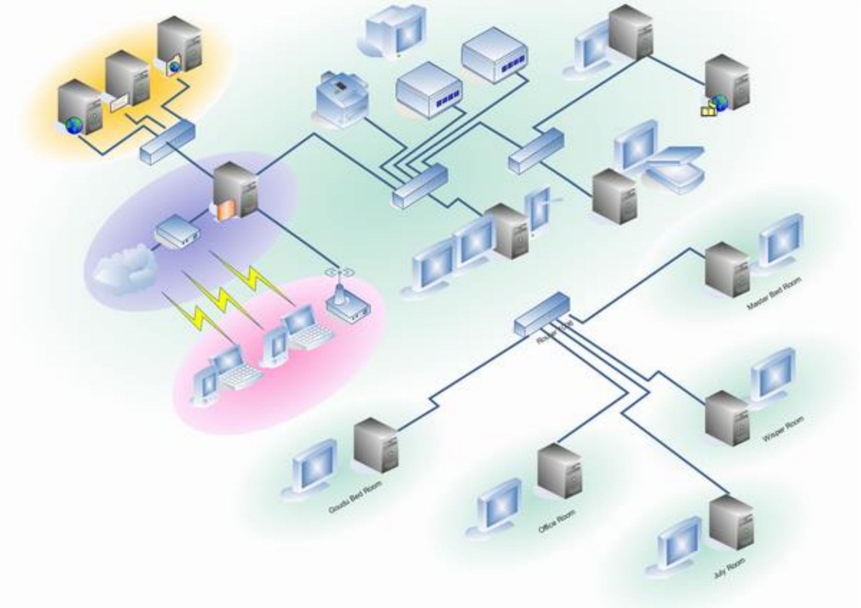
Linux For Embedded Systems

For Arabs

Course 102:

Understanding Linux

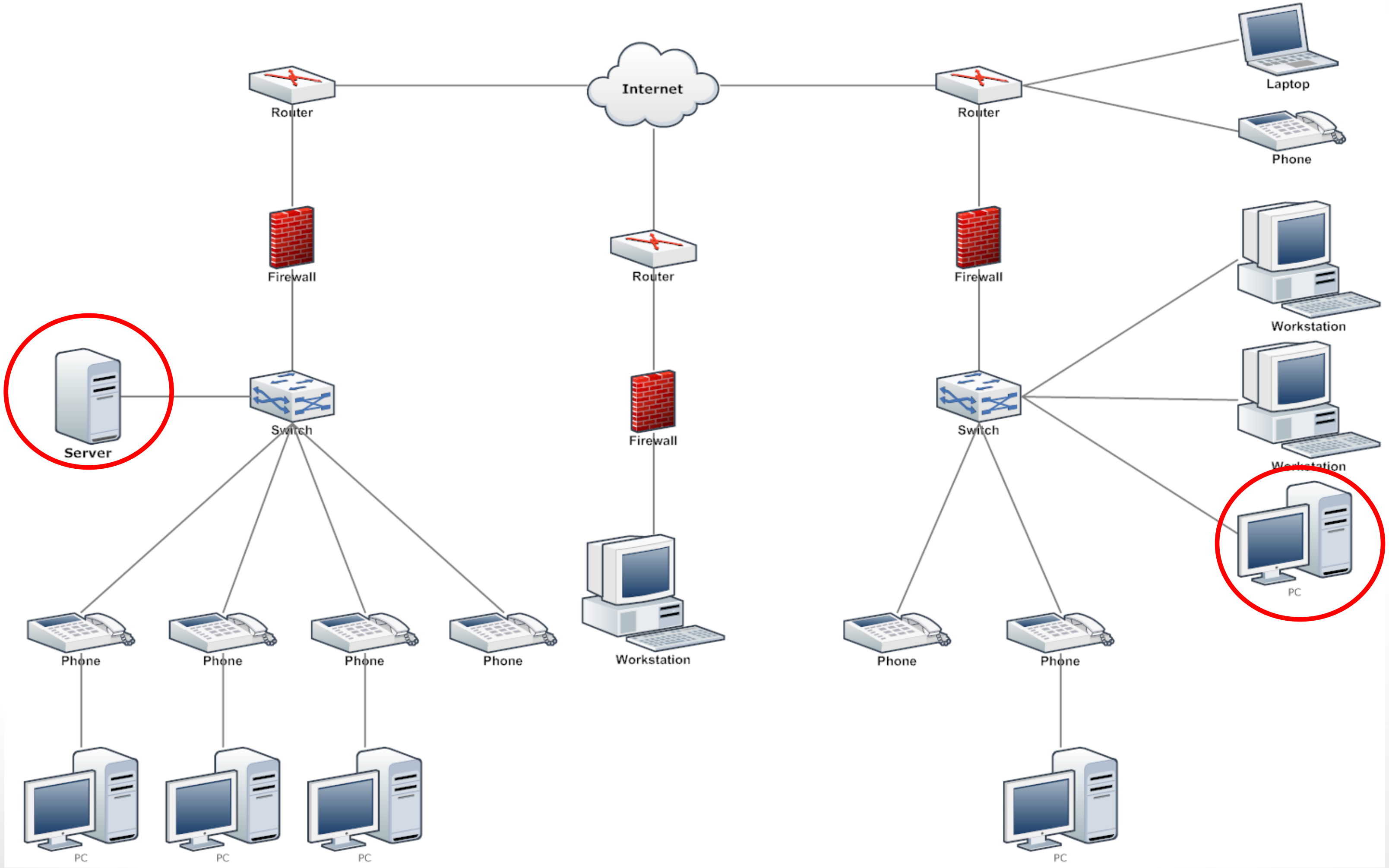
Ahmed ElArabawy



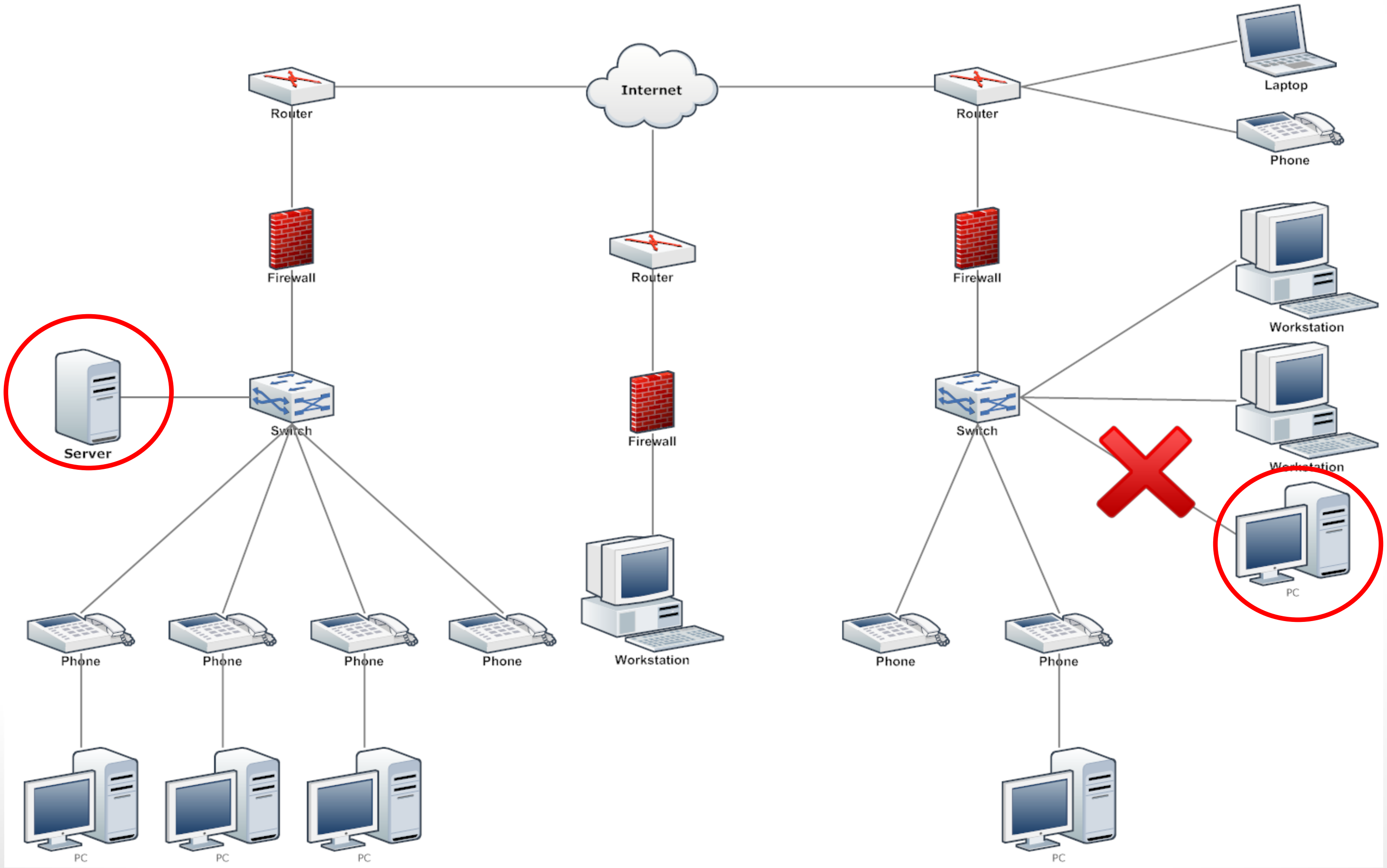
Lecture 21:
Networking in Linux (Applications)

Utility Applications

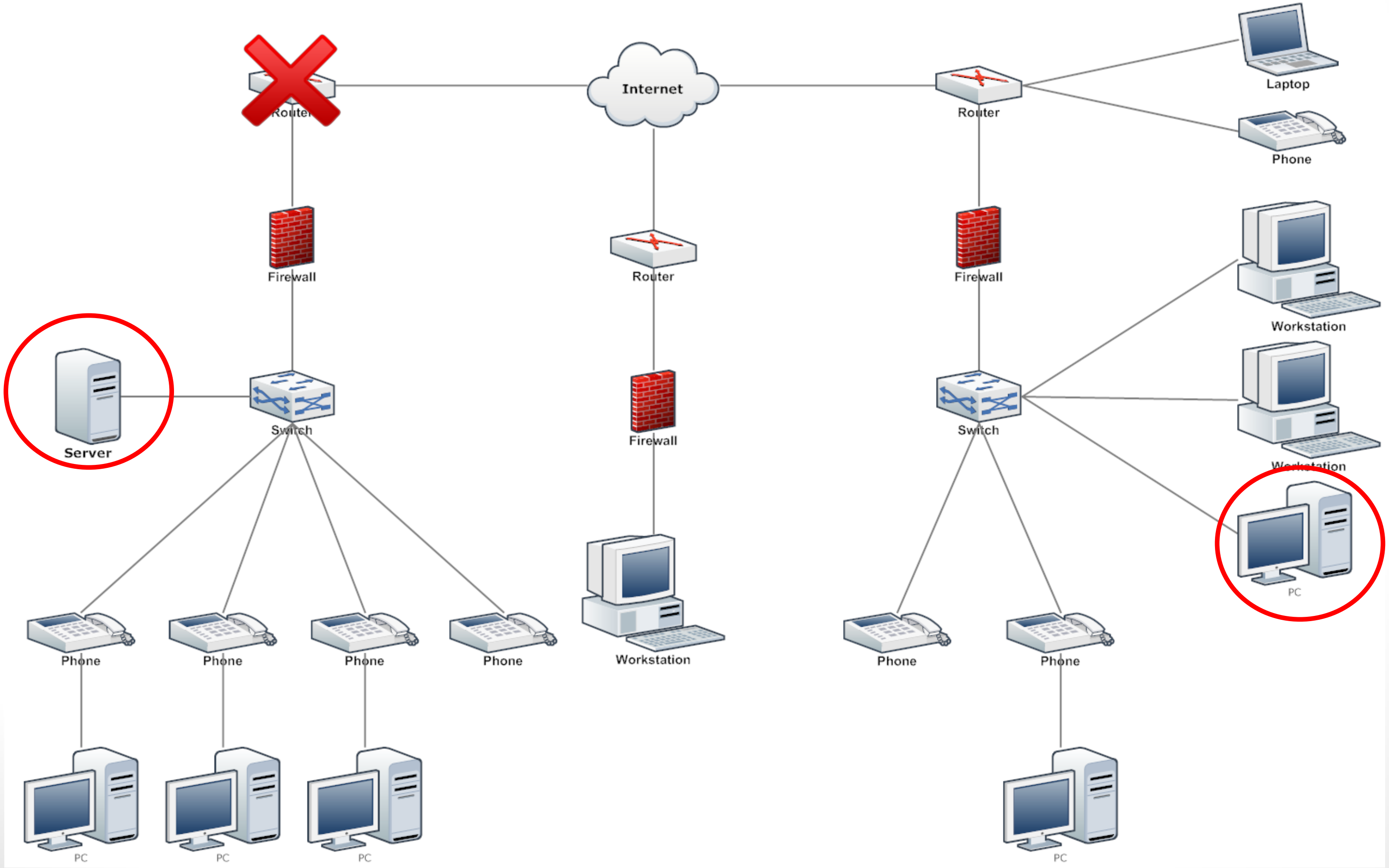
Check Network Connectivity (ping Command)



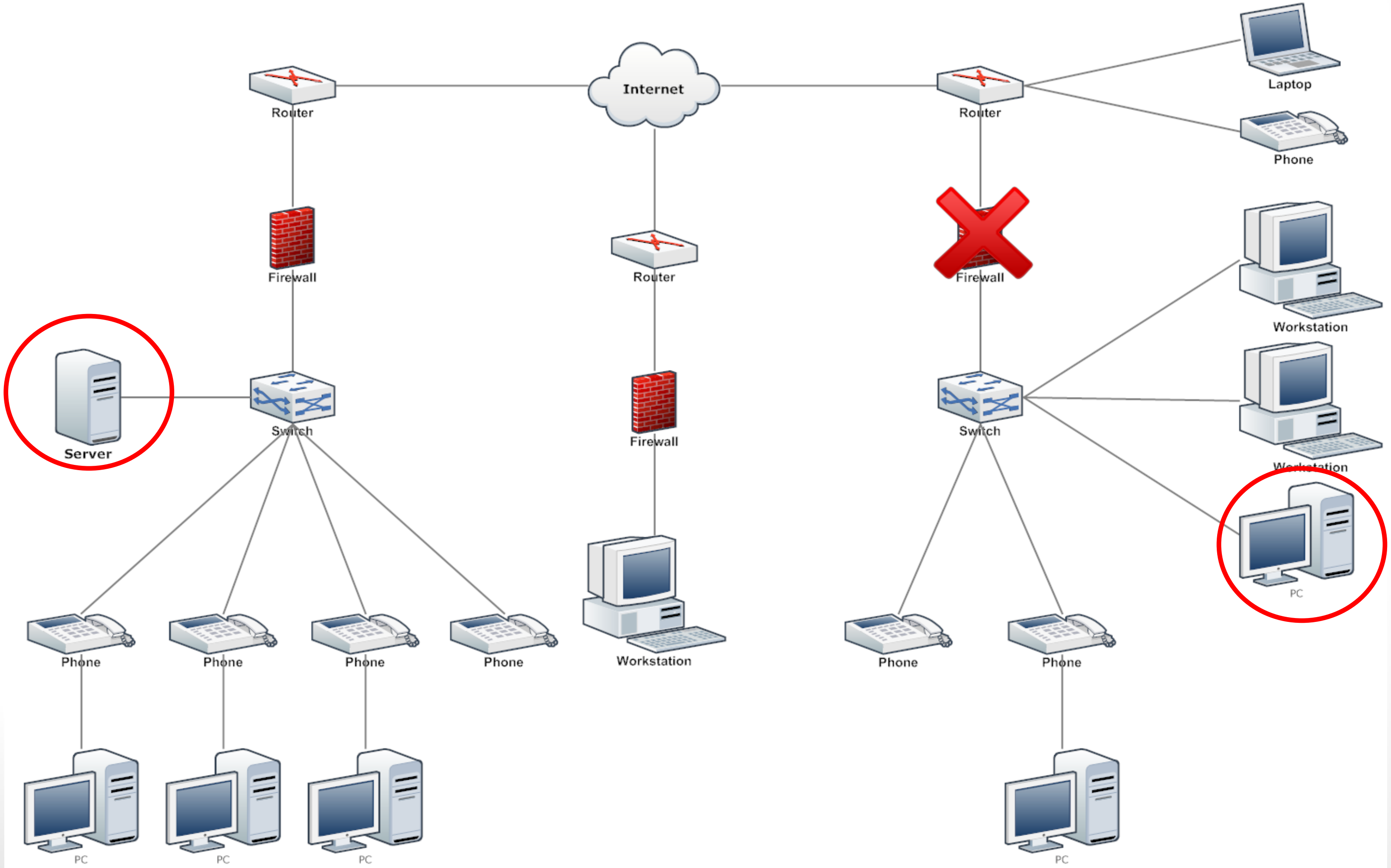
Check Network Connectivity (ping Command)



