

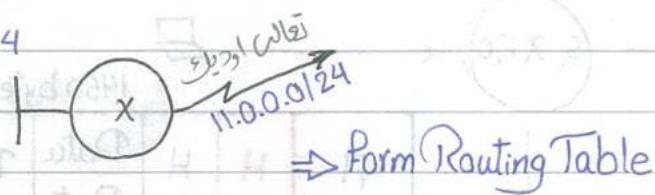
## Session 1 Part 1

### Routing Overview

#### Routing Protocol:

It is exchange of information between routers, so as each router tell others about network it can reach.

11.0.0.0/24



Control Plane

RIPv1, RIPv2
IGRP, EIGRP
<b>ISIS, OSPF</b>
EGP, BGP
LDP, RSVP
MP-BGP

#### Routed Protocol:

It is protocol that carry user data traffic from end to end.

ex: TCP, UDP, IPv4, MPLS

Forwarding Plane

(Data Plane)

FIB, LF-IB

CEF

Application layer

HTTP(80), RIP(520), LDP(646)  
FTP(20,21), BGP(179), TDP(711)  
SMTP(25) (Cisco)

Transport layer

TCP, UDP

Internet layer

IPv4

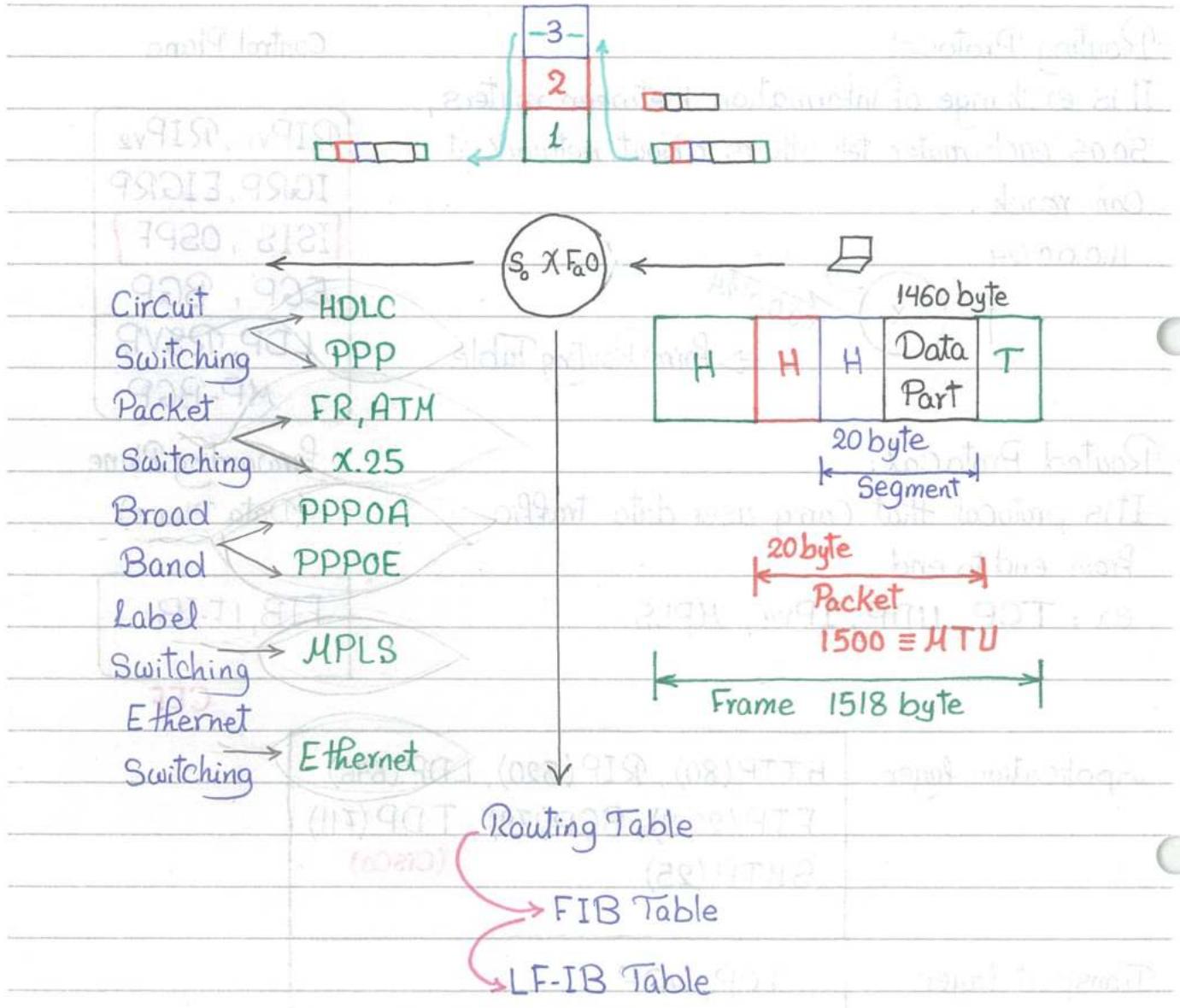
Network Access Layer

Ethernet

TCP/IP Model

