Computer Networks

LAB 1

Some common commands used in Linux and Windows:

Linux Command	Windows Command	Usage / Effect
ifconfig	ipconfig	to find ip address of the computer
hostname	hostname	to display host name
nmap	nmap	To scan what hosts are available on a network and what ports they have open.
nslookup	nslookup	to list variety of info about DNS and the computers that have joined the domain
ping	ping	to check if a host can be accessed (by IP or name)
traceroute	tracert	to trace route from a host through internet router to a destination. Useful to discover why a network cannot get access to internet, and internet routing problems.
netstat	netstat	to print status of network ports, routing tables and more

/all

ipconfig: MAC, IP, gateway, DHCP

hostname: hostname

netstart -ano: ports

Ping Hostname: IP connectivity

incorrect IP config

Physical connectivity issue

net view Hostname: shared resources

arp -a: available hosts

tracert URL: path

LAB 2

```python
#1: How to get Host name using IP address in python

import socket
hostname=socket.gethostname()
ip_address=socket.gethostbyname(hostname)
print(f"Hostname : {hostname}")
print(f"IP Address : {ip_address}")
```

```
Hostname : DESKTOP-19
IP Address : 192.168.56.1
```

```python
#4: How to find service name, given port number & protocol
import socket
def find_service_name():
    protocolname= 'tcp'
    for port in [80 , 25]:
        print ("Port: %s => service name: %s" %(port,socket.getservbyport(port,protocolname)))

    print ("Port: %s => service name: %s" %(53,socket.getservbyport(53,'udp')))


find_service_name()
```

```
Port: 80 => service name: http
Port: 25 => service name: smtp
Port: 53 => service name: domain
```

```python
#5.Port scanner
from socket import *
import time
startTime = time.time()

target = input("Enter the host to be scanned: ")
t_IP= gethostbyname(target)
print ('starting scan on host: ', t_IP)

for i in range (50,500):
    s=socket(AF_INET,SOCK_STREAM)

    conn = s.connect_ex((t_IP, i))  # Use connect_ex to check for connection success

    if (conn == 0):
        print('Port %d: OPEN' % (i,))

    s.close()

print('Time taken:', time.time() - startTime)
```

```
Enter the host to be scanned: localhost
starting scan on host:  127.0.0.1
```

```python
#6: Simple Client Server connecting program

#Server:

import socket # Import socket module

s = socket.socket() # Create a socket object
print('Socket Created')
s. bind(('localhost', 1604)) # Bind to the port


s.listen(5) # Now wait for client connection.
print('waiting for connection')

while True:
    c, addr = s.accept() # Establish connection with client.
    print ('Got connection from', addr)
    c.send(bytes ('Thank you for connecting','utf-8'))
    c.close()  # Close the connection
```

```
Socket Created
waiting for connection
Got connection from ('127.0.0.1', 62518)
```

```python
#Client:

import socket # Import socket module

s = socket.socket() # Create a socket object
s.connect(( 'localhost', 1604))
print(s.recv(1024).decode())
s.close()
```

```
Thank you for connecting
```

```python
#Design a simple Client Server chat application

#Server:

import socket

def start_server():
    host = 'localhost'
    port = 1555

    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind((host, port))
    server_socket.listen(5)

    print(f"Server listening on {host}:{port}")

    while True:
        client_socket, addr = server_socket.accept()
        print(f"Connection from {addr}")

        # Communication loop
        while True:
            message = client_socket.recv(1024).decode('utf-8')
            if not message:
                break  # Break the loop if no message received
            print(f"Client: {message}")

            reply = input("Server (you): ")
            client_socket.send(reply.encode('utf-8'))

        client_socket.close()
        print("Connection closed")


start_server()
```

```
Server listening on localhost:1555
Connection from ('127.0.0.1', 62749)
Client: hello
Server (you): hi hows you
Client: i am thak gaya
Server (you): mee too
```

```python
#Client

import socket

def start_client():
    host = 'localhost'
    port = 1555

    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect((host, port))

    # Communication loop
    while True:
        message = input("Client (you): ")
        client_socket.send(message.encode('utf-8'))

        reply = client_socket.recv(1024).decode('utf-8')
        print(f"Server: {reply}")

    client_socket.close()


start_client()
```

```
Client (you): hello
Server: hi hows you
Client (you): i am thak gaya
Server: mee too

Client (you):
```

LAB 3

network addresses for each pair:

BC

IP Address: 188.10.18.2
Subnet Mask: 255.255.0.0
Network Address: 188.10.0.0

188.10.255.255

IP Address: 10.10.48.80
Subnet Mask: 255.255.255.0
Network Address: 10.10.48.0

s

IP Address: 192.149.24.191
Subnet Mask: 255.255.255.0
Network Address: 192.149.24.0

IP Address: 150.203.23.19
Subnet Mask: 255.255.0.0
Network Address: 150.203.0.0

IP Address: 10.10.10.10
Subnet Mask: 255.0.0.0
Network Address: 10.0.0.0

IP Address: 186.13.23.110
Subnet Mask: 255.255.255.0
Network Address: 186.13.23.0

IP Address: 223.69.230.250
Subnet Mask: 255.255.0.0
Network Address: 223.69.0.0

IP Address: 200.120.135.15
Subnet Mask: 255.255.255.0
Network Address: 200.120.135.0

host addresses for each pair:

IP Address: 188.10.18.2
Subnet Mask: 255.255.0.0
Host Address: 0.0.18.2

IP Address: 10.10.48.80
Subnet Mask: 255.255.255.0
Host Address: 0.0.0.80

IP Address: 222.49.49.11
Subnet Mask: 255.255.255.0
Host Address: 0.0.0.11

IP Address: 128.23.230.19
Subnet Mask: 255.255.0.0
Host Address: 0.0.230.19

IP Address: 10.10.10.10
Subnet Mask: 255.0.0.0
Host Address: 0.10.10.10

IP Address: 200.113.123.11
Subnet Mask: 255.255.255.0
Host Address: 0.0.0.11

IP Address: 223.169.23.20
Subnet Mask: 255.255.0.0
Host Address: 0.169.23.20

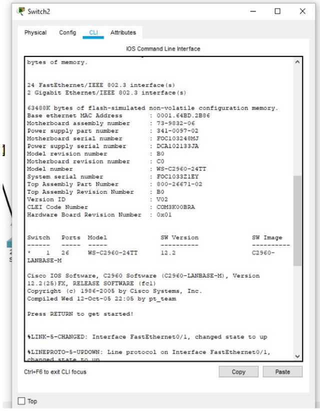
IP Address: 203.20.35.215
Subnet Mask: 255.255.255.0
Host Address: 0.0.0.215

default subnet masks for the given IP addresses:

177.100.18.4 (Class B) - Default Subnet Mask: 255.255.0.0
119.18.45.0 (Class A) - Default Subnet Mask: 255.0.0.0
191.249.234.191 (Class B) - Default Subnet Mask: 255.255.0.0
223.23.223.109 (Class C) - Default Subnet Mask: 255.255.255.0
10.10.250.1 (Class A) - Default Subnet Mask: 255.0.0.0
126.123.23.1 (Class A) - Default Subnet Mask: 255.0.0.0
223.69.230.250 (Class C) - Default Subnet Mask: 255.255.255.0
192.12.35.105 (Class C) - Default Subnet Mask: 255.255.255.0
77.251.200.51 (Class A) - Default Subnet Mask: 255.0.0.0
189.210.50.1 (Class B) - Default Subnet Mask: 255.255.0.0

A	B	C
1–127	128–191	192–223

USER EXEC Mode: show interfaces
Privilege : show running-config
Global Config : interface GigabitEthernet 0/0



switch/CLI

Privileged EXEC Mode: `show running-config`

router

IP Address Classes

Class A	1 – 127	(Network 127 is reserved for loopback and internal testing) Leading bit pattern	0	00000000.00000000.00000000.00000000 Network . Host . Host . Host
Class B	128 – 191	Leading bit pattern	10	10000000.00000000.00000000.00000000 Network . Network . Host . Host
Class C	192 – 223	Leading bit pattern	110	11000000.00000000.00000000.00000000 Network . Network . Network . Host
Class D	224 – 239	(Reserved for multicast)		
Class E	240 – 255	(Reserved for experimental, used for research)		

Private Address Space

Class A	10.0.0.0 to 10.255.255.255
Class B	172.16.0.0 to 172.31.255.255
Class C	192.168.0.0 to 192.168.255.255

Default Subnet Masks

Class A	255.0.0.0
Class B	255.255.0.0
Class C	255.255.255.0

Routers: chooses shortest Path, and forwards, routes packet to other network → Packet switching

Switch: intelligent broadcast data overtime

Hub: broadcast data

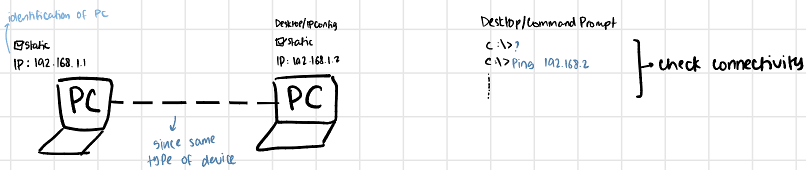
NIC: installed in computer, so computer can be connected to network

straight wire: Connect different types of devices

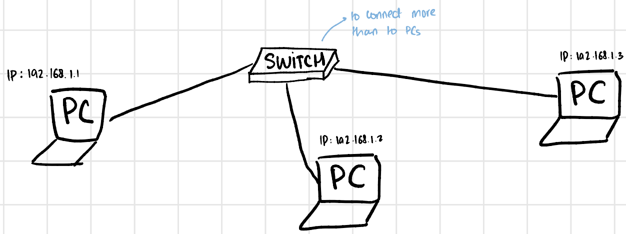
crossover wire: connect same types of devices

Lab 1

1.



2.



IPV4: Network Part + Host Part

192.168.1.3 /24

NP HP

32 = 8 + 24

2^8

Subnet: 255.255.0.0

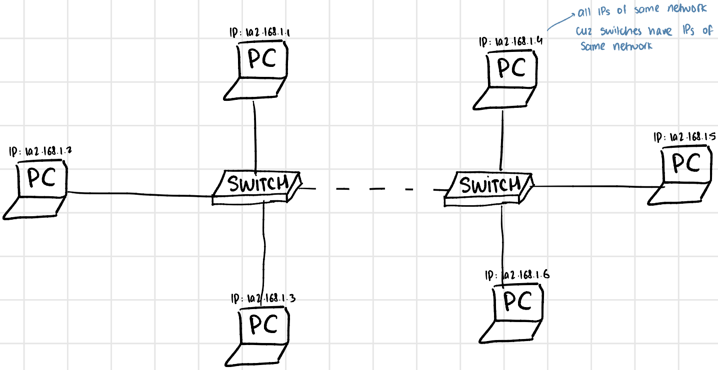
NP HP

same network

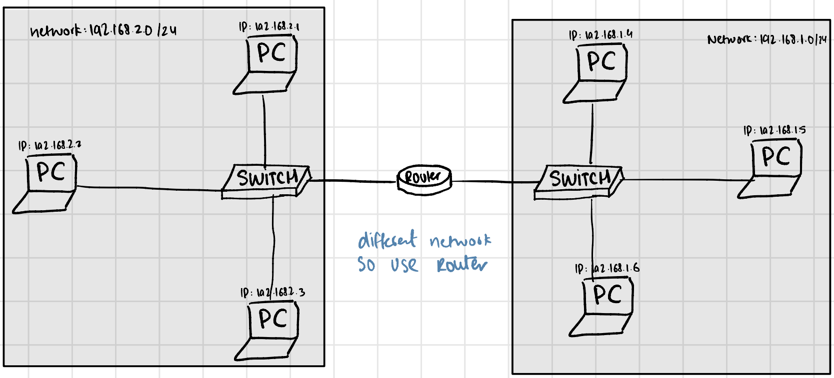
192.168.1.2 /24
192.168.1.55 /24
192.168.1.99 /24
192.168.1.4 /24

LAB2

3.

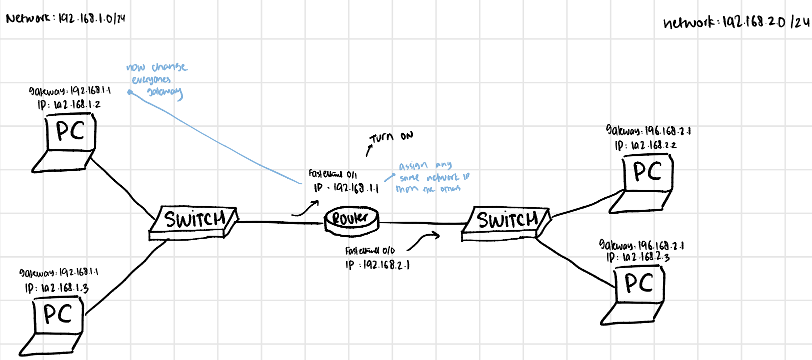


4.



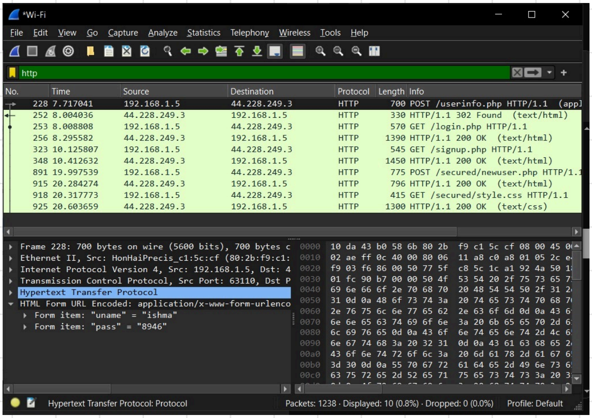
Lab3

5.