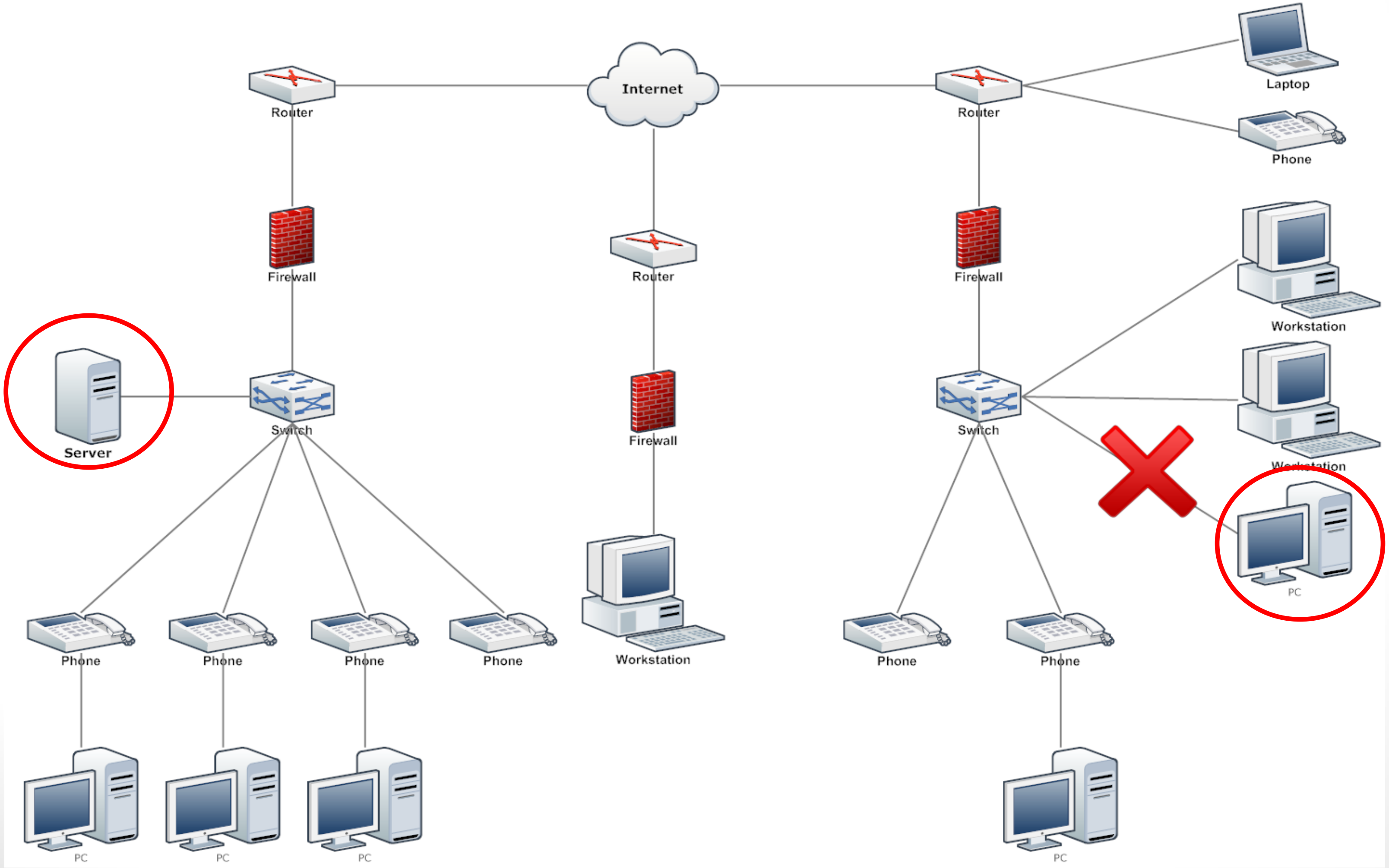
Check Network Connectivity (ping Command)



Check Network Connectivity (ping Command)



Check Network Connectivity (ping Command)



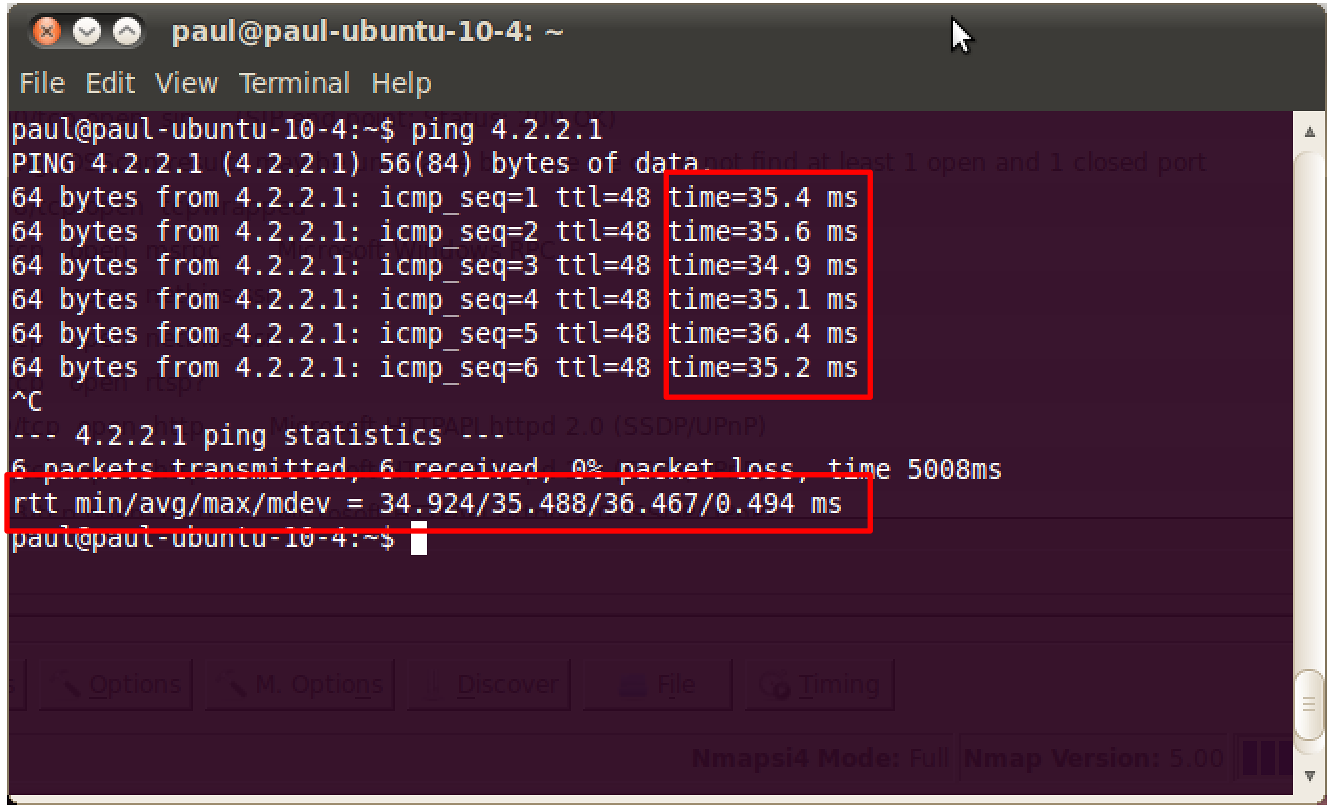
Check Network Connectivity (ping Command)

$ ping <remote machine Address>

This command is used to check connectivity to the remote machine

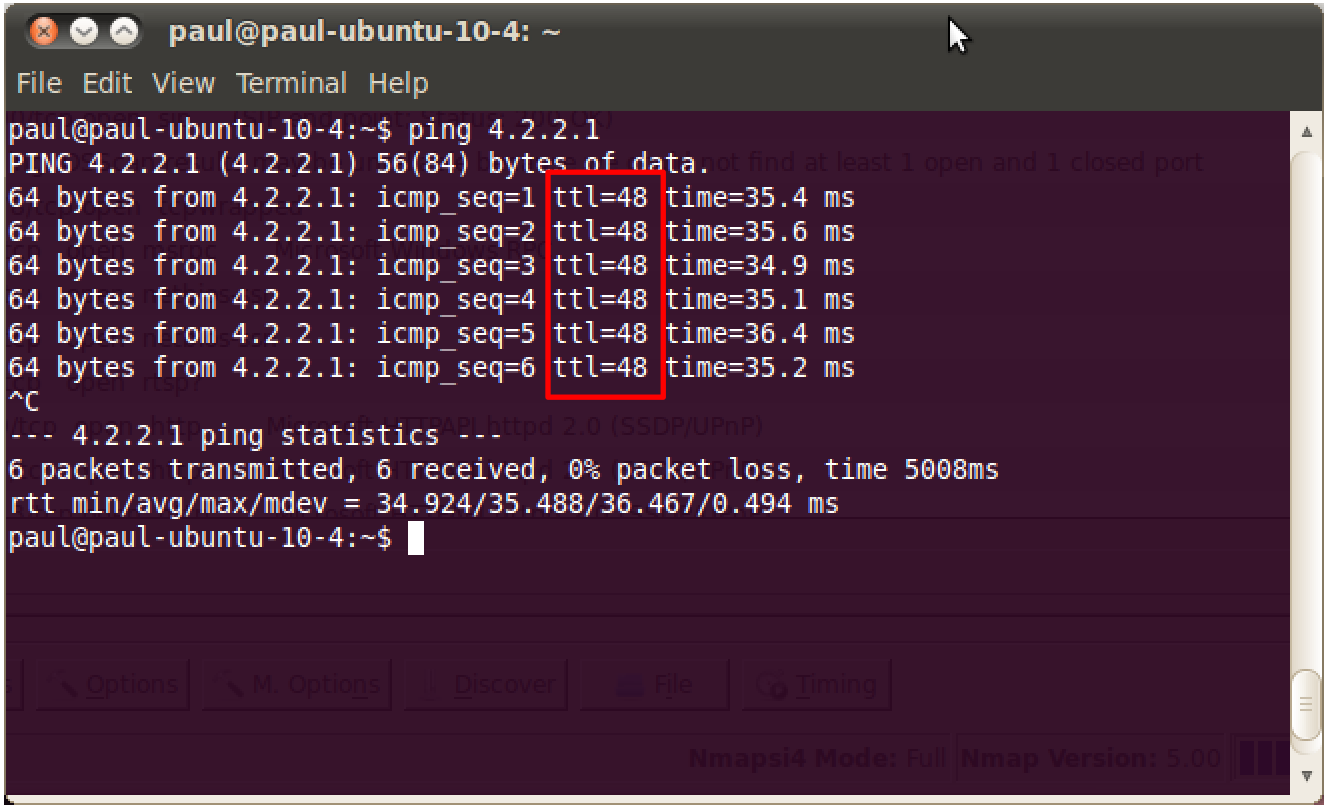
- When you ping a destination,
  - A message ***ICMP_ECHO_REQUEST*** is sent to this destination
  - And a reply ***ICMP_ECHO_RESPONSE*** should be received from the remote destination
- This continues every N second period for a number of times, or until it is interrupted with a Ctrl-C
- Ping can also be used if you want to check the round trip delay to the destination
- You can specify destination by the IP address or by the Domain name
- If you have a network problem, check the following
  - Ping the gateway to make sure it is accessible
  - Ping the DNS Server
  - Ping the destination by name and/or by address

Check Network Connectivity (ping Command)



