that need speed rather than reliability

UDP is short for User Datagram Protocol.

It is a connectionless protocol.
It is a stateless protocol.
It is an unreliable protocol.
It is a fast protocol.
UDP does not guarantee in order delivery.

UDP Header



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.

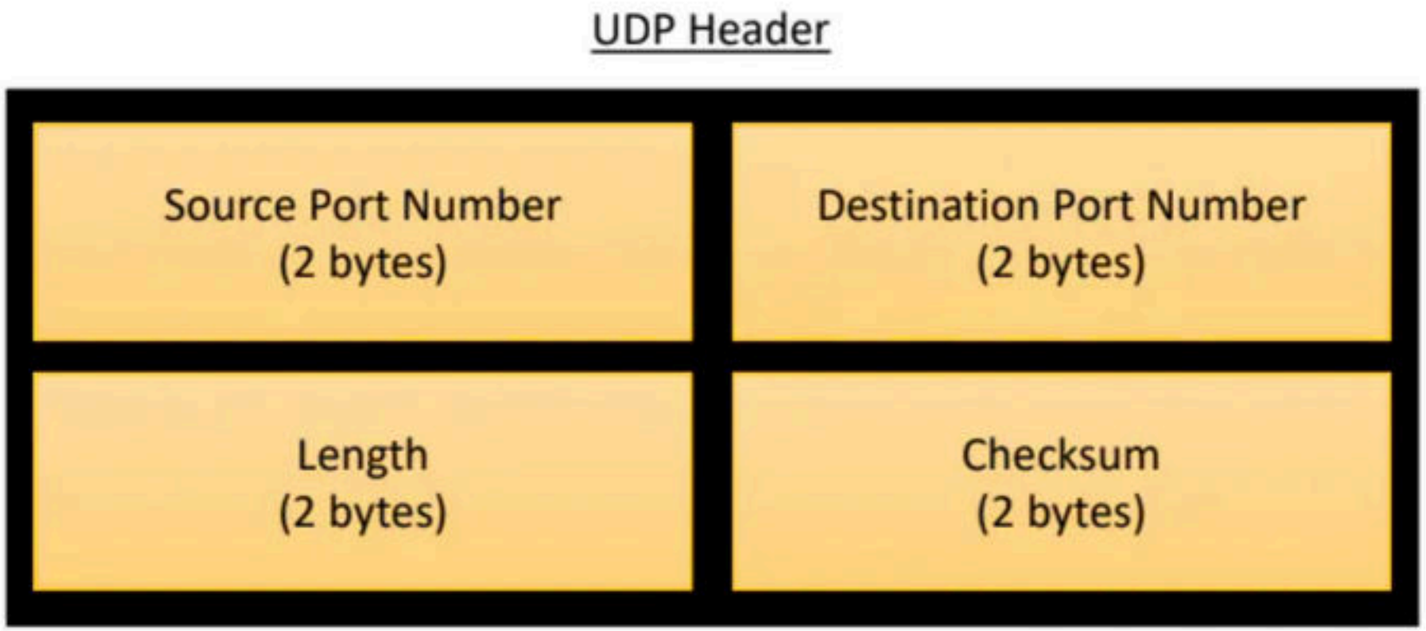
3. Length-

Length is a 16 bit field.
It identifies the combined length of UDP Header and Encapsulated data.

Length = Length of UDP Header + Length of encapsulated data

4. Checksum-

Checksum is a 16 bit field used for error control.
It is calculated on UDP Header, encapsulated data and IP pseudo header.
Checksum calculation is not mandatory in UDP.

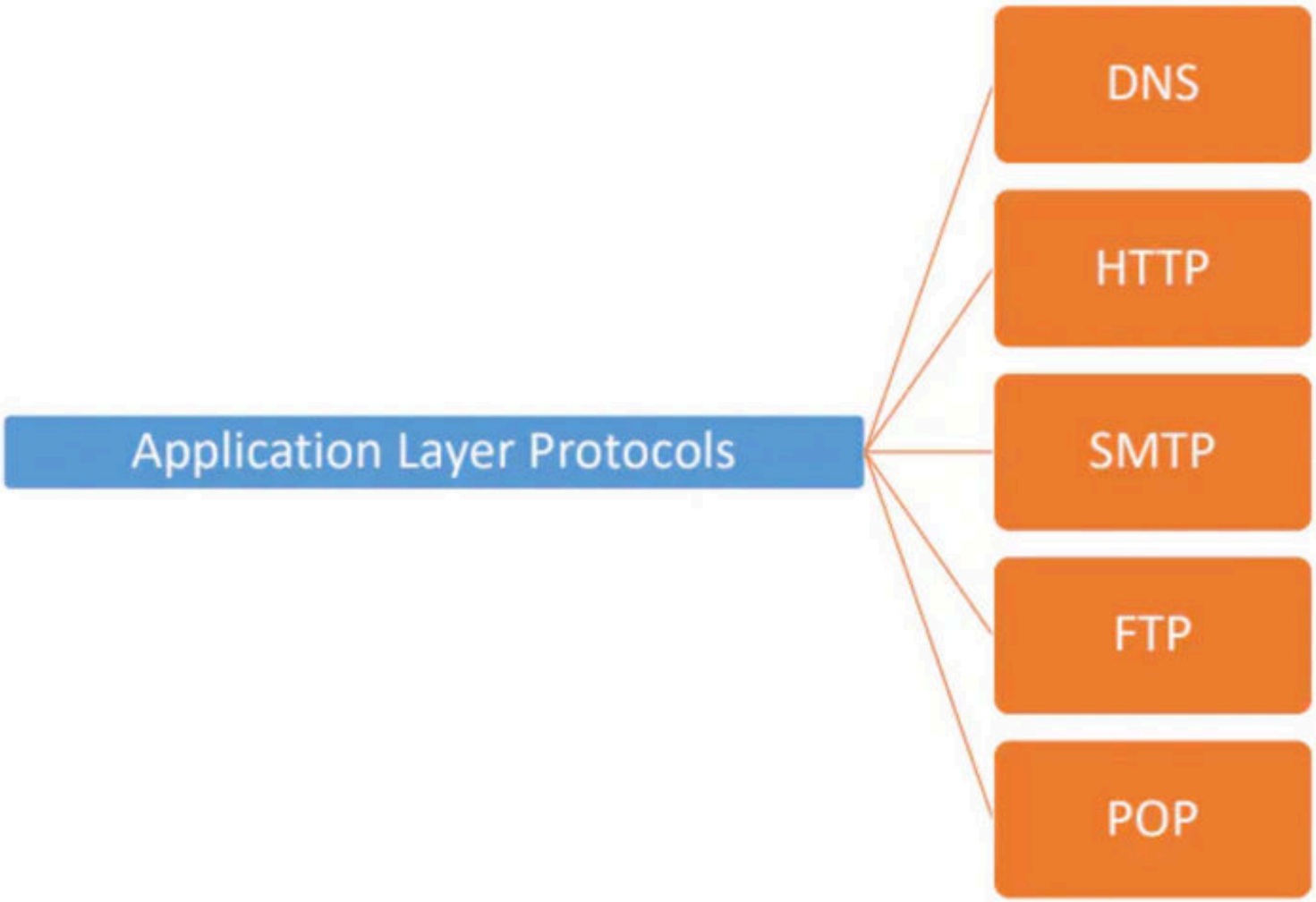


Application layer can do the following tasks through UDP-
1.Trace Route
2.Record Route
3.Time stamp

UDP acts like a messenger between the application layer and the IP datagram.

Computer Networks

Application Layer Protocols - DNS


Application Layer Protocols
DNS
HTTP
SMTP
FTP
POP

DNS

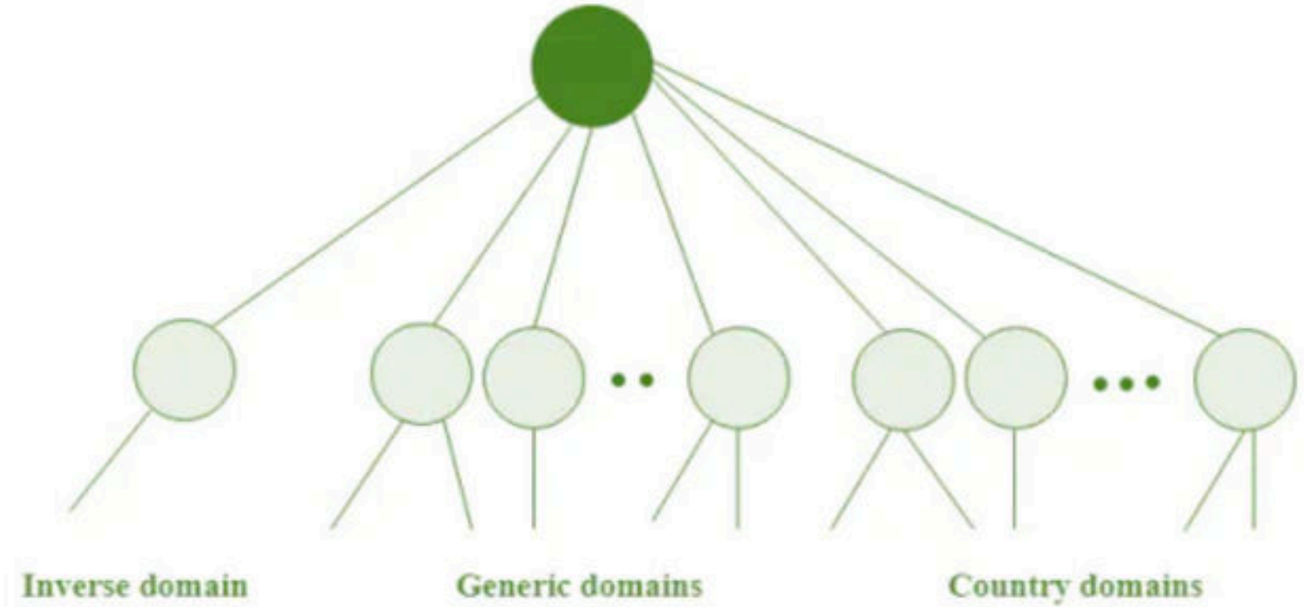
DNS is short for Domain Name Service or Domain Name System.

DNS Resolution is a process of resolving a domain name onto an IP Address.

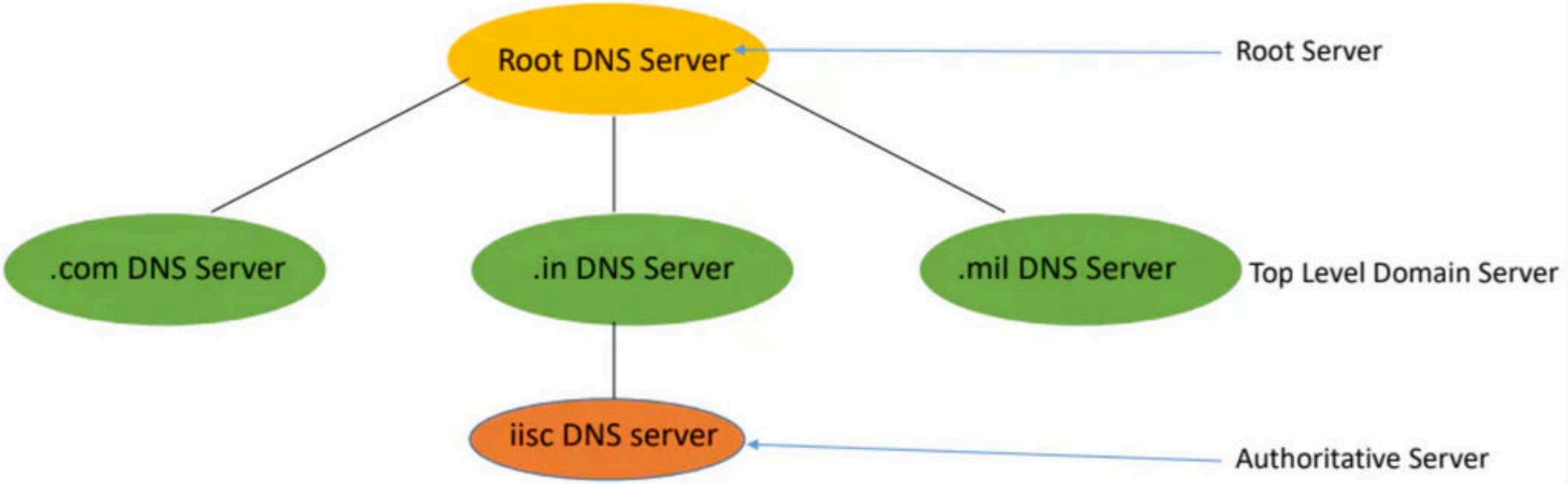
DNS uses UDP (port 53) at the transport layer.

DNS is a connection less protocol

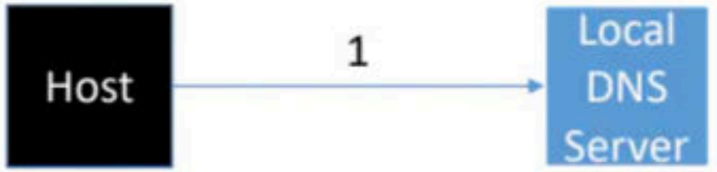
The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.