HTTP packet ON WIRE SHARK

Lab 4

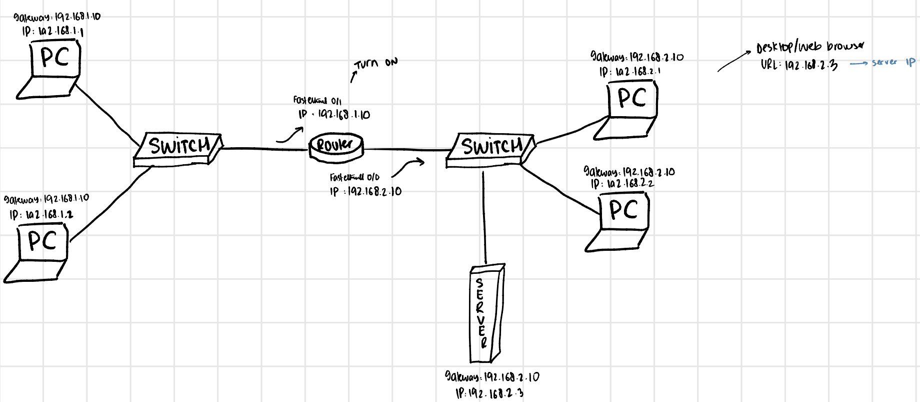


1. Start
2. testphp.vulnweb.com
3. Filter http

Lab 5

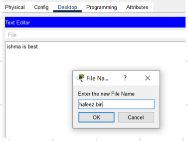
Network: 192.168.1.0/24

network: 192.168.2.0 /24



Lab 6

FTP server



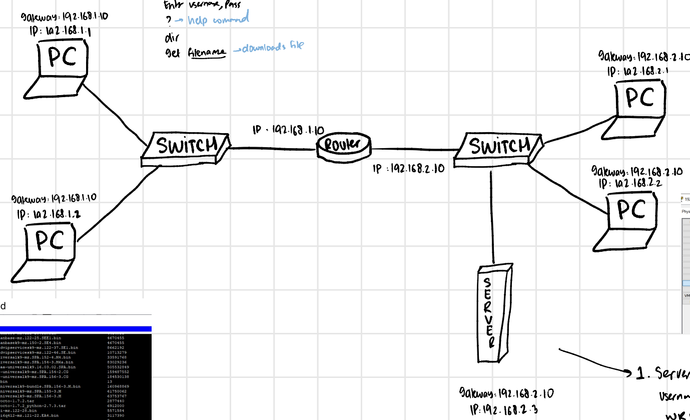
Network : 192.168.1.0/24

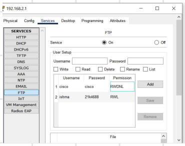
network : 192.168.2.0 /24

Upload File
Desktop / Text Editor
test.bin file
Desktop/CMD
ftp 192.168.2.3
Enter details
Put test.bin

Download file
2. Desktop/CMD
ftp 192.168.2.3 → server IP
Enter username, Pass
? → help command
dir
get filename → downloads file

Rename / Delete
ftp 192.168.2.3
Enter details
rename test.bin ishma.bin
delete ishma.bin





1. Server/Services/FTP
Username ☐ Pass ☐
WRDRL
☐☐☐☐☐
ADD

Upload

```
192.168.1.1
ftp 192.168.2.1
Trying to connect...192.168.2.1
Connected to 192.168.2.1
220- Welcome to PT Ftp server
Username:ishma
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put hafeez.bin

Writing file hafeez.bin to 192.168.2.1:
File transfer in progress...

[Transfer complete - 13 bytes]

13 bytes copied in 0.026 secs (500 bytes/sec)
ftp>
```

Top

Download

```
192.168.1.1
Command Prompt
17  : c2960-lanbase-mz.122-25.SEE1.bin                4670455
18  : c2960-lanbasek9-mz.150-2.SE4.bin                4670455
19  : c3560-advipservicesk9-mz.122-37.SE1.bin         8662192
20  : c3560-advipservicesk9-mz.122-46.SE.bin          10713279
21  : c800-universalk9-mz.SPA.152-4.M4.bin            33591768
22  : c800-universalk9-mz.SPA.154-3.M6a.bin           83029236
23  : cat3k_caa-universalk9.16.03.02.SPA.bin          505532849
24  : cgr1000-universalk9-mz.SPA.154-2.CG             159487552
25  : cgr1000-universalk9-mz.SPA.156-3.CG             184530138
26  : hafeez.bin                                      13
27  : ir800-universalk9-bundle.SPA.156-3.M.bin        160968869
28  : ir800-universalk9-mz.SPA.155-3.M                61750062
29  : ir800-universalk9-mz.SPA.156-3.M                63753767
30  : ir800_yocto-1.7.2.tar                           2877440
31  : ir800_yocto-1.7.2_python-2.7.3.tar              6912000
32  : pt1000-i-mz.122-28.bin                          5571584
33  : pt3000-i6q4l2-mz.121-22.EA4.bin                 3117390
34  : test.bin                                        8
ftp>get test.bin

Reading file test.bin from 192.168.2.1:
File transfer in progress...

[Transfer complete - 8 bytes]

8 bytes copied in 0.01 secs (800 bytes/sec)
ftp>
```

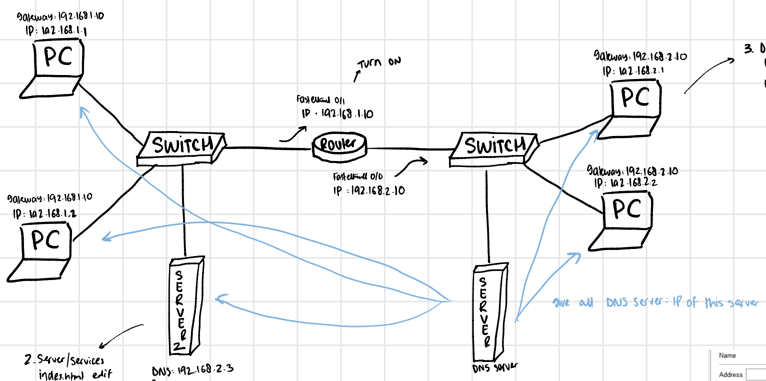
Top

Lab 7

DNS Server

Network : 192.168.1.0/24

network : 192.168.2.0 /24



3. Desktop/web browser
URL: google.com → out put google
URL: 192.168.2.3 → out put Cisco

2. Server/Services
index.html edit
add google

DNS: 192.168.2.3
gateway: 192.168.1.10
IP: 192.168.1.3
GOOGLE

gateway: 192.168.2.10
IP: 192.168.2.3
Cisco

Name ☐ Type A Record
Address ☐
Add Save Remove

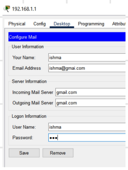
No.	Name	Type	Detail
0	gmail.com	A Record	192.168.1.3
1	google.com	A Record	192.168.1.3

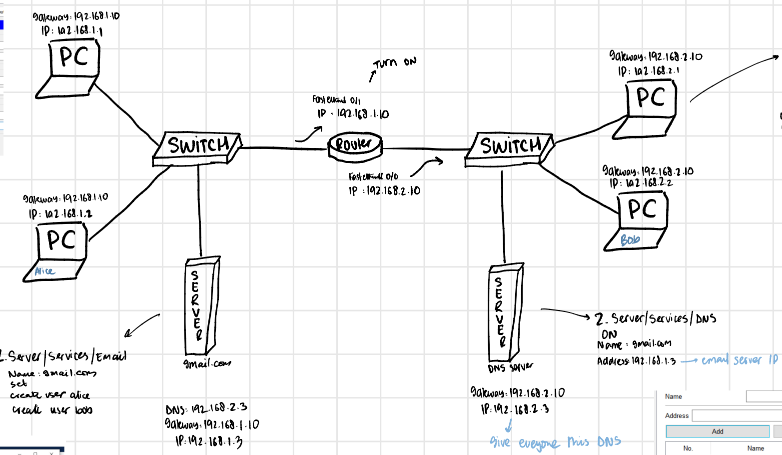
1. Server/services/DNS
ON
Name: google.com
Address: 192.168.1.3 → other server IP
Add

SMTP SERVER

Network: 192.168.1.0/24

network: 192.168.2.0/24

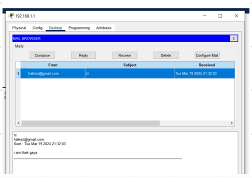


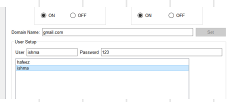


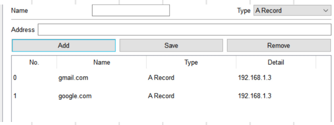
Alice
3. Desktop/Email
Name: Alice
Email: Alice@gmail.com
Incoming: gmail.com
Outgoing: gmail.com
User: alice
Pass: 123
Set

5. Receive

Desktop/Email
Name: bob
Email: bob@gmail.com
Incoming: gmail.com
Outgoing: gmail.com
User: bob
Pass: 123
Save
Compose
To: alice@gmail.com
Sub: Try
Send







DNS → Authori
name
servers

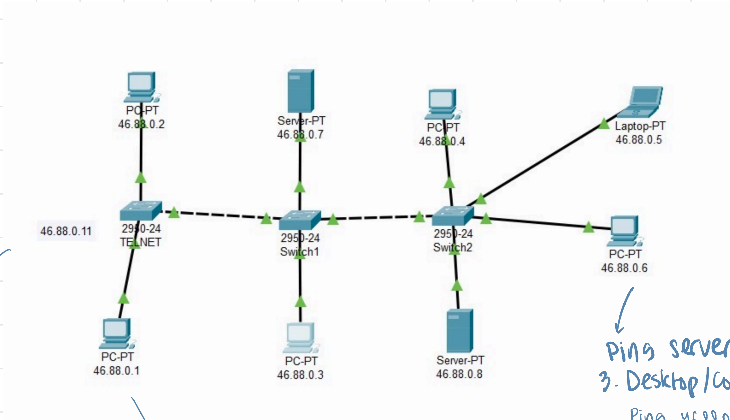
to lab 6

ishma hafeez
notes
represent

ACTUAL LAB 8

Telnet: a way to remotely control web servers
connects your PC to a server over the network

SSH (secure shell / secure socket shell)
↳ an application layer protocol
↳ in 7th layer of OSI (open systems interconnection)
↳ supports password and key based authentication



Ping server from any PC
3. Desktop / Command Prompt
Ping 46.88.0.8 → Ping server

Setup Telnet
1. CLI

assign IP to switch
en
conf t
interface vlan 1
ip address 46.88.0.11 255.0.0.0
no shutdown
exit

authentification
line vty 0 15 → gives Telnet acess within the virtual terminal
password 1234
login

Verify telnet connection
2. Desktop / Command Prompt
telnet 46.88.0.11 → gave in switch
→ enter password

user exct mode
en
enter password

TO shutdon a connection
conf t
interface fastethernet 0/3
shutdown
no shutdown

?

HOW to change IP address

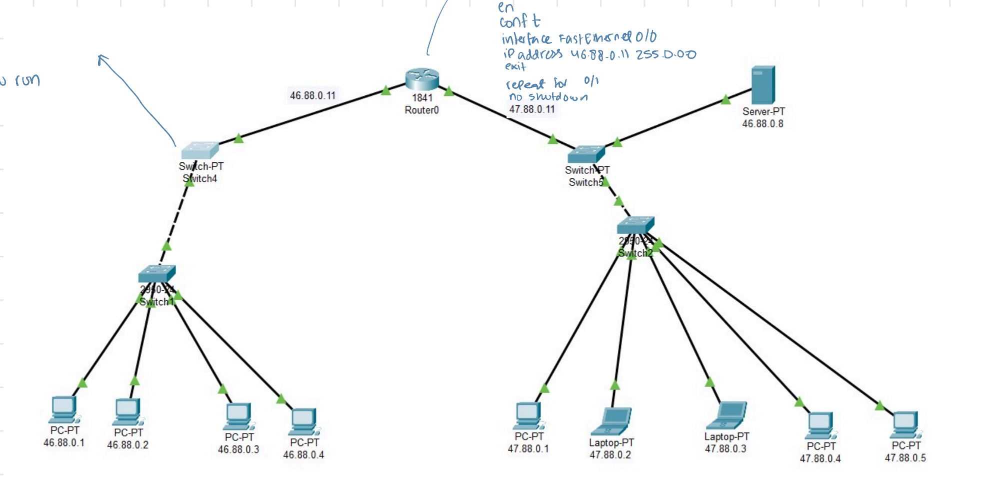
conf t
interface vlan1
ip address 46.88.0.12 255.0.0.0

CLI

user privilege mode
conf t
enable password ishma
exit

CLI
en
show run

ASSIGN IP statically
CLI
en
conf t
interface FastEthernet 0/0
ip address 46.88.0.11 255.0.0.0
exit
repeat for 0/1
no shutdown



en
conf t
hostname ishma
ip domain name cnlab
crypto key generate rsa
:1024
ip ssh version 2
line vty 0 15
transport input ssh

ACTUAL LAB 9 video LAB 8

Subnet mask: 32 bits
IPV4 address: 32 bits → written as 8 bit numbers

Octet - IP address 8 bits → in decimal → $2^7 \ldots 2^0$

max value: 255

min value: 0

same network

192.168.1.2	/24
192.168.1.55	/24
192.168.1.99	/24
192.168.1.4	/24

IPV4: Network Part + Host Part

$\underbrace{192.168.1.3}_{NP}\ \underbrace{/24}_{HP}$

32 = 8 + 24

2^8

Subnet: $\underbrace{255.255}_{NP}.\underbrace{0.0}_{HP}$

Subnet

↳ a network inside a network

↳ make networks more efficient

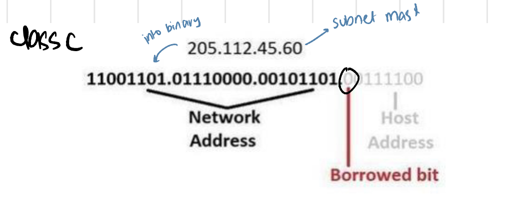
↳ a range of IP addresses

↳ all devices in same subnet can comm. directly with one another w/o going through any routers

↳ divides network into smaller portions called subnets class C

into binary

subnet mask

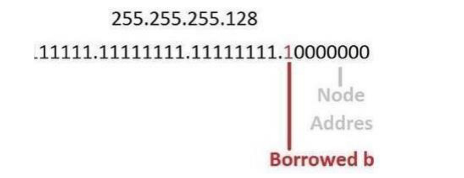


IP address components

- network part
- subnet part
- host part

create subnet by

- last bit of network : no of subnets req
- left most bit of host: identify subnet
  - 0 → one subnet
  - 1 → second subnet



Address Class	Value in First Octet	Classful Mask (Dotted Decimal)	Classful Mask (Prefix Notation)
Class A	1–126	255.0.0.0	/8
Class B	128–191	255.255.0.0	/16
Class C	192–223	255.255.255.0	/24
Class D	224–239	—	—
Class E	240–255	—	—

Table 3: Classification of IP address

Class	Representation based on Left Most Octet	Address Range	Support
A	0xxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx	1.0.0.1 to 126.255.255.254	127 networks, 16777214 hosts.
B	10xxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx	128.1.0.1 to 191.255.255.255	16384 networks, 65534 hosts.
C	110xxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx	192.0.1.1 to 223.255.254.254	2097152 networks, 254 hosts.