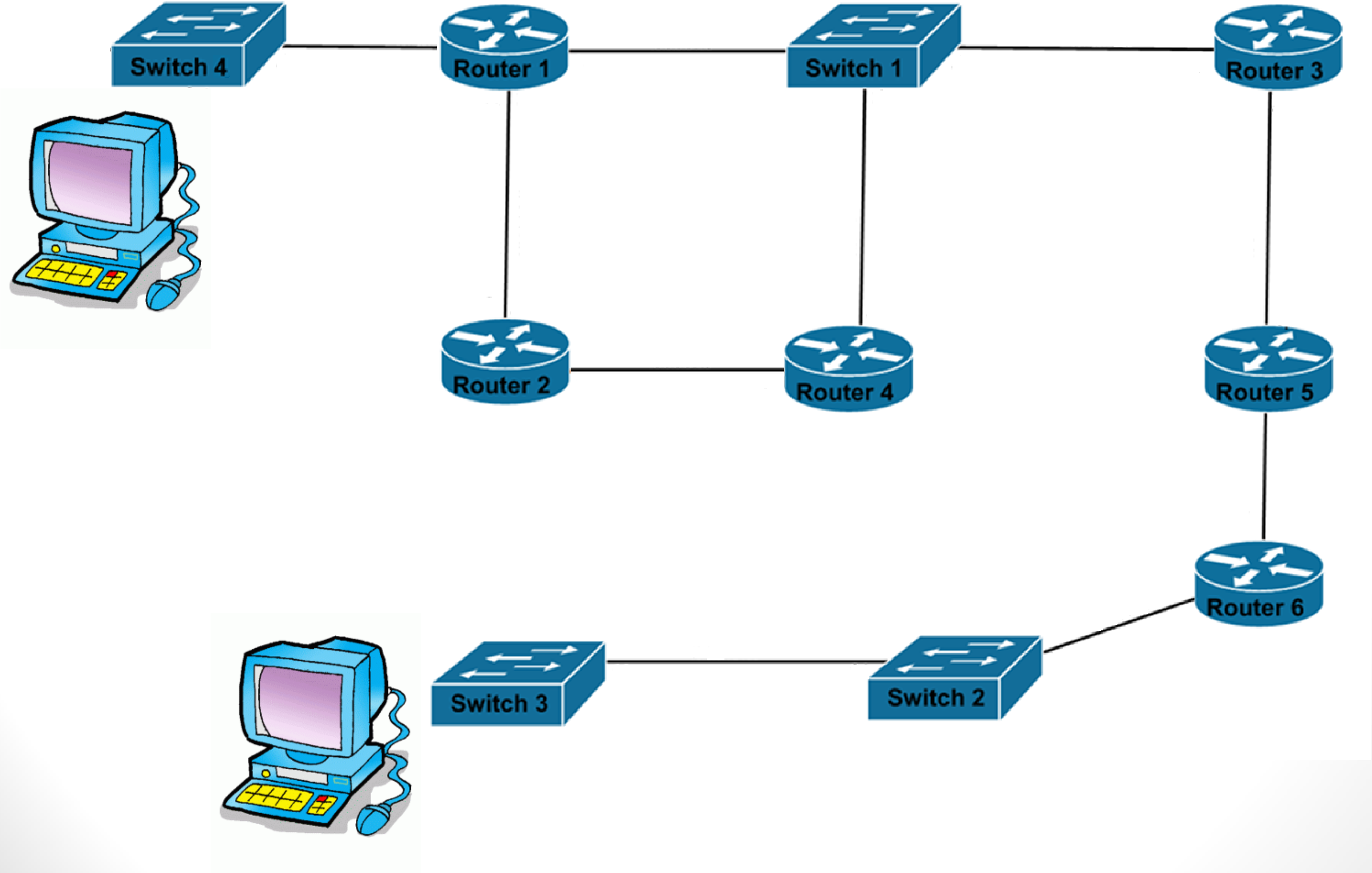
```
paul@paul-ubuntu-10-4:~$ ping 4.2.2.1
PING 4.2.2.1 (4.2.2.1) 56(84) bytes of data.
64 bytes from 4.2.2.1: icmp_seq=1 ttl=48 time=35.4 ms
64 bytes from 4.2.2.1: icmp_seq=2 ttl=48 time=35.6 ms
64 bytes from 4.2.2.1: icmp_seq=3 ttl=48 time=34.9 ms
64 bytes from 4.2.2.1: icmp_seq=4 ttl=48 time=35.1 ms
64 bytes from 4.2.2.1: icmp_seq=5 ttl=48 time=36.4 ms
64 bytes from 4.2.2.1: icmp_seq=6 ttl=48 time=35.2 ms
^C
--- 4.2.2.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 34.924/35.488/36.467/0.494 ms
paul@paul-ubuntu-10-4:~$ 
```

Check Network Connectivity (ping Command)

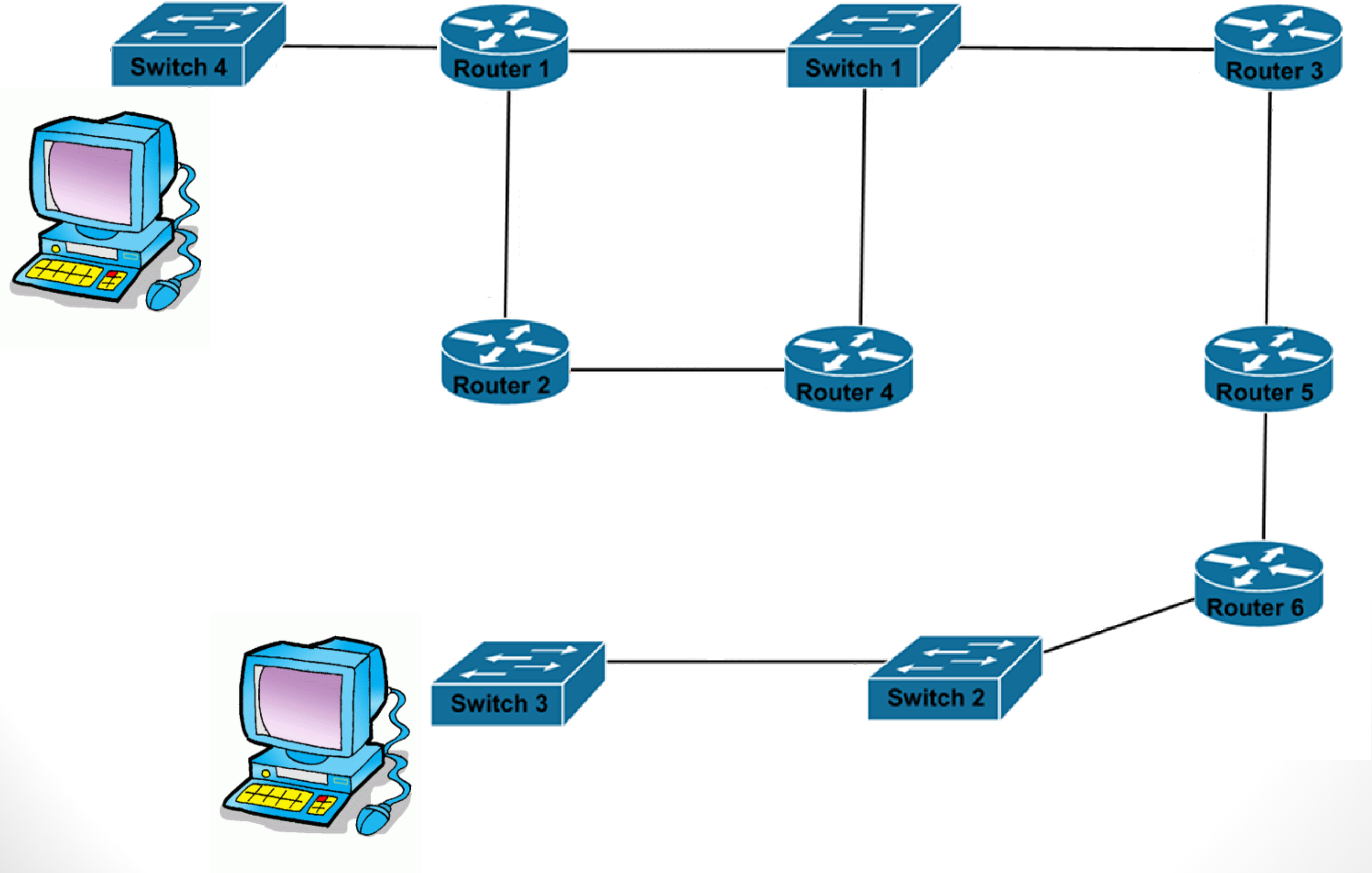


```
paul@paul-ubuntu-10-4:~$ ping 4.2.2.1
PING 4.2.2.1 (4.2.2.1) 56(84) bytes of data.
64 bytes from 4.2.2.1: icmp_seq=1 ttl=48 time=35.4 ms
64 bytes from 4.2.2.1: icmp_seq=2 ttl=48 time=35.6 ms
64 bytes from 4.2.2.1: icmp_seq=3 ttl=48 time=34.9 ms
64 bytes from 4.2.2.1: icmp_seq=4 ttl=48 time=35.1 ms
64 bytes from 4.2.2.1: icmp_seq=5 ttl=48 time=36.4 ms
64 bytes from 4.2.2.1: icmp_seq=6 ttl=48 time=35.2 ms
^C
--- 4.2.2.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 34.924/35.488/36.467/0.494 ms
paul@paul-ubuntu-10-4:~$ 
```

Time To Live (TTL)



Tracing the Route



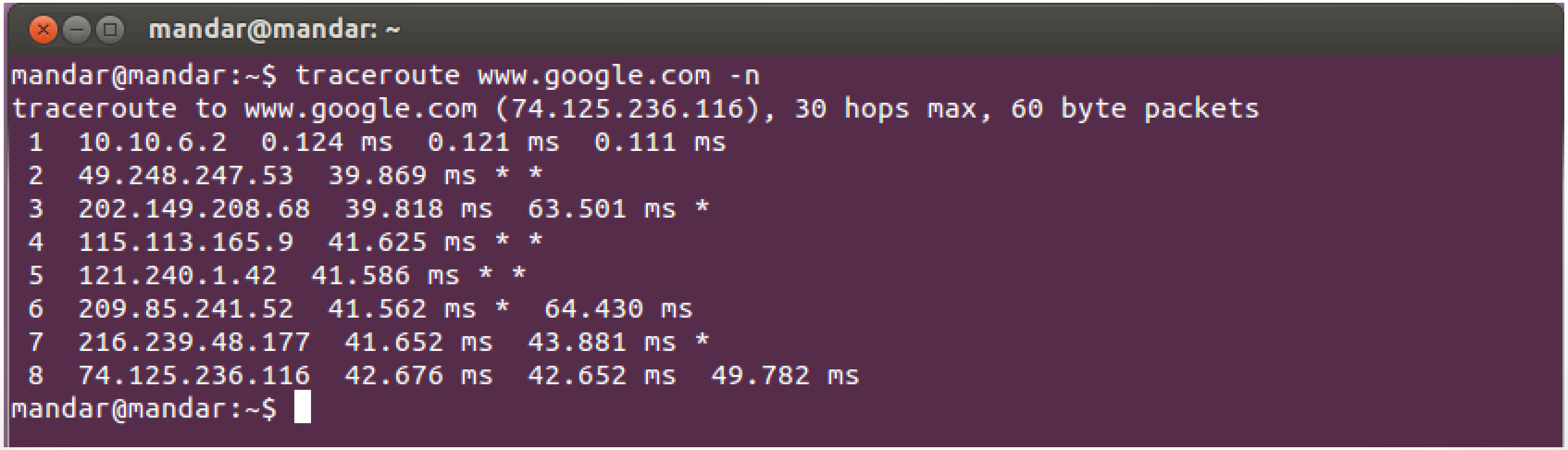
Tracing the Route (traceroute Command)

$ traceroute <destination Address>

- Same as ping, but this time, you get the whole route of the packet

  $ traceroute www.google.com

- Sometimes, certain nodes in the route remain hidden



mandar@mandar: ~

```
mandar@mandar:~$ traceroute www.google.com -n
traceroute to www.google.com (74.125.236.116), 30 hops max, 60 byte packets
 1  10.10.6.2  0.124 ms  0.121 ms  0.111 ms
 2  49.248.247.53  39.869 ms * *
 3  202.149.208.68  39.818 ms  63.501 ms *
 4  115.113.165.9  41.625 ms * *
 5  121.240.1.42  41.586 ms * *
 6  209.85.241.52  41.562 ms *  64.430 ms
 7  216.239.48.177  41.652 ms  43.881 ms *
 8  74.125.236.116  42.676 ms  42.652 ms  49.782 ms
mandar@mandar:~$ 
```

Collecting Network Statistics (netstat Command)

$ netstat [Options]

This command displays various network related information such as network connections, routing tables, interface statistics, etc.,

- To display the routing table information

  ```
  $ netstat -r
  ```

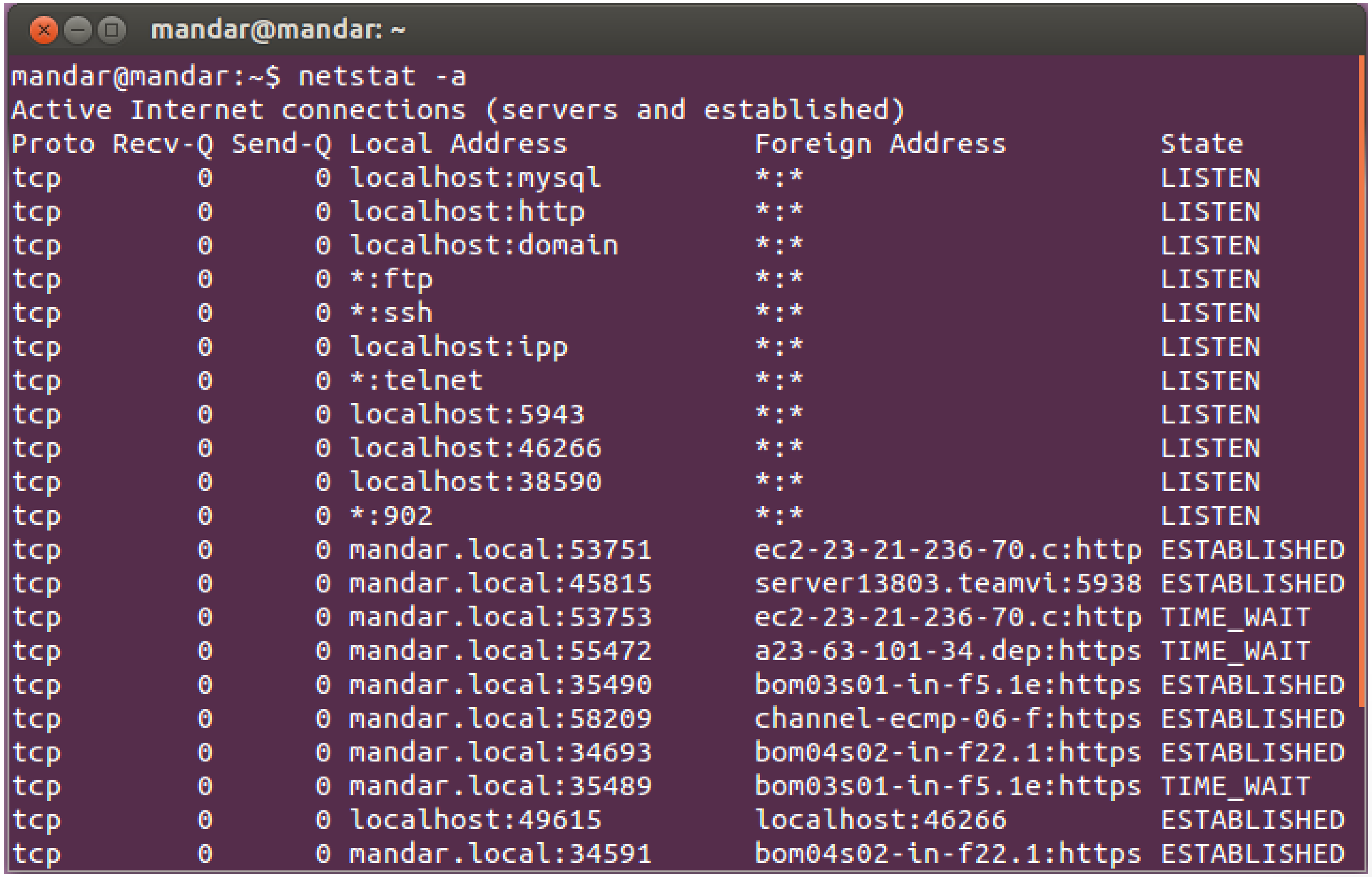
- To list network interfaces on the machine

  ```
  $ netstat -i
  $ netstat -ie    (output similar to ifconfig)
  ```