## Session 1 Part 2

### Routing Process



## Session 1 Part 3

### Routing Classification

Static Routing:

- If network is simple.

If Single path exists to DST.

- Built routing table manually.

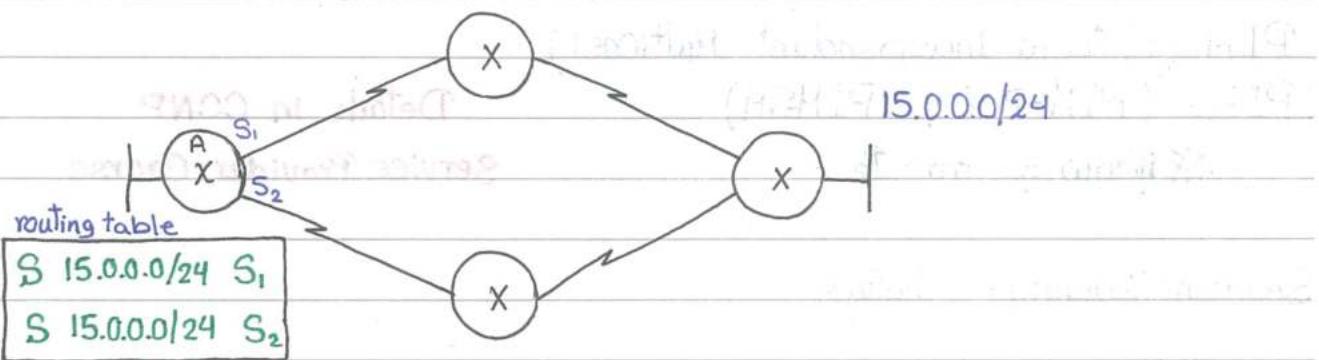
(Config) # ip route network mask vector [distance] [tag name] <sub>1-255</sub>

destination <sub>النهاية</sub>

exit interface  
ip of next router

Community  
refer to vrf

ex:



A(Config) # ip route 15.0.0.0 255.255.255.0 S<sub>1</sub> } load sharing

A(Config) # ip route 15.0.0.0 255.255.255.0 S<sub>2</sub> } (load Blancing)

OR

A(Config) # ip route 15.0.0.0 255.255.255.0 S<sub>1</sub> 10 → Primary Path

A(Config) # ip route 15.0.0.0 255.255.255.0 S<sub>2</sub> 20 → Backup Path

### Dynamic Routing:

IP network is Complex.

many paths exist to dst.