1 bit is reserved for prefix identification

network, broadcast ID

Networks	Hosts
$2^7 = 128$	$2^{24} = 16777216 - 2$
$2^{14} = 16384$	$2^{16} = 65536 - 2$
$2^{21} = 2097152$	$2^8 = 256 - 2 = 254$

192.168.4.0/24 create 3 subnets

Sunny Subnetting Table

Original networkID:
192.168.4.0/24

Subnet	1	2	4	8	16	32	64	128	256
Host	256	128	64	32	16	8	4	2	1
Subnet Mask	/24	/25	/26	/27	/28	/29	/30	/31	/32

Network ID	Subnet Mask	Host ID Range	# of Usable Host	Broadcast ID
192.168.4.0	/26	192.168.4.1-192.168.4.62	62	192.168.4.63
192.168.4.64	/26	192.168.4.65-192.168.4.126	62	192.168.4.127
192.168.4.128	/26	192.168.4.129-192.168.4.190	62	192.168.4.191
192.168.4.192	/26	192.168.4.193-192.168.4.254	62	192.168.4.255

192. 168. 100. 0 /24 → class C

4 sub networks ?

2^2 → no of bits to borrow

only convert host portion to binary

$\overbrace{192.\ 168.\ 100.}^{24}\ \overbrace{0000\ 0000}\ /24$

24+2 = 26

2 can change

192. 168. 100. 0000 0000 /26
192. 168. 100. 0100 0000 /26
192. 168. 100. 1000 0000 /26
192. 168. 100. 1100 0000 /26

to decimal ⟹

192.168.100.0 /26
192.168.100.64 /26
192.168.100.128 /26
192.168.100.192 /26

To find Range

1.) Host : $\frac{256}{4}$ → total no of host → class C; → no of sub networks

= 64 → add to host

2.) Host: 26 − 32 = 6

2^6 = 64

0 to 64 = 63

192.168.100.0/26 − 192.168.100.63/26
192.168.100.64 /26 − 192.168.100.127 /26
192.168.100.128 /26 − 192.168.100.191 /26
192.168.100.192 /26 − 192−168.100.255/26

To find subnet mask

Q) 192.168.100.0 /26 → first 26 bits 1, the rest 0

subnet mask: 1111 1111. 1111 1111. 1111 1111. 11 00 0000 → convert to decimal

64

255 . 255 . 255 . 192

NID (network ID)	BID (broadcast ID)	Usable Range	Subnet mask
192.168.100.0	192.168.100.63	100.1 − 100.62	255.255.255.192
192.168.100.64	192.168.100.127	100.65 − 100.126	255.255.255.192
192.168.100.128	192.168.100.191	100.129 − 100.190	subnet same for all
192.168.100.192	192.168.100.255	100.193 − 100.254	

8) 192. 168. 100. 0 /24 → class C
4 sub networks ?

Sunny Subnetting Table

Possible hosts

Subnet	1	2	4	8	16	32	64	128	256
Host	256	128	64	32	16	8	4	2	1
Subnet Mask	/24	/25	/26	/27	/28	/29	/30	/31	/32

Subnet = 4
Possible hosts - 64
Usable hosts : 62

network ID — +64 — +1 NID — -1 BID — broadcast ID — -1 NID

NID	Subnet mask	HID Range	BID
192.168.100.0	255.255.255.192	100.1 — 100.62	192.168.100.63
192.168.100.64	255.255.255.192	100.65 — 100.126	192.168.100.127
192.168.100.128		100.129 — 100.190	192.168.100.191
192.168.100.192		100.193 — 100.254	192.168.100.255

+64
+64
+64

subnet same for all

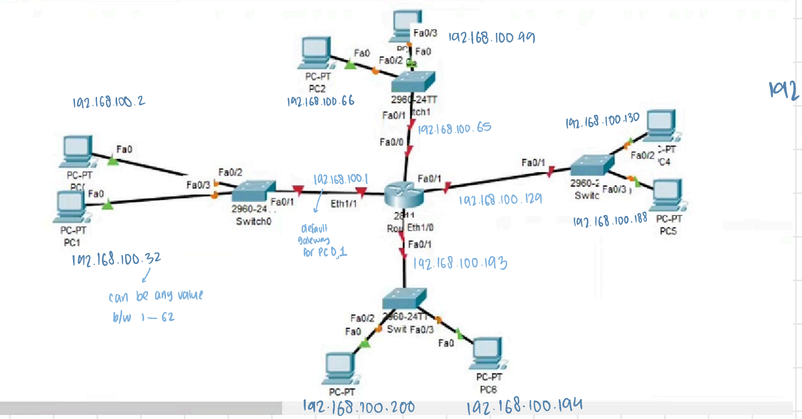
??

change subnet mask for all

192.168.100.64/26
65 — 126

192.168.100.0/26
1 — 62

192.168.128/26
129 - 190



192.168.192/26
193 — 254

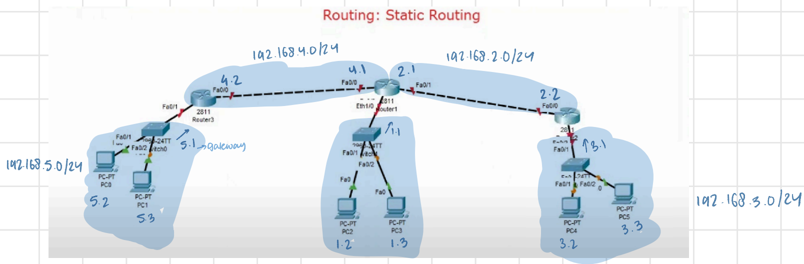
STATIC ROUTES

5 networks

24

1111 1111.1111 1111.11111111.0000 0000

255.255.255.0



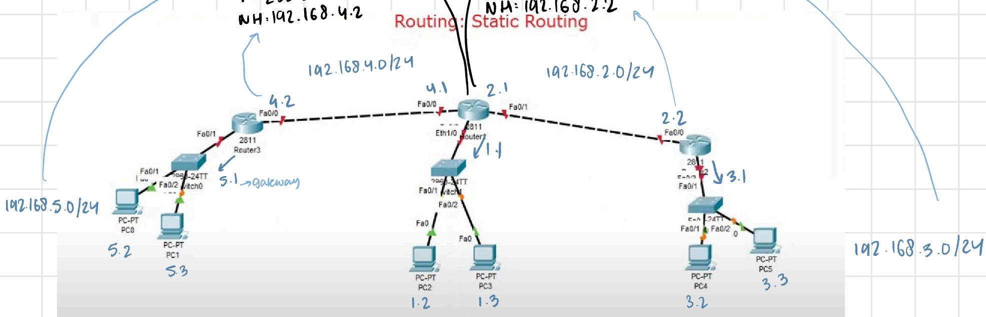
192.168.1.0/24

has no subnetting so use static routes

static 5.0 3.0

N: 192.168.5.0
M: 255.255.255.0
NH: 192.168.4.2

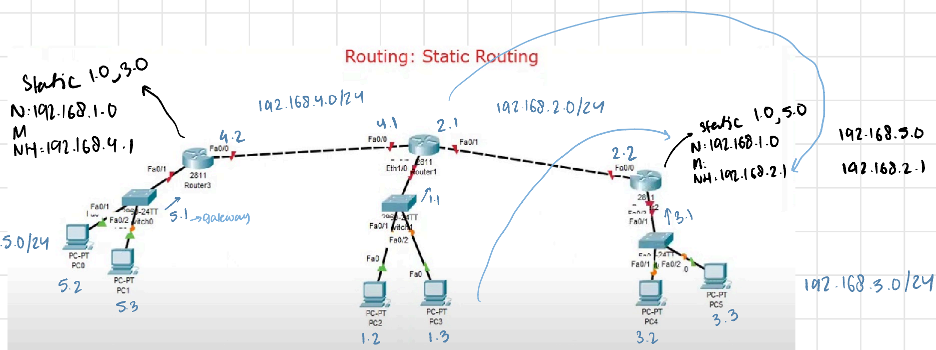
N: 192.168.3.0
M: 255.255.255.0
NH: 192.168.2.2



192.168.1.0/24

N: 192.168.3.0

NH: 192.168.4.1



192.168.5.0

192.168.2.1

192.168.1.0/24

Q) 192.168.1.0/27 of class C

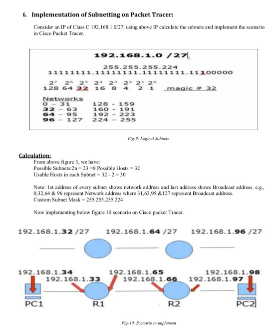
using sunny subnetting table

subnet: 8

Possible host: 32

Usable host: 32-2 = 30

NID	Subnet mask	HID Range (+1 NID ... -1 BID)		BID (broadcast ID) -1 NID
192.168.1.0	255.255.255.224	1.1	1.30	192.168.1.31
192.168.1.32	11111111 11111111 11111111 11100000	1.33	1.62	192.168.1.63
192.168.1.64		1.65	1.94	192.168.1.95
192.168.1.96		1.97	1.126	192.168.1.127



6. **Implementation of Subnetting on Packet Tracer:**

Consider an IP of Class C 192.168.1.0/27, using above IP calculate the subnets and implement the scenario in Cisco Packet Tracer.

Fig-9: Logical Subnets

Calculation:

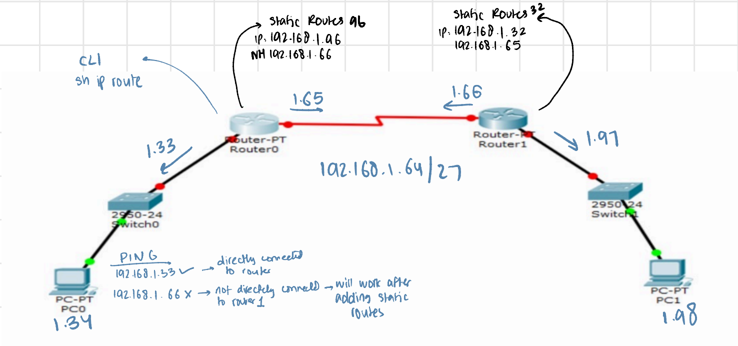
From above figure 3, we have:
Possible Subnets:2n = 23 =8 Possible Hosts = 32
Usable Hosts in each Subnet = 32 - 2 = 30

Note: 1st address of every subnet shows network address and last address shows Broadcast address. e.g., 0,32,64 & 96 represent Network address where 31,63,95 &127 represent Broadcast address.
Custom Subnet Mask = 255.255.255.224

Now implementing below figure 10 scenario on Cisco packet Tracer.