- To list statistics on sockets of different protocols

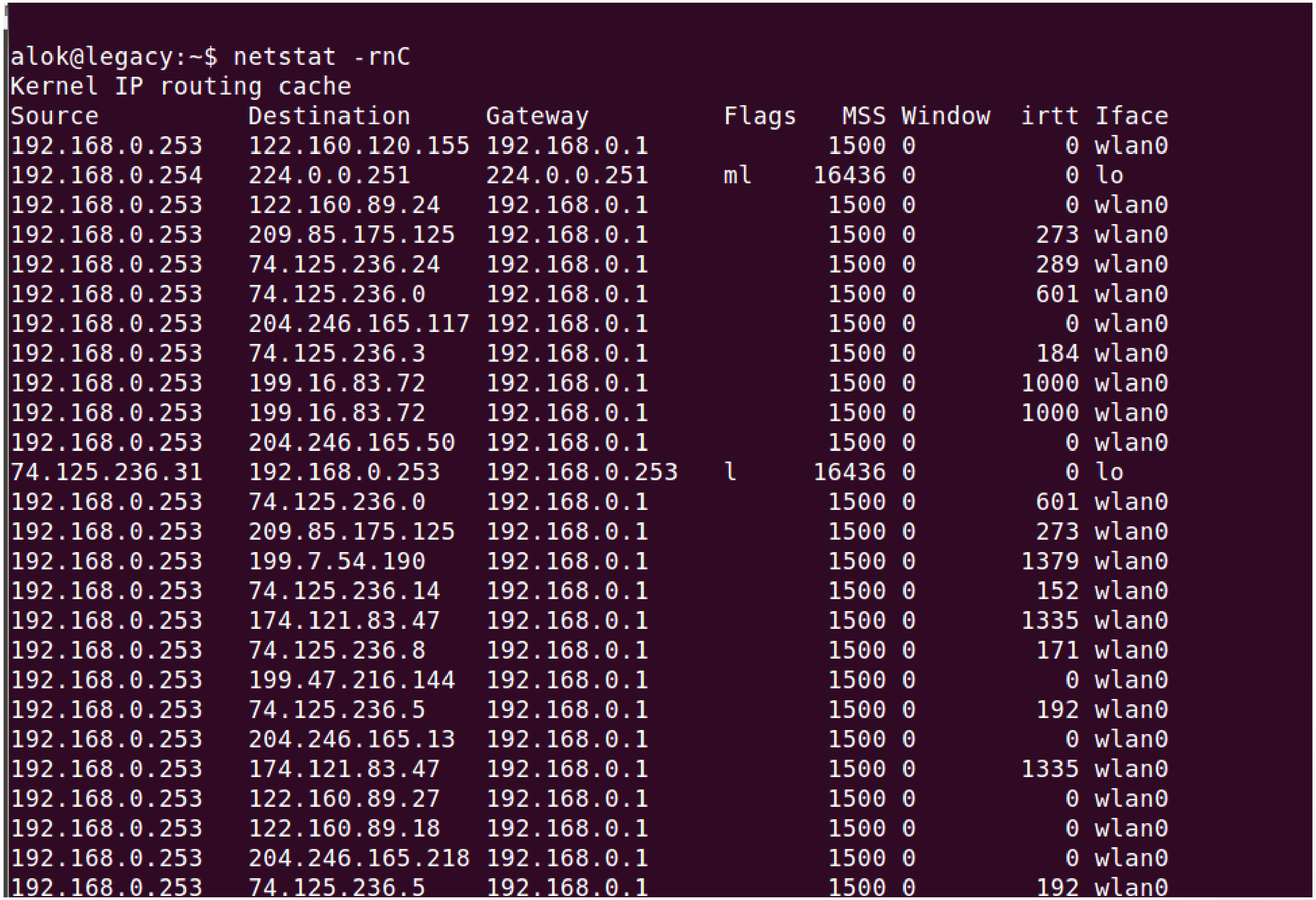
  ```
  $ netstat -s
  $ netstat -st     (only for TCP Protocol)
  $ netstat -su    (only for UDP Protocol)
  ```

Collecting Network Statistics (netstat Command)



```
mandar@mandar:~$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:mysql         *:*                     LISTEN
tcp        0      0 localhost:http          *:*                     LISTEN
tcp        0      0 localhost:domain        *:*                     LISTEN
tcp        0      0 *:ftp                   *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 *:telnet                *:*                     LISTEN
tcp        0      0 localhost:5943          *:*                     LISTEN
tcp        0      0 localhost:46266         *:*                     LISTEN
tcp        0      0 localhost:38590         *:*                     LISTEN
tcp        0      0 *:902                   *:*                     LISTEN
tcp        0      0 mandar.local:53751      ec2-23-21-236-70.c:http ESTABLISHED
tcp        0      0 mandar.local:45815      server13803.teamvi:5938 ESTABLISHED
tcp        0      0 mandar.local:53753      ec2-23-21-236-70.c:http TIME_WAIT
tcp        0      0 mandar.local:55472      a23-63-101-34.dep:https TIME_WAIT
tcp        0      0 mandar.local:35490      bom03s01-in-f5.1e:https ESTABLISHED
tcp        0      0 mandar.local:58209      channel-ecmp-06-f:https ESTABLISHED
tcp        0      0 mandar.local:34693      bom04s02-in-f22.1:https ESTABLISHED
tcp        0      0 mandar.local:35489      bom03s01-in-f5.1e:https TIME_WAIT
tcp        0      0 localhost:49615         localhost:46266         ESTABLISHED
tcp        0      0 mandar.local:34591      bom04s02-in-f22.1:https ESTABLISHED
```

Collecting Network Statistics (netstat Command)



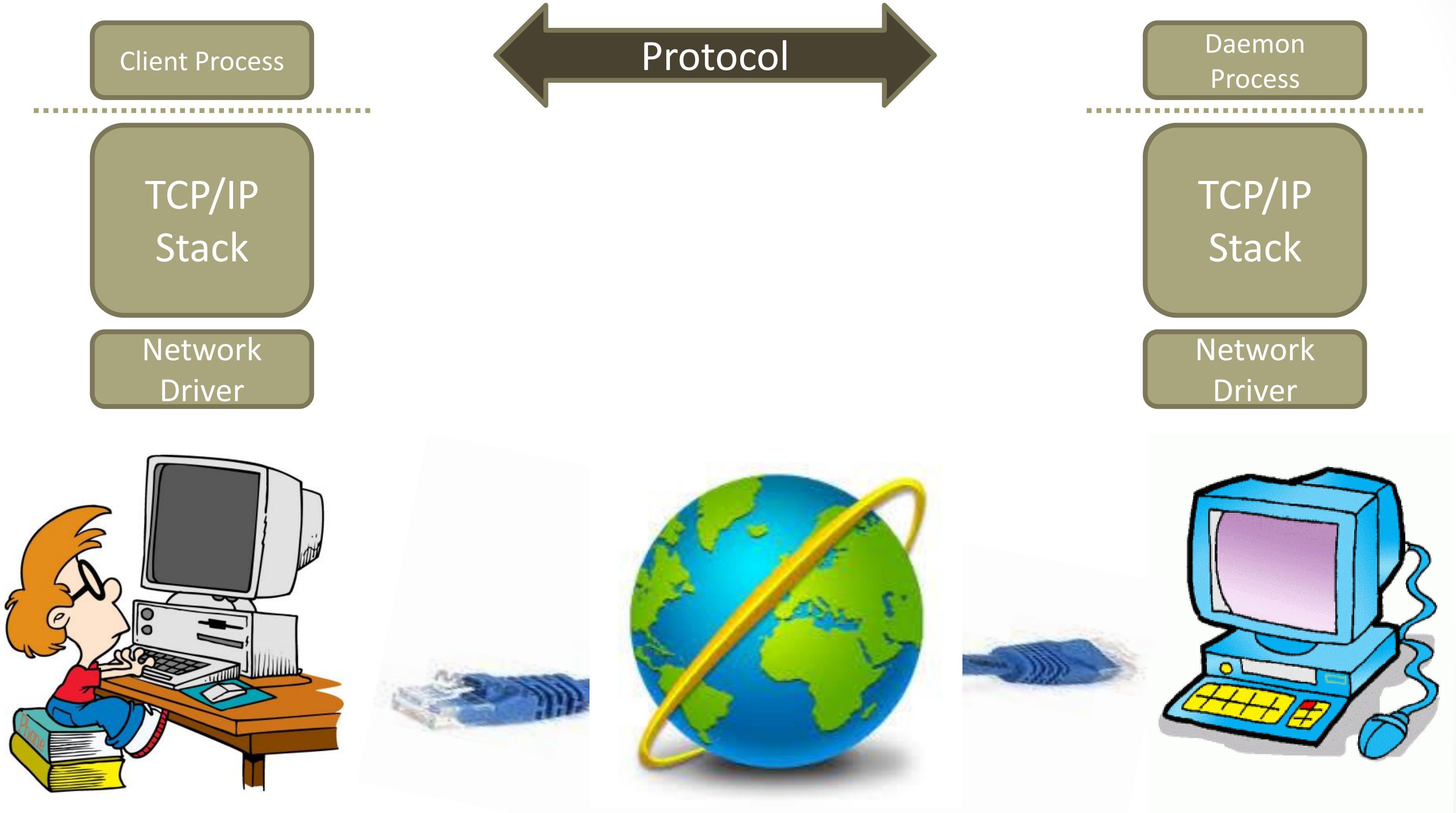
```
alok@legacy:~$ netstat -rnC
Kernel IP routing cache
Source          Destination     Gateway         Flags   MSS Window  irtt Iface
192.168.0.253   122.160.120.155 192.168.0.1            1500 0          0 wlan0
192.168.0.254   224.0.0.251     224.0.0.251     ml    16436 0          0 lo
192.168.0.253   122.160.89.24   192.168.0.1            1500 0          0 wlan0
192.168.0.253   209.85.175.125  192.168.0.1            1500 0        273 wlan0
192.168.0.253   74.125.236.24   192.168.0.1            1500 0        289 wlan0
192.168.0.253   74.125.236.0    192.168.0.1            1500 0        601 wlan0
192.168.0.253   204.246.165.117 192.168.0.1            1500 0          0 wlan0
192.168.0.253   74.125.236.3    192.168.0.1            1500 0        184 wlan0
192.168.0.253   199.16.83.72    192.168.0.1            1500 0       1000 wlan0
192.168.0.253   199.16.83.72    192.168.0.1            1500 0       1000 wlan0
192.168.0.253   204.246.165.50  192.168.0.1            1500 0          0 wlan0
74.125.236.31   192.168.0.253   192.168.0.253   l     16436 0          0 lo
192.168.0.253   74.125.236.0    192.168.0.1            1500 0        601 wlan0
192.168.0.253   209.85.175.125  192.168.0.1            1500 0        273 wlan0
192.168.0.253   199.7.54.190    192.168.0.1            1500 0       1379 wlan0
192.168.0.253   74.125.236.14   192.168.0.1            1500 0        152 wlan0
192.168.0.253   174.121.83.47   192.168.0.1            1500 0       1335 wlan0
192.168.0.253   74.125.236.8    192.168.0.1            1500 0        171 wlan0
192.168.0.253   199.47.216.144  192.168.0.1            1500 0          0 wlan0
192.168.0.253   74.125.236.5    192.168.0.1            1500 0        192 wlan0
192.168.0.253   204.246.165.13  192.168.0.1            1500 0          0 wlan0
192.168.0.253   174.121.83.47   192.168.0.1            1500 0       1335 wlan0
192.168.0.253   122.160.89.27   192.168.0.1            1500 0          0 wlan0
192.168.0.253   122.160.89.18   192.168.0.1            1500 0          0 wlan0
192.168.0.253   204.246.165.218 192.168.0.1            1500 0          0 wlan0
192.168.0.253   74.125.236.5    192.168.0.1            1500 0        192 wlan0
```

Network Applications

General Structure

- The different network applications described in this lecture share the following structure
  - The user accesses a remote machine for different purposes
    - Copy files to/from the remote machine
    - Access a terminal in the remote machine
    - Access the GUI of the remote machine
  - In all cases, the user runs a client application on his local machine
  - The remote machine will be running a server application
  - The server application will be running on a Daemon process waiting for a connection from the client side
  - Both the Client and the Daemon are user plane applications that communicate with the TCP/IP stack residing in the kernel

General Structure



Client Process

TCP/IP Stack

Network Driver

Protocol

Daemon Process

TCP/IP Stack

Network Driver

Remote Access of a Machine (telnet Protocol)

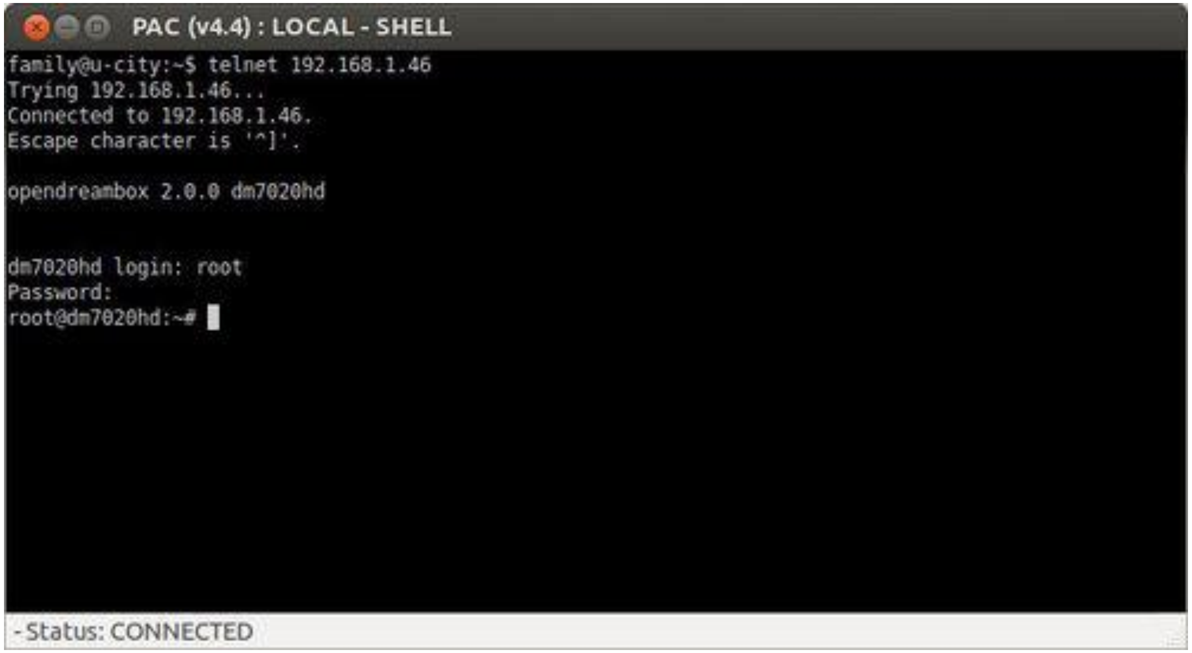
$ telnet <destination Address>

- The telnet is a protocol to enable the user at the client side to access a remote machine by opening a terminal on it

```
$ telnet 192.168.101.27
$ telnet bob@192.168.101.27
```

- The User will need to enter his login info
- A server application must be running on the destination machine to accept client connections
- Once connection is established, a **tty terminal** will be established on the remote machine that is controlled via the **Telnet session**
- Anything the user types is sent to the remote machine as if the user is using it

