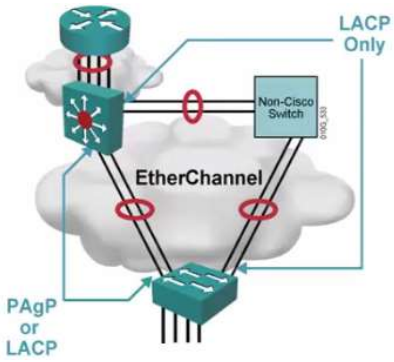
Ether channel

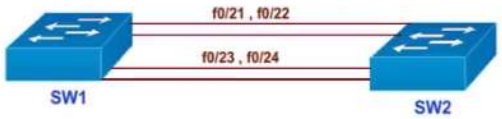
Etherchannel

- Used to aggregate bandwidth between multiple **L2/L3** interfaces.
- EtherChannel increases bandwidth and provides redundancy by aggregating individual links between switches.



Etherchannel

- EtherChannel load balances traffic over all the links in the bundle.
- Up to 8 links can be used to combine in to one logical link.
- Etherchannel can be configured as layer 2 or layer 3.
- Port-channel is the logical instance of the physical interfaces.

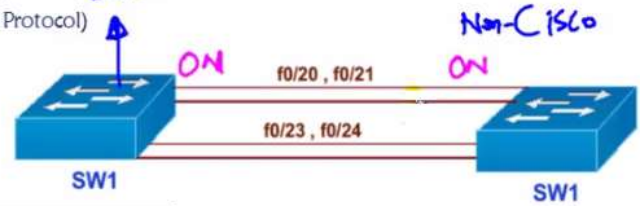


To configure etherchannel -both the switches should have same Speed,duplex,same Vlan and same set of configurations.

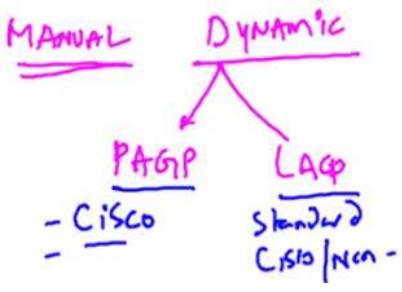
Etherchannel Modes:

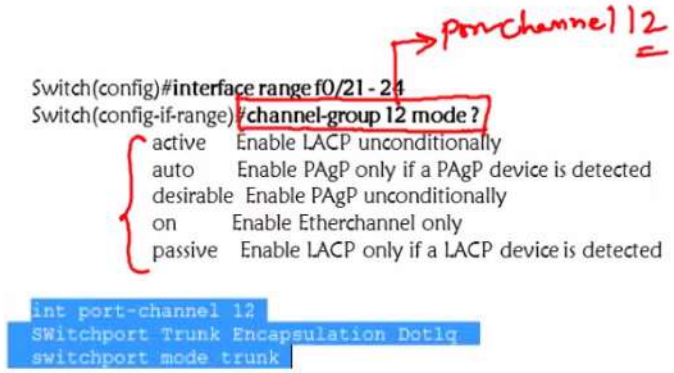
EtherChannel can be dynamically configured between switches using two protocols.

- -PAgP (Port Aggregation Protocol) Cisco
- -LACP (Link Aggregation Control Protocol)



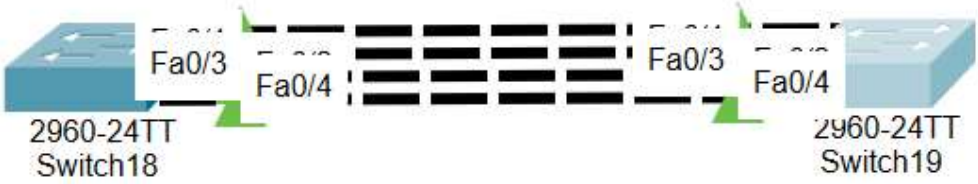
Mode	Result
On	PAgP and LACP disabled (negotiation disable) Manual
Auto	Passively listen for PAgP
Desirable	Actively negotiate PAgP
Passive	Passively listen for LACP
Active	Actively negotiate LACP





Whatever changes that we do on logical interface – same changes will happen to the physical interfaces.
Verification commands – Show ether channel summary , show ip int br ,
Any implementation on port channel will impact the indivisual switch

Switchport trunk encapsulation dot1q / ISL



SW1

SW1(config)#
interface range fastEthernet 0/1 - 4
channel-group 2 mode desirable
exit
interface port-channel 2
switchport mode trunk
exit

