Comparison between Dual-Core processors and Quad-Core Processors

Dual-Core Processors

Characteristics:

- Number of Cores: 2
- Threads: Typically 2 or 4 (with Hyper-Threading)
- Power Consumption: Generally lower than quad-core processors, making them more energy-efficient.
- Heat Generation: Produces less heat, requiring simpler cooling solutions.

Performance:

- Multitasking: Can handle basic multitasking well, suitable for running multiple light applications simultaneously.
- Applications: Ideal for everyday tasks like web browsing, office applications, media playback, and light gaming.
- Cost: Typically less expensive, making them a good choice for budget-conscious buyers.

Use Cases:

- General Computing: Perfect for users who perform basic computing tasks.
- Energy Efficiency: Suitable for devices where battery life and energy consumption are important factors.

Quad-Core Processors

Characteristics:

- Number of Cores: 4
- Threads: Typically 4 or 8 (with Hyper-Threading)
- Power Consumption: Higher than dual-core processors but still efficient, especially with modern architectures.
- Heat Generation: Generates more heat, necessitating better cooling solutions.

Performance:

- Multitasking: Handles more intensive multitasking efficiently, capable of running several demanding applications simultaneously.
- Applications: Ideal for more complex tasks like gaming, video editing, software development, and running virtual machines.
- Cost: Generally more expensive than dual-core processors but offers better performance for more demanding tasks.

Use Cases:

- Advanced Computing: Suitable for users who need more computing power for intensive applications.
- Mid-Range to High-End Systems: Common in mid-range to high-end laptops, desktops, and some high-performance tablets.
- Performance Demands: Ideal for gamers, content creators, and professionals who run resource-intensive applications.

Conclusion

Choose Dual-Core If:

- You have basic computing needs like web browsing, office work, and media consumption.
- You are budget-conscious and want a cost-effective solution.
- You prioritize battery life and energy efficiency.

Choose Quad-Core If:

- You require better performance for multitasking and running intensive applications.
- You are into gaming, content creation, or professional work that demands more processing power.
- You are willing to invest more for better overall performance and efficiency.

Comparison between Intel Core i5 and Intel Core i7 processors

Intel Core i5 Processors

Characteristics:

- Cores and Threads: Typically 4-6 cores and 8-12 threads (in newer generations, some may have more cores).
- Clock Speed: Moderate base clock speeds with Turbo Boost technology to increase performance during intensive tasks.
- Cache: Moderate-sized cache, usually around 6-12 MB.
- Integrated Graphics: Good integrated graphics performance suitable for casual gaming and multimedia.

Performance:

- Multitasking: Capable of handling multiple applications simultaneously but may struggle with extremely intensive multitasking compared to i7.
- Applications: Suitable for everyday tasks, moderate gaming, office applications, and some content creation like photo editing and casual video editing.
- Power Consumption: Balanced power consumption, making them efficient for laptops and desktops.

Use Cases:

- General Computing: Ideal for most users who need reliable performance for a variety of tasks.
- Mid-Range Systems: Common in mid-range laptops and desktops, offering a good balance of performance and cost.
- Cost: More affordable than i7 processors, providing good value for money.

Intel Core i7 Processors

Characteristics:

- Cores and Threads: Typically 4-8 cores and 8-16 threads .

- Clock Speed: Higher base and boost clock speeds compared to i5, with better Turbo Boost capabilities.
- Cache: Larger cache, usually around 8-16 MB, allowing for faster access to frequently used data.
- Integrated Graphics: Superior integrated graphics performance compared to i5, suitable for more demanding graphics tasks.

Performance:

- Multitasking: Excellent at handling intensive multitasking and running multiple demanding applications simultaneously.
- Applications: Ideal for heavy-duty tasks such as gaming, professional video editing, 3D rendering, and software development.
- Power Consumption: Generally higher power consumption due to increased performance, requiring better cooling solutions.

Use Cases:

- Advanced Computing: Suitable for power users who need robust performance for demanding applications.
- High-End Systems: Common in high-end laptops and desktops, catering to gamers, content creators, and professionals.
- Cost: More expensive than i5 processors, reflecting the higher performance and capabilities.

Conclusion

Choose Intel Core i5 If:

- You need reliable performance for everyday tasks, moderate gaming, and light content creation.
- You are budget-conscious and want good value for your money.
- You prefer a balance between performance and power efficiency.

Choose Intel Core i7 If:

- You require top-tier performance for intensive multitasking, high-end gaming, and professional-grade content creation.
- You are willing to invest more for superior performance and capabilities.
- You need a processor that can handle demanding applications and workloads efficiently.

Binary To Decimal Conversion

128	64	32	16	8	4	2	1	Answers
1	0	0	1	0	0	1	0	146
0	1	1	1	0	1	1	1	119
1	1	1	1	1	1	1	1	255
1	1	0	0	0	1	0	1	197
1	1	1	1	0	1	1	0	246
0	0	0	1	0	0	1	1	19
1	0	0	0	0	0	0	1	129
0	0	1	1	0	0	0	1	49
0	1	1	1	1	0	0	0	120
1	1	1	1	0	0	0	0	240
0	0	1	1	1	0	1	1	59
0	0	0	0	0	1	1	1	7

Scratch Area

128 64
16 32
2 16
146 4
2
1
119

Address Class Identification

Address	Class
10.250.1.1	A
150.10.15.0	B
192.14.2.0	C
148.17.9.1	B
193.42.1.1	C
126.8.156.0	A
220.200.23.1	C
230.230.45.58	D
177.100.18.4	B
119.18.45.0	A
249.240.80.78	E