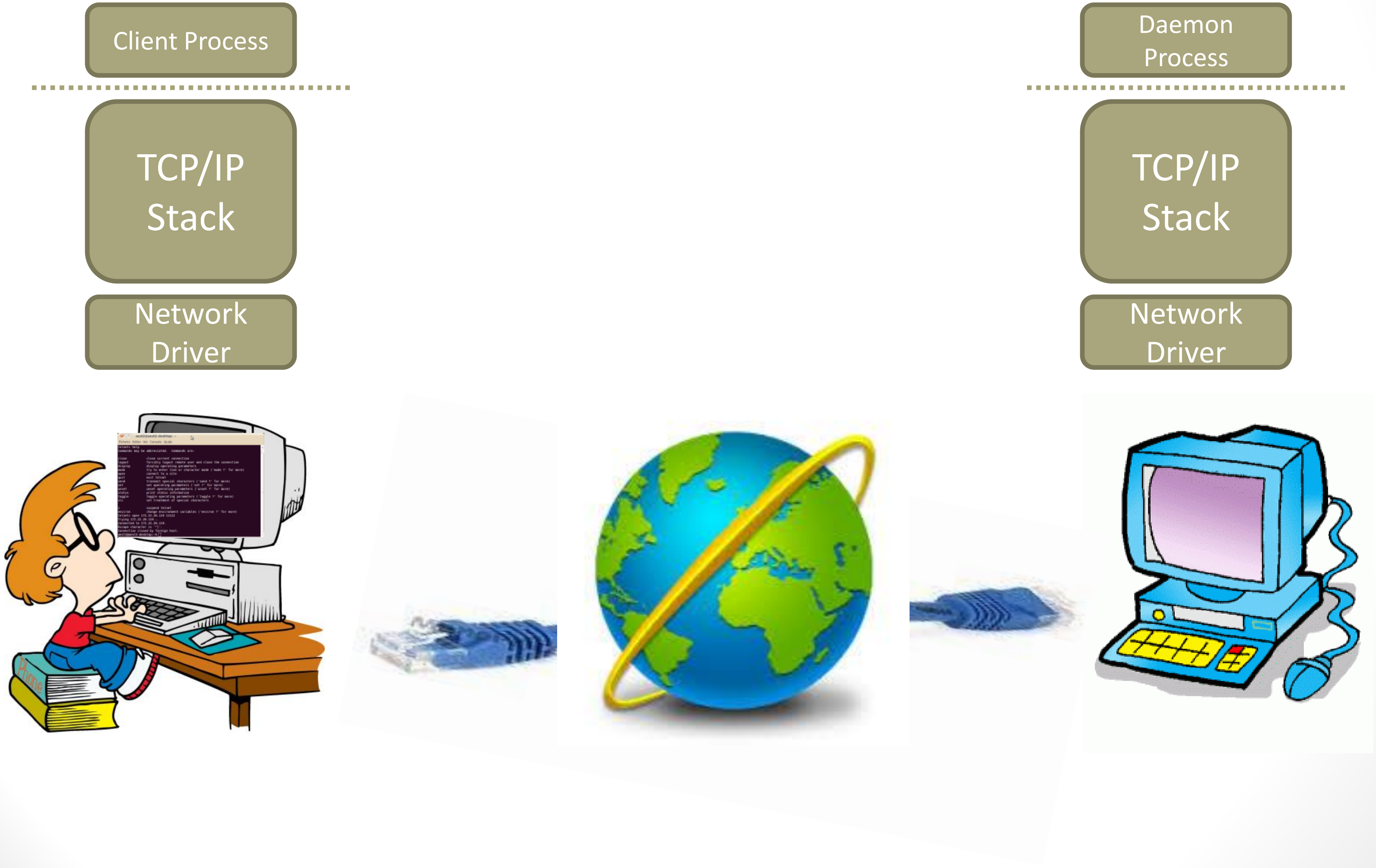
Remote Access of a Machine (telnet Protocol)



```
family@u-city:~$ telnet 192.168.1.46
Trying 192.168.1.46...
Connected to 192.168.1.46.
Escape character is '^]'.

opendreambox 2.0.0 dm7020hd


dm7020hd login: root
Password:
root@dm7020hd:~#
```

- After log-in is complete successfully, the user can perform any action that is done on normal terminal

Telnet



Client Process

TCP/IP Stack

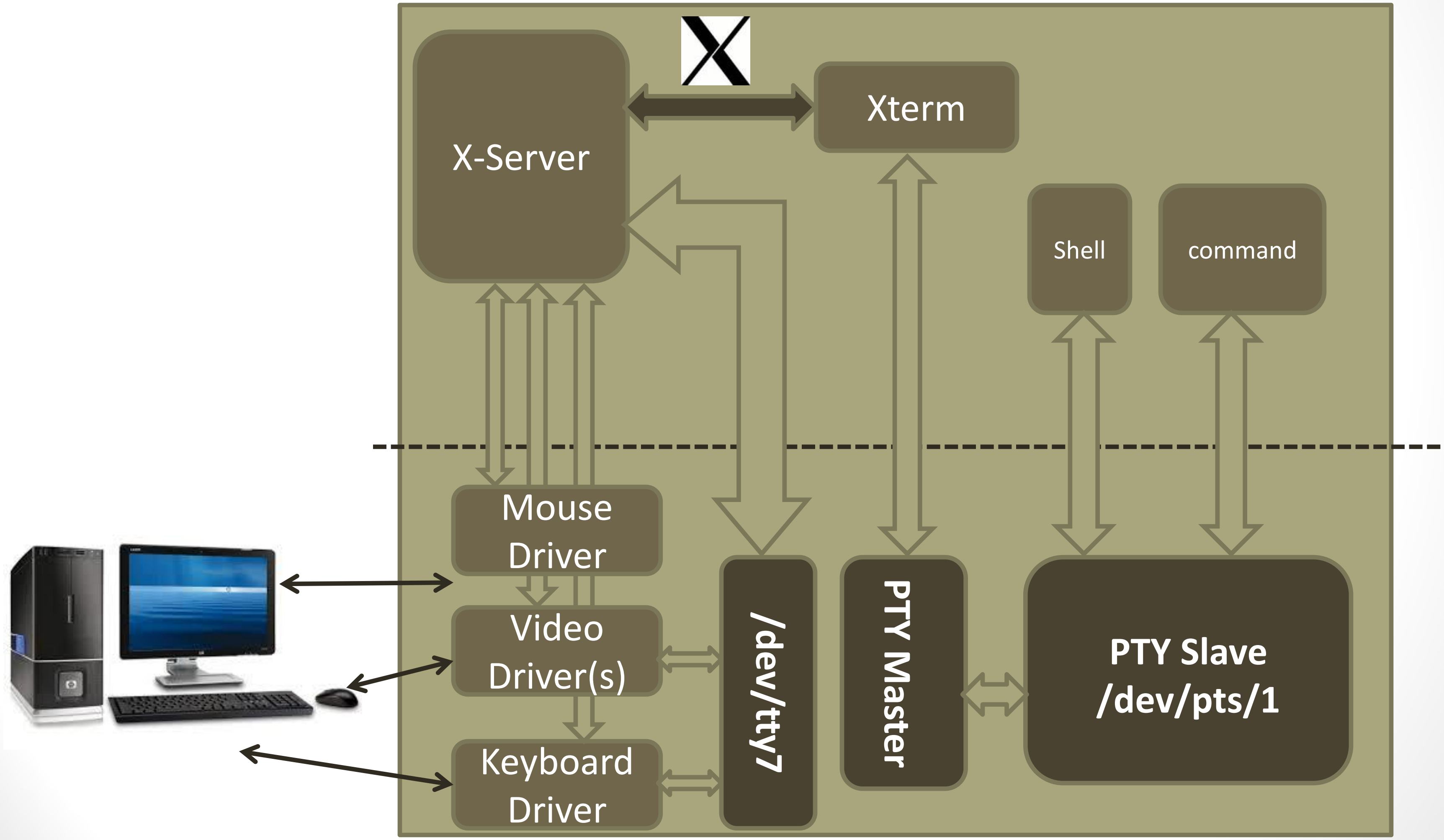
Network Driver

Daemon Process

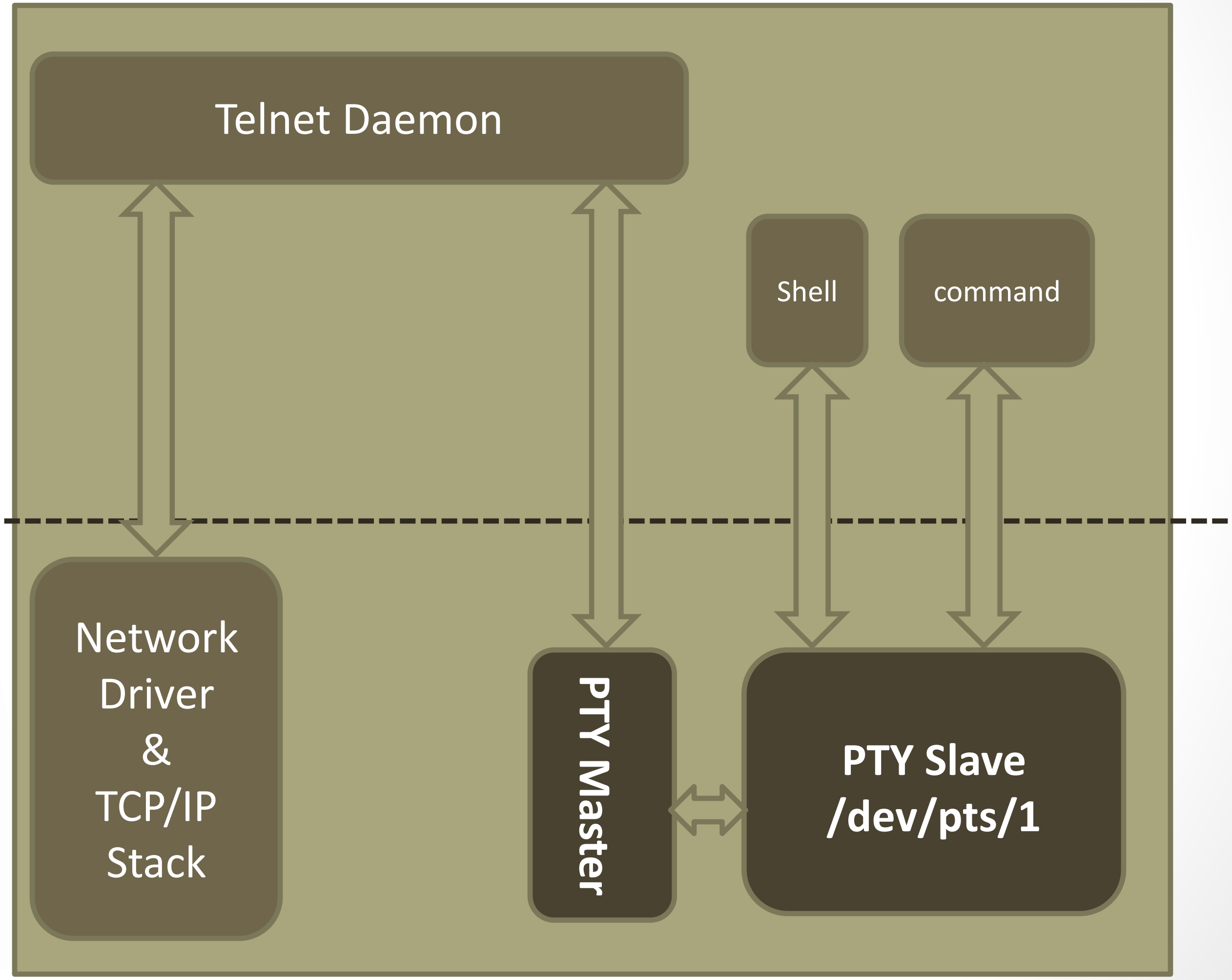
TCP/IP Stack

Network Driver

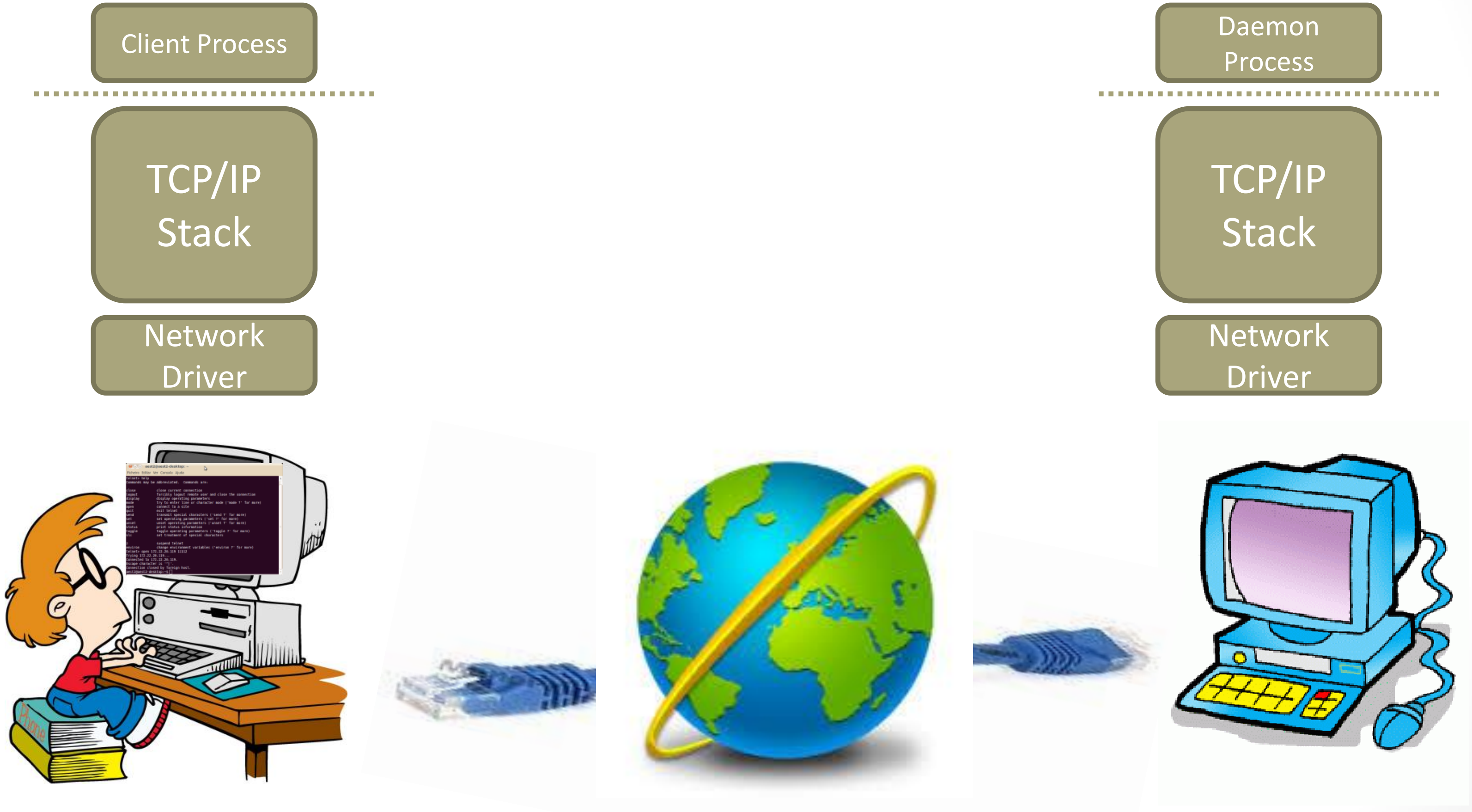
Logging in Using a Telnet Session…



Logging in Using a Telnet Session…



Transporting files (ftp protocol)



Client Process

TCP/IP Stack

Network Driver

Daemon Process

TCP/IP Stack

Network Driver

Transporting files (ftp protocol)

$ ftp <Remote Machine Addrress>

- This protocol enables the client to move files from/to the remote machine

```
$ ftp 192.168.101.12
$ ftp bob@192.168.101.12
```

