Switch -

\* A switch is a hardware component in a network that connects devices together.

\* It allows multiple devices to share a network while preventing each device's traffic from interfering with other devices' traffic.

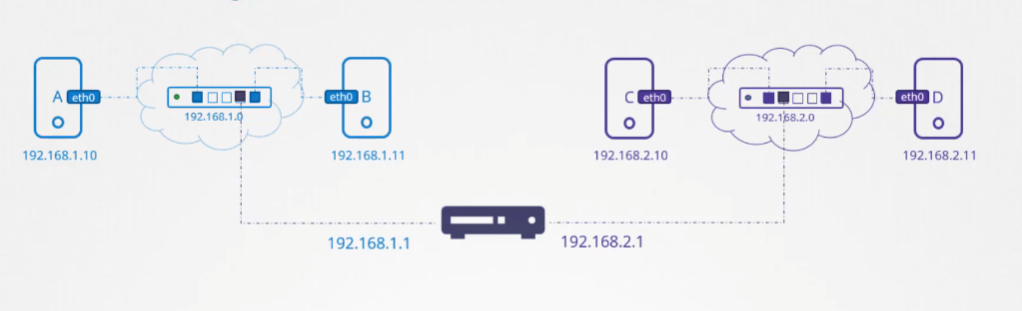
By using ip link command we can see the interfaces of devices

(the term "interfaces" refers to the points of interaction or connections that enable communication and data exchange between devices or between a device and its users.)

(No, the term "point of interaction" does not specifically refer to an IP address. In the context of devices, a "point of interaction" generally refers to a location or

mechanism where interaction occurs, and it can encompass various aspects, such as physical interfaces, user interfaces, or even software interfaces.)

-> Imagine a USB port on your computer. It's a physical interface that lets you connect a USB device, like a flash drive. The port serves as a bridge for the computer and the flash drive to share data.



1) For two devices to communicate with each other directly, they should be on the same network.

2) To be in same network, we need to connect them to a switch (this is only one case, we have another options also)

3) After they are connected to a switch, we will get an IP address for that network by DHCP (Dynamic Host Configuration Protocol). For devices also we will get IP address.

If we want to assign with our self, we will use static (manually assigned) IP addresses.

\*\* ip addr add <IP\_ADDRESS>/<SUBNET\_MASK> dev <INTERFACE\_NAME>

ip addr add 192.168.1.10/24 dev eth0

-> Command to assign a static IP address to a network interface

4) Now devices can communicate with each other, we can check by using ping command.

\*\* ping <IP\_ADDRESS>

5) Switch can only enable communication within same network.

How does a system in one network can communicate with other system in other network???

-> A router is a networking device that is commonly used to connect two or more different networks together

To a router 2 networks are assigned, then the router will have 2 different IP addresses. If there are three, then it will get 3 IP addresses.

By using router we are connected 2 different networks. In one network we have system A, B and in other network we have system C, D...

Now I want to system B to communicate with system C. but the router won't know where to forward the packets.

How does a system in one network can communicate with other system in other network???

1) First we need to connect both networks to a router.

2) We need to update the routing table, so that router will know where to send the packets.

3) If you don't update the table, the router may either drop the packet or attempt to forward it along a path that doesn't lead to Network B.

4) To see those routing configuration use route command.

\* route

-> Command to see the routing table

5) We have 2 types to add configurations to routing table - Static Routing - Dynamic Routing

6) ip route add is used to manually add a static route to the routing table on a Unix-like operating system.

\* ip route add [destination\_network\_or\_host] via [router\_ip\_address]

-> destination\_network\_or\_host - IP address of network B ----- network IP

-> router\_ip\_address - when we are saying to a router if 2 networks connected then the router will have 2 IP address. In that we need to give Network A router IP address ---- router IP

7) Router address on network A (router\_ip\_address) is called gateway

8) Run ‘route’ command to see the table again.

9) So by this we can send from system B to system C, but if you need to send from system C to B, again we need to update routing table

\* ip route add [destination\_network or host] via [router\_ip\_address]

-> destination\_network\_or\_host - IP address of network A ----- network IP

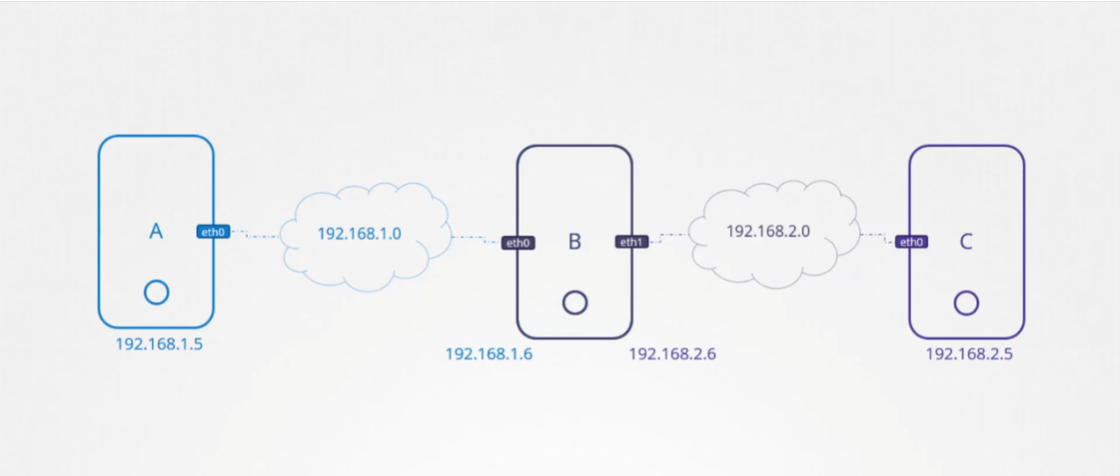
-> router\_ip\_address - when we are saying to a router if 2 networks conneted then the router will have 2 IP address. In that we need to give Network B router IP address ---- router IP

10) If you want to access internet also we need to update in routing table with ip route add

\* ip route add default via <ip address>

\* ip route add 0.0.0.0/0 via <ip address> (both same)

-> If you give this in routing table, that, means for that routing ip any destination network was not there then it will send the packet to the router's interface within its own network.