- Generic Domains – It defines the registered hosts according to their generic behavior. Each node in a tree defines the domain name, which is an index to the DNS database. It uses three-character labels, and these labels describe the organization type.
  Example: .com, .mil, .edu
- Country Domain - The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.
  Example: .in, .uk, .us
- The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name


Root DNS Server
Root Server
.com DNS Server
.in DNS Server
.mil DNS Server
Top Level Domain Server
iisc DNS server
Authoritative Server

Let us assume a scenario where we wish to go to www.csa.iisc.in
There are two ways DNS can contact the server –
1.)Iterative way

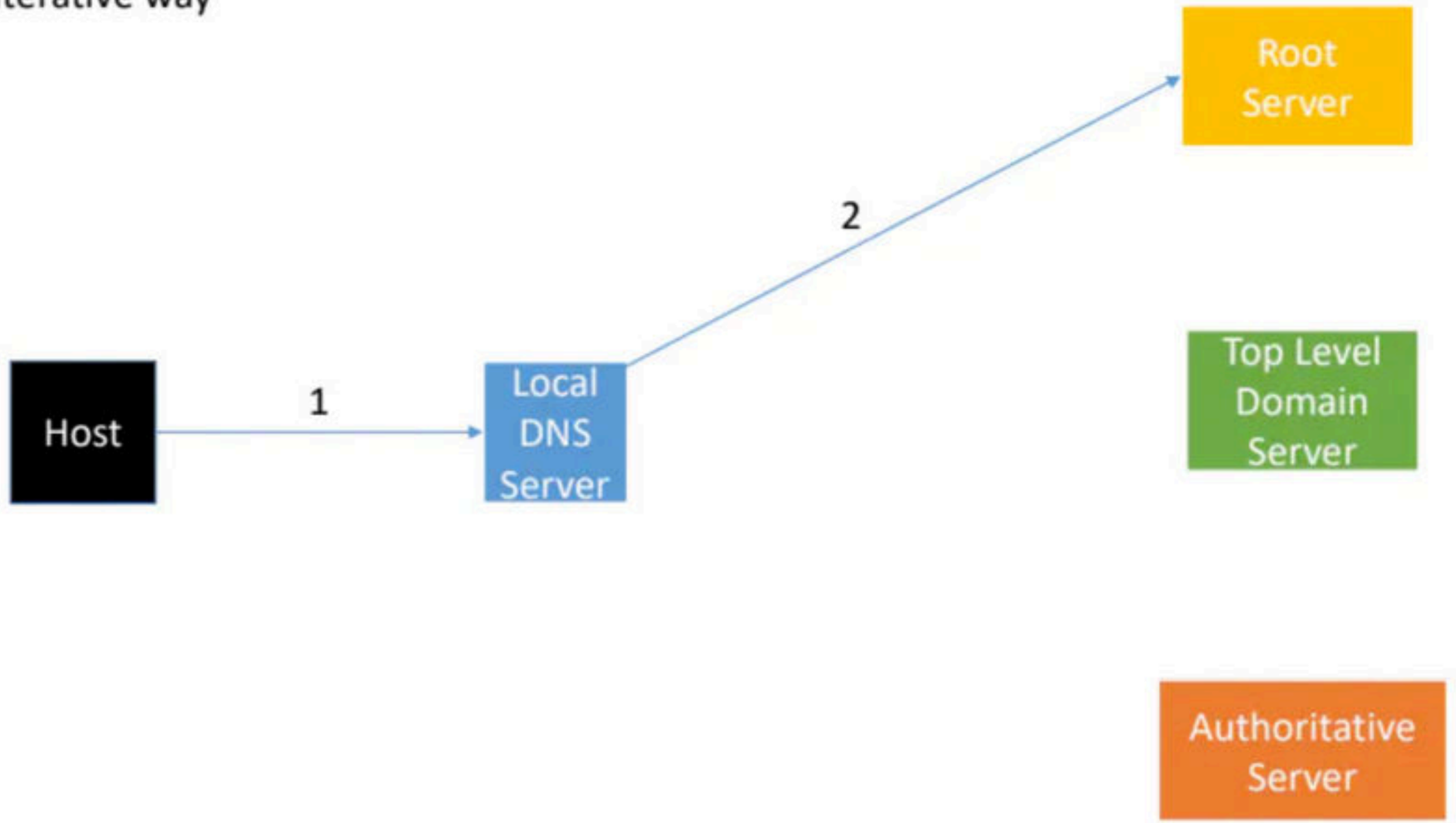


Root Server

Top Level Domain Server

Authoritative Server

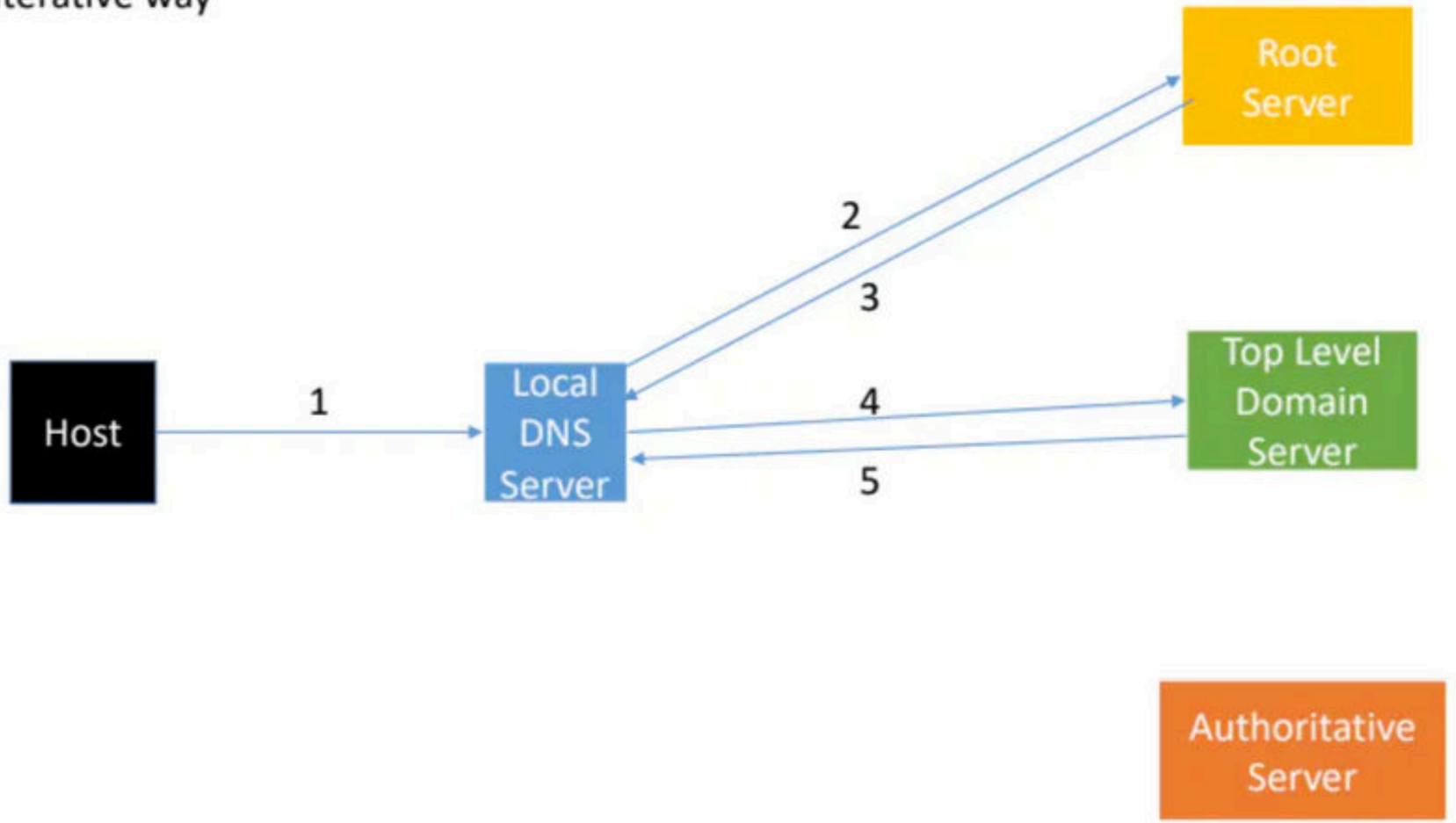
Let us assume a scenario where we wish to go to www.csa.iisc.in
There are two ways DNS can contact the server –
1.)Iterative way



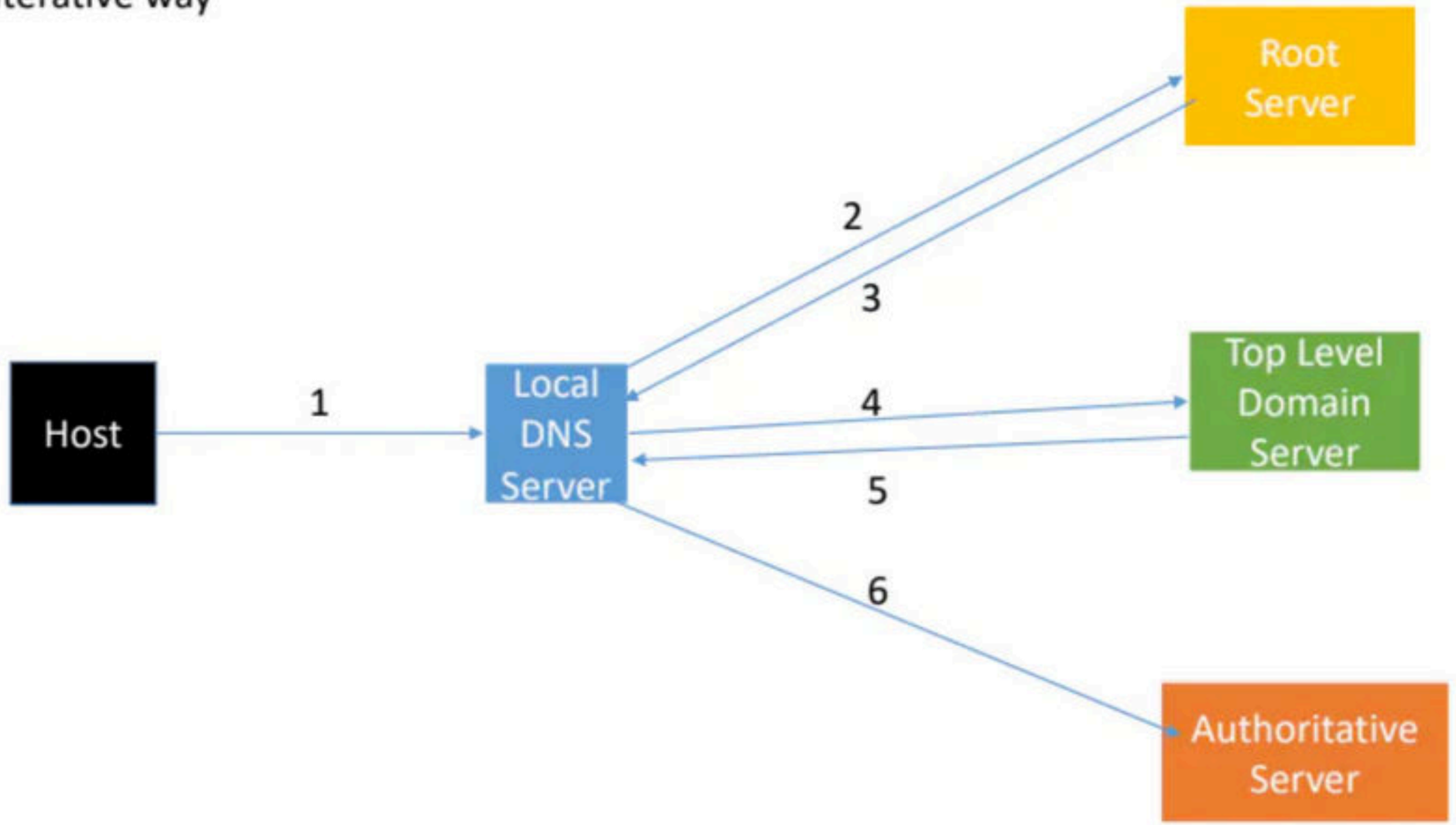
Let us assume a scenario where we wish to go to www.csa.iisc.in
There are two ways DNS can contact the server –
1.)Iterative way

Let us assume a scenario where we wish to go to www.csa.iisc.in
There are two ways DNS can contact the server –
1.)Iterative way

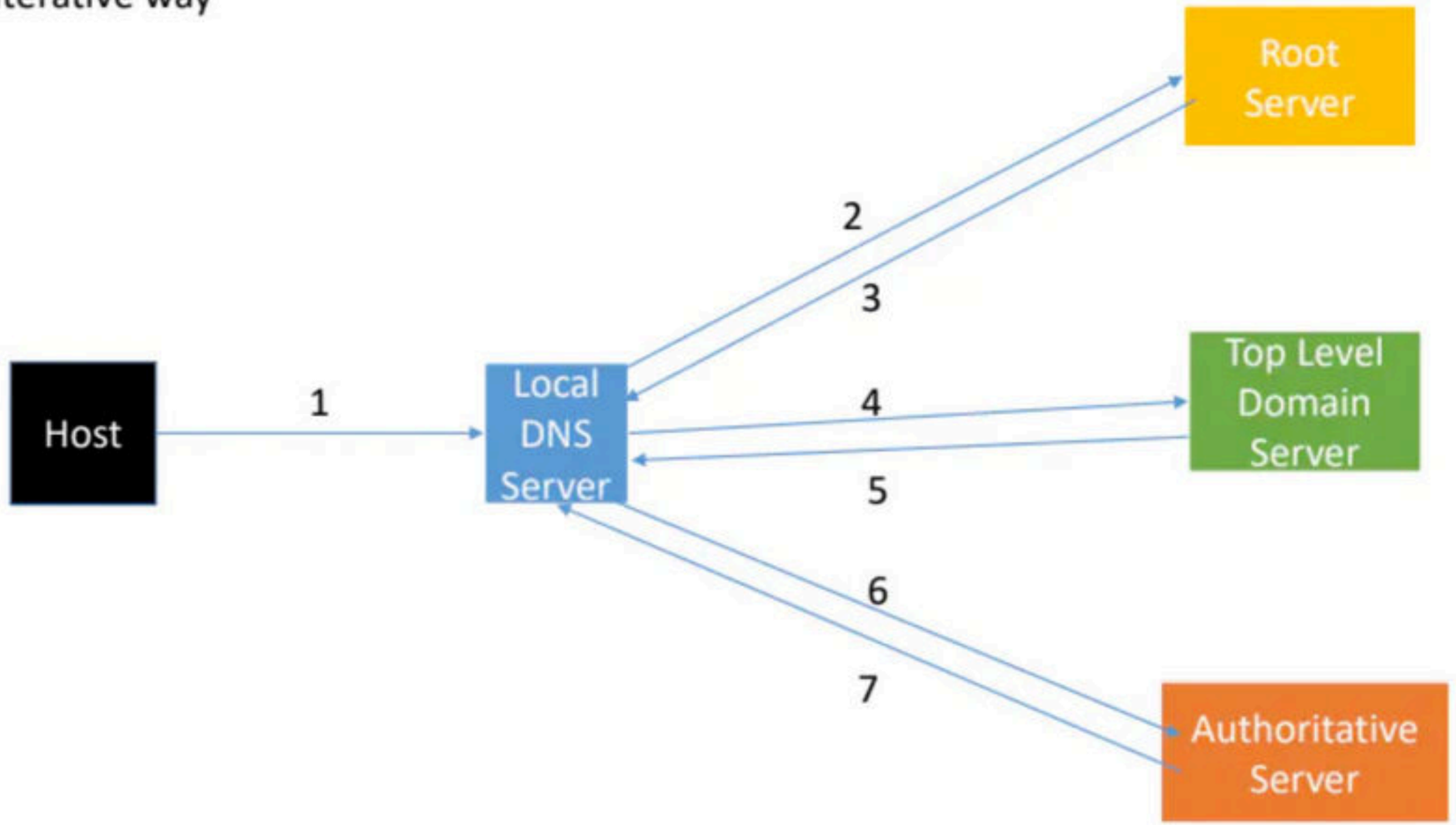
Let us assume a scenario where we wish to go to www.csa.iisc.in
There are two ways DNS can contact the server –
1.)Iterative way