Decimal To Binary Conversion

Use all 8 bits for each problem

128	64	32	16	8	4	2	1	= 255
1	1	1	0	1	1	1	0	238
0	0	1	0	0	0	1	0	34
0	1	1	1	1	0	1	1	123
0	0	1	1	0	0	1	0	50
1	1	1	1	1	1	1	1	255
1	1	0	0	1	0	0	0	200
0	0	0	0	1	0	1	0	10
1	0	0	0	1	0	1	0	138
0	0	0	0	0	0	0	1	1
0	0	0	0	1	1	0	1	13
1	1	1	1	1	0	1	0	250
0	1	1	0	1	0	1	1	107
1	1	1	0	0	0	0	0	224
0	1	1	1	0	0	1	0	114
1	1	0	0	0	0	0	0	192

Scratch Area

238
-128
110
-64
46
-32
14
-8
6
-4
2
-2
0

34
-32
2
-2
0

Default Subnet Masks

Write the correct default subnet mask for each of the following addresses:

177.100.18.4	255 . 255 . 0 . 0
119.18.45.0	255 . 0 . 0 . 0
191.249.234.191	255.255.255.0
223.23.223.109	255.255.255.0
10.10.250.1	255.0.0.0
126.123.23.1	255.0.0.0
223.69.230.250	255.255.0.0
192.12.35.105	255.255.255.0
77.251.200.51	255.255.255.0
189.210.50.1	255.255.255.0
88.45.65.35	255.255.255.0
128.212.250.254	255.255.255.0

Network Addresses

Using the IP address and subnet mask shown write out the network address:

188.10.18.2
255.255.0.0 — 188 . 10 . 0 . 0

10.10.48.80
255.255.255.0 — 10 . 10 . 48 . 0

192.149.24.191
255.255.255.0 — 192.149.24.0

150.203.23.19
255.255.0.0 — 150.203.0.0

10.10.10.10
255.0.0.0 — 10.0.0.0

186.13.23.110
255.255.255.0 — 186.13.23.0

223.69.230.250
255.255.0.0 — 223.69.0.0

200.120.135.15
255.255.255.0 — 200.120.135.0

Network & Host Identification

Circle the network portion of these addresses:	Circle the host portion of these addresses:
177.100.18.4	10.15.123.50
119.18.45.0	171.2.199.31
209.240.80.78	198.125.87.177
199.155.77.56	223.250.200.222
117.89.56.45	17.45.222.45
215.45.45.0	126.201.54.231
192.200.15.0	191.41.35.112
95.0.21.90	155.25.169.227
33.0.0.0	192.15.155.2
158.98.80.0	123.102.45.254
217.21.56.0	148.17.9.155
10.250.1.1	100.25.1.1

Host Addresses

Using the IP address and subnet mask shown write out the host address:

188.10.18.2
255.255.0.0

0 . 0 . 18 . 2

10.10.48.80
255.255.255.0

0 . 0 . 0 . 80

222.49.49.11
255.255.255.0

0.0.49.11

128.23.230.19
255.255.0.0

0.0.230.19

10.10.10.10
255.0.0.0

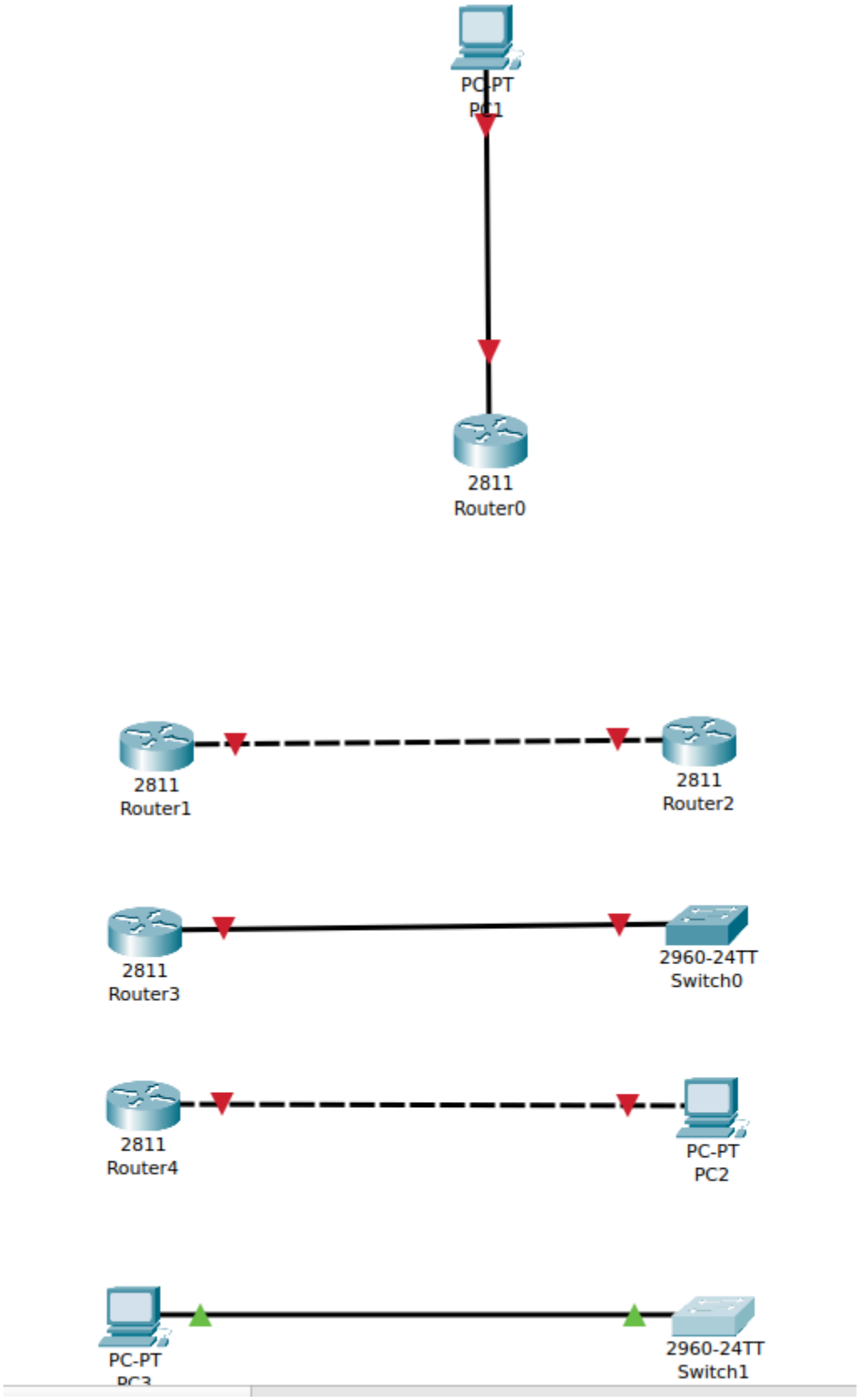
0.10.10.10

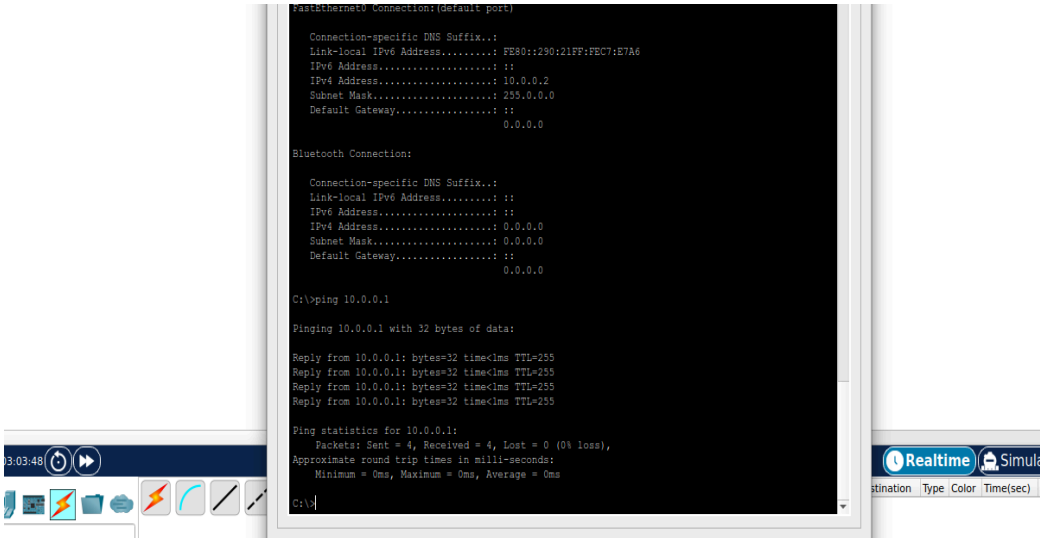