- Sometimes an FTP server can allow anonymous login. In this case use,

Username: anonymous

Password: your email

- Once you login, you will be able to get/put files

```
$ get myfile.txt
$ mget *.exe
$ put my_picture.png
$ mput *.jpg
```

- To exit

```
$ bye
```

Security Concern

- Both Telnet and FTP do not use a secured connection
- Information travel between the local machine and the remote one in clear text
- This can be a big problem. A hacker can,
  - Listen to the message being sent
  - Modify the messages being sent
  - Send its own messages with false identity
- Sometimes, this is not a problem
  - Connecting to a machine in a secured environment
  - Connecting to an embedded platform within a isolated network
  - You don’t care about security
- Other times, this is not acceptable

Secure login to remote machines (ssh protocol)

$ ssh <destination Address>

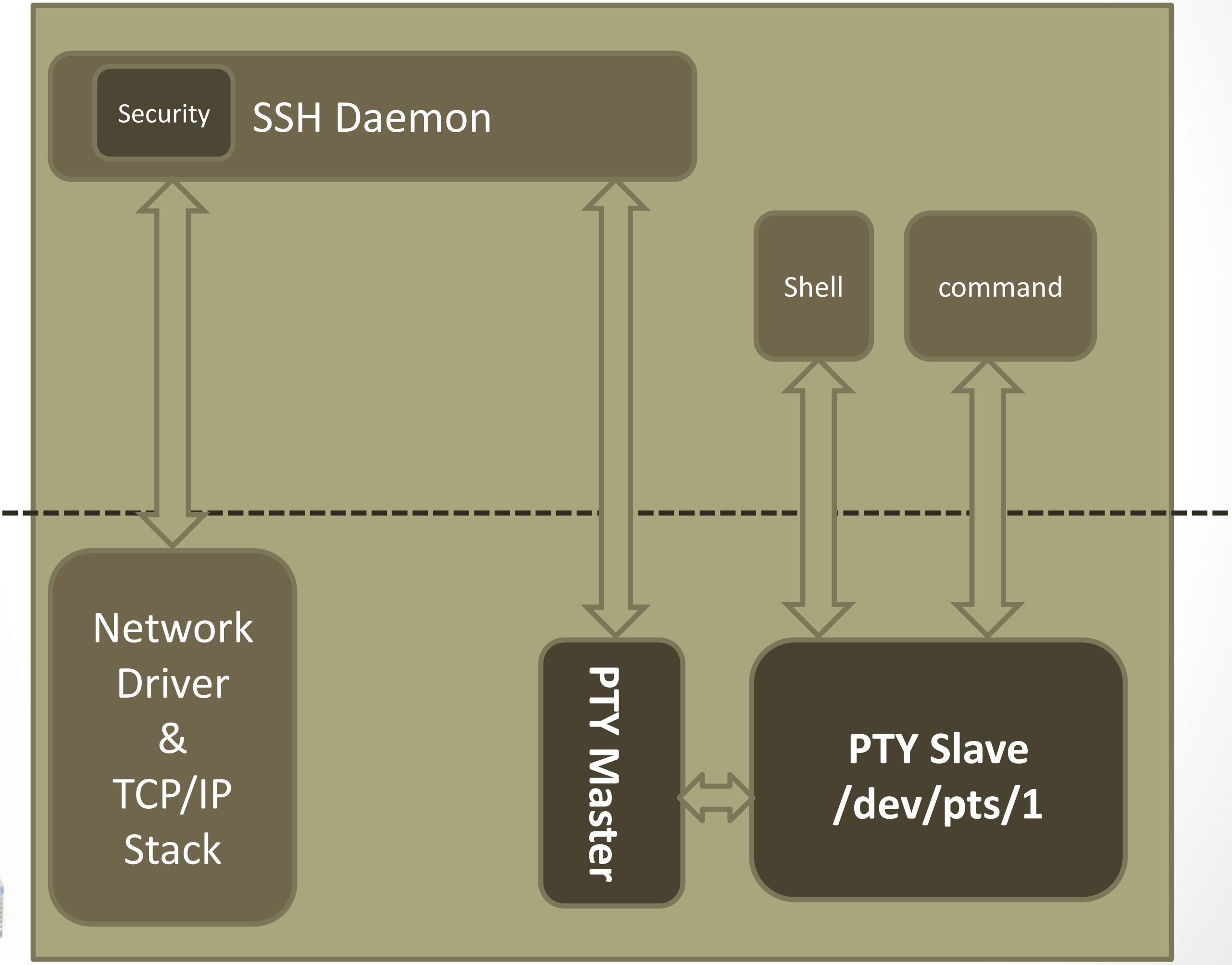
- This is similar to the telnet protocol except for that the connection will be secured (traffic will be encrypted)
- To login securely to a machine,

  $ ssh 192.168.101.100

  $ ssh bob@ 192.168.101.100

  $ ssh bob@tom-machine
- In the first time to connect to this machine, some confirmation will be requested to install the required keys for encryption
- Once connection is established, a **tty terminal** will be established on the remote machine that is controlled via the **SSH session**

Logging in Using a SSH Session…



Secure File Copy (scp Command)

```
$ scp <local filename> <user>@<remoteServer>:<remote-filename>
$ scp <user>@<remoteServer>:<remote-filename> <local filename>
```

- This command copies files from/to a remote machine
- It uses a secure channel similar to that of SSH
- Usage is similar to the ordinary copy command "***cp***" with the exception:
  - Remote filename is preceded by the remote server name, and optionally the user name
  - A username / password may need to be entered to complete the command

- The scp performs secure copy,

```
$ scp 192.168.101.13:my-doc.pdf  ./my-docs/
$ scp bob@192.168.101.13:my-doc.pdf  ./my-docs/
$ scp ./my-docs/*.pdf   bob@remoteServer:.
$ scp -r ./documents bob@202.11.1.20:.
```

Secure File Transfer (sftp Command)

$ sftp <remote Address>

- This command has a similar usage as the normal ***ftp*** command
- However, it uses an SSH connection to secure the file transfer

```
$ sftp 192.168.1.103
$ sftp bob@192.168.1.103
```

- It has the same interface as ftp
- Note that sftp does not require an ftp daemon on the remote machine since it uses the ssh connection

Downloading file from the Web (wget Command)

$ wget <URL of the file>

- Very useful tool for downloading files from the web from the command line

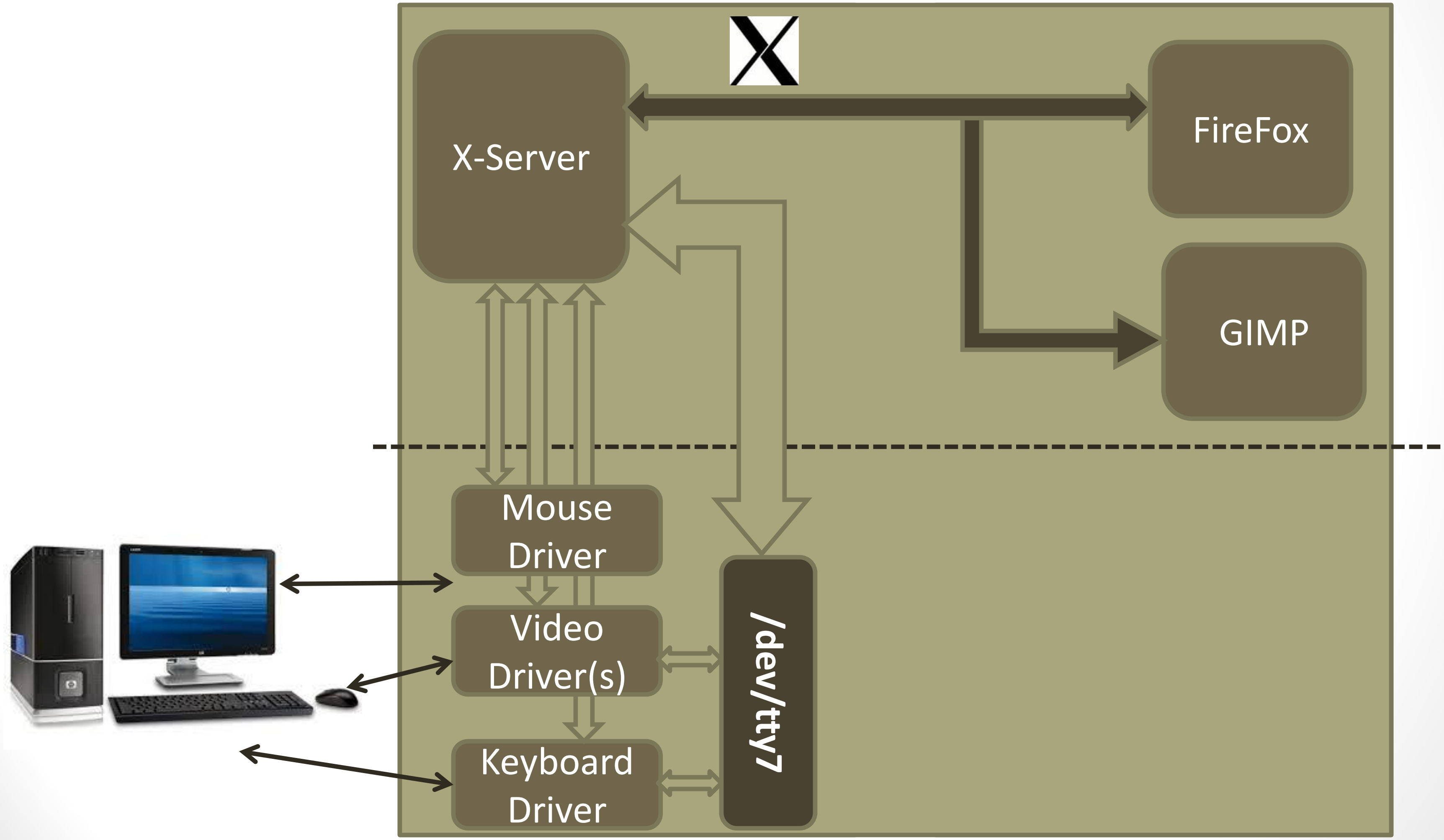
  $ wget http://www.my-web-site.com/file.xml

- Very useful in scripts that perform a download from the web

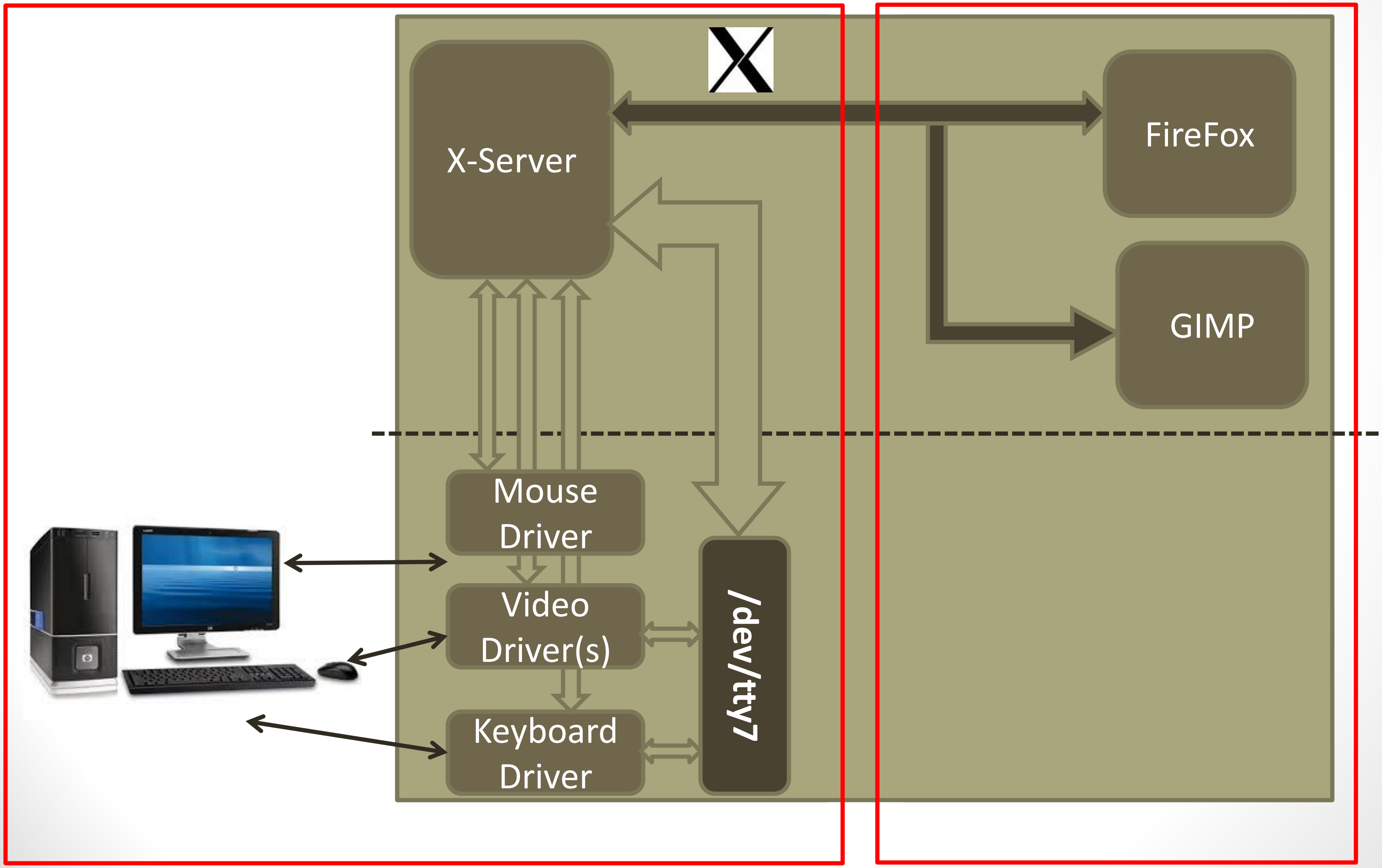
Remote Access the Desktop

- The access methods mentioned so far (Telnet and SSH) provide access to a text terminal in the remote machine
- However, sometimes we need to have remote access to the remote machine GUI
- We will need to access the GUI using our mouse and keyboard
- This can be achieved using two ways,
  - Running the X-server on the local machine, and connecting to the x-clients (applications) on the remote machine
  - Running a VNC Client-Server model