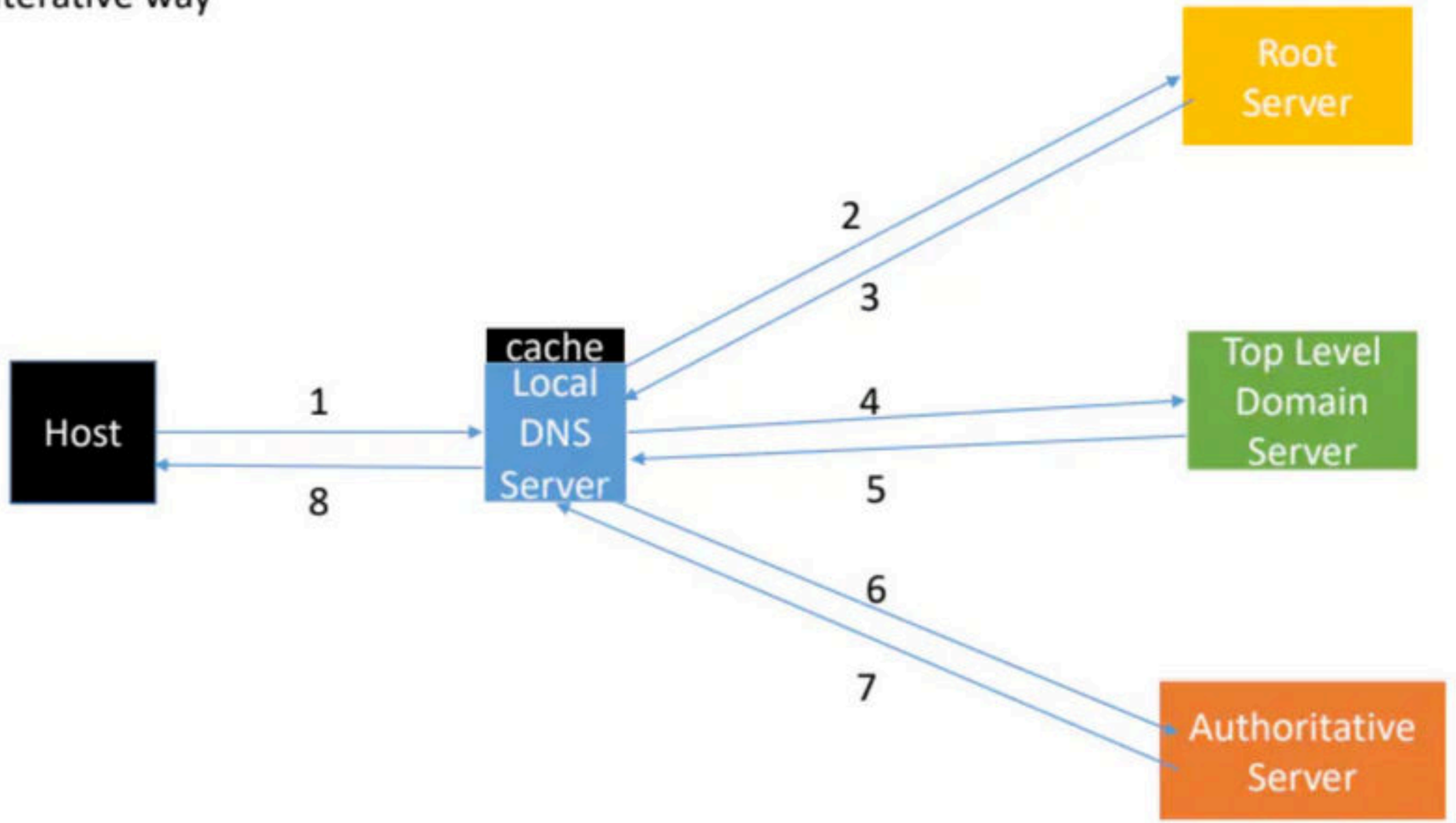
Let us assume a scenario where we wish to go to www.csa.iisc.in
There are two ways DNS can contact the server –
1.)Iterative way



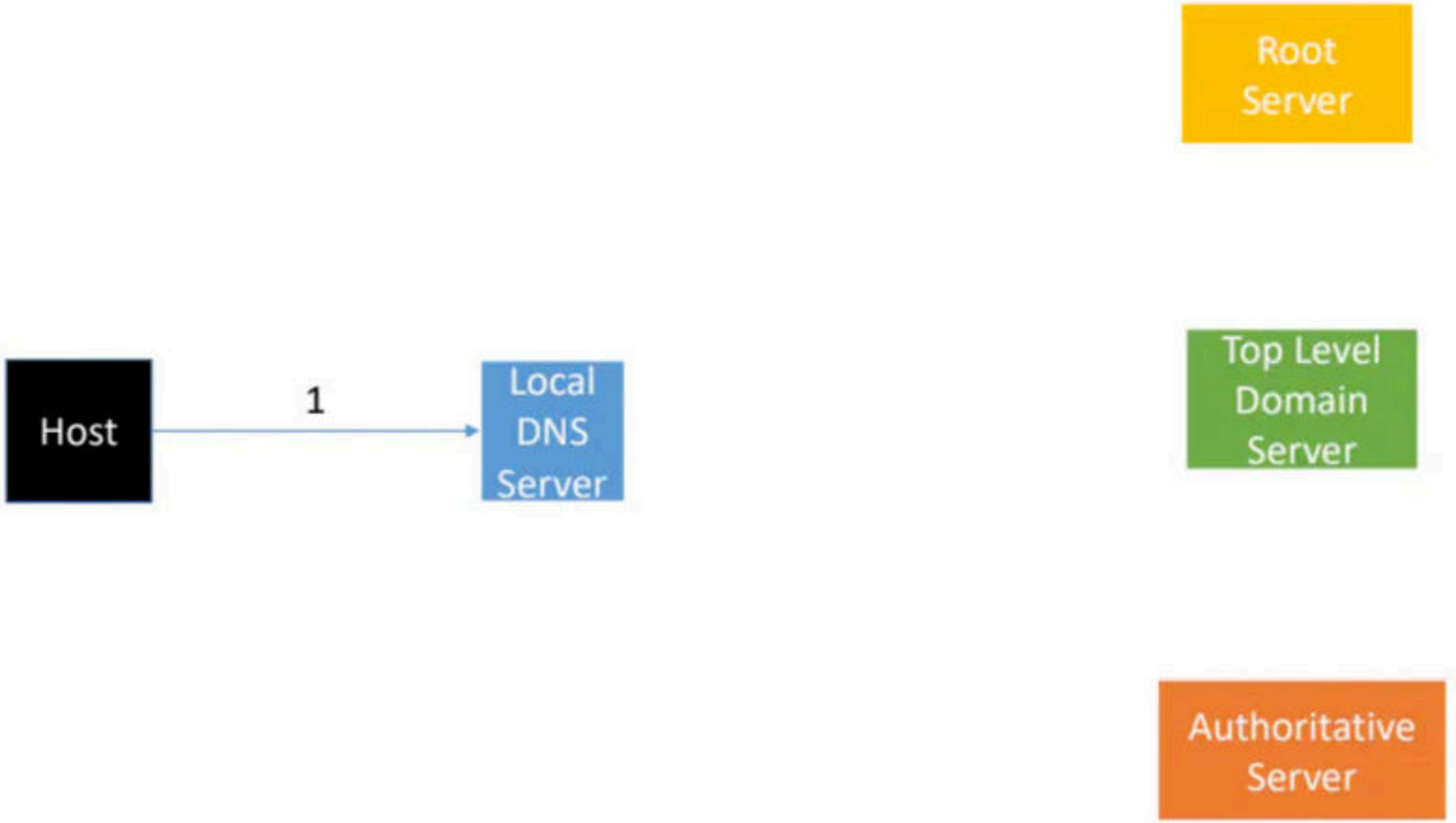
Let us assume a scenario where we wish to go to www.csa.iisc.in
There are two ways DNS can contact the server –
1.)Iterative way



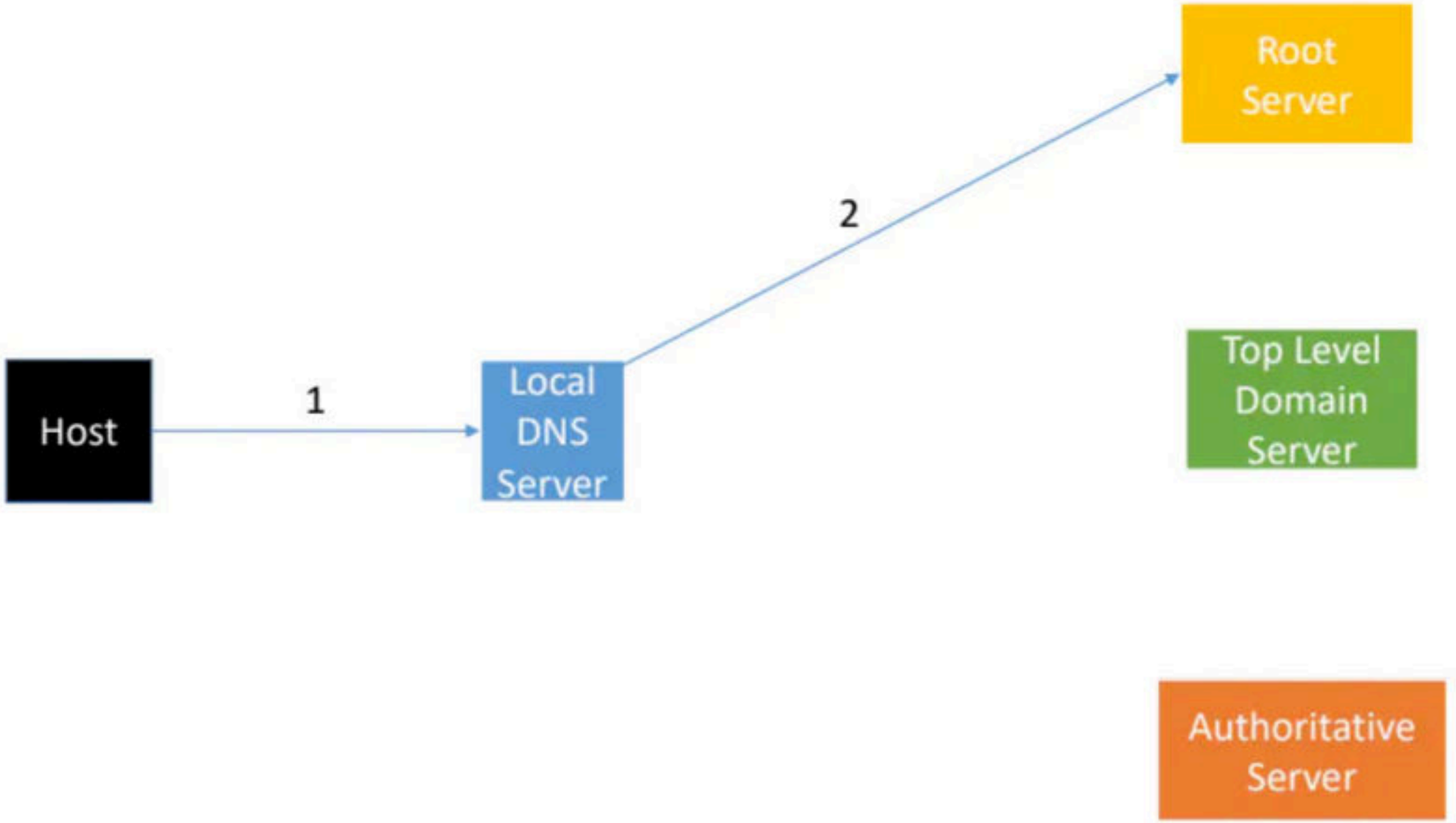
Let us assume a scenario where we wish to go to www.csa.iisc.in
There are two ways DNS can contact the server –
1.)Iterative way