SW2

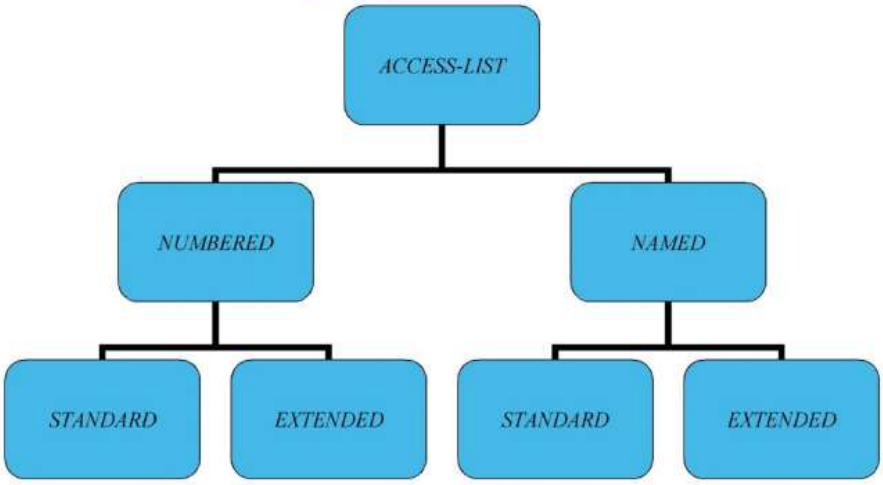
SW1(config)#
interface range fastEthernet 0/1 - 4
channel-group 2 mode desirable
exit
interface port-channel 2
switchport mode trunk
exit

ACCESS CONTROL LIST

ACCESS CONTROL LIST (ACL)

- ACL is a set of rules which will allow or deny the specific traffic moving through the router
- It is a Layer 3 security which controls the flow of traffic from one router to another.
- It is also called as Packet Filtering Firewall.

Types of Access-list



STANDARD ACCESS LIST	EXTENDED ACCESS LIST
1. The access-list number range is 1 – 99	1. The access-list number range is 100 – 199
2. Can block a Network, Host and Subnet	2. We can allow or deny a Network, Host, Subnet and Service
3. All services are blocked.	3. Selected services can be blocked.
4. Implemented closest to the destination.	4. Implemented closest to the source.
5. Filtering is done based on only source IP address	5. Filtering is done based on source IP , destination IP , protocol, port no

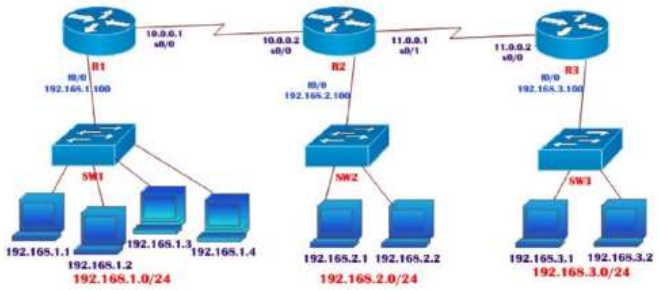
The main advantage we get in named ACL is we can add a specific line or delete a specific line form the list of rules written which is not available in numbered access list