200.113.123.11
255.255.255.0

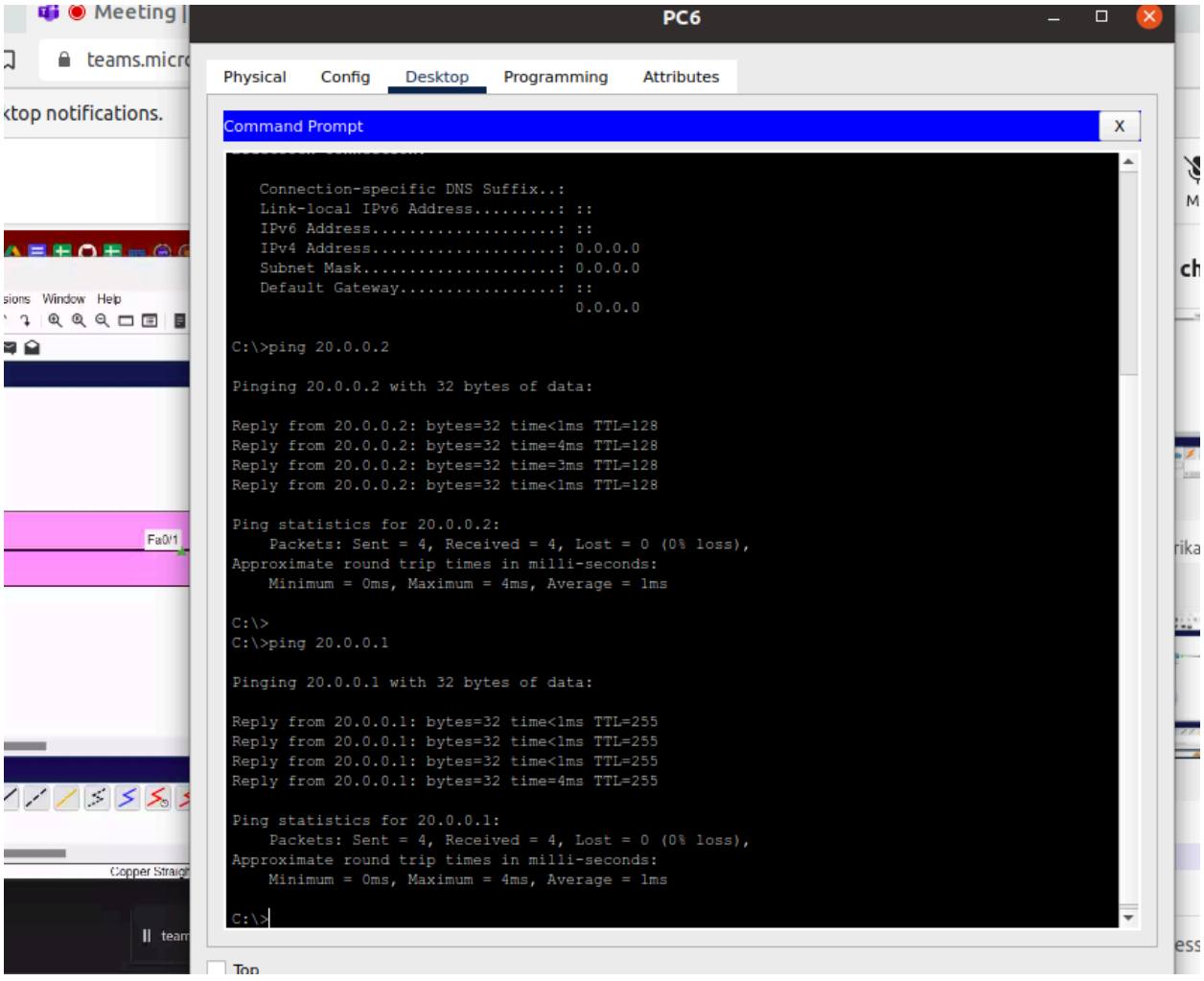
0.0.0.11

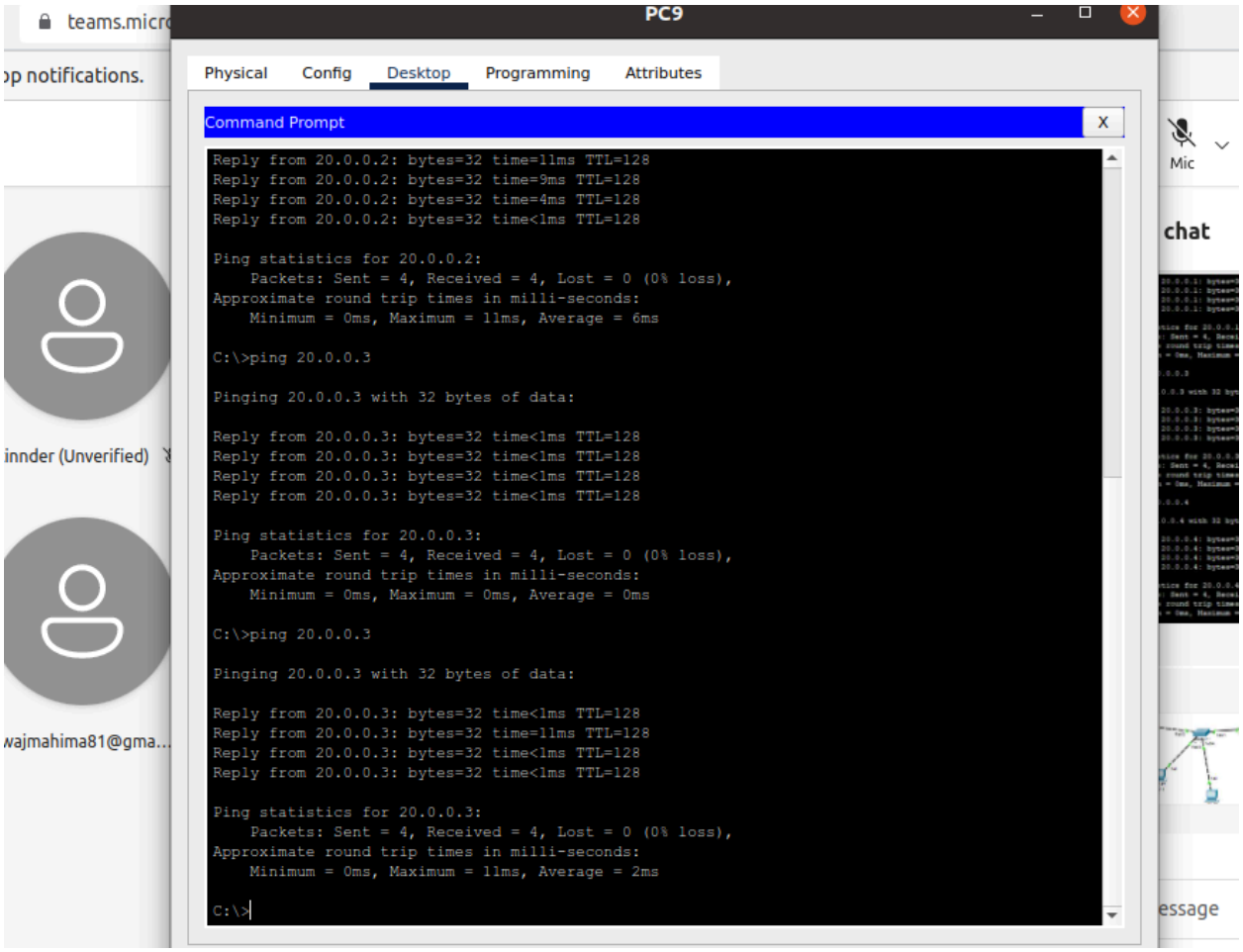
223.169.23.20
255.255.0.0

0.0.23.20


PC-PT
PC1
2811
Router0
2811
Router1
2811
Router2
2811
Router3
2960-24TT
Switch0
2811
Router4
PC-PT
PC2
PC-PT
PC3
2960-24TT
Switch1