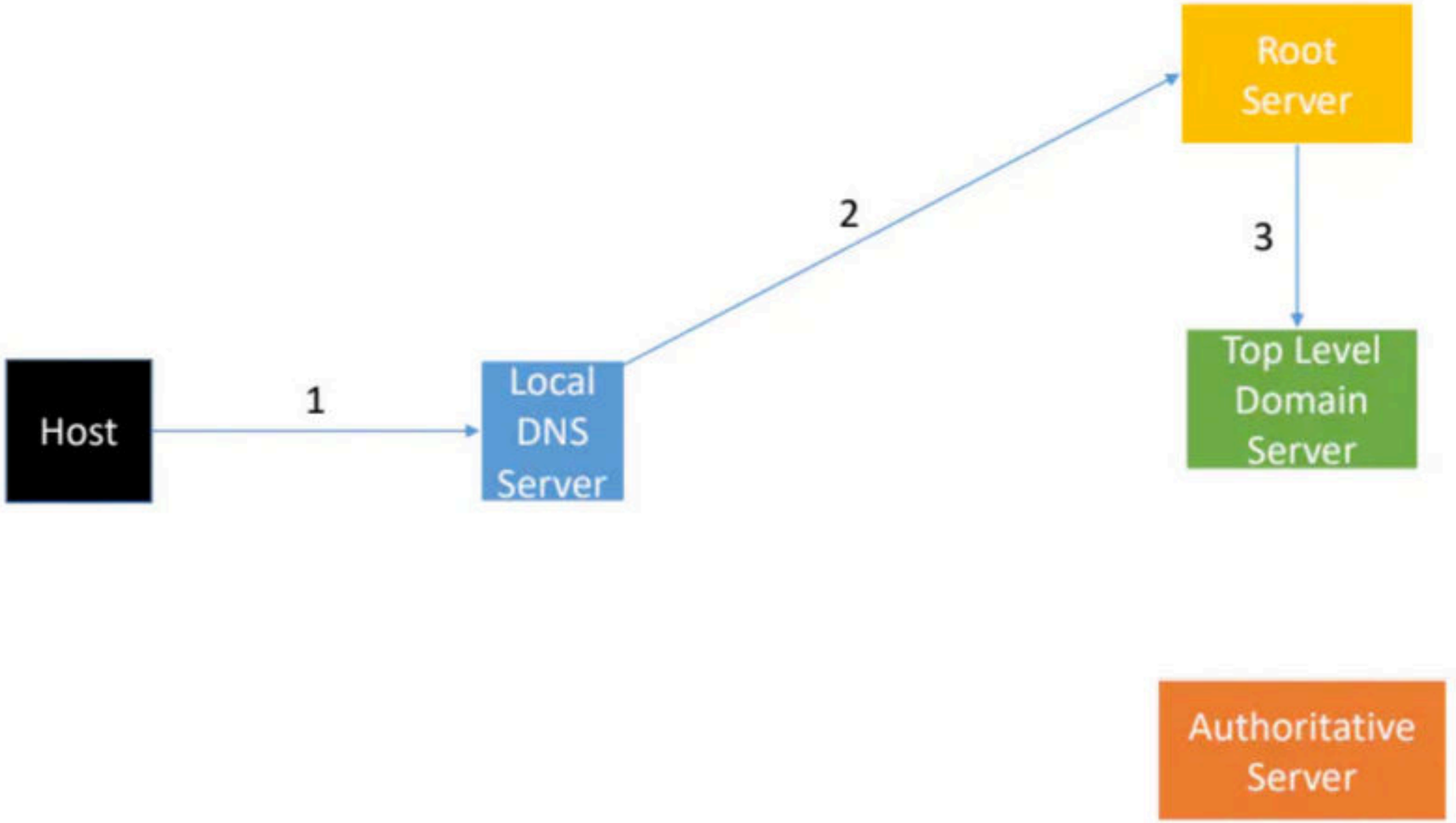
2.)Recursive way



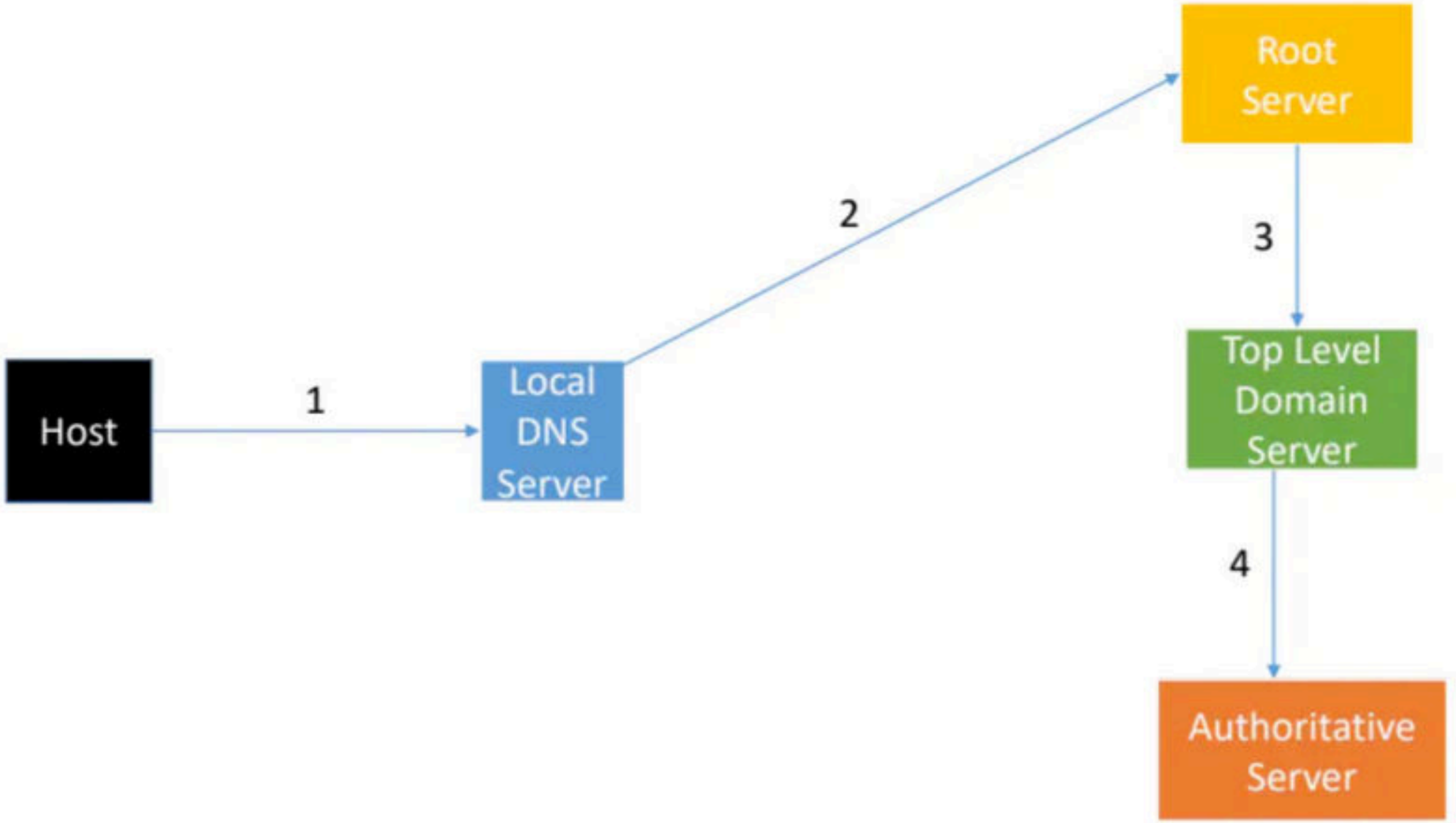
2.)Recursive way



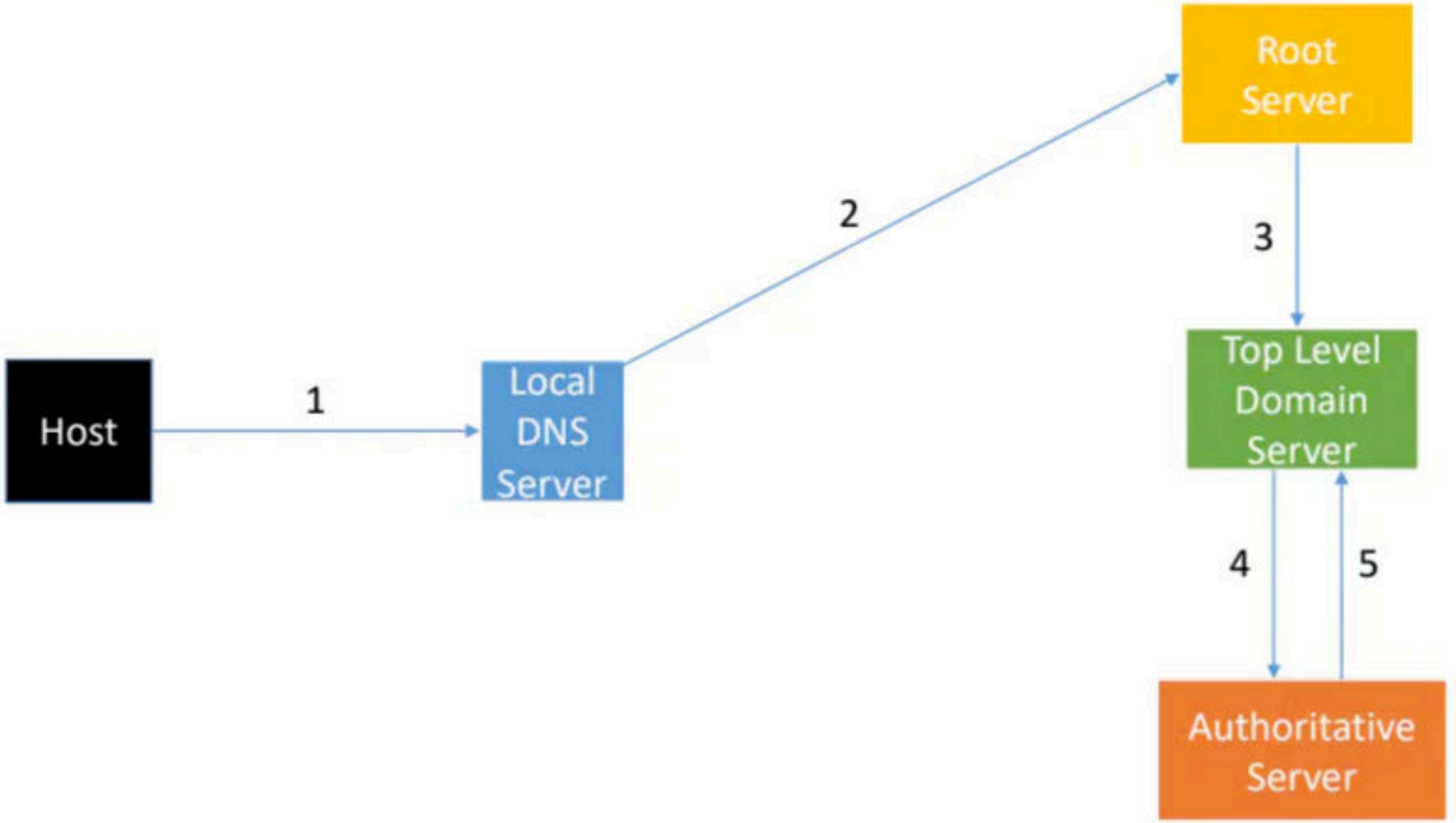
2.)Recursive way



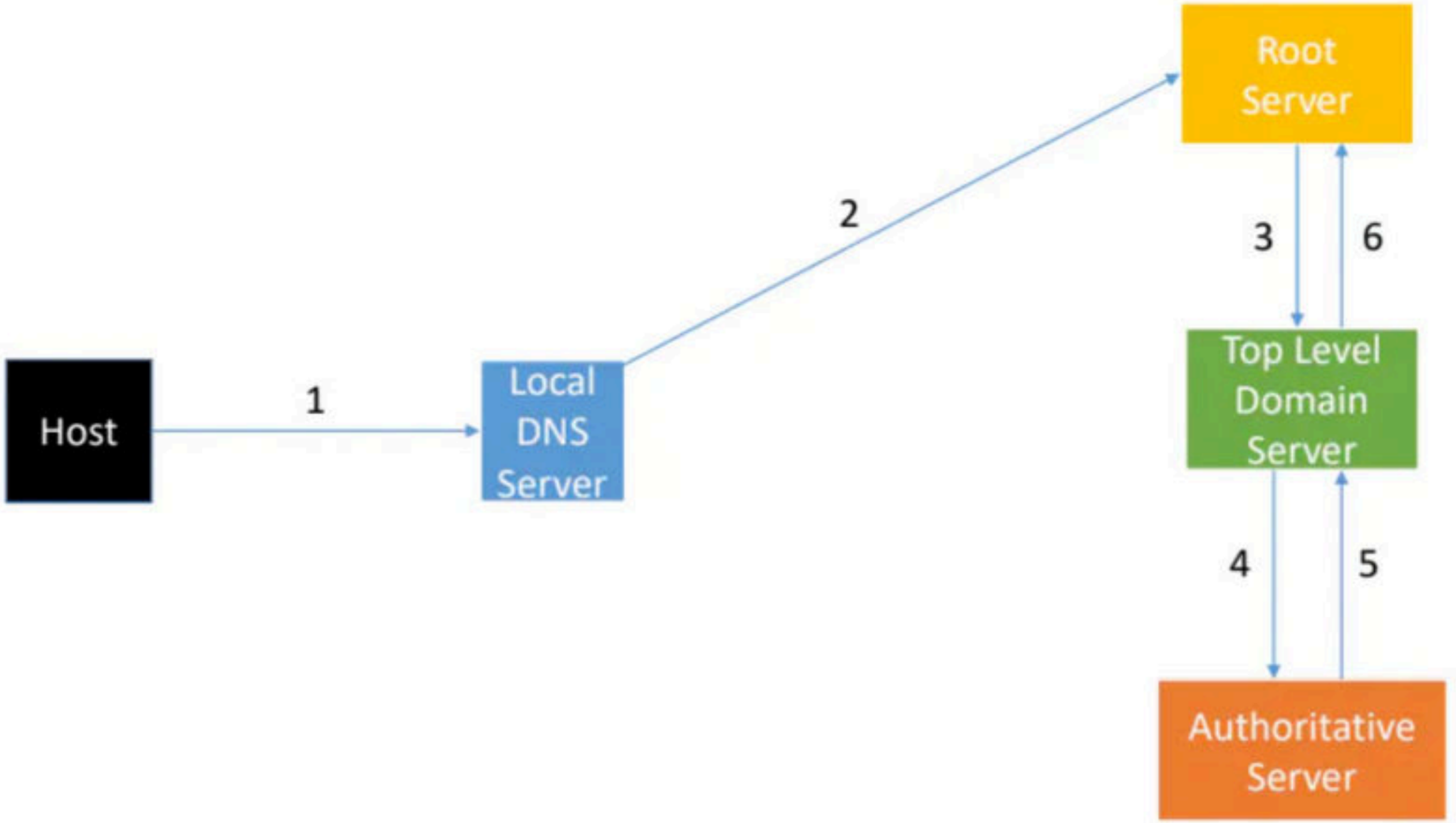
2.)Recursive way



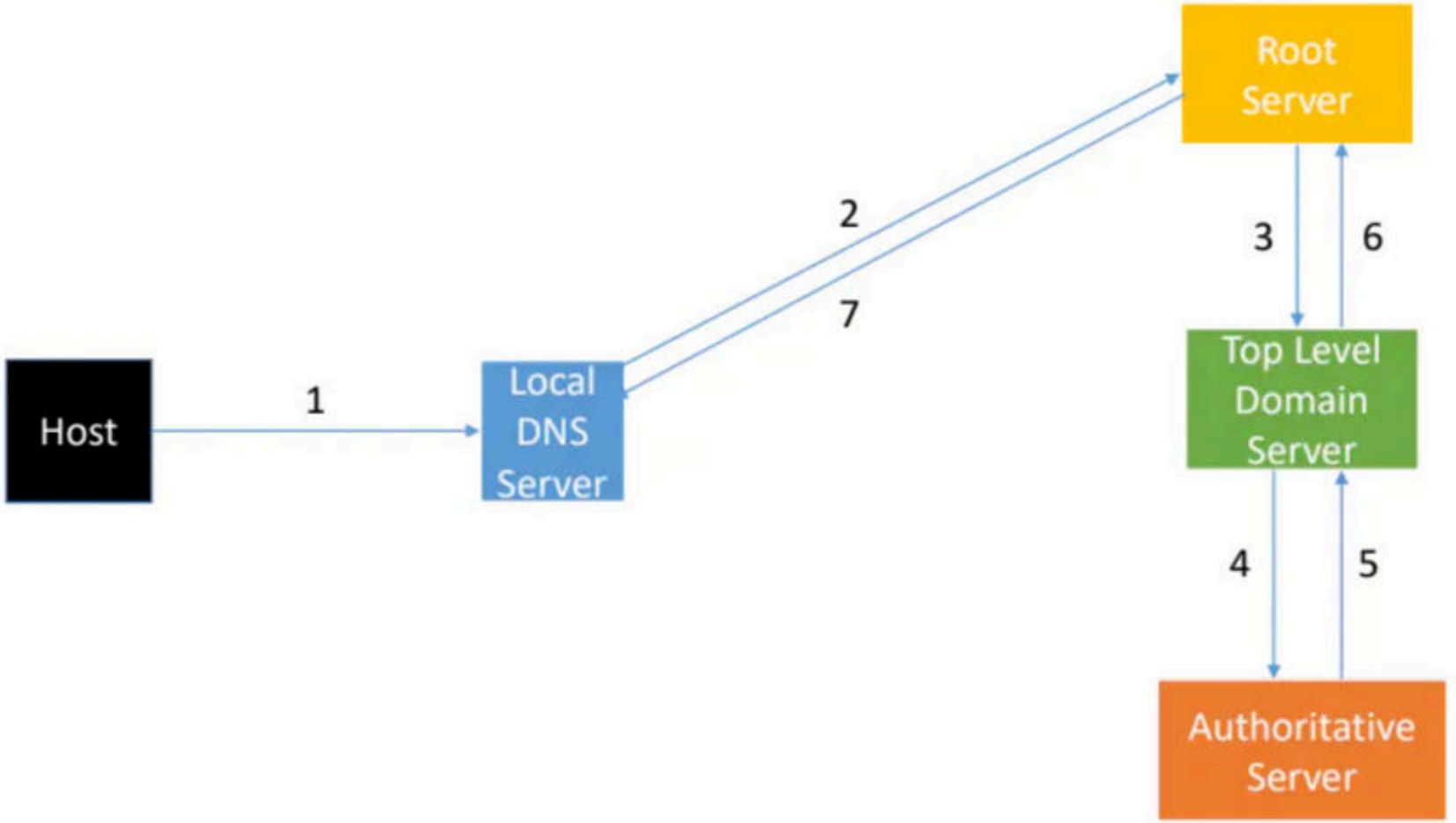
2.)Recursive way



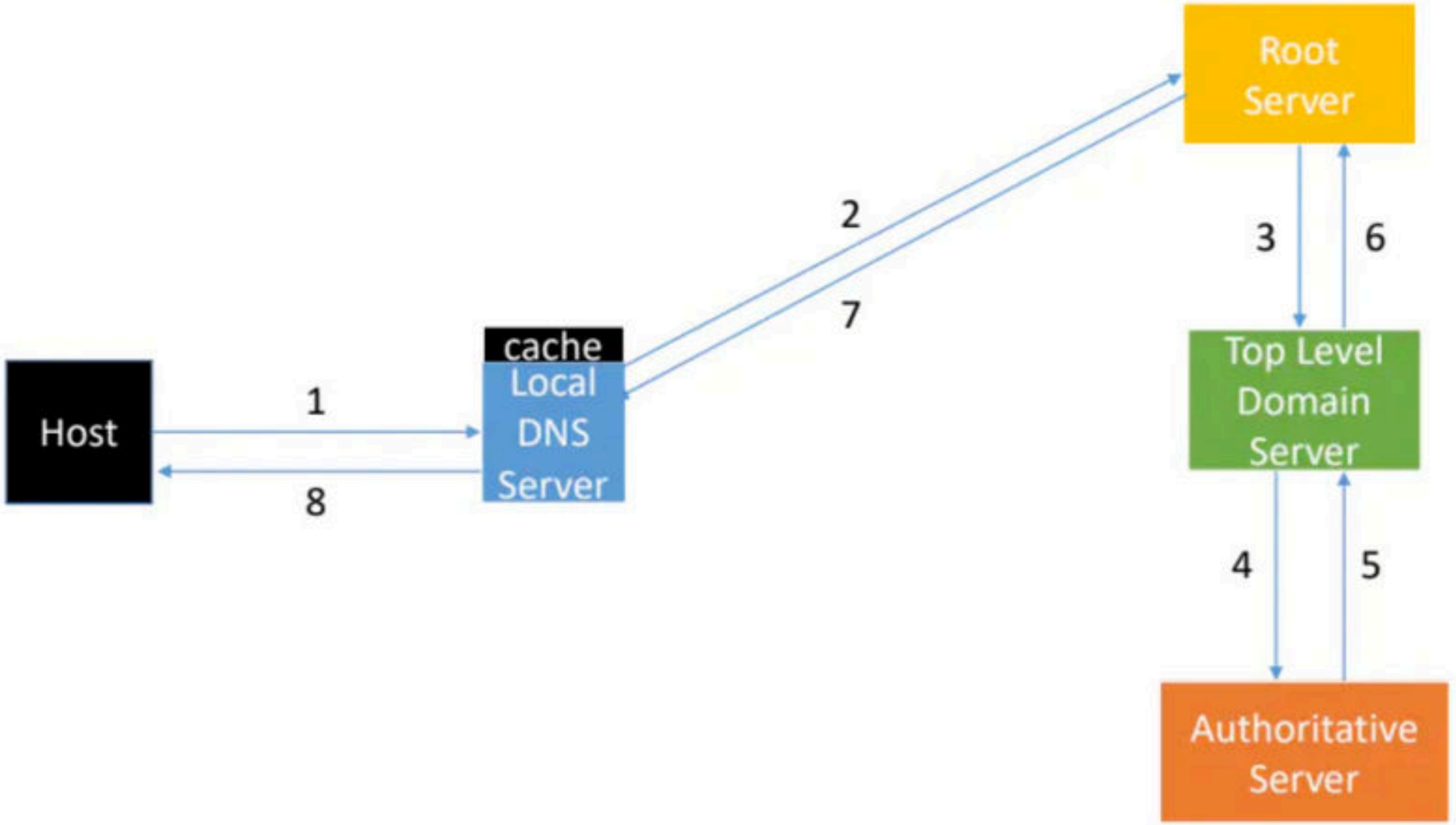
2.)Recursive way



2.)Recursive way



2.)Recursive way



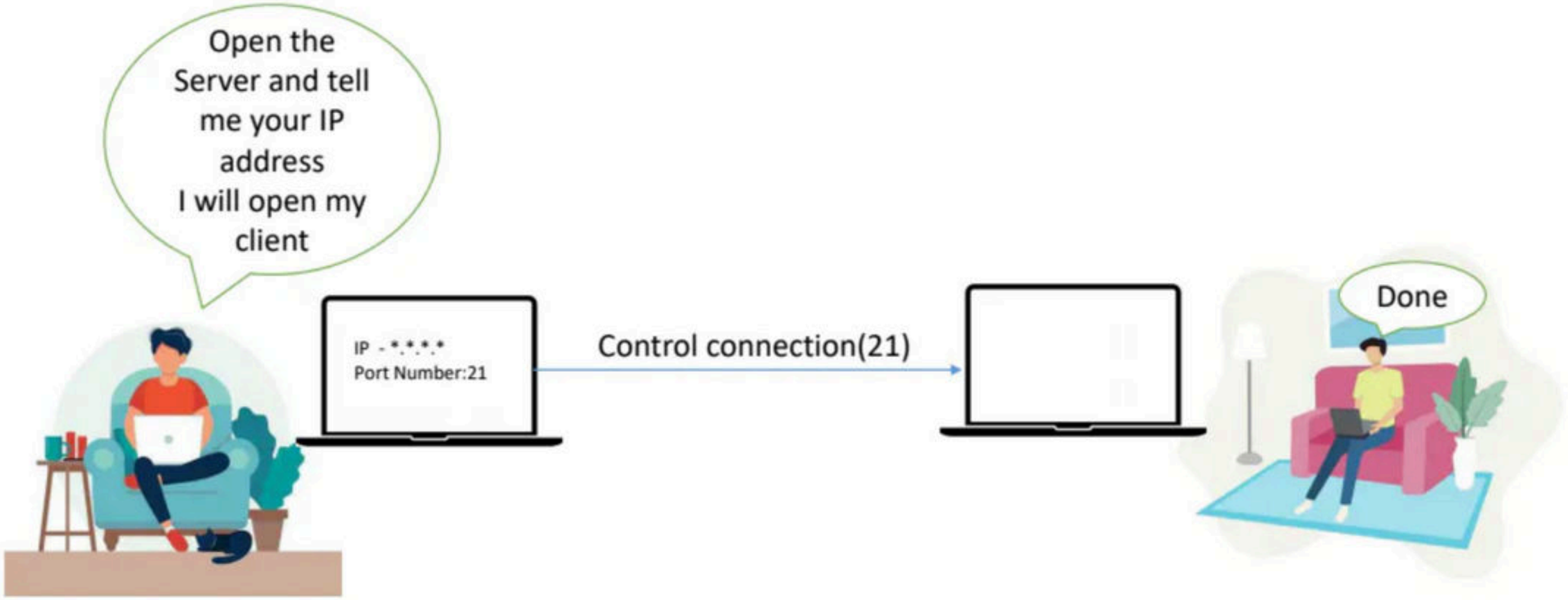
Computer Networks

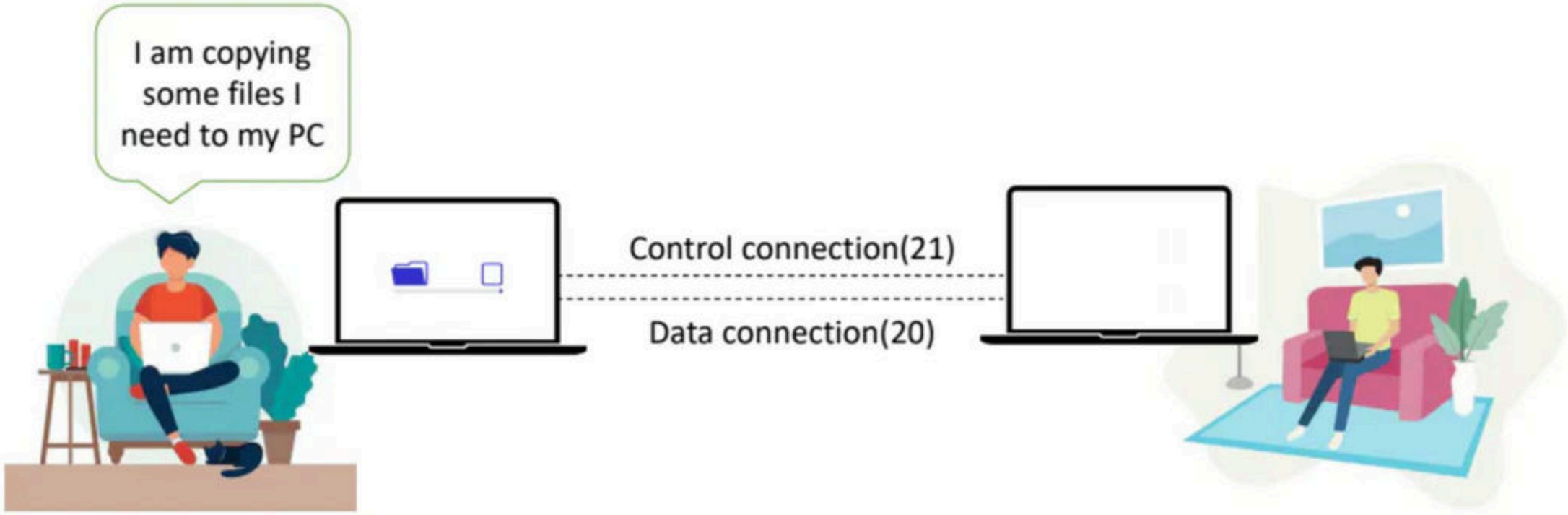
FTP

FTP

- FTP is short for File Transfer Protocol
- It is used for exchanging files over the internet.
- It enables the users to upload and download the files from the internet.
- FTP uses TCP at the transport layer.
- FTP uses port number 21 for control connection.
- FTP uses port number 20 for data connection.
- FTP uses persistent TCP connections for control connection.
- FTP uses non-persistent connections for data connection.
- FTP is a connection oriented protocol.
- FTP is an out-of-band protocol as data and control information flow over different connections.
- SMTP is a stateful protocol.


I wish to access your files ! Download Filezilla


Open the Server and tell me your IP address I will open my client
IP - *.*.*.*
Port Number:21
Control connection(21)
Done


I am copying some files I need to my PC
Control connection(21)
Data connection(20)

- FTP is the standard protocol provided by TCP/IP for copying a file from one host to another.

Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first.