Fig-10: Scenario to implement

Static route



SM: 255.255.255.224

/27

192.168.1.32/27

192.168.1.96/27

Network Address: 10.20.0.0 /24

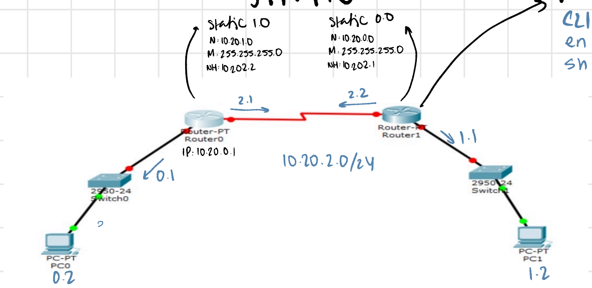
STATIC

TO SEE Route Table

CLI

en

sh ip route



10.20.0.0 /24

SM: 255.255.255.0

/24

10.20.1.0/24

Q) 3 Departments, perform Subnetting for the allocation of given req
Network address : 200.16.100.0/24

↳ D → 120 PCs
↳ F → 35 PCs
↳ HR → 10 PCs

subnet the given network address 200.16.100.0/24:

1. Development: Subnet size = 128 → Subnet mask = /25 (because 24 + 7 = 31)
   Subnet range: 200.16.100.0/25

2. Finance: Subnet size = 64 → Subnet mask = /26 (because 24 + 6 = 30)
   Subnet range: 200.16.100.128/26

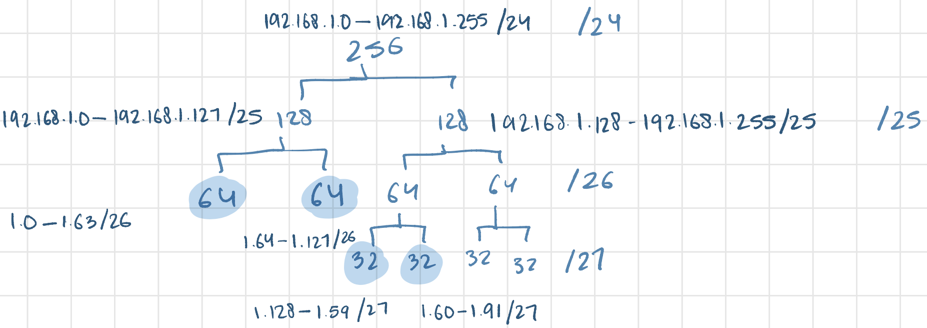
3. HR: Subnet size = 16 → Subnet mask = /28 (because 24 + 4 = 28)
   Subnet range: 200.16.100.192/28

The subnetting allocation for ABC Company's departments is:

- Development: 200.16.100.0/25 (Subnet mask: 255.255.255.128)
- Finance: 200.16.100.128/26 (Subnet mask: 255.255.255.192)
- HR: 200.16.100.192/28 (Subnet mask: 255.255.255.240)

Q) 192.168.1.0/24
2 D → 60 PCs $2^6 = 64$
1 F → 28 PCs $2^5 = 32$
1 HR → 16 PCs $2^5 = 32$
BID NID
-2

64+64+32+32 = 192
/24 = 256 resources
192 < 256



27 Subnet mask
1111 1111.1111 1111.1111 1111.111 00000
.224

[5 Marks]

Given the network address 10.0.0.0/8, you need to create subnets for six different departments: Administration (100 hosts), Development (50 hosts), Testing (30 hosts), Finance (20 hosts), HR (10 hosts), and Sales (5 hosts). Calculate the subnet mask for each department.

10.0.0.0/8

11111111

255.0.0.0

DA	100	$2^6 = 128$
D	50	$2^5 = 64$
T	30	$2^5 = 64$
F	20	$2^4 = 32$
HR	10	$2^4 = 16$
S	5	$2^3 = 8$

Q) 3 Departments , Network : 195.168.10.0 /24

CS → 90 PCs	2^7	: 128	10.0 – 10.127/25	255.255.255.128
EE → 50 PCs	2^6	: 64	10.128 – 10.10.191/26	255.255.255.192
BBA → 20 PCs	2^5	: 32	10.192 – 10.223/27	255.255.255.224

24

128 + 64 + 32 = 224

256 10.0 – 10.255/24 /24

128 — 10.0 – 10.127/25

128 10.128 – 10.255/25 /25

64 — 10.128 – 10.10.191/26

64 10.192 – 10.255/26 /26

32 — 10.192 – 10.223/27

32 — 10.224 – 10.255/27 /27

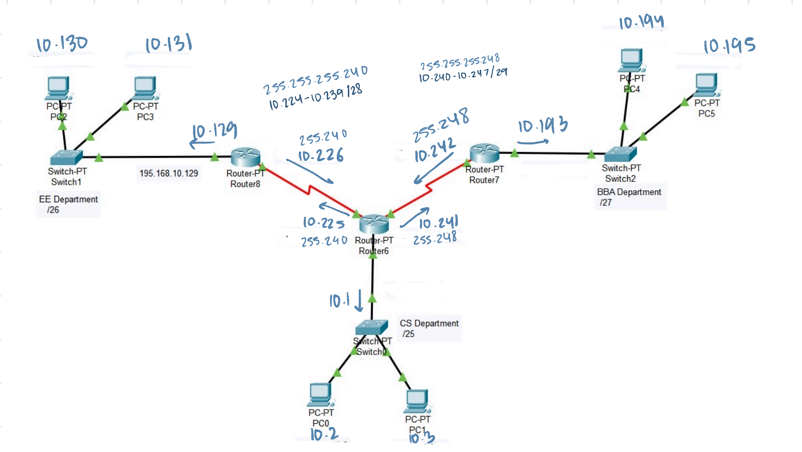
16 — 10.224 – 10.239/28

16 — 10.240 – 10.255/28

8 — 10.240 – 10.247/29

10.128 – 10.10.191/26
255.255.255.192

10.192 – 10.223/27
255.255.255.224



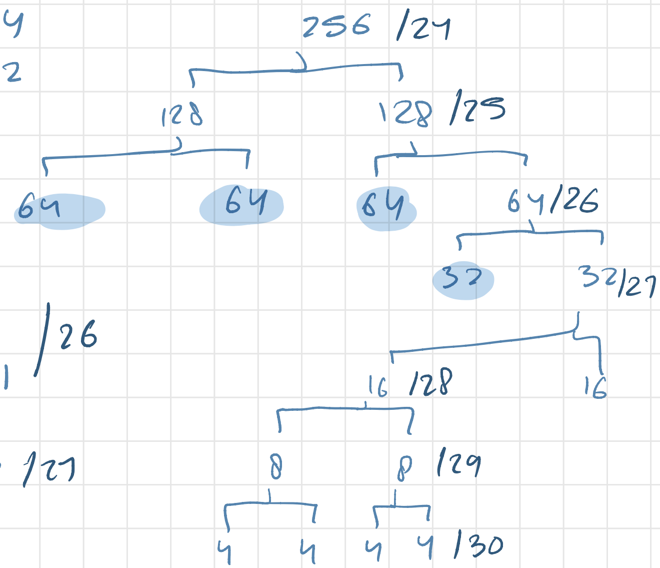
255.255.255.128

10.0 – 10.127/25

DYNAMIC ROUTING

Q) 3 Networks 42 PCs $2^6 = 64$
1 Network 22 PCs $2^5 = 32$

64+64+64+32 = 194



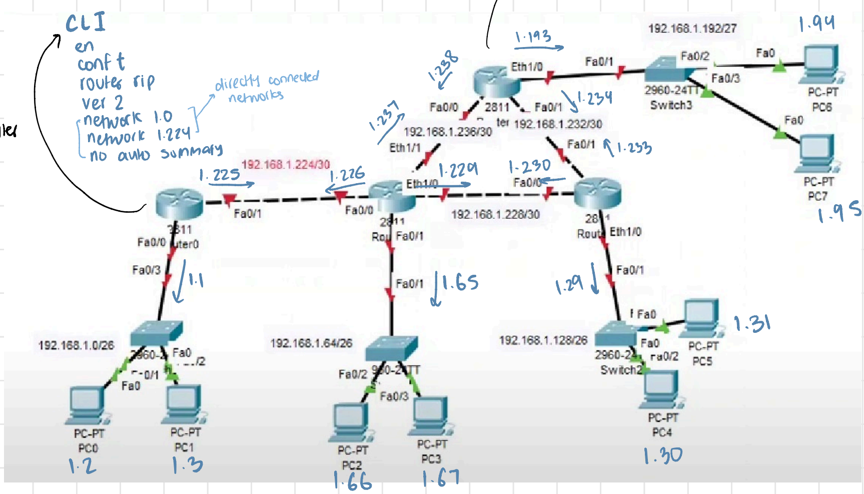
192.168.1.0 – 192.168.1.63
192.168.1.64 – 192.168.1.127 /26
192.168.1.128 – 192.168.1.191

192.168.1.192 – 192.168.1.223 /27

192.168.1.224 – 192.168.1.227
192.168.1.228 – 192.168.1.231
192.168.1.232 – 192.168.1.235 /30
192.168.1.236 – 192.168.1.239

To see route table
CLI
en
sh ip route

do for every router



1.192 – 1.223 /27
255.224

1.0 – 1.63 /26 1.64 – 1.127 /26 1.128 – 1.191 /26
255.255.255.192

Advantages	Disadvantages
Easy to implement in a small network.	Suitable for simple topologies or for special purposes such as a default static route.
Very secure. No advertisements are sent,unlike with dynamic routing protocols.	Configuration complexity increases dramatically as the network grows. Managingthe static configurations in large networks can become time consuming.
It is very predictable, as the route to thedestination is always the same.	If a link fails, a static route cannot reroute traffic. Therefore, manual intervention isrequired to re-route traffic.
No routing algorithm or update mechanisms are required. Therefore, extra resources (CPU and memory) are not required.	

Table-1: Advantages & Disadvantages of Static Routing

Route Source	Default AD
Connected interface	0
Static route	1
EIGRP	90
IGRP DV	100
OSPF LS	110
RIP 1 DV	120
External EIGRP	170

Table-2: Administrative Distances

The smaller the AD is, the more preferable to route is.

Distance vector (DV)
↳ router with least hops
each time a packet goes through a router

HYBRID
↳ DV + LS

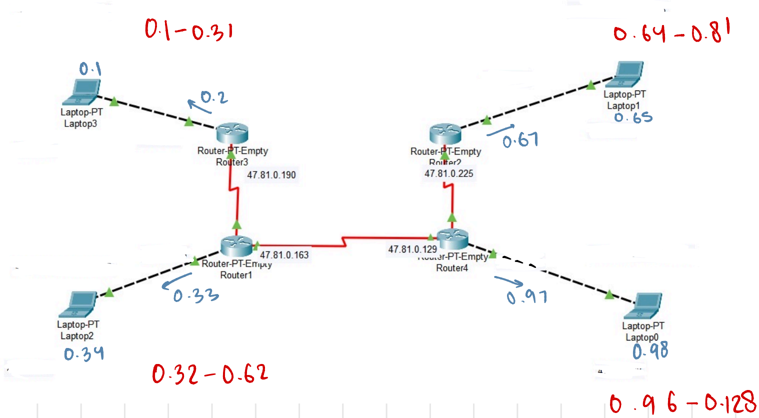
Link state (LS)
↳

Q) 47.81.0.0/24

0.0
0.32
0.64
0.128

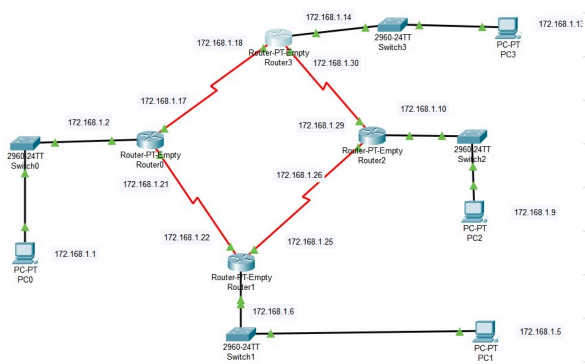
Q1: Implement Subnetting with IP address of XX.XX.0.0/24. where xx is your roll no same are midterm. Then assign ip address in such a way that very less ip address should waste. Last run RIP routing protocol in such a way that all devices can communicate easily. What will be the administrative distance of the routing? Use figure 8 for your reference.

subnet = 255.255.255.224



Q2: Implement Subnetting with IP address of 172.168.1.0/24. All the assignment of IP should be done dynamically in such a way that there should be less waste of Ips. Run the dynamic routing protocol with less administrative distance. What will be the administrative distance of the routing? Use Figure 9 as reference.

Subnet = 255.255.255.252



Q3: Implement the subnetting on the given scenario of figure 10. You have to implement the static routing on the same. What will be the administrative distance of the routing?
Use Network Address as follows: 192.168.4.0/24.

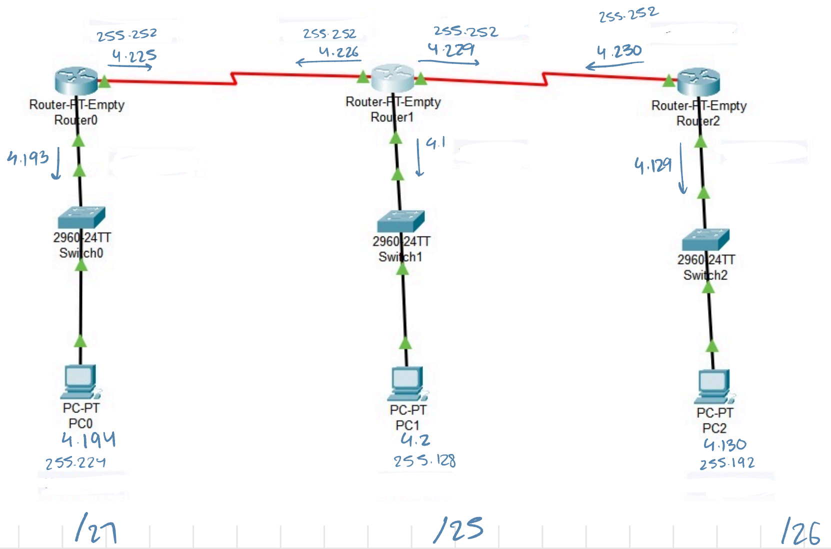


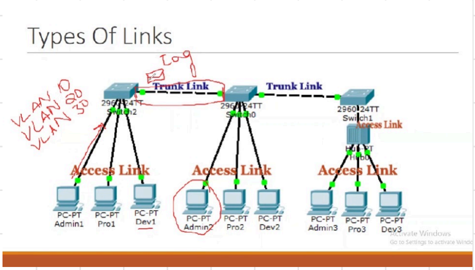
Fig-10: Network Topology for Q3

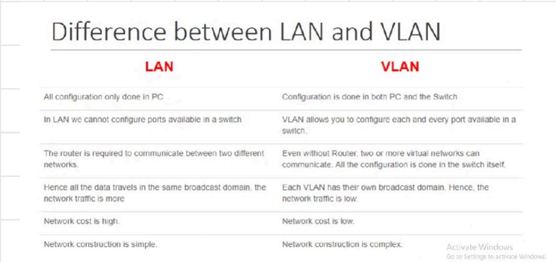
Q4: In what case we use static routing or dynamic routing given a topological reason for this question.