* Here we have 3 systems
* System A and B connected to network (IP – 192.168.1.0)
* System B and C connected to network (IP – 192.168.2.0)
* System A IP (192.168.1.5)
* System C IP (192.168.2.5)
* System B IP (192.168.1.6) & IP (192.168.1.6)

How system A communicate to System C?? We know router

But in this case system B will act as a router

We need to update the routing table

* Ip addr add 192.168.2.0/24 via 192.188.1.6

How system C communicate to System A??

* Ip addr add 192.168.1.0/24 via 192.168.2.6

So now we can communicate successfully but we won’t get response back.

By default in Linux packets are not forwarded from one interface to the next.

Here packets received on eth0 on system B are not forwarded to elsewhere through eth1. This is the way for security reasons.

But if we want we can forward packets from one network to the other.

In /proc/sys/net/ipv4/ip\_forward file by default it will have 0 (that means no forward)

If we modify that to 1 (It will forward to other interface one time)

echo 1 > /proc/sys/net/ipv4/ip\_forward

By changing the value in here does not persist the changes across reboots. So to make it permanent we need to change the value in /etc/sysctl.conf file (net.ipv4.ip\_forward = 1)

**DNS** (Domain Name System)

Generally we can give names to ip address in /etc/hosts file. But this is local file which is dedicated to only our local system.

Giving names like this is called Name Resolution.

Suppose we have 10 systems working for a project and we have 4 servers. To access those servers with names we need to update etc/hosts file in each system. If in case we change any server IP address in future, we need to update all 10 systems along with that. So this is complicated. There we can use DNS.

This is how it will work….

We have etc/reslov.conf file. That is used to specify the IP addresses of DNS servers that your system should use to resolve domain names. We will provide our IP address and names of our servers in DNS server and that DNS server address will be given in reslov.conf file.

And we are allowed to write direct ip address and names of our server in reslov.conf file, but it is not recommended.

In general, it's not advisable to add individual IP addresses directly to the reslov.conf file

* **Maintenance and Scalability**
* **Flexibility**
* **Troubleshooting**

Nslookup command is used to query the Domain Name System (DNS) to obtain information about domain names, including their corresponding IP addresses and other DNS records

Nslookup <domain name>

* nslookup <example.com

nslookup <ip address> This is called reverse lookup

(We can search both by domain and ip address with using nslookup)

Nslookup will not work for the records stored in /etc/hosts file. It is query the DNS server.

Dig is also used to search for DNS server

Ex. Dig [www.google.com](http://www.google.com)

Suppose, we have a server entry in both /etc/hosts and DNS server, then if you ping it… it will search in hosts file first and then dns server.

This searching order is mentioned in /etc/nsswitch.conf. We can modify it if we want.

DNS forwarding can be particularly useful in scenarios where your local DNS server might not have direct connections to the entire internet or where it's not configured to handle queries for external domains. This is known as DNS forwarding.

(Forward All to 8.8.8.8)

Domain Names

[www.facebook.com](http://www.facebook.com)

The reason they are in this format separated by dots is to group like things together.

Top-Level domains (They represent the intent of the website)

.com – commercial or general purpose

.net – network

.edu – educational organizations

.org – non-profit organization

[www.google.com](http://www.google.com)

. is the root

.com is top level domain

Google is domain name

www is a sub domain

The sub domain help in further grouping things under google.

For example, in the domain "google.com," "apps.google.com" is a subdomain.

Ex. Maps.google.com

Ex. Drive.google.com

When you try to reach apps.google.com… how the searching pattern will done

1. Organizational DNS (It will search in our own internal DNS) – it don’t know. So, it will forward that request to internet.
2. Root DNS ( so it knows that it is related to .com, then it will forward to .com DNS)
3. .com DNS
4. Google DNS (after in .com DNS, it will forward to google DNS & then this DNS provides the IP address of the server serving the apps)

In order to speed up future results, your organization DNS server may choose to cache the IP for a period of time. (Up to few seconds or few minutes)

You have some websites for your customers like web.mycompany.com, mail.mycompany.com

For outsiders this names are perfect, but to save time for internal organizational working people we can use them as web (as a shortcut)

For that we need to use ‘search’ domain in reslov.conf file. Search domain is to allow users to enter short, unqualified hostnames (without the full domain suffix) and still have their systems successfully resolve those names to fully qualified domain names.

We can give only one search in reslov.conf file… we can add multiple names to search

Ex. search stratos.xfusioncorp.com yahoo.com

Ex. In reslov.conf file

Search mycompany.com

So when we try to pig web next time, It actually means web.mycompany.com

First it will search in DNS server. Then it will search in ‘search domain’

How records are stored in DNS server??

A (Address) Record: Associates a domain name with an IPv4 address.

Example: example.com IN A 192.168.1.1

AAAA (IPv6 Address) Record: Associates a domain name with an IPv6 address.

Example: example.com IN AAAA 2001:db8::1

CNAME (Canonical Name) Record: Provides an alias for another domain name (canonical name).

Example: www.example.com IN CNAME example.com