Ex: Two systems may use different file name conventions

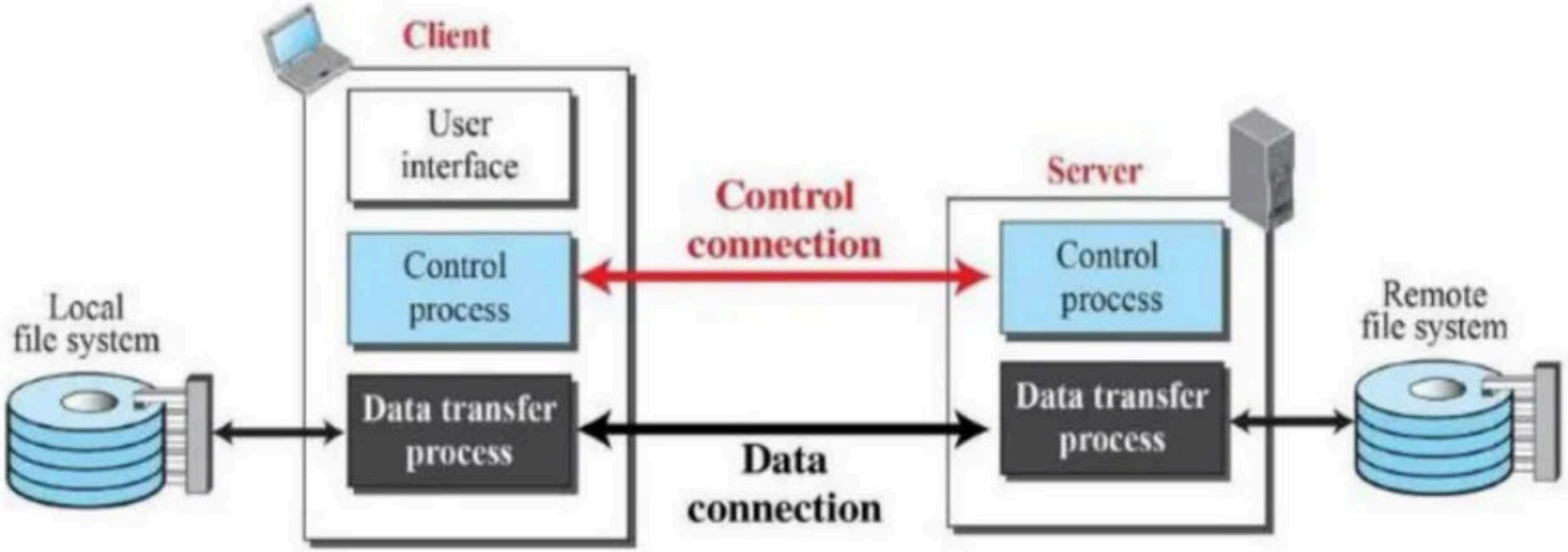
Two systems may have different ways to represent data.

Two systems may have different directory structures.

All of these problems have been solved by FFTP in a very simple and elegant approach.

Although we can transfer files using HTTP, FTP is a better choice to transfer large files or to transfer files using different formats

Basic model of FTP



- **The server has two components**: 1. The server control process.
  2. The server data transfer process

  The control connection is made between the control processes

  The data connection is made between the data transfer processes.

Separation of commands and data transfer makes FTP more efficient.

The control connection uses very simple rules of communication.

We need to transfer only a line of command or a line of response a time.

The data connection, on the other hand, needs more complex rules due to the variety of data types transferred.

Lifetimes of Two connections

The two connections in FTP have different lifetimes.
The control connection remains connected during the entire iterative FTP session.
The data connection is opened and then closed for each file transfer activity.
It opens each time commands that involve transferring files are used, and it closes when the file is transferred.

-> When a user starts an FTP session, the control connection opens.
While the control connections is open, the data connection can be opened and closed multiple times if several files are transferred.

Some FTP commands

Command	Argument(s)	Description
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subsidiaries or flies
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	Port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RNFR	File name(old)	Identify a file to be renamed

Continued.....

Command	Argument(s)	Description
RNTO	File name (new)	Rename the file
SNOR	File name(s)	Store files; file(s) are transferred from client to server
STRU	F, R, or P	Define data organization (F: file, R: record, or P: page)
TYPE	A, E, I	Default file type (A: ASCII, E: EBCDIC, I: image)`
USER	User ID	User information
MODE	S,B, or C	Define transmissionbhhhj

Some responses in FTP

Code	Description	Code	Description
125	Data connection open	250	Request file action OK
150	File status OK	331	User name OOK; password is needed
200	Command OK	425	Cannot open data connection
221	Service closing	452	Action aborted; insufficient storage
225	Data connection open	500	Syntax error; unrecognized command
226	Closing data connection	501	Syntax error in parameters or arguments
230	User login OK	530	User not logged in

Data connection:

The data connection uses the well-known port 20 as the server site.
However, the creation of a data connection is different from the control connection. The following shows the steps:

1. The client, not the server, issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.
2. The client sends this port umber to the server using the PORT command.
3. The server receives the port number and issues an active open using the well-known port 20 and the received ephemeral port number.

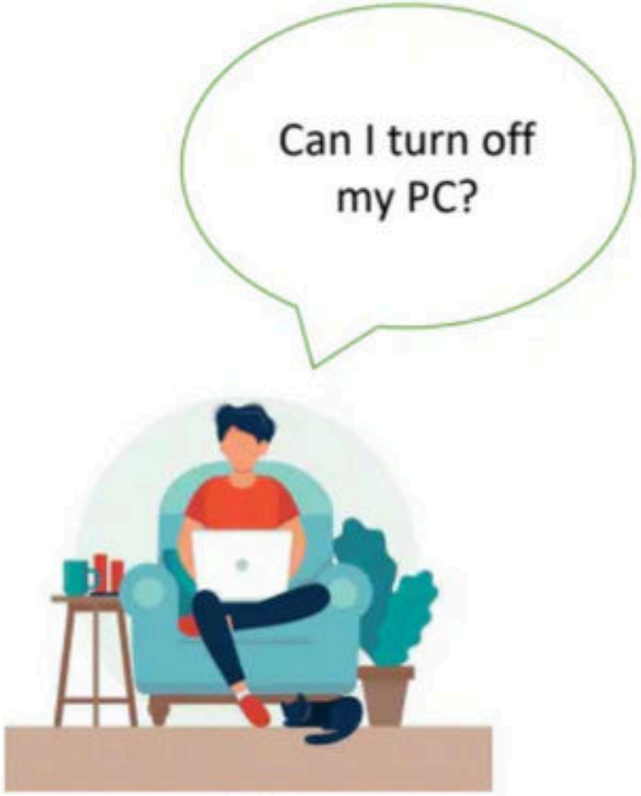
Communication over Data connection:

File type: ASCII file, EBCDIC file, or image file
Transmission Mode: Stream mode, Block mode, or compressed mode

Electronic Mail:

It is neither feasible nor logical for Bob to run a server program and wait until someone sends an email to him.

Bob may turn off his computer when he is not using it.



Bob

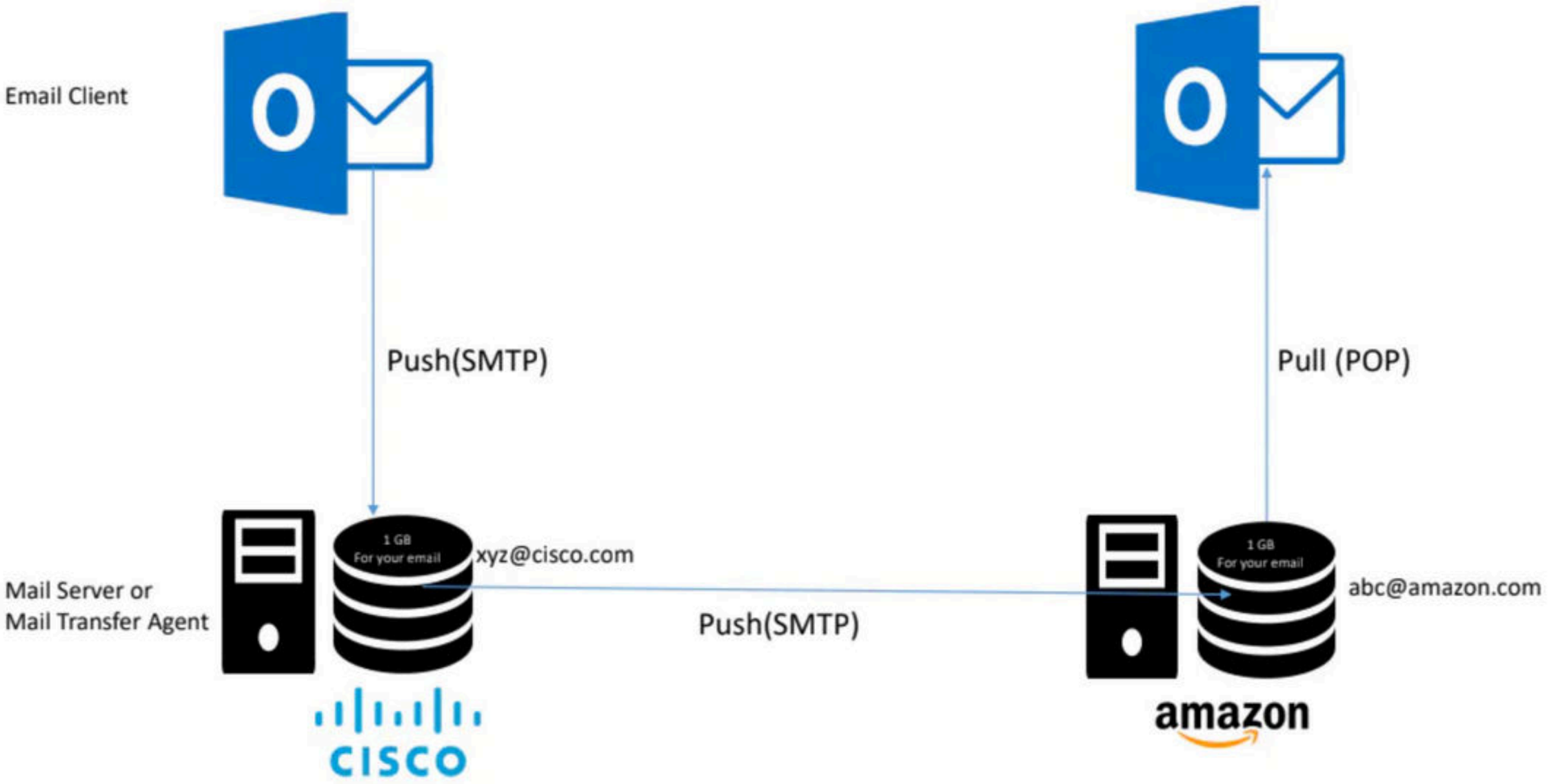
Computer Networks

SMTP - POP

SMTP and POP

- SMTP is short for Simple Mail Transfer Protocol.
- It is an application layer protocol.
- It is used for sending the emails efficiently and reliably over the internet.
- SMTP is a push protocol.
- SMTP uses TCP at the transport layer.
- SMTP uses port number 25.
- SMTP uses persistent TCP connections, so it can send multiple emails at once.
- SMTP is a connection oriented protocol.
- SMTP is an in-band protocol.
- SMTP is a stateless protocol.

- POP is a pull protocol.
- POP uses TCP at the transport layer.
- POP uses port number 110.
- POP uses persistent TCP connections.
- POP is a connection oriented protocol.
- POP is an in-band protocol.
- POP is a stateful protocol until the mail is downloaded as well as stateless across sessions.

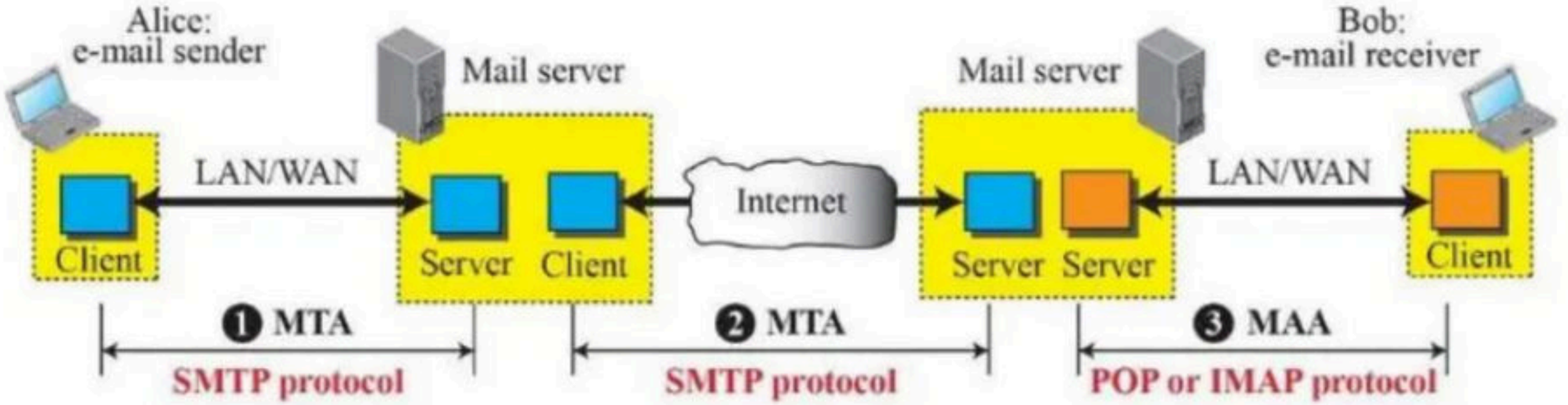

Email Client
Push(SMTP)
Pull (POP)
1 GB
For your email
xyz@cisco.com
Mail Server or
Mail Transfer Agent
Push(SMTP)
1 GB
For your email
abc@amazon.com
CISCO
amazon

Multipurpose Internet Mail Extension (MIME) Protocol

Later people needed to send images, videos, graphs and basically all types of non text content
For that purpose we moved on to MIME

- MIME is able to send multiple attachments with a single message.
- Unlimited message length.
- Binary attachments (executables, images, audio, or video files) which may be divided if needed.

Protocols used in electronic mail:



POP 3: Two modes

i) Delete mode: The mail is deleted from the mail box after each retrieval.

Normally used when the user is working at her permanent computer and can save and organize the received mail after reading or replying.

ii) Keep mode: The mail remains in the mailbox after retrieval.

Normally used when the user accesses his mail away from his primary computer.

The mail is read but kept in the system for later retrieval and organizing.

IMAP4:

Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.

POP 3 is deficient in the following ways:

- Does not allow user to organize her mail on the server.
- The user cannot have different folders on the server.
- Does not allow the
- User to partially check the contents of the mail before downloading.

IMAP4 provides the following extra functions

- A user can check the e-mail header prior to downloading.
- A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- A user can partially download e-mail. This is , or rename mailboxes in a folder for e-mail storage useful if bandwidth is limited and email contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes in a folder for e-mail storage.

INTRODUCTION

- HTTP is an acronym for Hypertext Transfer Protocol . It is an application layer protocol in the TCP/IP protocol suite.
- WWW is repository of resources (web pages) stored in different computers all over the world. The primary purpose of HTTP is to transfer web pages from one computer (web server) to another computer(web client).
- HTTP can be used to access virtually all types of resources on the web. It allows us to transfer a wide variety of data such as text ,image ,audio , video and even the result of a query.