TASK 4:

Static Routing: Static routing suits smaller, less complex networks, offering simple implementation and enhanced security with predictable routes. However, it becomes cumbersome in larger setups, requiring manual configuration and lacking automatic rerouting capabilities.

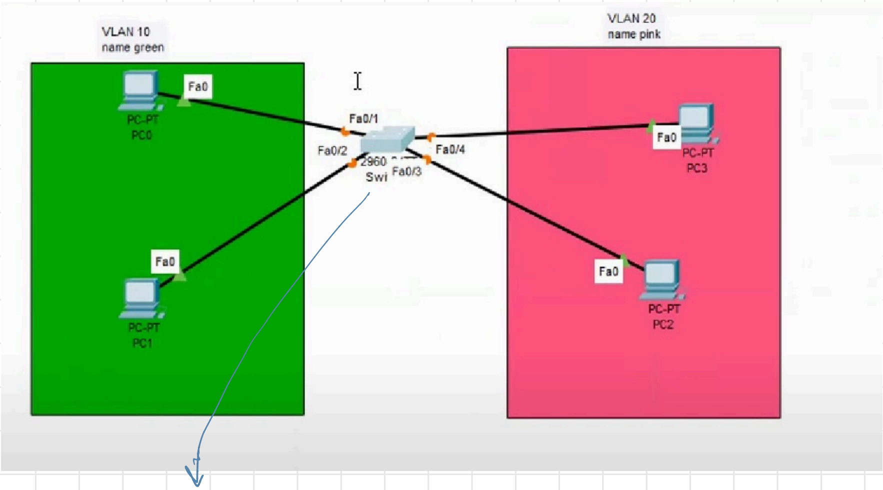
Dynamic Routing: Dynamic routing is ideal for larger networks with changing topologies, ensuring scalability and resilience. It adapts to network changes automatically but demands more resources and introduces potential security risks from routing advertisements. Configuration complexity may also arise in diverse network environments.





Difference between LAN and VLAN

LAN	VLAN
All configuration only done in PC	Configuration is done in both PC and the Switch
In LAN we cannot configure ports available in a switch	VLAN allows you to configure each and every port available in a switch.
The router is required to communicate between two different networks.	Even without Router, two or more virtual networks can communicate. All the configuration is done in the switch itself.
Hence all the data travels in the same broadcast domain, the network traffic is more	Each VLAN has their own broadcast domain. Hence, the network traffic is low.
Network cost is high.	Network cost is low.
Network construction is simple.	Network construction is complex.



CLI

en conf t

vlan 10
name green
ex

int fa0/1
switchport mode access
switchport access vlan 10
ex

int fa0/2
switchport mode access
switchport access vlan 10
ex

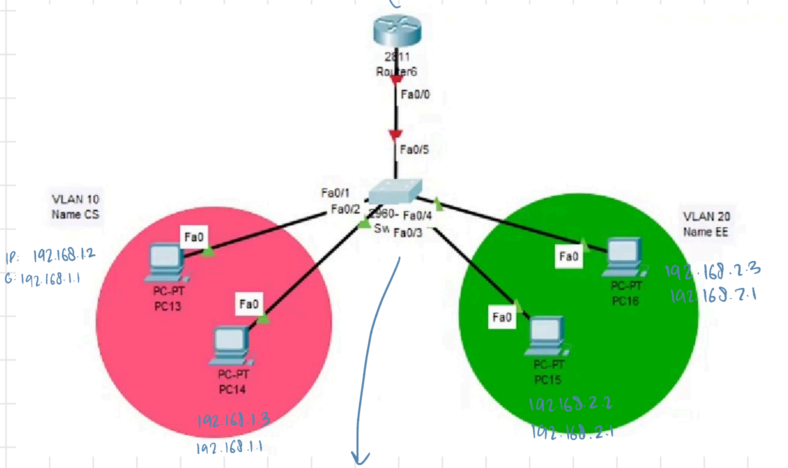
vlan 20
name pink
ex

int fa0/1
switchport mode access
switchport access vlan 20
ex

int fa0/2
switchport mode access
switchport access vlan
ex

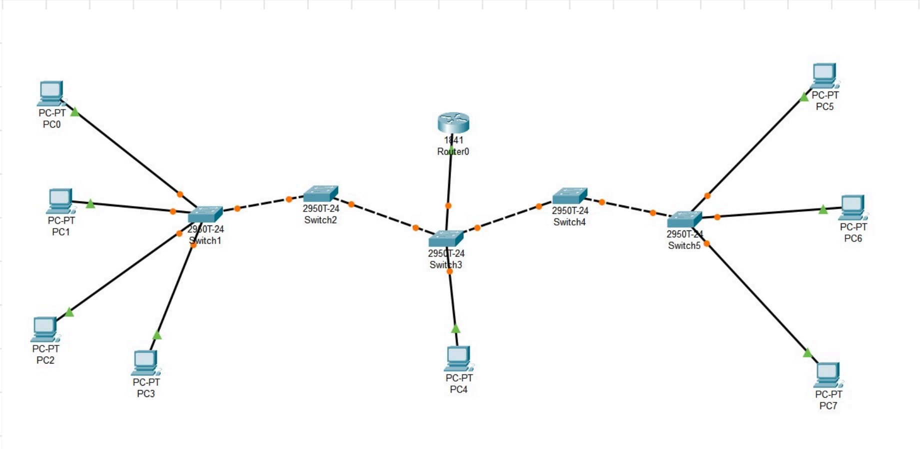
CLI
en
coff t
int fa0/0.10
encapsulation dot1Q 10
ip address 192.168.1.1 255.255.255.0
no sh
ex

int fa0/0.20
encapsulation dot1Q 20
ip address 192.168.2.1 255.255.255.0
no sh
ex



en
conf t
int fa0/5
switchport mode trunk
ex

sh interfaces trunk



Gufran Ahmed

Overview/roadmap:

- What *is* the Internet? What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Protocol layers, service models
- Security
- History

INTERNET: NUTS AND BOLTS VIEW

Packet switches

↳ chunks of data

e.g routers, switches

communication links

e.g fiber, radio, satellite

networks

collection of devices, routers, links managed by an organization

internet

↳ network of networks

interconnected ISPs

Protocols

↳ control sending, receiving of messages

e.g HTTP, skype, streaming video, wifi, 4G, Ethernet

Internet standards

↳ RCF (Request for comments

↳ IETF (Internet Engineering Task Force)

INTERNET: SERVICES VIEW

Infrastructure

↳ provides services to applications

e.g streaming video, social media, games e-commerce

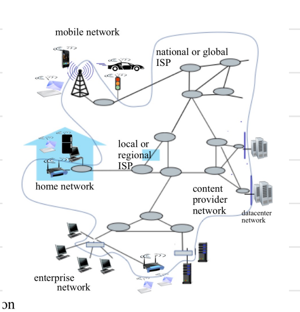
Programming interface

↳ provided to distributed applications

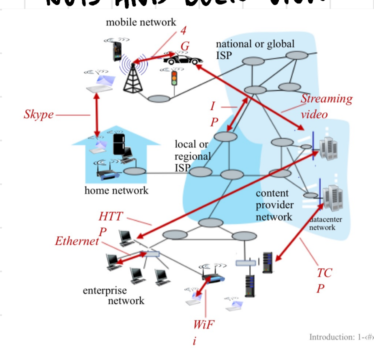
↳ hooks: allows apps to connect to internet transport service

↳ provides service options

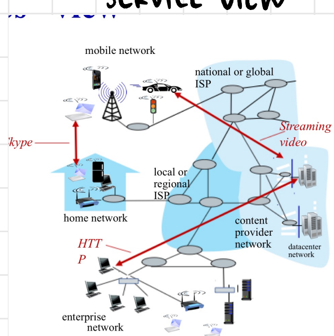
↳ analogous to postal service



NUTS AND BOLTS VIEW



SERVICE VIEW



PROTOCOLS

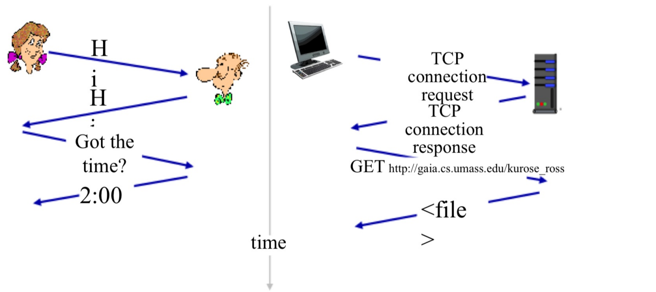
↳ defines format, order of mssgs sent and received, receipt, actions taken on mssg transmission

Human Protocol

"what's the time?"
"I have a question"
introductions

Network Protocol

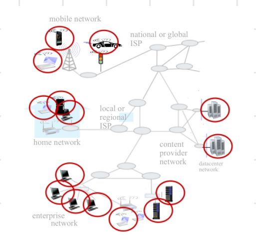
↳ b/w computers
↳ all comm activity on the internet is governed by protocols



Network edge

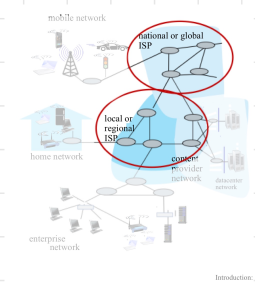
↳ hosts: clients and servers

often in data centers



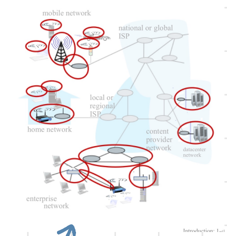
Network core

↳ inter connected routers
↳ network of networks



Access networks, physical media

↳ wired comm. link
↳ wireless comm. link



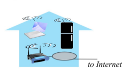
Q: How to connect end systems to edge router?

- residential access nets
- institutional access networks (school, company)
- mobile access networks (WiFi, 4G/5G)

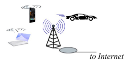
Wireless Access Networks

↳ connects end system to router

Wireless local area networks (WLANs)
- typically within or around building (~100 ft)
- 802.11b/g/n (WiFi): 11, 54, 450 Mbps transmission rate



Wide-area cellular access networks
- provided by mobile, cellular network operator (10's km)
- 10's Mbps
- 4G cellular networks (5G coming)



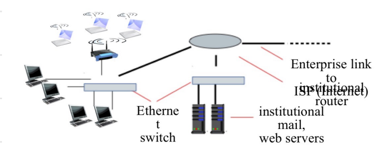
Access Networks

Enterprise Networks

↳ mix of wired + wireless technologies

↳ connects mix of switches and routers

e.g universities, companies

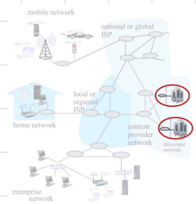


Data Centre Networks

↳ high bandwidth links

↳ connects thousands of servers together and to the internet

Courtesy: Massachusetts Green High Performance Computing Center (mghpcc.org)



Ethernet: wired acess at 100Mbps, 1 Gbps, 10 Gbps

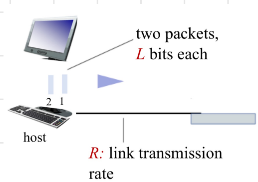
WiFi: wireless access at 11, 54, 450 MbPs

Host Sending Function

↳ sends packets of data

Steps

1. take application message
2. breaks into packets → smaller chunks → of length 'L' bits
3. transmits packet into acess network → at transmission rate 'R'



$$\boxed{\text{Packet transmission delay} = \frac{L}{R}}$$

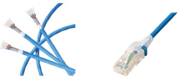
→ time to transmit L bit packet into link at R bps

One-hop numerical example:
- L = 10 Kbits
- R = 100 Mbps
- one-hop transmission delay = 0.1 msec

Links: physical media

- bit: propagates between transmitter/receiver pairs
- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: copper, fiber, coax
- unguided media:
  - signals propagate freely, e.g., radio

Twisted pair (TP)
- two insulated copper wires
  - Category 5: 100 Mbps, 1 Gbps Ethernet
  - Category 6: 10Gbps Ethernet



Introduct

Coaxial cable:
- two concentric copper conductors
- bidirectional
- broadband:
  - multiple frequency channels on cable
  - 100's Mbps per channel

Fiber optic cable:
- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (10's-100's Gbps)
- low error rate:
  - repeaters spaced far apart
  - immune to electromagnetic noise

Introduction: 1-..