FastEthernet0 Connection:(default port)
Connection-specific DNS Suffix..:
Link-local IPv6 Address.....: FE80::290:21FF:FEC7:E7A6
IPv6 Address.....: ::
IPv4 Address.....: 10.0.0.2
Subnet Mask.....: 255.0.0.0
Default Gateway.....: ::
0.0.0.0
Bluetooth Connection:
Connection-specific DNS Suffix..:
Link-local IPv6 Address.....: ::
IPv6 Address.....: ::
IPv4 Address.....: 0.0.0.0
Subnet Mask.....: 0.0.0.0
Default Gateway.....: ::
0.0.0.0
C:\>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time<1ms TTL=255
Reply from 10.0.0.1: bytes=32 time<1ms TTL=255
Reply from 10.0.0.1: bytes=32 time<1ms TTL=255
Reply from 10.0.0.1: bytes=32 time<1ms TTL=255
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
Realtime
Type
Color
Time(sec)

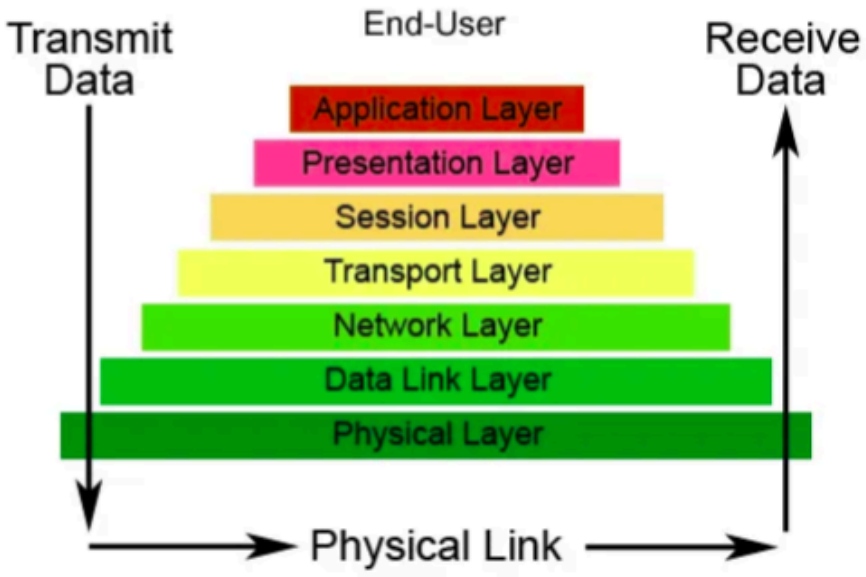

PC6
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
Connection-specific DNS Suffix..:
Link-local IPv6 Address.....: ::
IPv6 Address.....: ::
IPv4 Address.....: 0.0.0.0
Subnet Mask.....: 0.0.0.0
Default Gateway.....: ::
0.0.0.0
C:\>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Reply from 20.0.0.2: bytes=32 time<1ms TTL=128
Reply from 20.0.0.2: bytes=32 time=4ms TTL=128
Reply from 20.0.0.2: bytes=32 time=3ms TTL=128
Reply from 20.0.0.2: bytes=32 time<1ms TTL=128
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4ms, Average = 1ms
C:\>
C:\>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time<1ms TTL=255
Reply from 20.0.0.1: bytes=32 time<1ms TTL=255
Reply from 20.0.0.1: bytes=32 time<1ms TTL=255
Reply from 20.0.0.1: bytes=32 time=4ms TTL=255
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4ms, Average = 1ms
C:\>
Top



DAY 5 [20JUNE]

OSI MODEL

- The OSI model is a conceptual framework that divides network communications into seven layers.
- OSI stands for Open Systems Interconnection.
- Created by the International Standards Organization.
- Was created as a framework and reference model to explain how different networking technologies work together and interact.



Physical Layer (Layer 1)

- Responsible for transmitting raw bits over a physical medium (e.g., cable, wireless)
- Defines the physical means of data transmission (e.g., voltage levels, frequency)
- Provides bit synchronization and bit rate control

Data Link Layer (Layer 2)

- Responsible for error-free transfer of data frames between two devices
- Provides framing, error detection and correction, and flow control
- Uses MAC (Media Access Control) addresses to identify devices

Network Layer (Layer 3)

- Responsible for routing data between devices on different networks
- Provides logical addressing (IP addresses) and routing

- Segments and reassembles data into packets

Transport Layer (Layer 4)

- Responsible for ensuring reliable data transfer between devices
- Provides segmentation and reassembly, flow control, and error detection and correction
- Uses port numbers to identify applications

Session Layer (Layer 5)

- Establishes, maintains, and terminates connections between applications
- Manages dialogues between applications
- Provides session establishment, maintenance, and termination

Presentation Layer (Layer 6)

- Converts data into a format that can be understood by the receiving device
- Provides data compression, encryption, and formatting