WEB SERVERS AND CLIENTS

- HTTP protocol is basically a request- response protocol between clients and servers.
- According to this protocol , a process is run which creates and stores resources such as HTML files, images ,etc. This process provides the resources on request and is called a web server or HTTP server.

To access resources stored on the web server, a process is designed to communicate with the server. The process is called web client or user agent.

- Web browsers are nothing but those web clients . They use HTTP protocol to communicate with web server to access resources specified by the URL at the address bar of the web browsers.

- Web servers typically runs on port 80 though any available port number may be used.

In TCP/IP protocol suite, a process in a machine is assigned a locally unique positive integer called port number.

- IP address together with the port number uniquely identifies a process all over the world.

So, IP address together with the port number uniquely identifies a process all over the world. This (IP address , port) pair is called a socket address of the process . To communicate with a web server , web clients need to specify the socket address of the web server.

URL AND ITS ANATOMY

A resource on the web is identified by an address called Uniform Resource Locator (URL).

An HTTP URL has the following form:

protocol : //host : [port]/[path[? params] [#anchor]]

protocol : //host : [port]/[path[? params] [#anchor]]

It indicates the protocol to be used for this URL. For HTTP URL , the protocol is http. Other possible protocols are ftp, gopher, mailto ,news ,nntp , telnet , wais ,file and prospero.

protocol : //host : [port]/[path[? params] [#anchor]]

This is the Fully Qualified Domain Name (FQDN) or the IP address of the computer where the web server runs.

Domain names are case –insensitive .So, www.google.com and WWW.GOOGLE.COM refer to the same host.

protocol : //host : [port]/[path[? params] [#anchor]]

Web servers typically run on port 80, if no port is specified ,port 80 is assumed.

protocol : //host : [port]/[path[? params] [#anchor]]

- This is the location of a file or a program (CGI ,Perl ,PHP ,JSP , etc) on the server relative to a document root specified by the web server. The document root is a directory where resources are stored.
- In the Apache web server ,the document root is usually set to /var /www/html. However , it can be configured.

protocol : //host : [port]/[path[? params] [#anchor]]

- This portion of the URL contains the parameters to be passed to web applications such as CGI, PERL, PHP or JSP.

- The path and params are separated by ? character. The params consists of a (name=value) pair separated by an ampersand(&).

For example, login= ukr&sid=145321& page=inbox specifies three parameters , login, sid and page whose values are “ukr ”, “145321” and “inbox” respectively.

- The server - side program typically extracts this information from the URL for processing.

protocol : //host : [port]/[path[? params] [#anchor]]

This part indicates a specific location in the web page. For example, #appendix specifies the appendix section of the web page. This location is created using <a> tag as follows:

 Appendix
The named section can then be referred to as :
http:// www.it.jusl.ac.in/httphelp.htlm#appendix

URL EXAMPLE


protocol
host
http://www.unacademy.com
protocol
host
port
path
http://www.unacademy.com:8080/@ravula
protocol
host
port
path
http://123.456.789.012:8080/@ravula
protocol
host
port
path
anchor
http://www.unacademy.com:8080/@ravula#subscribe
protocol
host
port
path
anchor
params
http://www.unacademy.com:8080/@ravula#subscribe?referral_code=rrcs

MESSAGE FORMAT

It specifies a set of rules that clients and servers use to communicate:

- An HTTP server process is created on a port (usually 80) , which waits for clients to establish a TCP connection.

- An HTTP client initiates a TCP connection with the HTTP server (process) at the designated port.

- The HTTP server accepts this connection.

- The HTTP client then sends a request for a resource to the server.

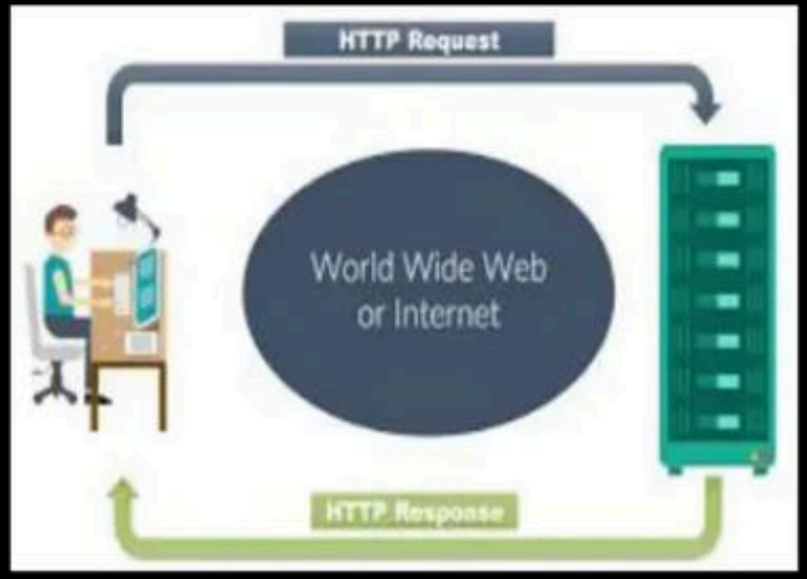
- Upon receiving the request , the server processes the request ,performs the desired task, and sends a response back to the client.

- The HTTP server closes the TCP connection.

- The HTTP client receives the response containing information and processes it.

Note that every time the HTTP client wants to get resource from the server, it has to follow these steps .

This makes HTTP stateless .This means that web server treats every request as a new request . There is no way to specify that some requests are related.



REQUEST MESSAGE

A request message is sent by a web client to the web server . It consists of the following parts:

- A request line.
- A header.
- An empty line.
- An optional body.

Request line
Header
Empty Line
Body(available for some messages)

HTTP request message format

REQUEST LINE

A request line consists of three parts : request type, URL, and HTTP version . Two consecutive parts are separated by a space.

Request Type		URL		HTTP Version

REQUEST TYPE (METHOD)

- It indicates the type of request , a client wants to send. They are also called methods.
- A method makes a message either a request or a command to the server.
- Request messages are used to retrieve data from the server whereas a command tells the server to do a specific task.

GET

- This is the most frequently used method in the WWW. It is specified when client wants to retrieve (GET) a resource(document) from the server.
- The URL in the request line identifies the resource . If the URL specified is a valid one , the server reads the content of the resource and sends the content back to the client; otherwise an error message is sent back to the client.
- If the resource being requested is a server-side program such as CGI script or ASP or JSP , the result generated by the program is returned instead of the content of the resource.
- The message body is empty for the GET method.
- The GET method may be a “conditional GET” method. In such a case, the request message includes an “If-Modified-Since” header that specifies a date . This header specifies that the identified resource has to be transferred only if it is modified after the specified date. So users can avoid downloading resources that were downloaded on some earlier date and have not been modified since.
- The conditional GET method effectively reduces network bandwidth and helps increasing network performance.

HEAD

- It is used when the client wants to know the header information (meta-information) about a resource but not the resource content.

POST

- It is used when a client wants to send (POST) some information (possibly large) to the server.
- The actual information is included in the body part of the request message instead of appending it to the URL as done in the GET method.
- The headers describe the message body such as content type and content length.
- The commonest form of the POST method is to submit an HTML form to the server.

PUT

- It is used to upload a new resource or replace an existing document. The actual document is specified in the body part.
- As the PUT method can modify or replace an existing document, it is vulnerable and is not permitted by most of the web servers.
- However, it is not recommended to configure web servers in such a way without any valid reasons.

PATCH

- This is similar to PUT method except that it specifies a list of differences that must be applied on the existing file.

COPY

- The HTTP protocol may be used to copy a file from one location to another.
- The method COPY is used for this purpose. The URL specified in the request line specifies the location of the source file.
- The location of the target file is specified in the entity header.
- Note that the target web server must be configured properly to accept the COPY method.
- This method is also vulnerable.

MOVE

- It is similar to the COPY method except that it deletes the source file.
- The location of the source file is specified by the URL in the request line.
- The entity header specifies the location of the target file.
- Note that the target web server must be configured properly to accept the MOVE method.
- This method is also vulnerable.

DELETE

- This method is used to remove a document from the server.
- The location of the document to be deleted is specified by the URL in the request line.
- This method is vulnerable.

LINK

- This is used to create a link or links from one document to another.
- The URL in the request line specifies the location of the source file and the entity header specifies the location of the target document.

UNLINK

- It is used to remove a link or links created by the LINK method.

OPTIONS

- It is used to retrieve the set of methods supported by the server.
- It is used to check whether a server is functioning properly before performing other tasks.
- It is used to check whether the web server really supports a method before actually using that method.

CONNECT

- It is used to convert a request connection into the transparent TCP/IP tunnel.
- It is usually done to facilitate Secured Socket Layer(SSL) encrypted communication(eg HTTPS) through an unencrypted HTTP proxy server.

TRACE

- It is used to instruct the web server to echo the request back to the client.
- The client can then see what additions or change are done by the immediate servers.

SAFE AND UNSAFE METHODS

- Among the methods discussed ,some methods such as GET ,HEAD, OPTIONS and TRACE are used only to retrieve information from the server . Such methods are called safe. They cannot change the state of the server . This means, they do not have any harmful side effects such as caching ,logging ,etc .
- On the other hand, methods such as DELETE,MOVE,UNLINK take actions that may change the state of the server.

These methods have harmful side effects and hence are vulnerable.

Sensitive web servers are usually not configured to accept these methods.

HTTP VERSIONS

This field specifies the version of the HTTP protocol being used. It can have the following values: HTTP/1.0 and HTTP/1.1 .

The current version is HTTP/1.1.

RESPONSE MESSAGE

In response to the request message , a response message is sent by a server to the client. It consists of the following parts:

- A status line
- A header
- An empty line
- An optional body

Status line
Header
Empty Line
Body(available for some messages)

HTTP response message format

STATUS LINE

Status line consists of three parts: HTTP version ,Status code and Status phrase. Two consecutive parts are separated by a space.

HTTP Version		Status Code		Status Phrase

STATUS CODE

It is a three –digit code that indicates the status of the response. The status codes are classified with respect to their functionality into five groups:

- 1 xx series (Informational) – This class of status codes represents provisional responses.
- 2 xx series (Success) – This class of status codes indicates that the client's requests are received, understood and accepted successfully.
- 3xx series(Redirectional) – These status codes indicates that additional actions must be taken by the client to complete the request. The user agent may take further actions in order to fulfill the request automatically provided that it uses either the HEAD or the GET method.
- 4xx series(Client error) – These status codes are used to indicate that the client request had an error and therefore it cannot be fulfilled. Except for the HEAD method, the body of the response message contains the explanation that caused the error. The user agent should display the error message to inform the user.
- 5xx series(Server Error) – This set of status code indicates that the server encountered some problem and hence the request cannot be satisfied at this time. The reason of the failure is embedded in the message body . It also indicates whether the failure is temporary or permanent.

HEADERS

- HTTP headers are very important part of both request message and response message.
- They collectively specify the characteristics of the resource requested and the data that are provided.
- For Example, a client may want to accept image files only in specified format.
- The headers are separated by an empty line from the request and response body.

General Header
Request Header
Entity Header

HTTP Request Header Format

General Header
Response Header
Entity Header

HTTP Response Header Format

Computer Networks

Devices in Computer Networks

CABLES

The cable is a physical media, through which an analog and digital data transfer take place

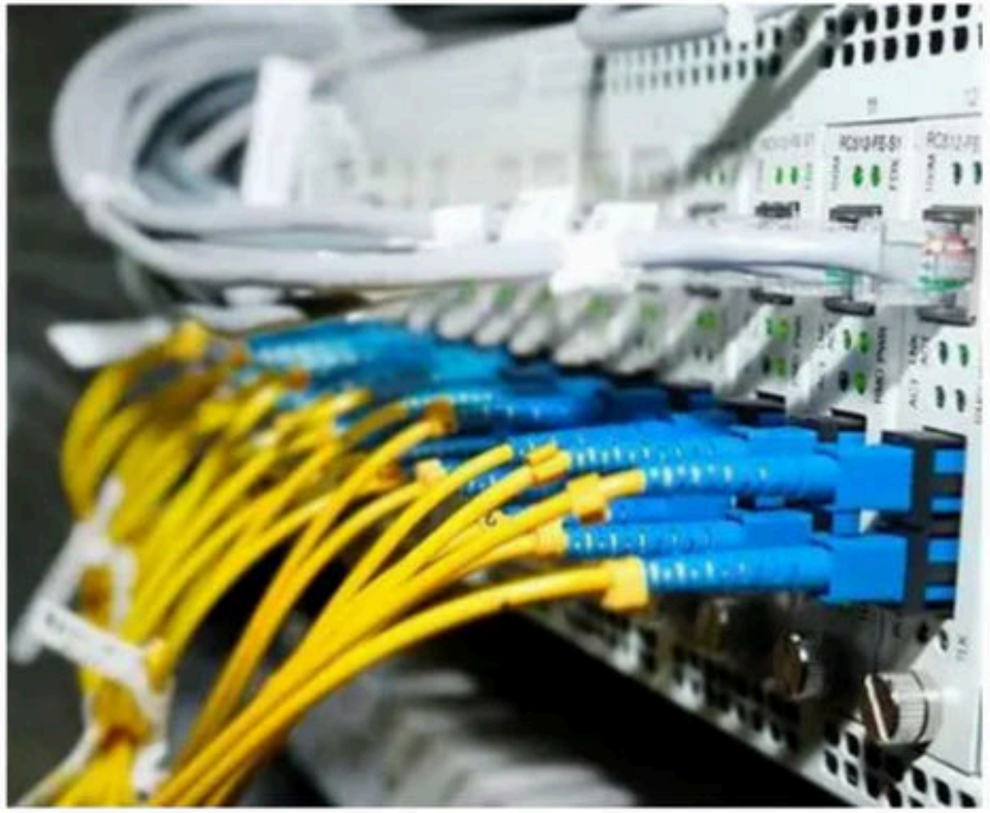
Types of Cables

- Twisted pair
- Coaxial
- Optical fiber

Operate at Physical Layer.

Have a problem of attenuation.

Collisions possible (Collision Domain is n)



REPEATER

A repeater operates at the physical layer.

Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network.

An important point to be noted about repeaters is that they do not amplify the signal.