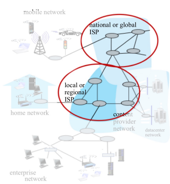
Wireless radio
- signal carried in various "bands" in electromagnetic spectrum
- no physical "wire"
- broadcast, "half-duplex" (sender to receiver)
- propagation environment effects:
  - reflection
  - obstruction by objects
  - Interference/noise

Radio link types:
- Wireless LAN (WiFi)
  - 10-100's Mbps; 10's of meters
- wide-area (e.g., 4G cellular)
  - 10's Mbps over ~10 Km
- Bluetooth: cable replacement
  - short distances, limited rates
- terrestrial microwave
  - point-to-point; 45 Mbps channels
- satellite
  - up to 45 Mbps per channel
  - 270 msec end-end delay

Intro

Network core

↳ mesh of interconnected routers



Two key network core functions

1. Routing → Global action

↳ determines source destination paths taken by packets

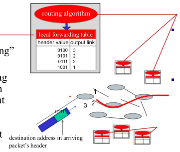
↳ routing algos

2. Forwarding → Local action

aka switching

↳ moves arriving packets from routers input link to appropriate routers output link

Forwarding:
- aka "switching"
- *local* action: move arriving packets from router's input link to appropriate router output link



Routing:
- *global* action: determine source-destination paths taken by packets
- routing algorithms

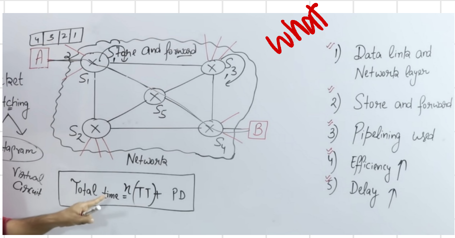
Introduction

Packet switching

↳ host breaks application layer into packets

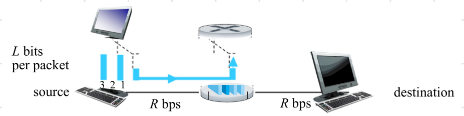
↳ network forwards packets router to router across links from source to destination

what



Store and forward

↳ entire packet must arrive at router before it can be transmitted on next line

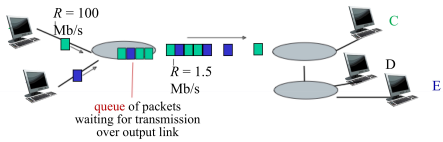


Queueing

↳ when work arrives faster than it can be served

↳ if arrival rate exceeds transmission rate

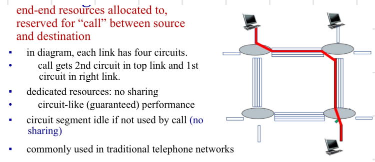
1. packets will queue → waiting to be transmitted on output link
2. packets can be lost → if buffer in router fills up

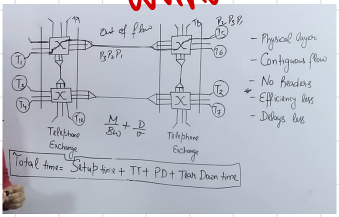


alternate to packet switching

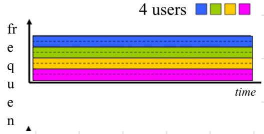
Circuit switching

whaa?





Frequency Division Multiplexing (FDM)

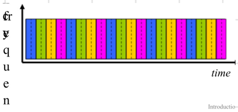


↳ optical, electromagnetic frequencies divided into frequency bands

↳ each call allocated its own band

↳ can transmit max rate of that band

Time Division Multiplexing (TDM)



↳ time divid into slots

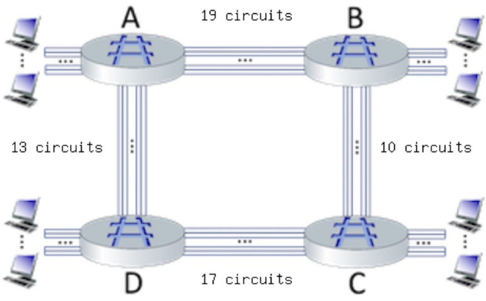
↳ each call allocated periodic slots

↳ can transmit max rate of frequency band only during its time slot

Question 1: Circuit Switching **(10) Points**

Explain Circuit Switching. Consider the circuit-switched network shown in the figure below, with circuit switches A, B, C, and D. Suppose there are 19 circuits between A and B, 10 circuits between B and C, 17 circuits between C and D, and 13 circuits between D and A.
What is the maximum number of connections that can be ongoing in the network at any one time?

\



There are no bottlenecks in the given network so all you have to do is sum the individual circuit counts.
Total Circuits = 13 + 19 + 17 + 10 = 59

Packet switching vs circuit switching

PROS

↳ good for bursty data
sometimes has data to send
sometimes not

↳ resource sharing

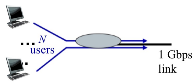
↳ simple → no call setup

CONS

↳ packet delay
↳ packet loss → due to buffer overflow

not good for excessive congestion

example:
- 1 Gb/s link
- each user:
  - 100 Mb/s when "active"
  - active 10% of time



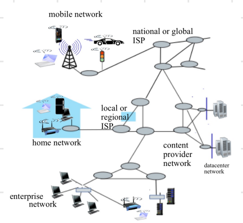
Q: how many users can use this network under circuit-switching and packet switching?

- *circuit-switching:* 10 users
- *packet switching:* with 35 users, probability > 10 active at same time is less than .0004 *

Q: how did we get value 0.0004?
A: HW problem (for those with course in probability only)

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive

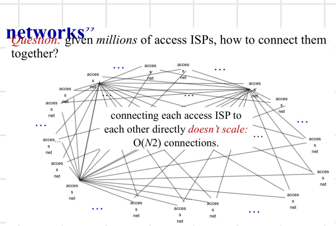
Internet structure: a network of networks

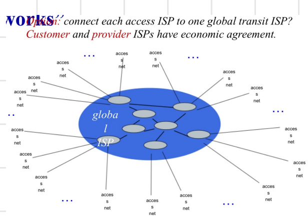


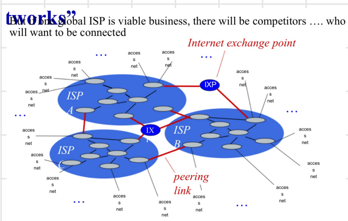
↳ host connects to Internet via access Internet Service Providers (ISPs)

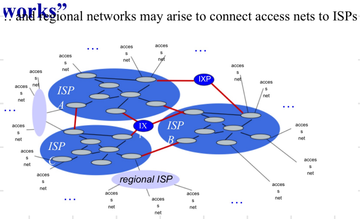
↳ ISPs must be interconnected so that any two hosts can send packets to each other
from anywhere

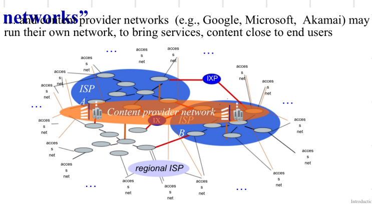
↳ results in network of networks

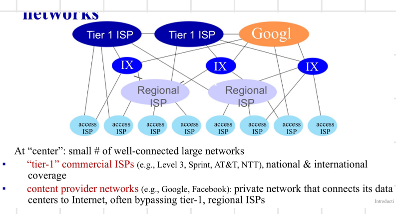




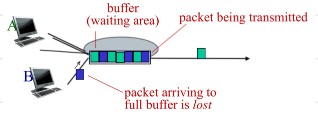








How Packet delay and loss occurs

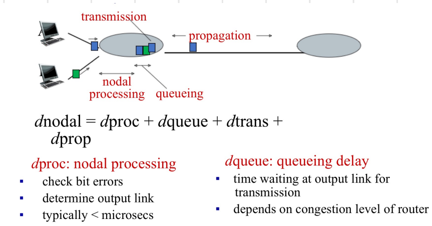


↳ packets queue in routers buffers, waiting for their turn for transmission

↳ when a packet arrives and buffer is full the packet is lost

↳ lost packet may be retransmitted → by previous node → by source end system
or not at all

Packet delay: 4 resources



1. Transmission Delay: packet being transmitted → dtrans
2. Queueing Delay: packets in buffers → dqueue
3. nodal Processing Delay: → dproc
4. Propagation Delay: → dprop $= \frac{d}{s}$

Total Delay: Transmission + Propogation

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

Total delay

$$d_{trans} = \frac{L}{R}$$

L → packet length bits
R → link transmission rate bps

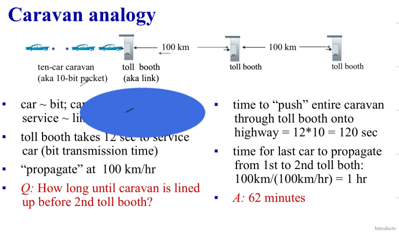
$$d_{prop} = \frac{d}{s}$$

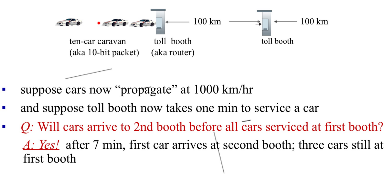
d → length of physical link
s → propagation speed 2×10^8 m/s

$$\text{traffic intensity} = \frac{L \cdot a}{R}$$

a → avg packet arrival rate

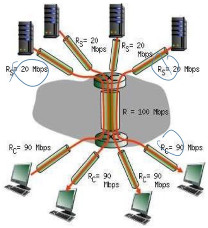
= 0: avg queueing delay small
→ 1 avg queueing delay large
> 1 avg queueing delay infinite
work arriving is more than can be serviced





Pd = 30 ms

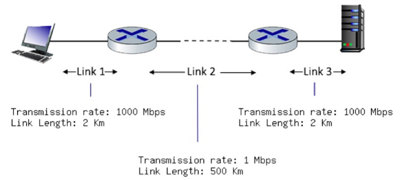
Question #5: Consider the scenario shown in figure 1, with four different servers connected to four different clients over four three-hop paths. The four pairs share a common middle hop with a transmission capacity of R = 100 Mbps. The four links from the servers to the shared link have a transmission capacity of RS = 20 Mbps. Each of the four links from the shared middle link to a client has a transmission capacity of RC = 90 Mbps per second. You might want to review Figure 1 in the text before answering the following questions:



a) What is the maximum achievable end-end throughput (in Mbps) for each of four client-to-server pairs, assuming that the middle link is fair-shared (i.e., divides its transmission rate equally among the four pairs)?
Answer:
The maximum achievable end-end-throughput is 20 Mbps.

b) Which link is the bottleneck link for each session?
Answer:
This is the transmission capacity of the first hop, which is the bottleneck link, since the first-hop transmission capacity of 20 Mbps is less than one quarter of the shared-link transmission capacity (100/4 = 25 Mbps) and less than the third-hop transmission capacity of 90 Mbps.

c) Assuming that the senders are sending at the maximum rate possible, what are the link utilizations for the sender links (RS), client links (RC), and the middle link (R)?
Answer:
The utilization of sender links is 100%. The utilization of receiver links is 22.22%. The utilization of the middle link is 80%.



Assume the length of a packet is 16000 bits. The speed of light propagation delay on each link is 3x10^8 m/sec
L = 16000 s=3x10^8

$\frac{L}{R}$ $\frac{16000}{1\times10^9}$ $\frac{16}{1\times10^6}$

Answer the following questions:

1) What is the transmission delay of link 1?
   R = 1000 Mbps = 1000x10^6 bps
   Transmission Delay = L / R = 16000/1x10^9 = 1.6x10^-5 seconds
2) What is the propagation delay of link 1?
   d = 2Km = 2000 m
   Propagation delay = d / s = 2000/3x10^8 = 6.66x10^-6 seconds

   $\frac{d}{s}$ $\frac{2\times10^3}{3\times10^8}$
3) What is the total delay of link 1?
   Total Delay = Transmission + Propagation = 1.6x10^-5 + 6.66x10^-6 = 2.267x 10^-5 sec
4) What is the transmission delay of link 2?
   R = 1 Mbps = 1x10^6 bps
   Transmission Delay = L/R = 16000/1x10^6 = 0.016 seconds
5) What is the propagation delay of link 2?
   d = 500 Km = 500 x 10^3 m
   Propagation delay = d / s = 500x10^3/3x10^8 = 1.667 x 10^-3 seconds
6) What is the total delay of link 2?
   Total Delay = Transmission + Propagation = 0.016 + 1.667x10^-3 = 0.017 seconds
7) What is the transmission delay of link 3?
   R = 1000 Mbps = 1000x10^6 bps
   Transmission Delay = L / R = 16000/1x10^9 = 1.6x10^-5 seconds
8) What is the propagation delay of link 3?
   d = 2Km = 2000 m
   Propagation delay = d / s = 2000/3x10^8 = 6.66x10^-6 seconds
9) What is the total delay of link 3?
   Total Delay = Transmission + Propagation = 1.6x10^-5 + 6.66x10^-6 = 2.267x 10^-5 sec
10) What is the total delay?
   Total Delay = Sum of all individual (total delays) = 0.01704 seconds

Question #3: Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates R1 = 1800 kbps, R2 = 4.6 Mbps, and R3 = 1.8 Mbps.
a) Assuming no other traffic in the network, what is the throughput for the file transfer?
Answer:
Throughput = Min (1800kbps, 4600kpbs, 1800kbps) = 1800 kbps

b. Suppose the file is 8 MB. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B?
Answer:
[(8*10^6)*8] / (1800*10^3) = 35.55 seconds.

$\frac{8}{1800\times10^3}$ 8×10^6

L d

Question #4: How long does it take a packet of length 50Kbytes to propagate over a link of distance 2, 500 km, propagation speed 1.8 x 10^8 m/s, and transmission rate 200.5 Mbps? Recalculate for distance = 3000m.
Answer:
Propagation delay = d/s = (2500 x 10^3/ 1.8 x 10^8) = 13.8 msec and
Propagation delay = d/s = (3000 / 1.8 x 10^8) = 16.6 usec

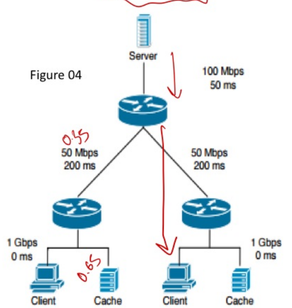
Part b) Consider the scenario as shown in Figure 04 in which a server is connected to a router by a 100Mbps link with a 50ms propagation delay. Initially this router is also connected to two routers, each over a 50Mbps link with a 200ms propagation delay. A 1 Gbps link connects a host and a cache (if present) to each of these routers and we assume that this link has 0 propagation delay. All packets in the network are 20,000 bits long.

i) What is the end-to-end delay from when a packet is transmitted by the server to when it is received by the client? In this case, we assume there are no caches, there's no queuing delay at the routers, and the packet processing delays at routers and nodes are all 0?

ii) Here we assume that client hosts send requests for files directly to the server (caches are not used or off in this case). What is the maximum rate at which the server can deliver data to a single client if we assume no other clients are making requests?

iii) Again we assume only one active client but in this case the caches are on and behave like HTTP caches. A client's HTTP GET is always first directed to its local cache. 65% of the requests can be satisfied by the local cache. What is the average rate at which the client can receive data in this case?