Building routing table automatic. (SW Protocol)

IGP/EGP

Interior /Exterior  
Gateway / Gateway  
Protocol / Protocol

Inside AS/between AS

ex: RIPv1 ex: EGP ex: RIPv2 ex: RIPv1 Link State: exchange LSA with

RIPv2, ISIS BGP ISIS, OSPF IGRP mask using multicast ex: isis, OSPF

OSPF, IGRP

EIGRP

Classless/Classful  
Protocols

Understand  
Subnetting

Doesn't  
send  
mask in  
updates

Distance vector: exchange routing

table without mask ex: RIP, IGRP

Advanced Distance Vector:

exchange routing table with mask  
using multicast ex: RIP2, EIGRP

Path vector: exchange routing table

with rich attributes ex: EGP, BGP

Multicast Routing:

Class D 224.x.x.x → 239.x.x.x

MOSPF (Multicast OSPF)

DVMRP (Distance Vector Multicast Routing Protocol).

PIM (Protocol Independant Multicast).

PIM<sub>2</sub> (PIM-DM, PIM-SM)

# Show ip mroute

Details in CCNP

Service Provider Course

Segment Routing: Future

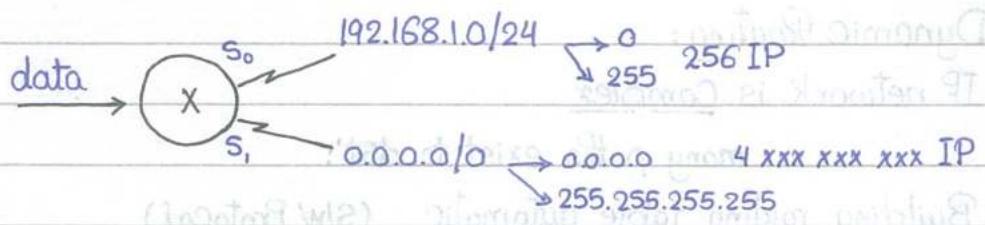
## Session 1 Part 4

### Routing Table Search

Best Path is:

1. Longest bit match:

longer mask = more accurate



2. Least admin. distance:

Protocol	Admin. Distance	It's for Cisco only Default & can be changed
Connect	0	
Static	0, 1	vector is Interface name, IP of next hop
RIP	120	
OSPF	110	
EIGRP	90, 170, 5	Internal, External, Summary
ISIS	115	
BGP	20, 200	External, Internal

### 3. Least Metric:

Hop (RIP), Cost (OSPF), Bandwidth (EIGRP)

Expense, Error (ISIS).

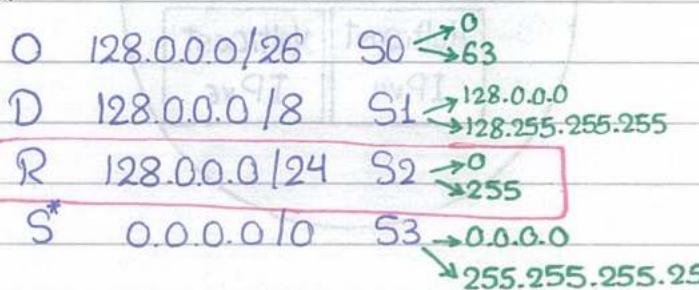
### 4. Equal Loadsharing:

Up to 4 paths by default, maximum by configuration 16 or more

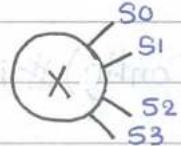
(Config) # router Protocol

(Config-router) # maximum-paths 1-16 or 256  
default 4

ex:



Data dst  
ip 128.0.0.5



worst bit match

آخر ملائم علشان نوصل

## Session 2 Part 1

### Routing Procedure

1. Check IOS Capability for routing & routed protocol.

IOS XE } Cisco      ISIS      IPv4

IOS XR }            BGP      IPv6

JunOS } Juniper

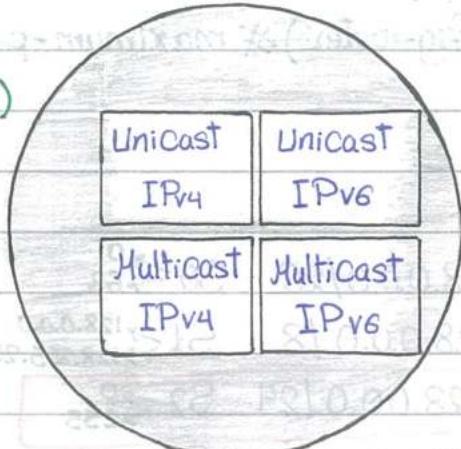
2. Enable router processor.