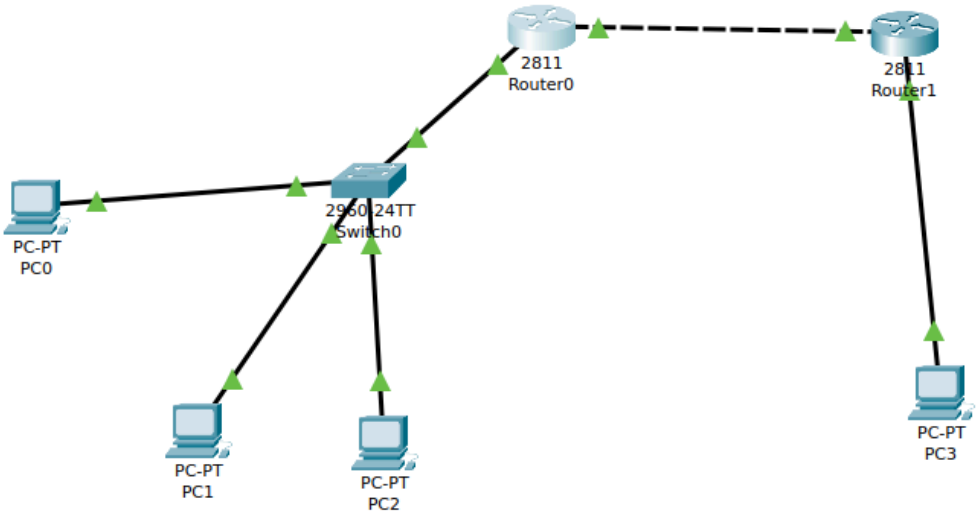
Application Layer (Layer 7)

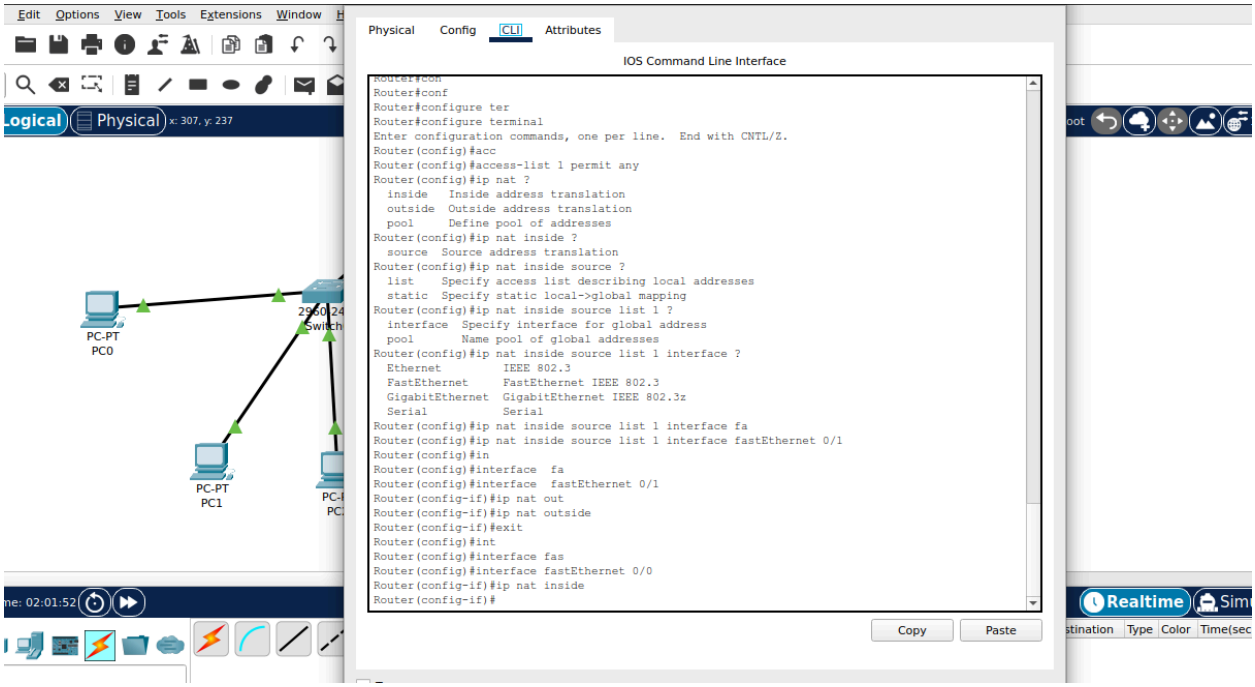
- Provides services to end-user applications (e.g., email, file transfer)
- Supports functions such as email, file transfer, and virtual terminals