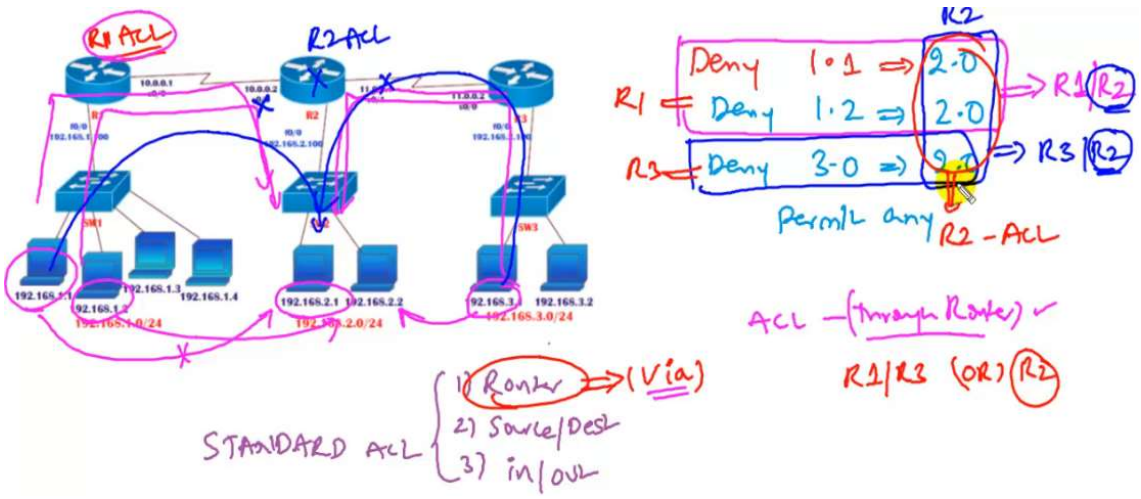
Lab : standard access-list

TASK: Configure the Appropriate router as per the rules given

1. Deny the host **192.168.1.1** communicating with **192.168.2.0**
2. Deny the host **192.168.1.2** communicating with **192.168.2.0**
3. Deny the network **192.168.3.0** communicating with **192.168.2.0**
4. Permit all the remaining traffic

NOTE: the Above ACL rules should not affect the other communication





We can implement ACL on R2 since it is common or we can also implement ACL on R1 as well as R2

Wild card mask

Tells the router which portion of the bits to match or ignore.

0 = must match
1 = ignore

Global Subnet Mask
– Customized Subnet Mask
==========
Wild Card Mask
==========

255.255.255.255
–255.255.255. 0
==========
0. 0. 0. 255
==========

255.255.255.255
–255.255.255.240
==========
0. 0. 0. 15
==========

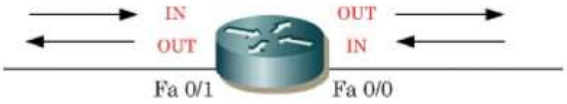
- Wild Card Mask for Network will be Inverse mask
- Wild Card Mask for a Host will be always **0.0.0.0**

To write ACL Statement

1. On which Router to implement ACL
2. Identify Source & Destination
3. In/out

Understanding IN / OUT

- In to the router
- Out of the router



Creation of Standard Access List

Router(config)# access-list <acl no> <permit/deny> <source address> <source WCM>

R-2(config)# access-list 15 deny 192.168.1.1 0.0.0.0
R-2(config)#access-list 15 deny host 192.168.1.2
R-2(config)#access-list 15 deny 192.168.3.0 0.0.0.255
R-2(config)#access-list 15 permit any

Implementation:
R-2(config)#interface fastEthernet 0/0
R-2(config-if)#ip access-group 15 out

R-2#sh access-lists
Standard IP access list 15
deny host 192.168.1.1
deny host 192.168.1.2
deny 192.168.3.0 0.0.0.255
permit any

Access-list Rules

- Works in Sequential order.
- All deny statements have to be given First (preferable most cases)
- There should be at least one Permit statement (mandatory)
- An implicit deny blocks all traffic by default when there is no match (an invisible statement).
- Can have one access-list per interface per direction. (i.e.) Two access-lists per interface, one in inbound direction and one in outbound direction.
- Any time a new entry is added to the access list, it will be placed at the bottom of the list. Using a text editor for access lists is highly suggested.
- You cannot remove one line from an access list.

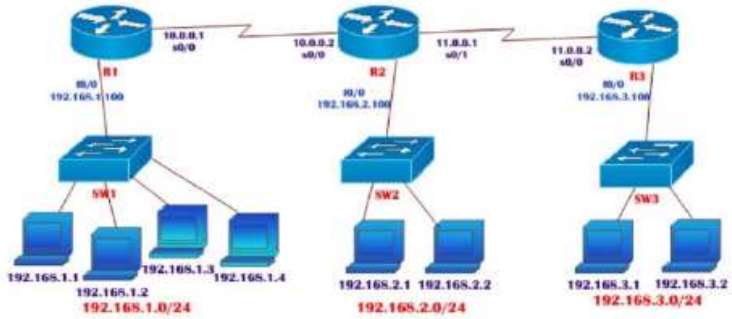
Extended Access-list

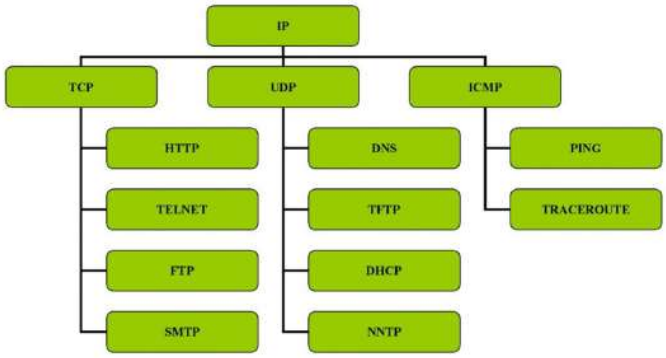
1. The access-list number range is 100 – 199
2. We can allow or deny a Network, Host, Subnet and Service
3. Selected services can be blocked.
4. Implemented closest to the source.
5. Filtering is done based on source IP , destination IP , protocol, port no