Figure 04



Part c) Explain how a higher education institution, such as FAST-NU Karachi campus, can use private IP addresses for its 3000 devices as PCs and mobile devices and still manage internet connectivity using NAT and one public IP address. First, give a label diagram explaining how your setup NAT within a campus network of three labs, and office spaces for faculty, admin and accounts. Later explain, how an IP datagram from a private IP address are send and received from the campus network to access bbc.co.uk website.

i) $\frac{20\,000}{100\times10^9} + 50\times10^6 + \frac{20\,000}{50\times10^9} + 200\times10^6 + \frac{20000}{1\times10^9}$

ii) 50

iii) $50\times10^6 \times 0.35 + 1\times10^9 \times 0.65$

$= 667.5$ Mbps

Traceroute Program

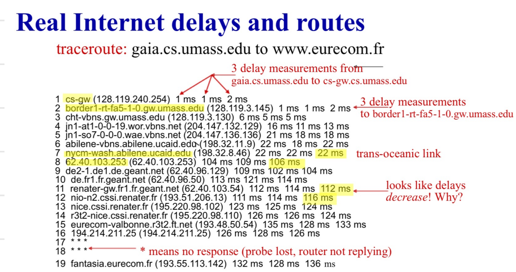
↳ provides delay measures from source to router along end-end Internet path towards destination

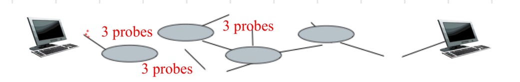
For all i

↳ sends 3 packets to router i

↳ router i will return packets to sender

↳ sender measures time interval b/w transmission and delay

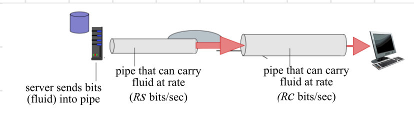


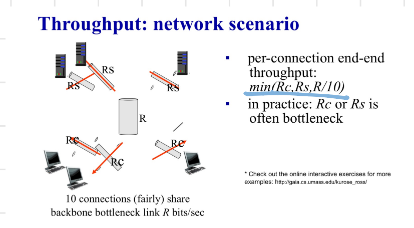


Throughput

↳ rate at which bits are being sent from sender to receiver

$$\boxed{\min\left(R_c, R_s, \underbrace{R}_{\text{no. of connections}}\right)}$$





instantaneous: rate at any given point in time

average: rate over long period of time

bottleneck link: link on end path that constrains end end throughput

Network Security

1. How bad guys attack computer networks
2. How to defend networks againts attacks
3. How to design architech that are immune to attacks

BAD GUYS

Packet interception

↳ Packet sniffing

↳ broadcast media e.g shared ethernet, wireless

↳ Promiscous network interface reads all packets e.g passwords

Fake Identity

↳ IP spoofing

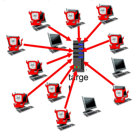
↳ injection of packet with false source address

Denial of Service (DOS)

↳ attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

↓

1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Lines of defence

- authentication: proving you are who you say you are
  - cellular networks provides hardware identity via SIM card; no such hardware assist in traditional Internet
- confidentiality: via encryption
- integrity checks: digital signatures prevent/detect tampering
- access restrictions: password-protected VPNs
- firewalls: specialized "middleboxes" in access and core networks:
  - off-by-default: filter incoming packets to restrict senders, receivers, applications
  - detecting/reacting to DOS attacks
- *... lots more in this course (throughout, Chapter 8)*

Introduc

PROTOCOL LAYERS

Protocol "layers" and reference models

Networks are complex, with many "pieces":
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question: is there any hope of *organizing* structure of network?
- and/or our *discussion* of networks?

Why layering?

dealing with complex systems:
- explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?

identify functionality

1. Application Layer [M]

↳ supporting network applications

e.g HTTP, IMAP, SMTP, DNS

↳ exchanges mssgs to implement some application service using services of transport layer

2. Transport Layer [Ht|M]

↳ process process data transfer

process to process delivery

e. TCP, UDP

↳ transfers mssgs from one process to another using services of network layer

↳ encapsulates application layer mssg M with transport layer header Ht to create transport layer segment [Ht|M]

used by transport layer to implement its service

3. Network Layer [Hn|[Ht|M]]

↳ routing of datagrams from source to destination

host to host delivery

e.g IP, routing protocols

↳ transfers Transport layer segment [Ht|M] from one host to another, using layer links service

↳ encapsulates transport layer segment [Ht|M] with network layer header Hn to create network layer datagram [Hn|[Ht|M]]

used by network layer to implement its service

4. Link Layer [Hl|[Hn|[Ht|M]]]

↳ data transfers b/w neighbouring network elements

e.g Ethernet, WiFi, PPP

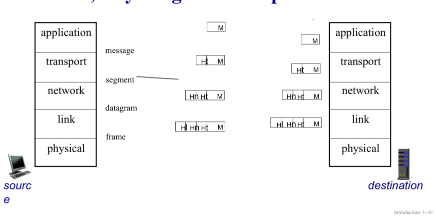
↳ transfers datagram [Hn|[Ht|M]] from host to neighbouring hosts, using network layer services

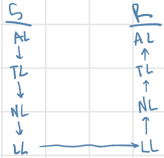
↳ encapsulates network layer datagram [Hn|[Ht|M]] with link layer header Hl to create link layer frame [Hl|[Hn|[Ht|M]]]

5. Physical Layer

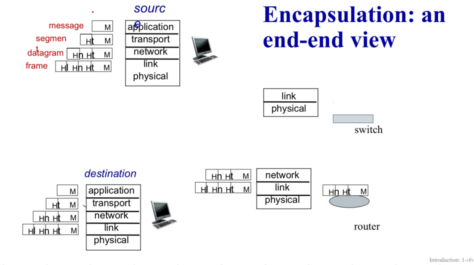
↳ bits on the wire

Services, Layering and Encapsulation





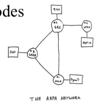
Encapsulation: an end-end view



INTERNET HISTORY

1961-1972: Early packet-switching principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational
- 1972:
  - ARPAnet public demo
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



1972-1980: Internetworking, new and proprietary networks

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- 1979: ARPAnet has 200 nodes

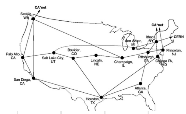
Cerf and Kahn's internetworking principles:
- minimalism, autonomy - no internal changes required to interconnect networks
- best-effort service model
- stateless routing
- decentralized control

define today's Internet architecture

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control
- new national networks: CSnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

NSFNET T1 Network 1991



1990, 2000s: commercialization, the Web, new applications

- early 1990s: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990s: commercialization of the Web

late 1990s – 2000s:
- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

2005-present: scale, SDN, mobility, cloud

- aggressive deployment of broadband home access (10-100's Mbps)
- 2008: software-defined networking (SDN)
- increasing ubiquity of high-speed wireless access: 4G/5G, WiFi
- service providers (Google, FB, Microsoft) create their own networks
  - bypass commercial Internet to connect "close" to end user, providing "instantaneous" access to social media, search, video content, …
- enterprises run their services in "cloud" (e.g., Amazon Web Services, Microsoft Azure)
- rise of smartphones: more mobile than fixed devices on Internet (2017)
- ~18B devices attached to Internet (2017)

Chapter 1: summary

We've covered a "ton" of material!

- Internet overview
- what's a protocol?
- network edge, access network, core
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models
- security
- history

You now have:

- context, overview, vocabulary, "feel" of networking
- more depth, detail, *and fun* to follow!

Network apps

- social networking
- Web
- text messaging
- e-mail
- multi-user network games
- streaming stored video (YouTube, Hulu, Netflix)
- P2P file sharing
- voice over IP (e.g., Skype)
- real-time video conferencing (e.g., Zoom)
- Internet search
- remote login
- …

Q: *your* favorites?

creating Network apps

↳ write programs that

↳ run on diff end systems

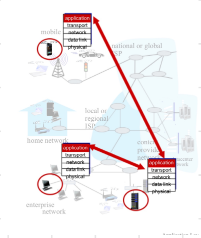
↳ comm. over network

e.g web server software comm. with browser software

↳ dont write softwares for network core devices

↳ as they do not run user application

↳ apps on end system allows for rapid app development, propagation



Client-Server Paradigm

server

↳ always on host

↳ permanent IP address

e.g data center for scalling

Clients

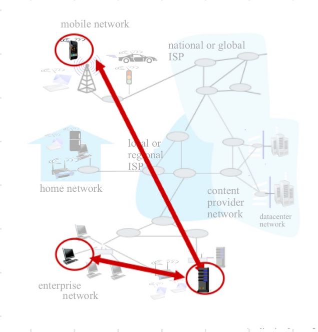
↳ comunicates with server

↳ may be intermittently connected

↳ may have dynamic IP address

↳ dosent comm. directly with eachother

e.g HTTP, IMAP, FTP



Peer peer architecture

↳ no always on server

↳ arbitary end systems directly comm.

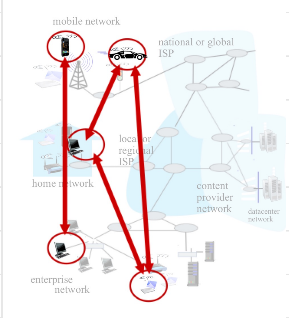
↳ are intermittently connected

↳ can change IP address → complex management

↳ peers request service from other peers, provide service in return to other peers → self scalibility

self scalability – new peers bring new service capacity, as well as new service demands

e.g P2P file sharing



Process communicating

Process
↳ a program running within a host

Client Process
↳ initiates communication

Server process
↳ waits to be contacted

inter process communication
↳ used by two processes to comm. within the same host

↳ process in diff host communicate by exchanging messages

* applications with P2P architecture have client and server processes

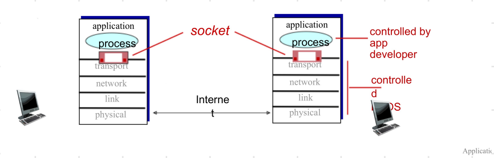
Sockets
↳ process sends/receives mssgs to/from its sockets

↳ sending process

↳ shoves mssg out door

↳ it relies on transport infrastructure to deliver mssg to socket at receiving process

two sockets involved : one on each side



Addressing Processes

identifier
↳ IP address + Port numbers

↳ every process must have an identifier → to receive mssg

- example port numbers:
  - HTTP server: 80
  - mail server: 25
- to send HTTP message to gaia.cs.umass.edu web server:
  - IP address: 128.119.245.12
  - port number: 80
- more shortly…

- *Q:* does IP address of host on which process runs suffice for identifying the process?
- *A:* no, *many* processes can be running on same host

Application Layer Protocol defines

1. Types of mssg exchange — request / response
2. mssg syntax — what feilds in mssgs / how feilds are delineated
3. mssg semantics — meaning of info in feilds
4. rules — when / how → process sends and responds to mssgs
5. Open Protocols → allows for interoperability e.g. HTTP, SMTP
6. Proprietary Protocols e.g. Skype zoom

What transport service does an app need?

data integrity
- some apps (e.g., file transfer, web transactions) require 100% reliable data transfer
- other apps (e.g., audio) can tolerate some loss

timing
- some apps (e.g., Internet telephony, interactive games) require low delay to be "effective"

throughput
- some apps (e.g., multimedia) require minimum amount of throughput to be "effective"
- other apps ("elastic apps") make use of whatever throughput they get

security
- encryption, data integrity, …

Applicatio

Transport service requirements: common apps

application	data loss	throughput	time sensitive?
file transfer/download	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5Kbps-1Mbps video:10Kbps-5Mbps	yes, 10's msec
streaming audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	Kbps+	yes, 10's msec
text messaging	no loss	elastic	yes and no

Application Layer:

Internet Transport Protocols services

TCP service
Transmission Control Protocol

- reliable transport → b/w sending and receiving process
- flow control → sender dosent overwhelm receiver
- congestion control → throttle sender when network overloaded
- connection oriented → setup req b/w client and server processes
- does not provide
  - timming
  - min throughput guarantee
  - security

UDP service
User datagram Protocol

- unreliable data transfer → b/w sending receiving process
- does not provide
  - reliable transport
  - flow control
  - congestion control
  - connection setup
  - timing
  - throughput guarantee
  - security

why is there a UDP?
- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small header size
- no congestion control: UDP can blast away as fast as desired

25

why then?
- fast data transfer
- for video streaming, real time applications

Securing TCP

Vanilla TCP & UDP sockets:
- no encryption
- cleartext passwords sent into socket traverse Internet in cleartext (!)