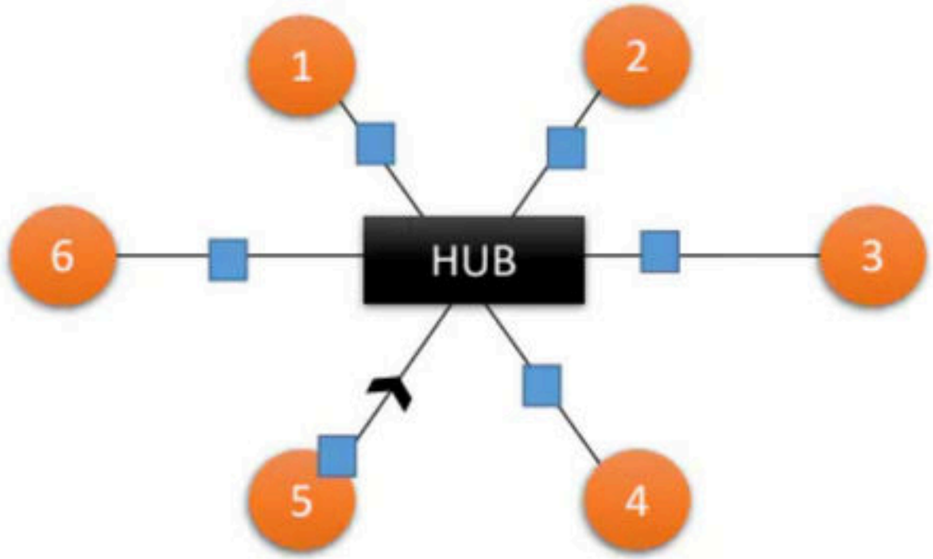
When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength.

It is a 2 port device.

Collisions possible

HUBS

- A hub is basically a multiport repeater.
- A hub connects multiple wires coming from different branches
- Hubs cannot filter data, so data packets are sent to all connected devices.
- Traffic is High
- Collision are possible

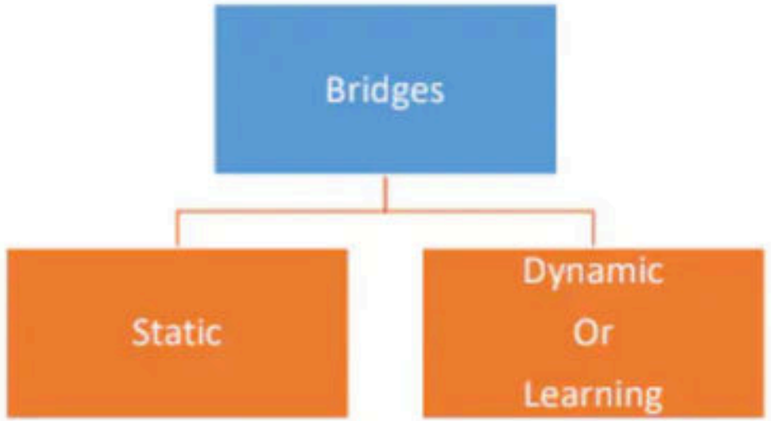


BRIDGE

- A bridge operates at Physical and data link layer.
- A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of source and destination.
- Features in include Filtering ,forwarding and flooding
- It is also used for interconnecting two LANs working on the same protocol.
- Collision Domain is reduced
- Store and Forward

BRIDGE

MAC	PORT
1	a
2	a
3	a
4	b
5	b
6	b



SWITCH

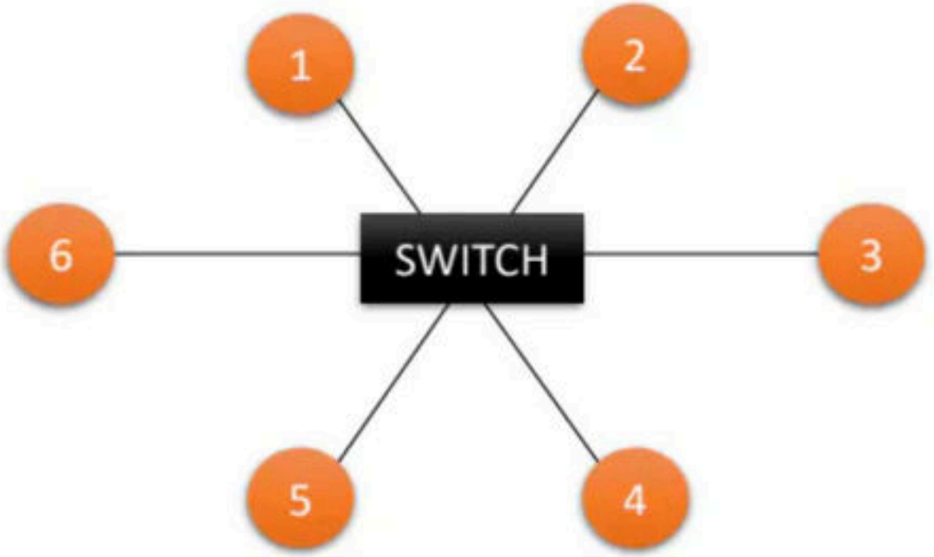
A switch is a multiport bridge with a buffer and a design that can boost its efficiency(a large number of ports imply less traffic) and performance.

A switch is a Physical and data link layer device.

The switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors.

No collisions

Traffic is low



ROUTER

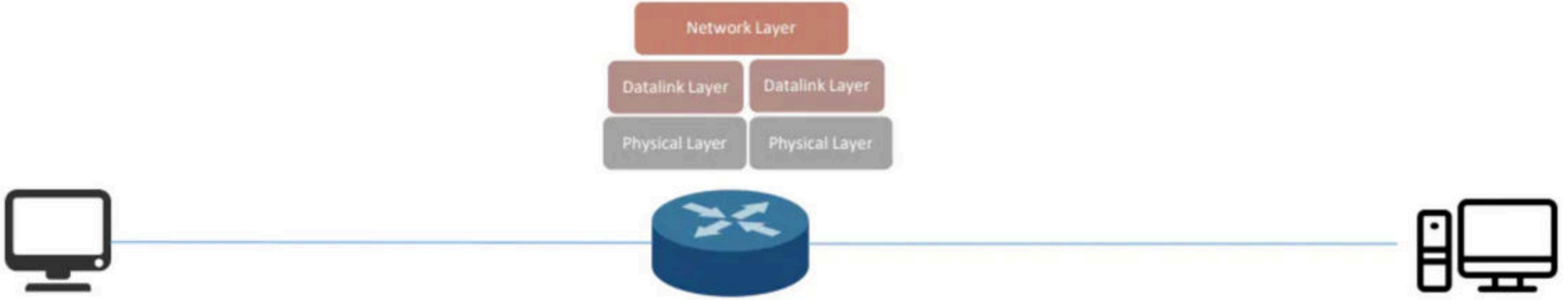
A router is a device that routes data packets based on their IP addresses.

Router is mainly a Network Layer device.

Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets.

Features include Filtering, Flooding and Forwarding

No collisions



DEVICES	COLLISION DOMAIN	BROADCAST DOMAIN
Repeater	Same	Same
Hub	Same	Same
Bridge	Same	Reduces
Switch	Same	Reduces
Routers	Reduces	Reduces

Computer Networks

Revision of Application Layer and GATE Questions Part 3

REVISION

	DNS	HTTP	SMTP	POP	FTP
Stateful / Stateless	Stateless	Stateless	Stateless	Stateful	Stateful
Transport Protocol Used	UDP	TCP	TCP	TCP	TCP
Connectionless / Connection Oriented	Connectionless	Connectionless	Connection Oriented	Connection Oriented	Connection Oriented
Persistent / Non-persistent	Non-persistent	HTTP 1.0 is non-persistent. HTTP 1.1 is persistent.	Persistent	Persistent	Control connection is persistent. Data connection is non-persistent.
Port Number Used	53	80	25	110	20 for data connection. 21 for control connection.
In band / Out-of-band	In band	In band	In band	In band	Out-of-band

1.) Consider a TCP connection between a client and a server with the following specifications: the round trip time is 6 ms, the size of the receiver advertised window is 50 KB, slow start threshold at the client is 32 KB, and the maximum segment size is 2 KB. The connection is established at time t=0. Assume that there are no timeouts and errors during transmission. Then the size of the congestion window (in KB) at time t+60 ms after all acknowledgements are processed is _____. [GATE 2020]

1.) Consider a TCP connection between a client and a server with the following specifications: the round trip time is 6 ms, the size of the receiver advertised window is 50 KB, slow start threshold at the client is 32 KB, and the maximum segment size is 2 KB. The connection is established at time t=0. Assume that there are no timeouts and errors during transmission. Then the size of the congestion window (in KB) at time t+60 ms after all acknowledgements are processed is _____. [GATE 2020]

SOLUTION:
Threshold = 32 Kb, MSS = 2KB, RTT = 6ms
Here, t + 60 is nothing but at the 10 RTT (60/6 = 10), but here
it's asking after all acknowledgement are processed it means
after the 10th RTT, .i.e at the 11RTT
1st transmission: 2 KB
2nd transmission: 4 KB
3rd transmission: 8 KB
4th transmission: 16 KB
5th transmission: 32 KB (Threshold reached)
6th transmission: 34 KB
7th transmission: 36 KB
8th transmission: 38 KB
9th transmission: 40 KB
10th transmission: 42 KB
At the completion of 10th transmission RTT = 10*6 = 60 ms
For the 11th transmission, The congestion window size is 44 KB

2.)Consider the following statements regarding the slow start phase of the TCP congestion control algorithm. Note that cwnd stands for the TCP congestion window and MSS denotes the Maximum Segment Size.

(i) The cwnd increase by 2 MSS on every successful acknowledgement.
(ii) The cwnd approximately doubles on every successful acknowledgedment.
(iii) The cwnd increase by 1 MSS every round trip time.
(iv) The cwnd approximately doubles every round trip time.
Which one of the following is correct?

A.)Only (ii) and (iii) are true
B.)Only (i) and (iii) are true
C.)Only (iv) is true
D.)Only (i) and (iv) are true [GATE 2018]

2.)Consider the following statements regarding the slow start phase of the TCP congestion control algorithm. Note that cwnd stands for the TCP congestion window and MSS denotes the Maximum Segment Size. [GATE 2018]

(i) The cwnd increase by 2 MSS on every successful acknowledgement.
(ii) The cwnd approximately doubles on every successful acknowledgement.
(iii) The cwnd increase by 1 MSS every round trip time.
(iv) The cwnd approximately doubles every round trip time.
Which one of the following is correct?

A.)Only (ii) and (iii) are true
B.)Only (i) and (iii) are true
C.)Only (iv) is true
D.)Only (i) and (iv) are true

SOLUTION:
In Slow-start, the value of the Congestion Window will be increased by 1 MSS with each acknowledgement (ACK) received, and effectively doubling the window size each round-trip time
Initially, TCP starts with cwnd of 1 MSS. On every ack, it increases cwnd by 1 MSS.
That is, cwnd doubles every RTT.
Initially sends 1 segment. On ack, sends 2 segments.
After these 2 acks come back, sends 4 segments etc.
TCP rate increases exponentially during slow start.
Slow start continues till cwnd reaches threshold.
After threshold is reached, cwnd increases more slowly, by one 1 MSS every RTT.

3.) Which one of the following protocols is NOT used to resolve one form of address to another one?

A.)DNS
B.)ARP
C.)DHCP
D.)RARP [GATE 2016]

3.) Which one of the following protocols is NOT used to resolve one form of address to another one?

A.)DNS
B.)ARP
C.)DHCP
D.)RARP [GATE 2016]

SOLUTION:
DHCP is dynamic host configuration protocol: allocates one of the unused IP address.
Except DHCP, remaining all the protocols are used to resolve one form of address to another one.
I. DNS is going to convert hostname to IP address.
II. ARP is going to convert IP to MAC.
III. DHCP is going to assign IP dynamically.
IV. RARP is going to convert MAC to IP.

4.)Which of the following is/are example(s) of stateful application layer protocols?
(i) HTTP
(ii) FTP
(iii) TCP
(iv) POP3
A.) (i) and (ii) only
B.) (ii) and (iii) only
C.) (ii) and (iv) only
D.) (iv) only [GATE 2016]

4.)Which of the following is/are example(s) of stateful application layer protocols?
(i) HTTP
(ii) FTP
(iii) TCP
(iv) POP3
A.) (i) and (ii) only
B.) (ii) and (iii) only
C.) (ii) and (iv) only
D.) (iv) only [GATE 2016]

SOLUTION:
Stateless protocol is a communications protocol in which no information is retained by either sender or receiver.
A protocol that requires keeping of the internal state on the server is known as a stateful protocol.
Stateless - HTTP, IP
Stateful - FTP, SMTP, POP3, TCP
TCP is stateful as it maintains connection information across multiple transfers, but TCP is a Transport layer protocol.
FTP and POP3 is stateful Application layer protocol.

5.)For a host machine that uses the token bucket algorithm for congestion control, the token bucket has a capacity of 1 megabyte and the maximum output rate is 20 megabytes per second. Tokens arrive at a rate to sustain output at a rate of 10 megabytes per second. The token bucket is currently full and the machine needs to send 12 megabytes of data. The minimum time required to transmit the data is seconds _____. [GATE 2016]

A.)1.1 sec

B.)1.2 sec

C.)1.3 sec

D.)1.4 sec

5.)For a host machine that uses the token bucket algorithm for congestion control, the token bucket has a capacity of 1 megabyte and the maximum output rate is 20 megabytes per second. Tokens arrive at a rate to sustain output at a rate of 10 megabytes per second. The token bucket is currently full and the machine needs to send 12 megabytes of data. The minimum time required to transmit the data is seconds _____. [GATE 2016]

A.)1.1 sec

B.)1.2 sec

C.)1.3 sec

D.)1.4 sec

SOLUTION:

According to the token bucket algorithm, the minimum time required sending 1 MB of data or the maximum rate of data transmission is given by:

$S = C / (M - P)$

Where,

M = Maximum output rate,

C = capacity of the bucket,

P = Rate of arrival of a token,

Given, M=20 Mb, C=1Mbps, P=10 Mbps

Therefore, S= 1 Mb / (20-10) Mbps = 1/10 = 0.1 sec

Since, the bucket is initially full, it already has 1 Mb to transmit so it will be transmitted instantly.

So, we are left with only (12 – 1) Mb, i.e. 11 Mb of data to be transmitted.

Therefore, time required to send the 11 MB will be 11 * 0.1 = 1.1 sec