TASK: Configure the Appropriate router as per the rules given below

1. Deny the users on LAN 192.168.2.0 should not access 192.168.1.3 HTTP service
2. Deny the users on LAN 192.168.3.0 should not access 192.168.1.4 FTP service
3. Deny the users on LAN 192.168.3.1 should not access 192.168.1.3 HTTP service
4. Deny the users on LAN 192.168.2.0 should not get DNS service from DNS server 192.168.1.4
5. Deny the users from the host between 192.168.3.2 and 192.168.1.2 should not be able to send ICMP (ping /trace) messages
6. Remaining hosts and services should be permitted

NOTE: the Above ACL rules should not affect the other communication





Operators: *eq (equal to)*
neq (not equal to)
lt (less than)
gt (greater than)

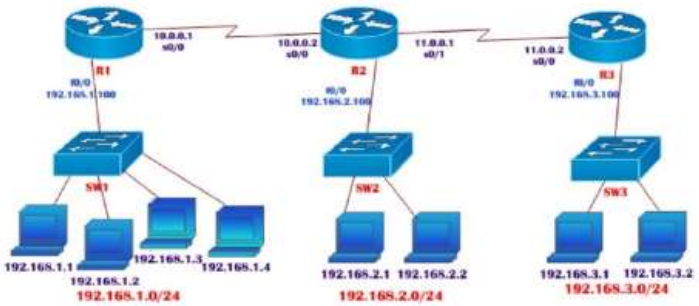
Eq = match only the packets which is equal to 80 example
Neq= other than 80 deny everything
Lt = deny/allow all port numbers lesser than 2000
Gt= deny/allow all port numbers greater than 2000

Extended ACL Syntax

Router(config)#

access-list <acl no> <permit/deny> <protocol> <source address> <source wildcard mask> <destination address> < destination wildcard mask> <operator> <service>

Router(config)#interface <interface type> <interface no>
Router(config-if)#ip access-group <number> <out/in>



R-1(config)#access-list 145 deny tcp 192.168.2.0 0.0.0.255 host 192.168.1.3 eq www

R-1(config)#access-list 145 deny tcp 192.168.3.0 0.0.0.255 host 192.168.1.4 eq ftp

R-1(config)#access-list 145 deny tcp host 192.168.3.1 host 192.168.1.3 eq www

R-1(config)#access-list 145 deny udp 192.168.2.0 0.0.0.255 host 192.168.1.4 eq domain

R-1(config)#access-list 145 deny icmp host 192.168.3.2 host 192.168.1.2 echo

R-1(config)#access-list 145 deny icmp host 192.168.3.2 host 192.168.1.2 echo-reply

R-1(config)#access-list 145 permit ip any any

Implementation:

R-1(config)# **interface fastEthernet 0/0**
R-1(config-if)# **ip access-group 145 out**
OR
R-1(config)# **interface serial 0/0**
R-1(config-if)# **ip access-group 145 in**

Named ACL

- Access-lists are identified using Names rather than Numbers.
- Names are Case-Sensitive
- No limitation of Numbers here.
- One Main Advantage is Editing of ACL is Possible (i.e) Removing a specific statement from the ACL is possible.
- IOS version 11.2 or later allows Named ACL

Creation of Standard Named Access List

Router(config)# ***ip access-list standard <name>***

Router(config-std-nacl)# ***<permit/deny> <source address> <source wildcard mask>***

Implementation of Standard Named Access List

*Router(config)#****interface <interface type><interface no>***

*Router(config-if)#****ip access-group <name> <out/in>***

```
R-1#sh access-lists
Standard IP access list CCIE
    10 deny host 192.168.1.1
    20 deny host 192.168.1.2
    30 deny 192.168.3.0 0.0.0.255
    40 permit any
```

```
R-1#sh access-lists
Standard IP access list CCIE
    10 deny host 192.168.1.1
    12 deny host 192.168.1.3
    20 deny host 192.168.1.2
    30 deny 192.168.3.0 0.0.0.255
    40 permit any
```