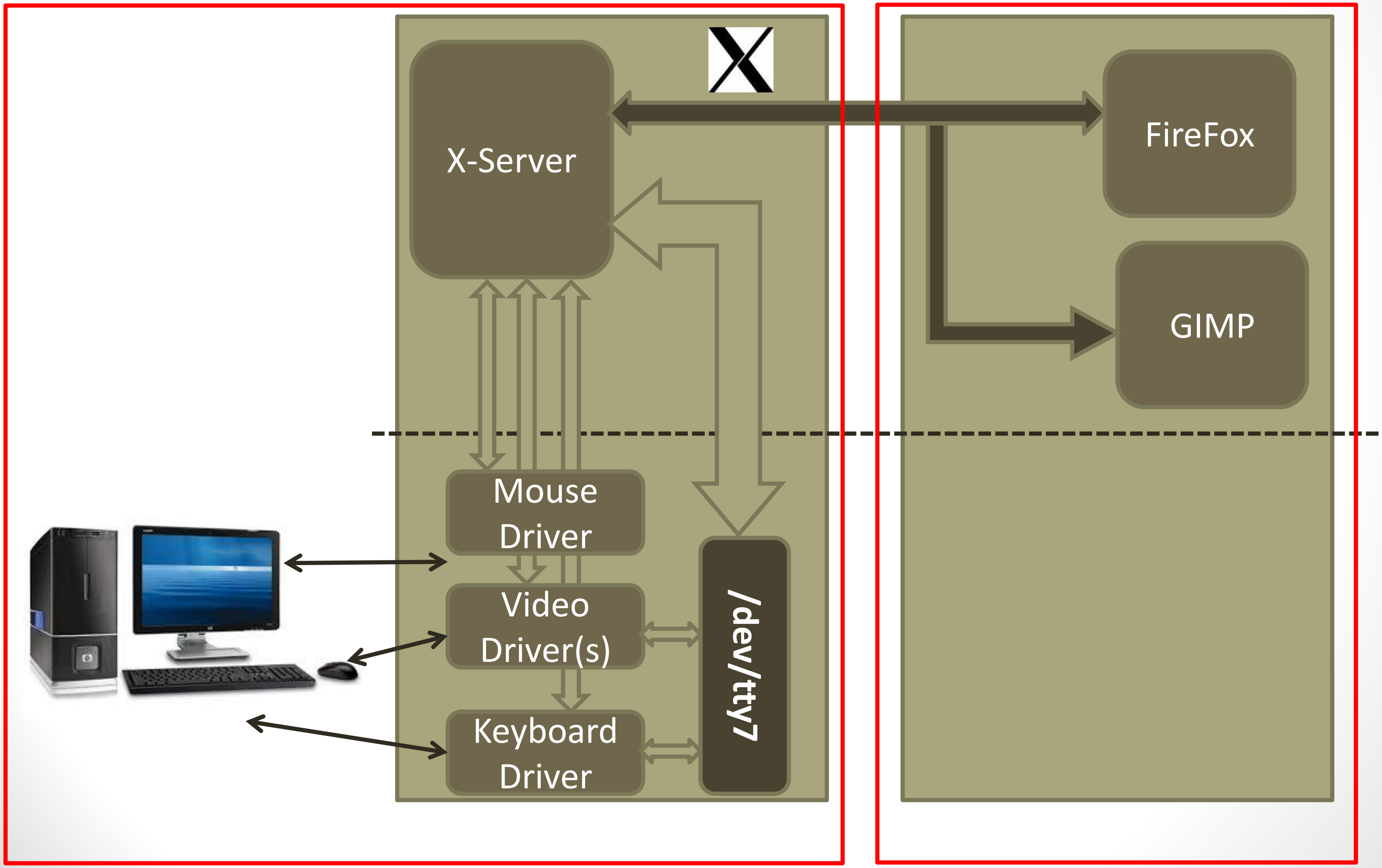
Using X-Server on the Local Machine



Using X-Server on the Local Machine



Using X-Server on the Local Machine



Using X-Server on the Local Machine (ssh –X Command)

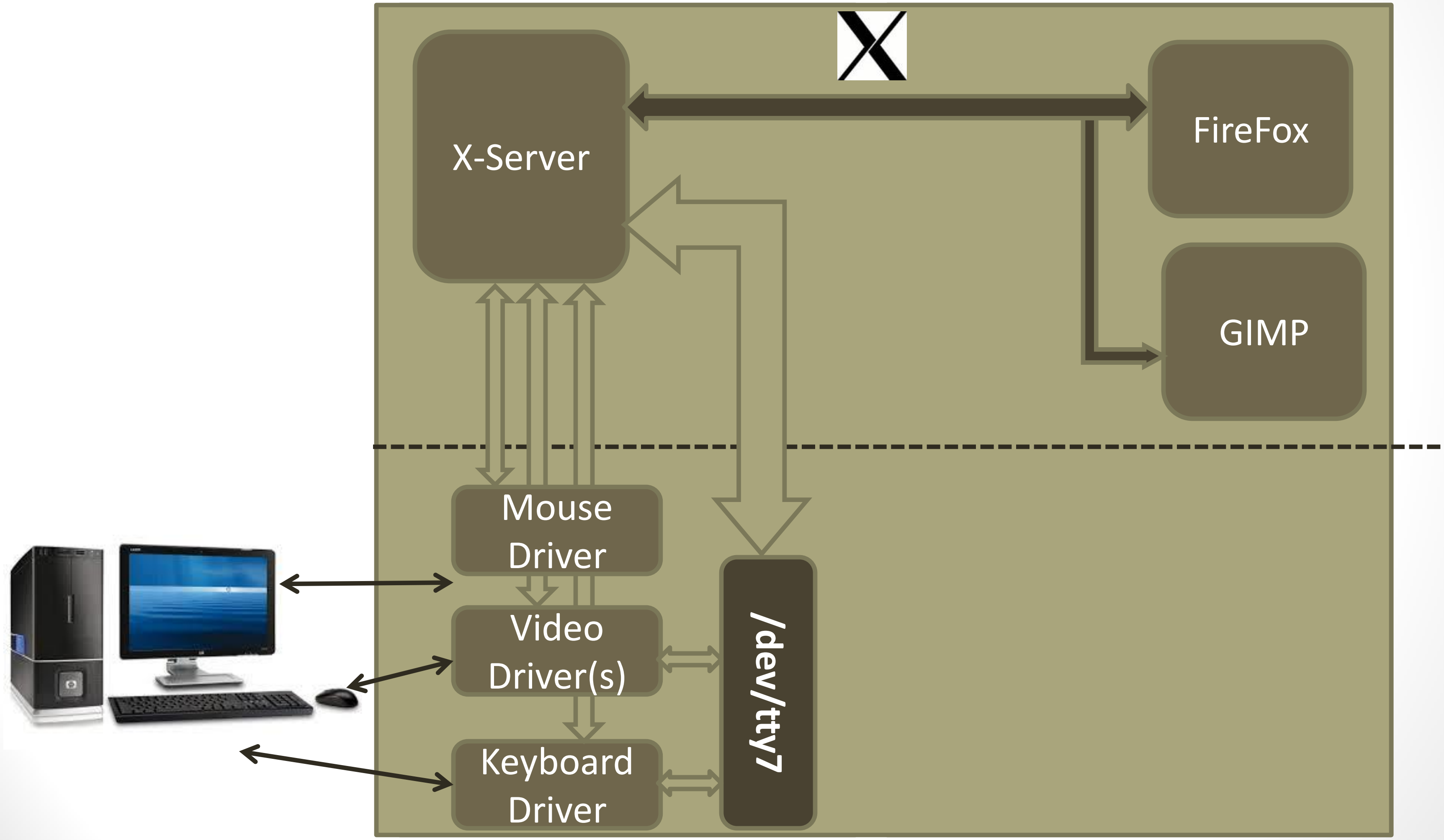
$ ssh -X <remote machine address>

- To establish SSH connection, along with X-Server running on the local machine, and the X-clients (applications) running on the remote machine

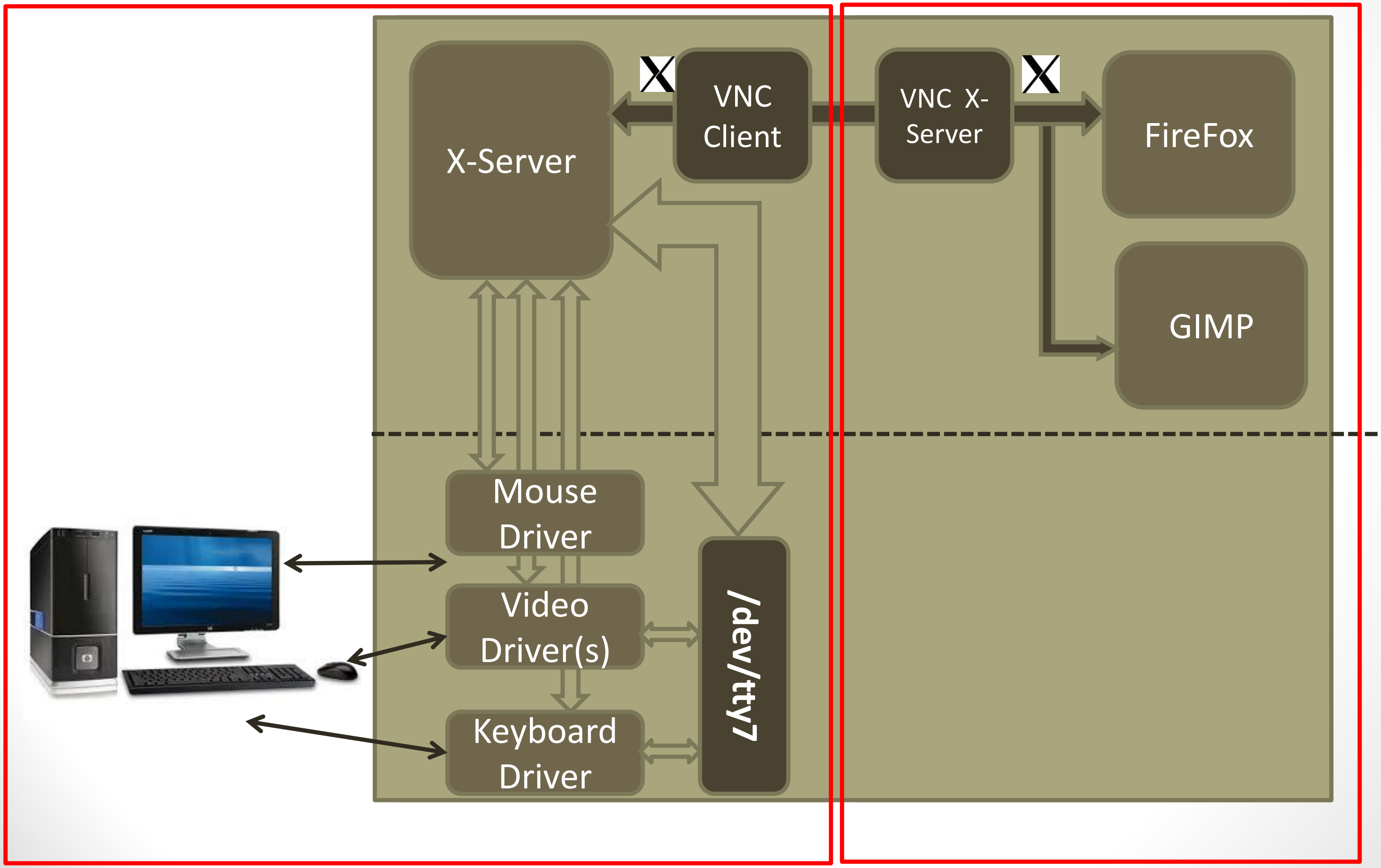
```
$ ssh -X bob@192.168.101.13
```

- This tunnels the X-Protocol massaging between the X-server, and the X-Clients in the SSH connection

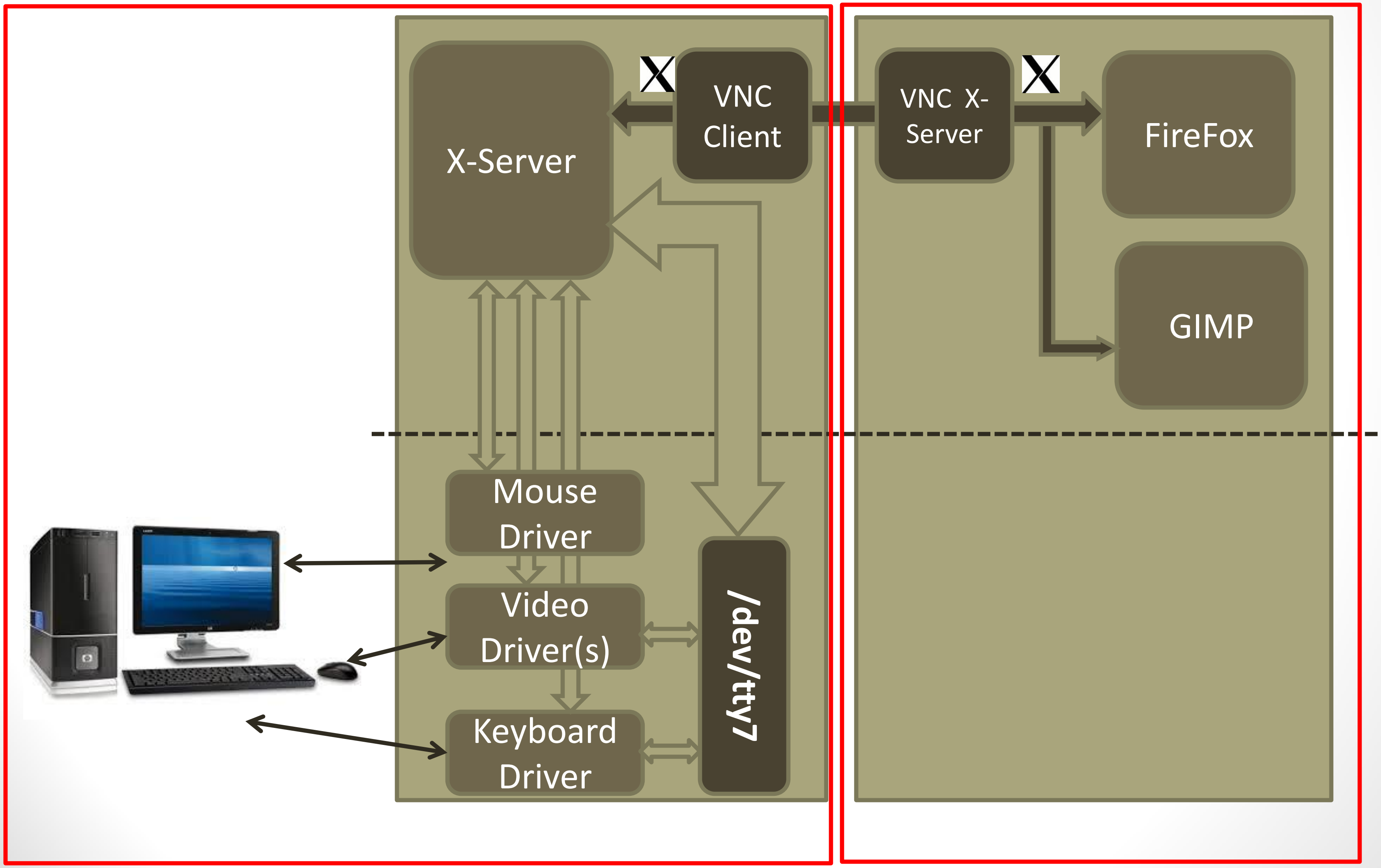
Using VNC


X-Server
FireFox
GIMP
Mouse Driver
Video Driver(s)
Keyboard Driver
/dev/tty7

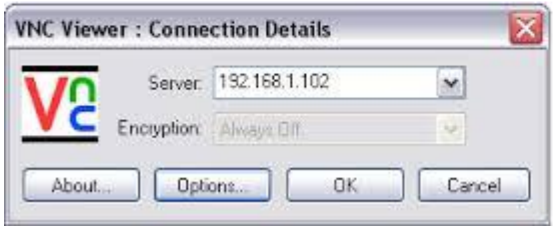
Using VNC


X-Server
X
VNC Client
VNC X-Server
X
FireFox
GIMP
Mouse Driver
Video Driver(s)
Keyboard Driver
/dev/tty7

Using VNC


X-Server
VNC Client
VNC X-Server
FireFox
GIMP
Mouse Driver
Video Driver(s)
Keyboard Driver
/dev/tty7