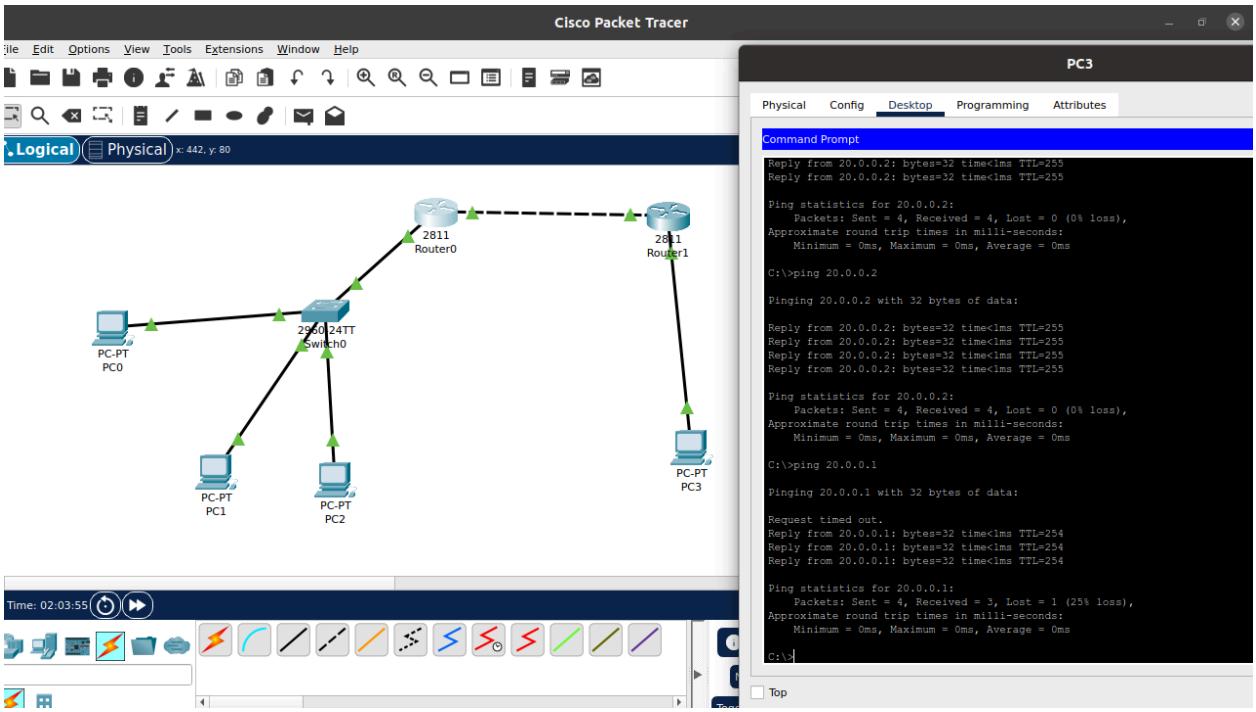
Layer	Description	Use case	Central Device/Protocols
Application **(7)**	Provides services to end-user	Resource sharing, Remote file access , Remote printer access, email, file transfer	**User Application SMTP**
Presentation **(6)**	Provides data compression, encryption, and formatting.	Character code translation , Data conversion , Data encryption.	JPEG/EBCDIC/TIFF/GIF
Session **(5)**	Provides session establishment, maintenance, and termination.	Session establishment , session support, logging,	**Logical Ports RPC/UDP**
Transport **(4)**	Responsible for ensuring reliable data transfer between devices.	Message segmentation, message traffic control .	TCP/UDP
Network **(3)**	Responsible for routing data between devices on different networks	Routing , Subnet traffic control , logica-physical address mapping , subnet usage accounting.	**Routers IP/IPX/ICMP**
Data Link **(2)**	Responsible for error-free transfer of data frames between two devices.	Establish and terminate the logical link between nodes , media access control .	**Switch Bridge Wap PPP/SLIP**
Physical **(1)**	Responsible for transmitting raw bits over a physical medium.	Data Encoding , Physical medium attachment , physical medium transmission bits and volts.	**HUb**

Major DIFFRENCE BETWEEN OSI MODEL AND TCP/IP:

- Number of Layers: OSI has 7 layers, while TCP/IP has 4 layers.
- Purpose: OSI is a conceptual
- model, while TCP/IP is a practical implementation.
- Implementation: OSI is not implemented in real-world networks, while TCP/IP is used in most networks.
- Layer Merging: TCP/IP combines the Physical and Data Link Layers into a single Network Access Layer.
- The OSI model is low in usage while TCP/IP is most widely used.
- The OSI model is less reliable than the TCP/IP Model.


2811
Router0
2811
Router1
2960-24TT
Switch0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3


Edit Options View Tools Extensions Window
Physical Config CLI Attributes
IOS Command Line Interface
Router#conf
Router#configure ter
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#acc
Router(config)#access-list 1 permit any
Router(config)#ip nat ?
inside Inside address translation
outside Outside address translation
pool Define pool of addresses
Router(config)#ip nat inside ?
source Source address translation
Router(config)#ip nat inside source ?
list Specify access list describing local addresses
static Specify static local->global mapping
Router(config)#ip nat inside source list 1 ?
interface Specify interface for global address
pool Name pool of global addresses
Router(config)#ip nat inside source list 1 interface ?
Ethernet IEEE 802.3
FastEthernet FastEthernet IEEE 802.3
GigabitEthernet GigabitEthernet IEEE 802.3z
Serial Serial
Router(config)#ip nat inside source list 1 interface fa
Router(config)#ip nat inside source list 1 interface fastEthernet 0/1
Router(config)#in
Router(config)#interface fa
Router(config)#interface fastEthernet 0/1
Router(config-if)#ip nat out
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#int
Router(config)#interface fas
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip nat inside
Router(config-if)#
Copy
Paste
Realtime



DAY 6 [20 JUNE]

LINUX OPERATING SYSTEM

LINUX → Linux is an open source O.S developed by Linus torvalds in 1991. mainly it is designed for system, serves, DB.

Various Linux distributor :- Linux Mint, fedora, Debian, ubuntu, solus.

Command	Description
1) ls	Display the content of present working direct
2) cd	change the directory
3) pwd	present working directory (Display)
4) mkdir	make/create a new working directory
5) rm	Remove object, file, directory, symbolic link
6) cp	Creat copy of source file at the specified destination
7) mv	move/Rename the file
8) touch	create a Empty file in current working directory
9) Cat	View the content of a file
10) Tac	view the content of a file in Reverse order.
11) grep-	~~filter~~ filter command in Linux. the grep filter search a file for a particulare pattern of character & display.
12) find	find command in Linux is a dynamic utility design for comprehensive file & directory searches within a hierachical structure.

[find [path] [option] [expression]]

Eg. [find ./download -name sample.txt]

* Tar (tape archieve) ⇒ Used to create archieve & Extract the Archieve files.

[Tar [option] [archieve file] [file/dir]]

Archieve file - An archieve file is a file that is composed of one or more file along with meta data.

⇒ Creating an uncompressed tar

[tar cvf file.tar *.c]
- c → create new archieve.
- v → display verbose o/p showing progress
- f → specify the file Name

⇒ Extracting file from Archieve

[tar xvf file.tar]
- x → extract file.
- v → display verbose o/p
- f → file

& software.