By default whenever we create acl sequence numbers (always in multiples of 10)will be added by default it is use to identify the acl s written , now lets say I want to add a new rule inbetween then I will use these sequence numbers , usually when I add a new acl it will add in the last but if I want to add it inbetween then *R1() ip access-list extended CCIE*
If you want to remove any line then *12 deny host 192.168.1.3*
R1() ip access-list extended CCIE
No 20
So this way we can selectivily add or remove a specific line in case of ACL

IPV6

IPV6 was introduced to overcome the shortage of IPV4 addresses

IP Address

- *IP Address is Logical Address.*
- *It is a Network Layer address (Layer 3).*
- *IP address is given to every device in the network and it is used to identify the device with in the network.*

- ***Two Versions of IP:***
  - ***IP version 4 is a 32 bit address***
  - ***IP version 6 is a 128 bit address***

What is the Need of IPV6 address - We have 2 ^ 32 = 4.3 billion ip addresses but these ip addresses are not enough to meet the growing requirements , as the internet is growing at a very faster rate .

http://www.ipv6now.com.au/primers/IPv6Myths.php

- Larger address space.
- Simpler header for increased router efficiency.
- No more broadcasts.
- Stateless auto-configuration.
- Built-in support for Mobile IP.
- Built-in support for IPsec security.
- Rich transition features.
- Easy IP address renumbering.
- Capability to have multiple addresses per interface.

2001:0db8:0000:0000:1234:0000:0000:3c4d

Global prefix interface ID

Unicast , Multicast and Anycast

Unicast Address

1) Global unicast

- like public IP (routable)
- starts with 2000::/3 (the first three bits 001) assigned by IANA

2) unique local

- like private ip (routable)
- FC00::/7
- They are not routable in the global IPv6 Internet.
- Starts wth either FC or FD in the first two numbers

3)link local

- default IPV6 address on every ipv6 enabled interface(non routable) FE80::/10
- Routers do not forward packets with link-local addresses.

Multicast

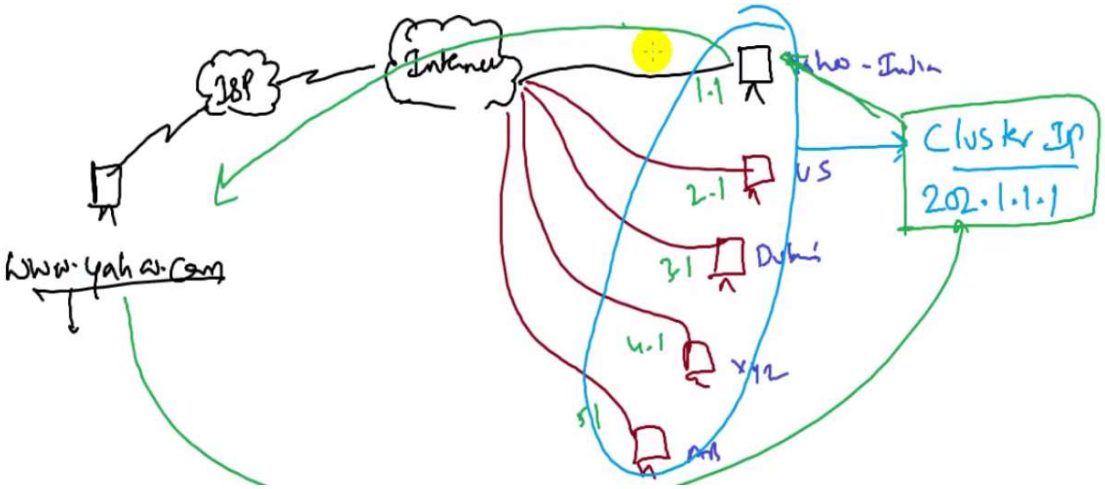
In IPV6 multicast address will be starting with FF (FF00::/8)

Any cast

- An anycast address is an address that is assigned to a set of interfaces that typically belong to different nodes.
- similar to multicast , identify multiple interfaces but sends to only one whichever it finds first.
- unique local and Global unicast addresses can be used as anycast.