Transport Layer Security (TLS)
- provides encrypted TCP connections
- data integrity
- end-point authentication

TSL implemented in application layer
- apps use TSL libraries, that use TCP in turn
- cleartext sent into "socket" traverse Internet *encrypted*
- more: Chapter 8

Hyper Text Transfer Protocol (HTTP)

↳ webs application layer protocol

↳ client/server model

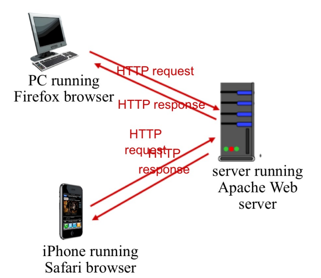
client

↳ browsers that request receives and displays web objects

using HTML protocol

server

↳ web server sends objects in response to requests



Application Layer: 2-

HTTP uses TCP

↳ client initiates TCP connection to server
creates socket
port 80

↳ server accepts TCP connection from client

↳ HTTP mssgs exchanged b/w browser and web server
Application Layer Protocol mssgs
HTTP client
HTTP server

↳ TCP connection closed

webpage

↳ consists of

↳ objects
HTML file
JPeg img
audio file

↳ base HTML file
www.someschool.edu/someDept/pic.gif

HTPP is stateless

↳ server maintains no info about past clients requests

HTTP Connections

1. Non Persistent HTTP

↳ TCP connection opened

↳ 1 object sent to TCP connection

↳ TCP connection closed

CON

↳ downloading multiple objects require multiple connections

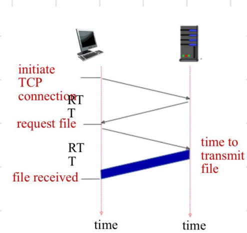
↳ requires 2 RTTs per object

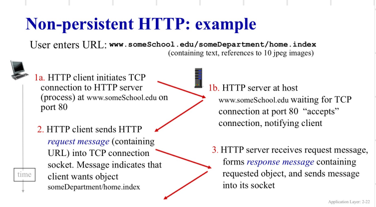
↳ OS overhead for each TCP connection

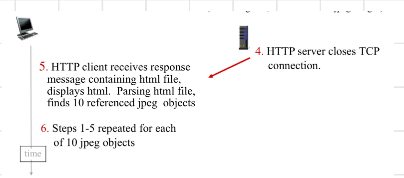
Response Time

↳ 2RTT + file transmission time

- one RTT to initiate TCP connection
- one RTT for HTTP request and first few bytes of HTTP response to return
- obect/file transmission time







2. Persistent HTTP

↳ TCP connection opened to server

↳ multiple objects can be sent to TCP connection → done over single TCP connection

↳ TCP connection closed

Persistent HTTP (HTTP1.1):

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects (cutting response time in half)

RTT

↳ time for packet to travel from client to server and back

Non Persistent

$= 2RTT\ (\text{connection time}) + \overbrace{2\ RTT\ (\text{no of images})}^{\text{file transmission time}}$

$= 2(23.2) \qquad 2(23.2)(1036) = 48116.8$

Persistent

$2RTT\ (\text{connection time}) + RTT\ (\text{no of images})$

Question 3: Persistent and Non-Persistent HTTP **(10+10) Points**

Suppose 1036 images need to be downloaded from the HTTP server. Assume RTT = 23.2 seconds, the total time taken to request and download 1036 images in a non-persistent and persistent connection is:

Always remember that in any case 2RTTs are a must-need to establish a connection.
For Non-Persistent Connections,
We need to establish a connection each time we want to receive an image thus
Non-Persistent Time = 2RTT(connection time) + 2 * RTT * # of images $2(23.2)+2(1036)$
Non-Persistent Time = 2RTT + 2RTT*1036 = 2RTT+ 2072RTT = 2074RTT
Non-Persistent Time = 2074(23.2) = 48116.8 seconds
For Persistent Connections,
We only need to establish a connection at the beginning and thats it.
Persistent Time = 2RTT(connection time) + RTT * # of images
Persistent Time = 2RTT + 1036 RTT = 1038 RTT
Persistent Time = 1038(23.2) = 24081.6 seconds

Non Persisted
= 2RTT (connection time) + 2 RTT (no of images)
= 2 (23.2) 2(23.2)(1036) = 48116.8

Persistent
2RTT(connection time) + RTT(no of images)

Part b) Consider the scenario as shown in Figure 04 in which a server is connected to a router by a 100Mbps link with a 50ms propagation delay. Initially this router is also connected to two routers, each over a 50Mbps link with a 200ms propagation delay. A 1 Gbps link connects a host and a cache (if present) to each of these routers and we assume that this link has 0 propagation delay. All packets in the network are 20,000 bits long.

i) What is the end-to-end delay from when a packet is transmitted by the server to when it is received by the client? In this case, we assume there are no caches, there's no queuing delay at the routers, and the packet processing delays at routers and nodes are all 0?

ii) Here we assume that client hosts send requests for files directly to the server (caches are not used or off in this case). What is the maximum rate at which the server can deliver data to a single client if we assume no other clients are making requests?

iii) Again we assume only one active client but in this case the caches are on and behave like HTTP caches. A client's HTTP GET is always first directed to its local cache. 65% of the requests can be satisfied by the local cache. What is the average rate at which the client can receive data in this case?

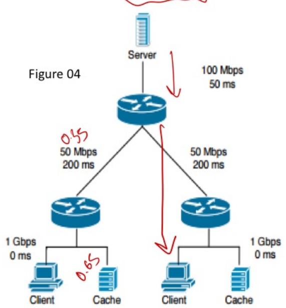


Figure 04

Part c) Explain how a higher education institution, such as FAST-NU Karachi campus, can use private IP addresses for its 3000 devices as PCs and mobile devices and still manage internet connectivity using NAT and one public IP address. First, give a label diagram explaining how your setup NAT within a campus network of three labs, and office spaces for faculty, admin and accounts. Later explain, how an IP datagram from a private IP address are send and received from the campus network to access bbc.co.uk website.

i) $\frac{20\,000}{100\times10^9} + 50\times10^6 + \frac{20\,000}{50\times10^9} + 200\times10^6 + \frac{20000}{1\times10^9}$

ii) 50

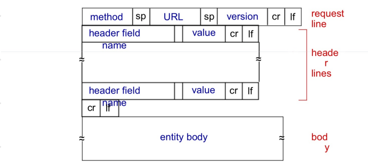
iii) $50\times10^6 \times 0.35 + 1\times10^9 \times 0.65$
$= 667.5$ Mbps

HTTP Messages Types

1. Request message

↳ ASCII → human readable format

HTTP request message: general format



Other HTTP request messages

POST method:
- web page often includes form input
- user input sent from client to server in entity body of HTTP POST request message

GET method (for sending data to server):
- include user data in URL field of HTTP GET request message (following a '?'):

`www.somesite.com/animalsearch?monkeys&banana`

HEAD method:
- requests headers (only) that would be returned *if* specified URL were requested with an HTTP GET method.

PUT method:
- uploads new file (object) to server
- completely replaces file that exists at specified URL with content in entity body of POST HTTP request message

2. Response message

HTTP response message

status line (protocol status code status phrase) → `HTTP/1.1 200 OK`

HTTP response status codes

- status code appears in 1st line in server-to-client response message.
- some sample codes:

200 OK
- request succeeded, requested object later in this message

301 Moved Permanently
- requested object moved, new location specified later in this message (in Location: field)

400 Bad Request
- request msg not understood by server

404 Not Found
- requested document not found on this server

505 HTTP Version Not Supported

Trying out HTTP (client side) for yourself

1. netcat to your favorite Web server:

```
% nc -c -v gaia.cs.umass.edu 80
```

- opens TCP connection to port 80 (default HTTP server port) at gaia.cs.umass. edu.
- anything typed in will be sent to port 80 at gaia.cs.umass.edu

2. type in a GET HTTP request:

```
GET /kurose_ross/interactive/index.php HTTP/1.1
Host: gaia.cs.umass.edu
```

- by typing this in (hit carriage return twice), you send this minimal (but complete) GET request to HTTP server

3. look at response message sent by HTTP server!

(or use Wireshark to look at captured HTTP request/response)

↳ Web cache
↳ cookies
missing

DNS: Domain Name System

uses UDP connection

identify using
↳ host name → human
↳ IP address → router easy process

↳ translate IP adress → name and vice versa

people: many identifiers:
- SSN, name, passport #

Internet hosts, routers:
- IP address (32 bit) - used for addressing datagrams
- "name", e.g., cs.umass.edu - used by humans

Q: how to map between IP address and name, and vice versa ?

Domain Name System (DNS):
- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol:* hosts, DNS servers communicate to *resolve* names (address/name translation)
  - *note:* core Internet function, implemented as application-layer protocol
  - complexity at network's "edge"

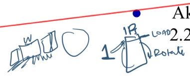
DNS: services, structure

DNS services:
- hostname-to-IP-address translation
- host aliasing
  - canonical, alias names
- mail server aliasing
  - real name   nick name
- load distribution
  - replicated Web servers: many IP addresses correspond to one name

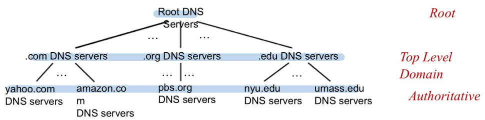
Q: Why not centralize DNS?
- single point of failure — one down all down
- traffic volume — too much traffic
- distant centralized database
- maintenance

↳ is at 1 particular location
so high response time
if far location

A: doesn't scale!
- Comcast DNS servers alone: 600B DNS queries/day
- Akamai DNS servers alone: 2.2T DNS queries/day



DNS: a distributed, hierarchical database



Client wants IP address for www.amazon.com; 1st approximation:
- client queries root server to find .com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

Local DNS name servers

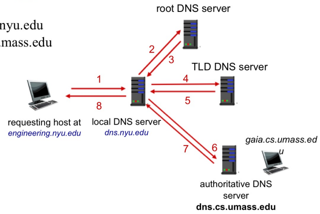
LDNS → Root DNS
A   doesn't know IP   gives IP   B
www.Amazon   TCP connection server

- when host makes DNS query, it is sent to its *local* DNS server
  - Local DNS server returns reply, answering:
    - from its local cache of recent name-to-address translation pairs (possibly out of date!)
    - forwarding request into DNS hierarchy for resolution
  - each ISP has local DNS name server; to find yours:
    - MacOS: `% scutil --dns`
    - Windows: `>ipconfig /all`
- local DNS server doesn't strictly belong to hierarchy

DNS name resolution: iterated query

Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

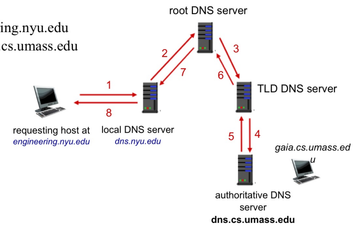
Iterated query:
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution: recursive query

Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

Recursive query:
- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?



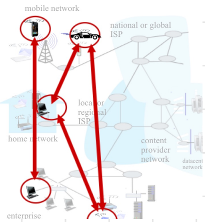
DNS security

DDoS attacks
- bombard root servers with traffic
  - not successful to date
  - traffic filtering
  - local DNS servers cache IPs of TLD servers, allowing root server bypass
- bombard TLD servers
  - potentially more dangerous

Spoofing attacks
- intercept DNS queries, returning bogus replies
  - DNS cache poisoning
  - RFC 4033: DNSSEC authentication services

Peer-to-peer (P2P) architecture

- *no* always-on server
- arbitrary end systems directly communicate
- peers request service from other peers, provide service in return to other peers
  - *self scalability* – new peers bring new service capacity, and new service demands
- peers are intermittently connected and change IP addresses
  - complex management
- examples: P2P file sharing (BitTorrent),



Transmission time: $\frac{L}{R} \geq \frac{NF}{u_s}$

no of copies

min download time $= \frac{F}{d_{min}}$

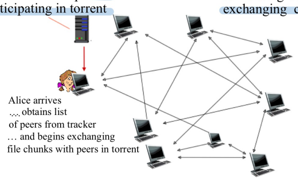
Distribution time $= \max\left[\frac{NF}{u_s}, \frac{F}{d_{min}}\right]$

P2P file distribution: BitTorrent

- file divided into 256Kb chunks
- peers in torrent send/receive file chunks

tracker: tracks peers participating in torrent

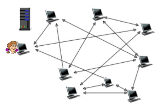
torrent: group of peers exchanging chunks of a file



Application La

P2P file distribution: BitTorrent

- peer joining torrent:
  - has no chunks, but will accumulate them over time from other peers
  - registers with tracker to get list of peers, connects to subset of peers ("neighbors")



- while downloading, peer uploads chunks to other peers
- peer may change peers with whom it exchanges chunks
- *churn:* peers may come and go
- once peer has entire file, it may (selfishly) leave or (altruistically) remain in torrent

Application Layer: 2

BitTorrent: requesting, sending file chunks

Requesting chunks:

- at any given time, different peers have different subsets of file chunks
- periodically, Alice asks each peer for list of chunks that they have
- Alice requests missing chunks from peers, rarest first

so if they leave from network chunk still avl

Sending chunks: tit-for-tat

- Alice sends chunks to those four peers currently sending her chunks *at highest rate*
  - other peers are choked by Alice (do not receive chunks from her)
  - re-evaluate top 4 every10 secs
- every 30 secs: randomly select another peer, starts sending chunks
  - "optimistically unchoke" this peer
  - newly chosen peer may join top 4

Question 1: Please calculate the LAN and Access link utilization in following scenario. **[10 points]**

a. Average object size is 450 Kilo Bytes
b. Average request rate from the browsers to origin server is 17 requests/seconds
c. Cache Hit ratio is 0.37

Average Requests/Sec = 17
Avg Size = 450kb * 8 = 3.600 Mbits
Total load = (3.6*17) = 61.2 Mbits/sec
Lan Utilization = 61.2%
Cache hit ratio = 0.37 which would put load of 22.65 Mbits/sec to access link
So it will be 100% utilized and packets will drop due to queuing.

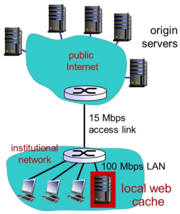


Figure 1

Avg Req/sec = 17
Avg size = 3.600 Mbits
Cache hit ratio = 0.37

Total Load = Avg Req/sec x Avg size
= 17 x 3.6Mbits
= 61.2 Mbits

LAN Utilization = Total load %
= 61.2 %

load on access link = total load x cache hit ratio
= 61.2 x 0.37
= 22.64 Mbits/sec