⇒ create tar file (file.tar.gz) which is the archieve of .c file

[tar cvzf file.tar.gz *.c]

↳ uses gzip compression

⇒ Extract file from tar archieve

[tar xvzf file.tar.gz]

* Gzip - used for compress file

[gzip [option] [fileName]]

compression

gzip mydoc.txt → mydoc.txt.gz

decompression

gzip -d mydoc.txt.gz → mydoc.txt

option

-f - forcefully compression

-k ⇒ compress but keep the original file.

-v = show the name & % reduction

*gunzip - used to compress or expand a file or a list of file in Linux. (.gz, .z, -z, -gz, -z, -z)

gunzip mydoc.txt → mydoc.txt.gz

gunzip mydoc.txt.gz → mydoc.txt

-c - view the content of file without uncompressing it.

[gunzip -c mydoc.txt.gz]

*zip - used to compress file to Reduce file size & its also used as a file package utility.

[zip [filename]]

*unzip - extract/uncompress file from a zip Archieve.

[unzip [zip file Name]]

*chmod (change mode) - used to modify permission so that it can grant or restrict access to directories & files.

chmod [option] [mode] [file Name]

Numbes	Permition	Symbol
0	No "	---
1	Execute	--x
2	Write	-w-
3	Exe+Write	-wx
4	Read	r--
5	Read+Exe	r-x
6	Read+Write	rw-x
7	Read+Write+Execute	rwx

Chmod 721 Semple.txt
- 7 → User
- 2 → group
- 1 → other

*PS (Process Status)

→ it show the processes for the current shell.

PID	TTY	TIME	CMD
↓	↓	↓	↓
Process ID	Terminal type that the user is logged into	amount of CPU in a Min.	Name of command that launched the process.

PS-A - showing all process.

*TOP - provide real time information about System process, used for monitoring Showing process, system Resource utilization & other critical system metrics.

Info including :-

PID, USER, PR, NV, VIRT, RES, SHR, S, +- CPU, +-MEM, Time

Top -n 10 → automatically exit after 10 No. of Repetation.

Top -u user1 → Display specific user process.

*Kill - Kill command is used to terminate process Manually. Kill send the signal to process that terminates the process

[kill [signal] PID]

Signal can be specified in three ways:-

1) By Number - Specify a signal using Number

[kill -9 1212]

2) By SIG prefix - Send signal as a [Term]

[kill -Term 1212]

Signal Name	Signal No.	Description
SIGHUP	1	- It hangup detect on controlling terminal or death of controlling ps
SIGINT	2	- Interrupt from keyboard
SIGKILL	9	- Kill signal
SIGTERM	15	- Terminate signal

Kill -l → Show all signal available

Killall firefox → Kill all process with the Name firefox.

Ping (packet internat groper) :- used to check Network connectivity b/w host & server.

Syntax

[Ping [option] host_or_IP_address]

[Ping www.google.com]

Continously monitor Network connectivity

[Ping -c 4 www.google.com]

↳ allow to set packet (4)

[Ping -f www.google.com]

↳ continous ping

diff → compare the content of two file & display the difference b/w them.

`diff file1 file2`

Head → print the top N No. of data of given I/P by default it print 10 line.

`Head text.txt`

Tail → complementry of head display the bottom 10 line by default of file in Reverse order.

Sort → used to sort the file, arranging the Record in particular way.

`[Sort text.txt]` `[Sort -r text.txt]` → Rev. decending

Echo - allow user to display line of text or string, basically used in shell scripting

echo "sunny"

History - used to check the history of terminal

uname (unix Name) - provide key detail of system

uname or uname -a.

df (disk free) - it provide the info about disk space utilization

df

Check particular

df text.txt

du (disk uses) - analyze & report on disk usage within directory & files.

du

du text.txt

nohup (No hang up) :- If you want your process keep running even after closing the terminal than we can use nohup

`nohup ./echo.sh &`

bg (background) - used to place foreground jobs in background.

`[bg [job spec.]]`

fg (foreground) :- used to put a background job in foreground.

`[fg (job specification)]`

Here is the history of terminal which i have done

```
486  sudo apt-get update
487  sudo apt-get upgrade
488  history
489  man ls
490  mkdir demodir
491  cd demodir/
492  touch test123.txt
493  cat > test123.txt
494  cat test123.txt
495  ls
496  pwd
497  mv test123.txt dem.txt
498  ls
499  touch myfie.txt
500  cp dem.txt myfie.txt
501  cat myfie.txt
502  open myfie.txt
503  find myfie.txt
504  ln
505  ln myfile
506  uname
507  ping www.google.com
508  who
509  whoami
510  vi file1.txt
511  tail file1.txt
512  less file1.txt
513  wc file1.txt
514  gzip file1.txt
515  ls
516  gunzip file1.txt.gz
517  ls
```

```
508  who
509  whoami
510  vi file1.txt
511  tail file1.txt
512  less file1.txt
513  wc file1.txt
514  gzip file1.txt
515  ls
516  gunzip file1.txt.gz
517  ls
518  sort dem.txt
519  echo sunny
520  chmod 777 myfile.txt
521  ls
522  chmod 777 dem.txt
523  du
524  df
525  basename
526  man basename
527  tar -cvf archieve_name.tar
528  tar -cvf archieve_name.tar file12
529  tar -cvf archieve_name.tar dem.txt
530  ps
531  top
532  type dem.txt
533  su
534  passwd
535  alias ll='ls -la'
536  ll
537  env
538  printenv
539  history
stree@opstree-Latitude-3460:~/demodir$
```