Device(config)# interface f0/0

Device(config-if)# IPv6 address ipv6-prefix/prefix-length anycast



Stateless auto configuration – EUI 64 bit format

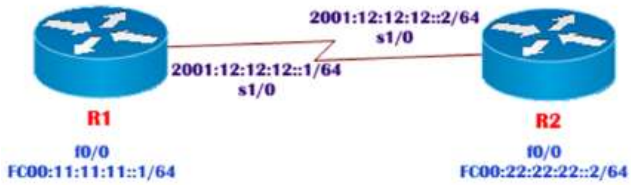
Assigning IPV6 address

Static (Manual)

- R-1(config)#**interface fastEthernet 0/0**
- R-1(config-if)#**ipv6 address fc00:11:11:11::1/64**

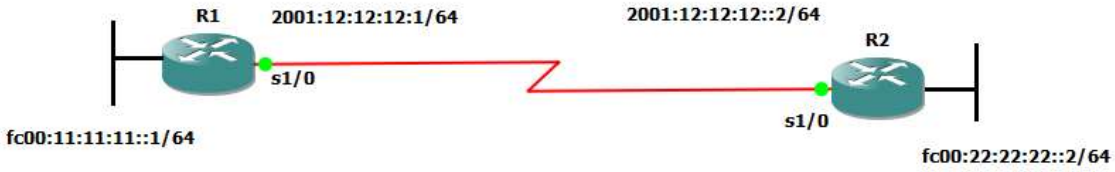
Auto-configuration

- Statefull (via DHCP)
- Stateless(Device gets IPv6 add by including the MAC add)



IPV6 routing types – Static , RIP ng , OSPF V3 , EIGRP – IPV6 unicast routing
Static routing

```
R-1(config)#ipv6 route fc00:22:22:22::/64 2001:12:12:12::2
R-2(config)#ipv6 route fc00:11:11:11::/64  2001:12:12:12::1
```



```
interface serial 1/0
 ipv6 address 2001:12:12:12::1/64
 no sh
 exi

 interface serial 1/1
 ipv6 address fc00:11:11:11::1/64
 no sh
 no keepalive
 exi

 ipv6 route FC00:22:22:22::/64 2001:12:12:12::2
 exi
```

R2

```
interface serial 1/0
  ipv6 address 2001:12:12:12::2/64
 no sh
 exi
```

interface serial 1/1
ipv6 address fc00:22:22:22::2/64
no sh
no keepalive
exi

ipv6 route FC00:11:11:11::/64 2001:12:12:12::1
exi

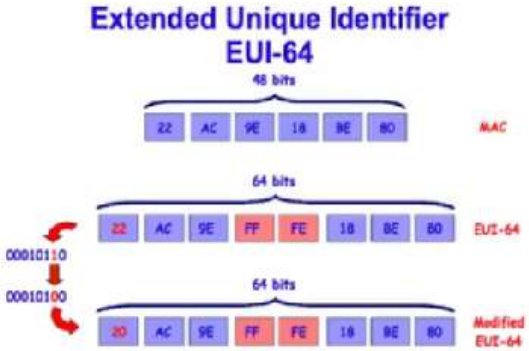
****

EUI-64 (extended Unique Identifier)

A host can automatically assign itself a unique 64-bit IPv6 interface identifier without the need for manual configuration or DHCP.

R-1(config)#**int gigabitEthernet 0/0**
R-1(config-if)#**ipv6 enable**

R-1#sh ipv6 int brief
GigabitEthernet0/0 [up/up]
FE80::20C:85FF:FE04:D001

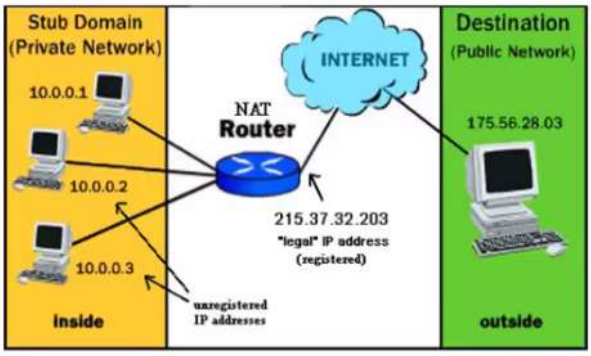


Network address translation

- NAT is the method of Translation of private IP address into public IP address ".
- In order to communicate with internet we must have registered public IP address.

Address translation was originally developed to solve two problems:

1. to handle a shortage of IPv4 addresses
2. Hide network addressing schemes.