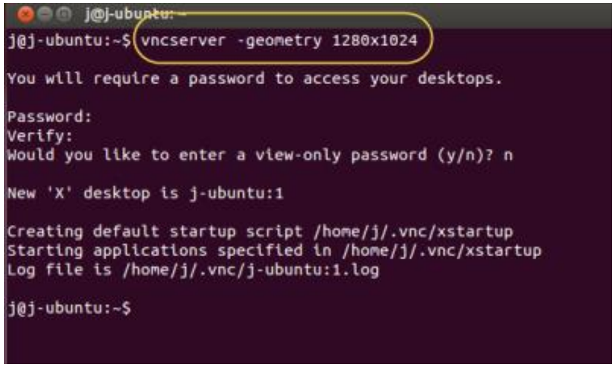
Using VNC



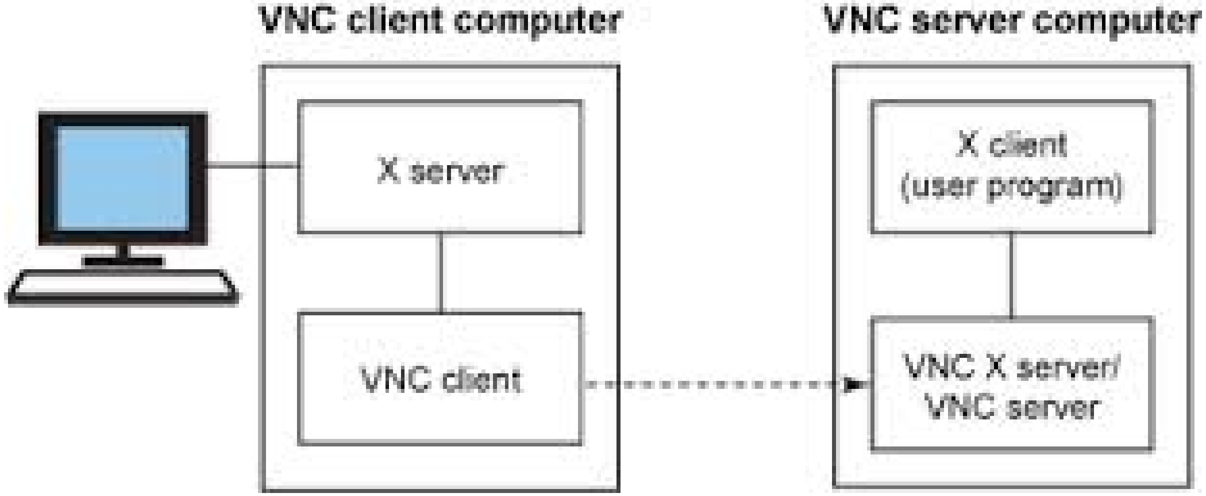
VNC Viewer : Connection Details

Server: 192.168.1.102

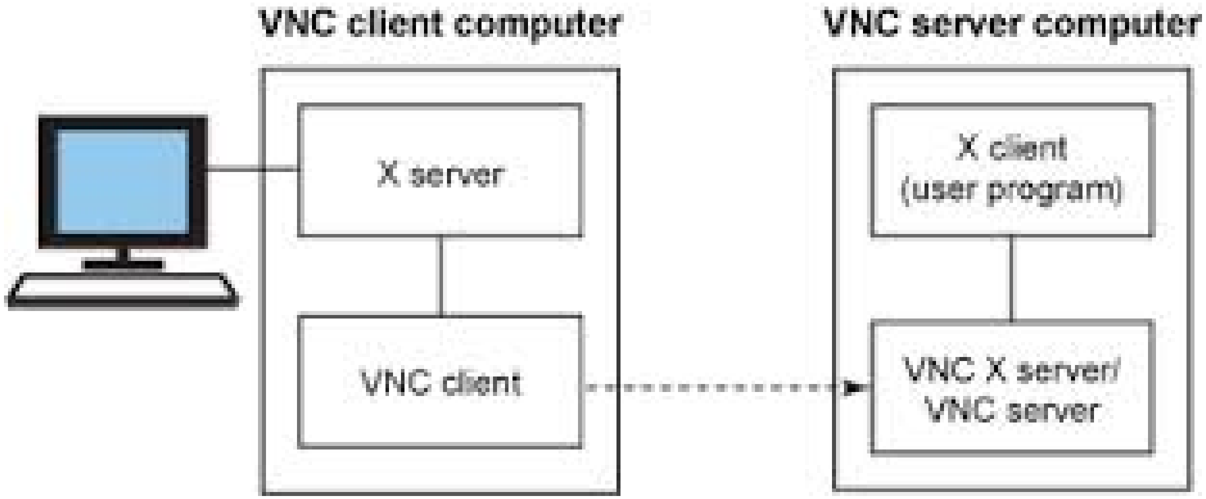
Encryption: Always Off

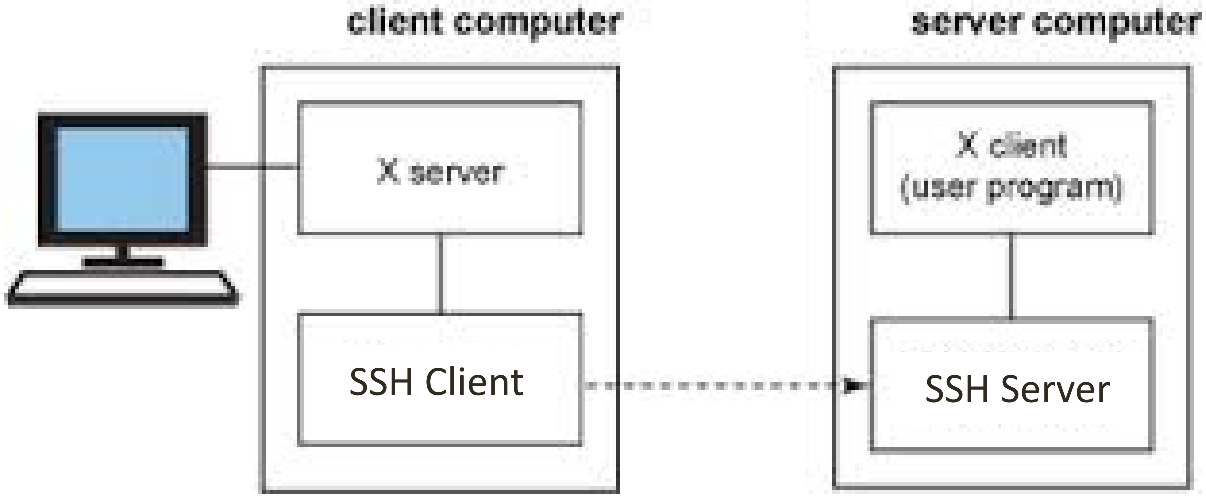


```
j@j-ubuntu: ~
j@j-ubuntu:~$ vncserver -geometry 1280x1024

You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? n

New 'X' desktop is j-ubuntu:1

Creating default startup script /home/j/.vnc/xstartup
Starting applications specified in /home/j/.vnc/xstartup
Log file is /home/j/.vnc/j-ubuntu:1.log

j@j-ubuntu:~$
```

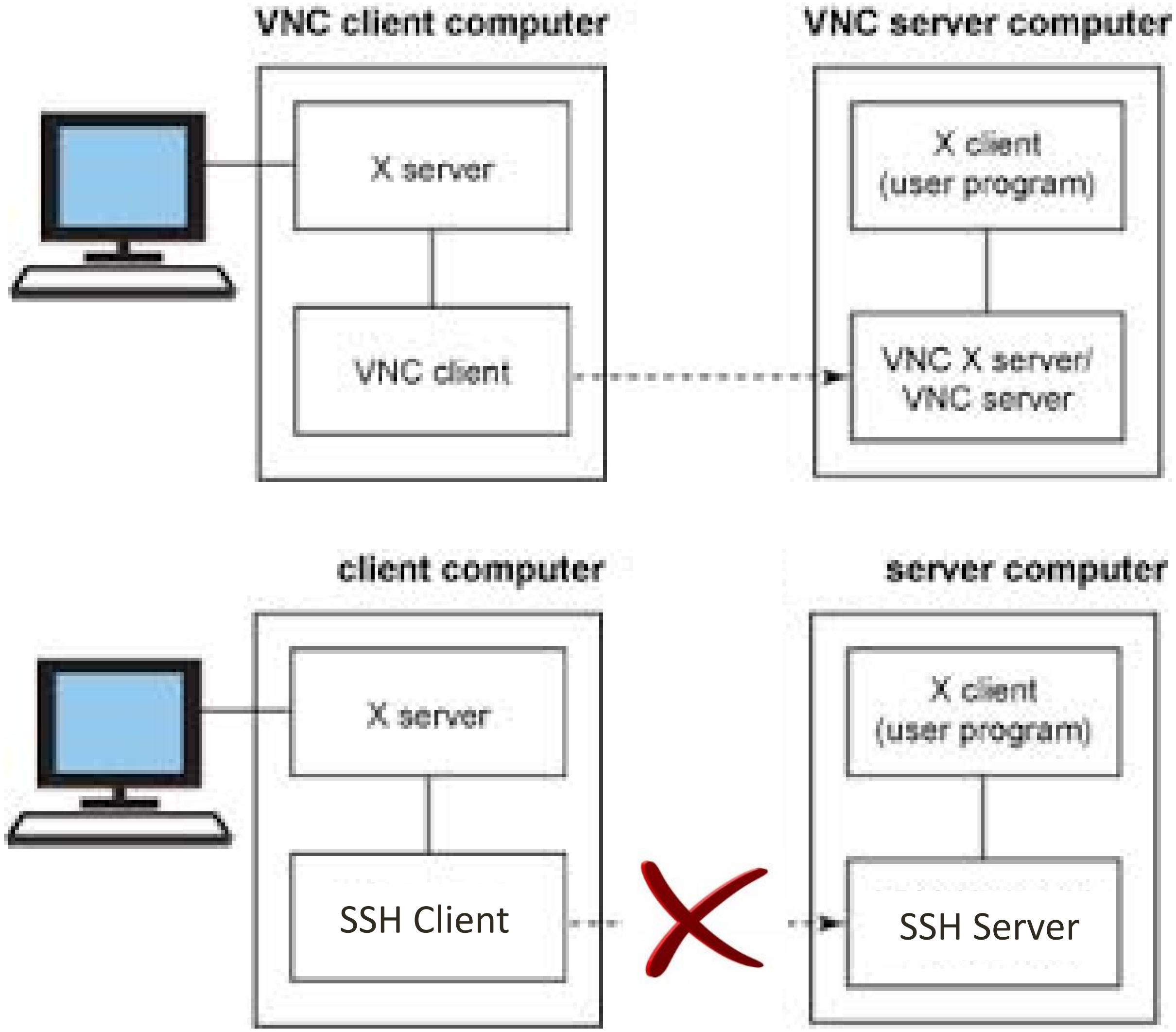


Comparison

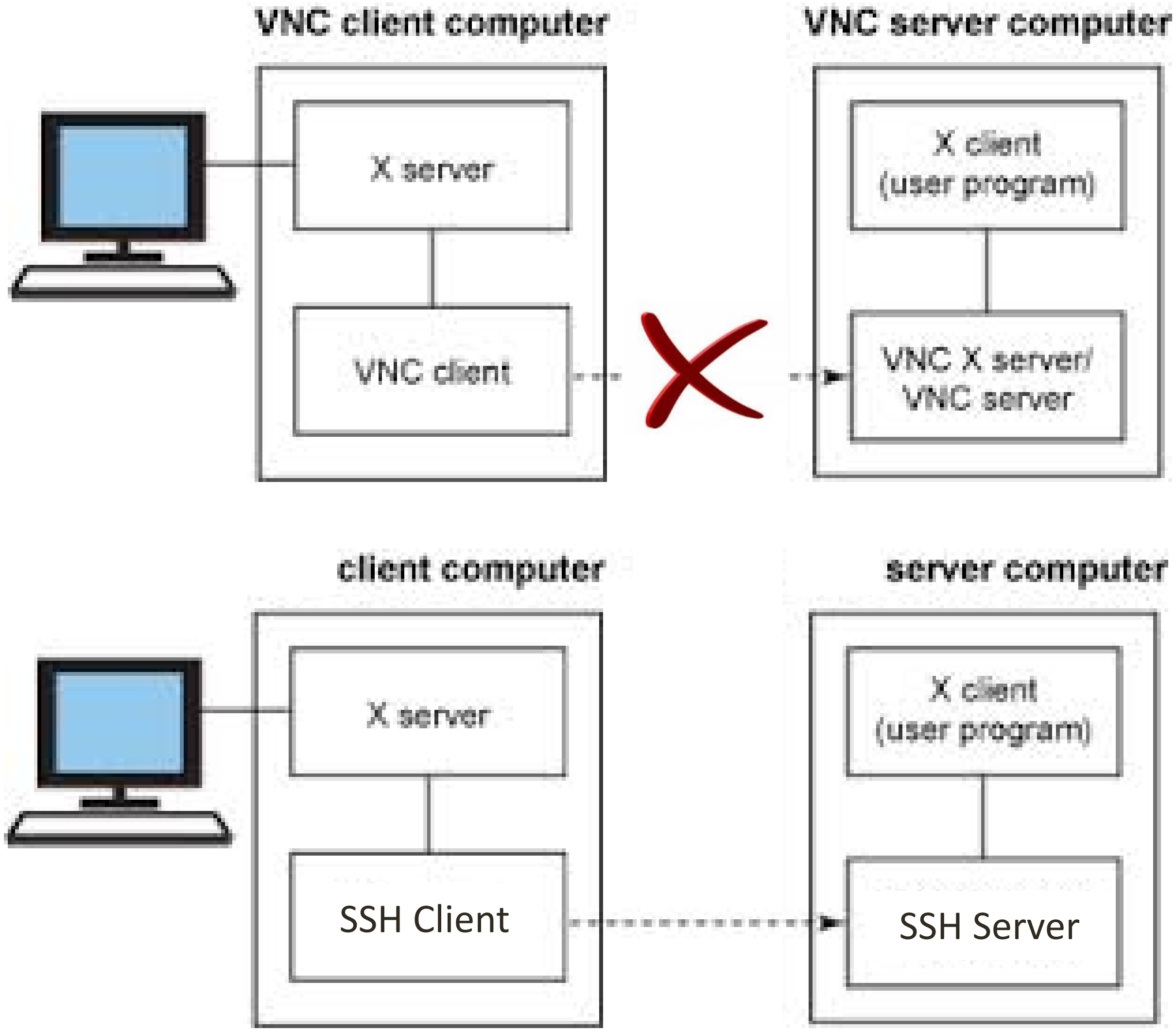

VNC client computer
VNC server computer
X server
VNC client
X client
(user program)
VNC X server/
VNC server


client computer
server computer
X server
SSH Client
X client
(user program)
SSH Server

Comparison


VNC client computer
VNC server computer
X server
X client
(user program)
VNC client
VNC X server/
VNC server
client computer
server computer
X server
X client
(user program)
SSH Client
SSH Server

Comparison


VNC client computer
VNC server computer
X server
VNC client
X client
(user program)
VNC X server/
VNC server
client computer
server computer
X server
SSH Client
X client
(user program)
SSH Server

Linux4

Embedded Systems

http://Linux4EmbeddedSystems.com