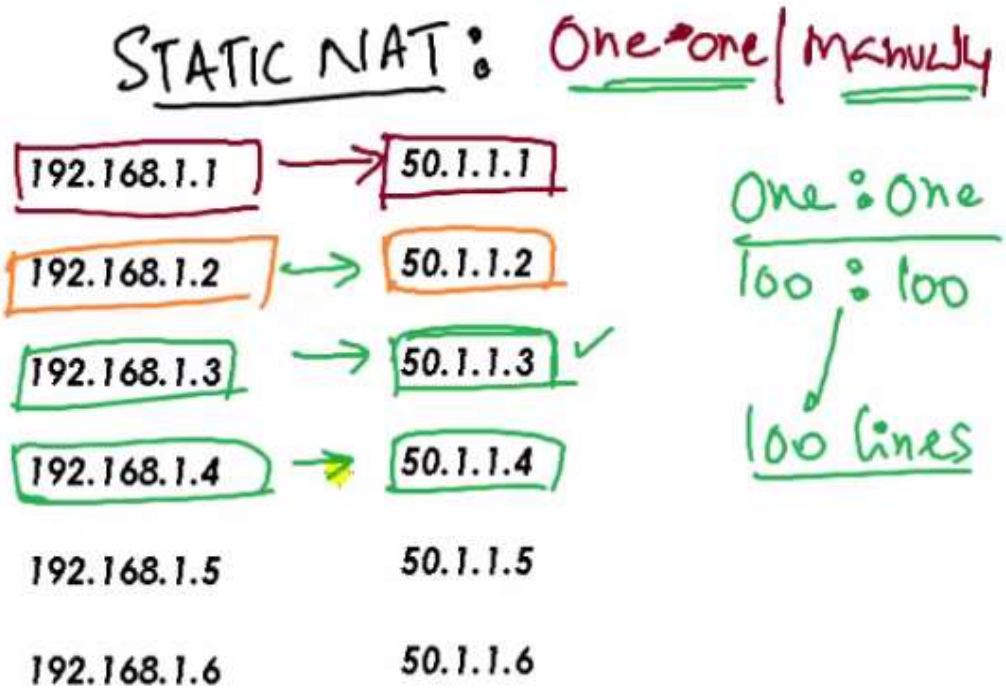
Private Address range

There are certain addresses in each class of IP address that are reserved for Private Networks. These addresses are called private addresses.

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

Types of NAT:-

- Static NAT
- Dynamic NAT
- Port Address Translation (PAT)



Static NAT

- One to one mapping done Manually
- For every private IP needs on registered IP address (one : one)

(Config)# **IP nat inside source static** <private IP> <public IP>

Configuration of static NAT

R-1(config)#**ip nat inside source static 192.168.1.1 50.1.1.1**
R-1(config)#**ip nat inside source static 192.168.1.2 50.1.1.2**
R-1(config)#**ip nat inside source static 192.168.1.3 50.1.1.3**

Implementation

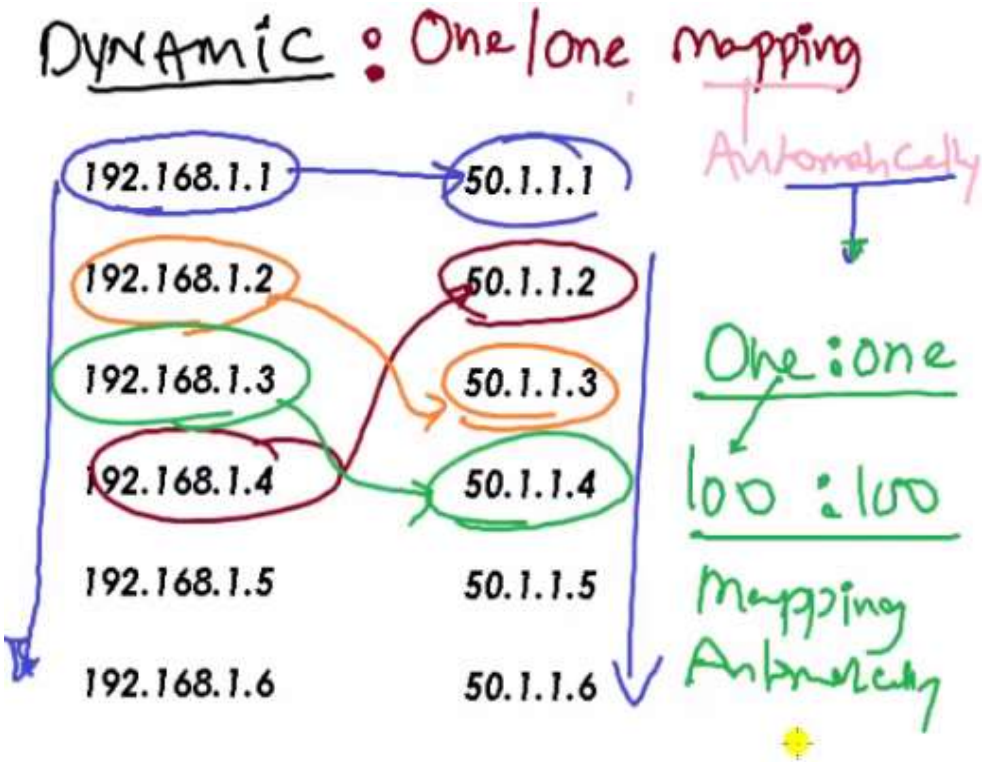
R-1(config)#**interface fastEthernet 0/0**
R-1(config-if)#**ip nat inside**
R-1(config-if)#**exit**

(interface facing towards LAN)

R-1(config)#**interface serial 0/0**
R-1(config-if)#**ip nat outside**

Lab setup for NAT

1. Configure IP address as per the diagram.
2. Configure default route towards ISP from R1
3. Configure static route from ISP to public IP used for translation



LAB: Dynamic NAT

Syntax :

(Config) # access-list < ACL-NO> permit <NET.ID> <WCM>

(Config) #ip nat pool <NAME> <starting Public IP> <end Public IP> netmask <mask>

(Config) # ip nat inside source list <ACL-NO> pool <NAME>

Configuration of DYNAMIC NAT

R-1(config)#access-list 55 permit 192.168.1.0 0.0.0.255

R-1(config)#ip nat pool CCNA 50.1.1.1 50.1.1.200 netmask 255.255.255.0

R-1(config)#ip nat inside source list 55 pool CCNA

Implementation

R-1(config)#interface fastEthernet 0/0

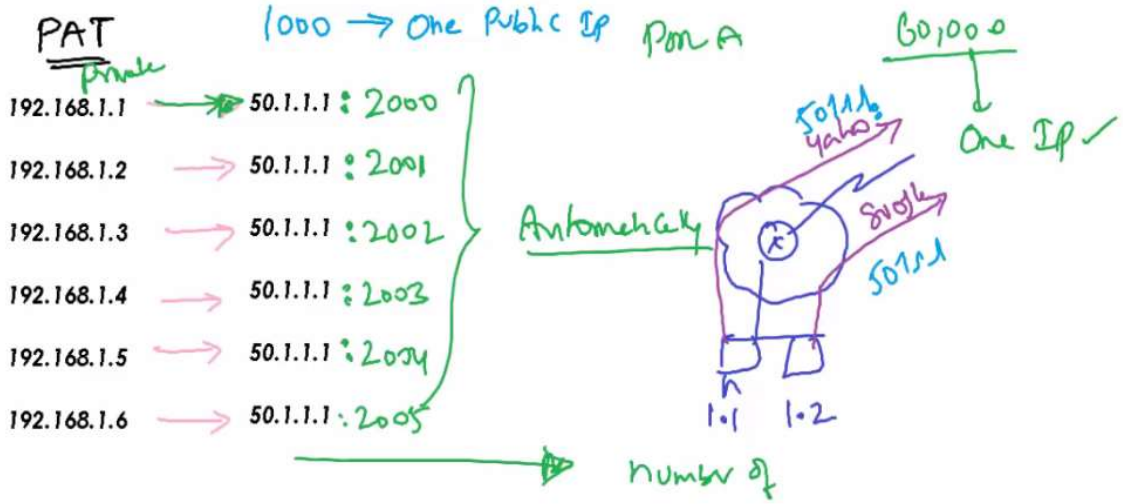
R-1(config-if)#ip nat inside

R-1(config-if)#exit

(Interface facing towards LAN)

R-1(config)#interface serial 0/0

R-1(config-if)#ip nat outside



Syntax:

(Config) # access-list < ACL-NO> permit <NET.ID> <WCM>

(Config) #ip nat inside pool <NAME> <starting Public IP> <end Public IP> netmask < mask>

(Config) # ip nat inside source list <ACL-NO> pool <NAME> overload

PAT Configuration

R-1(config) #access-list 55 permit 192.168.1.0 0.0.0.255

R-1(config) #ip nat pool CCNA 50.1.1.1 50.1.1.1 netmask 255.255.255.255

R-1(config) #ip nat inside source list 55 pool CCNA overload

Implementation

R-1(config) #interface fastEthernet 0/0

R-1(config-if) #ip nat inside

R-1(config-if) #exit