(Config) # ip routing (enable by default)

(Config) # ip multicast-routing

(Config) # ip vs unicast-routing

(Config) # ipvs multicast-routing



3. Create routing process (routing instance).

(Config) # router Protocol

4. Activate routing process on interface.

(Config-router) # network ip of interface

OR...

(Config) # interface name

(Config-if) # ip Protocol

5. Exchange of updates.

Send / Receive

6. Building routing table.

### Routing Table:

Protocol	Network / mask Prefix / Prefix length	Distance	Vector	Time
		[admin. / metric] distance	exit interface & IP of next hop	Period from add route to routing Table

### ARP Table

IP	MAC	IP	DLCI

### IARP Table

address store & forward

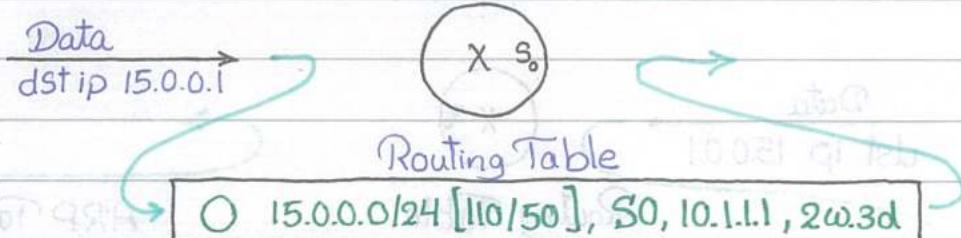
### Session 2 Part 2

### Routing Generations

1<sup>st</sup> generation:

Always routing (normal routing) = Sequential search  
Route thousands of pps.

(Config) # ip routing



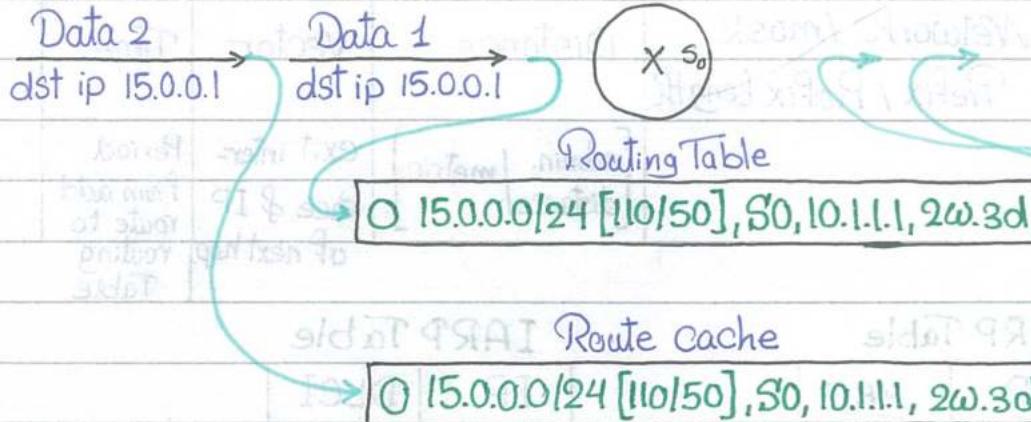
2<sup>nd</sup> generation:

Route once, Switch many.

Route tens of thousands of pps.

(Config) # ip routing

(Config) # ip route-cache



#Show route-cache

#Show ip cache

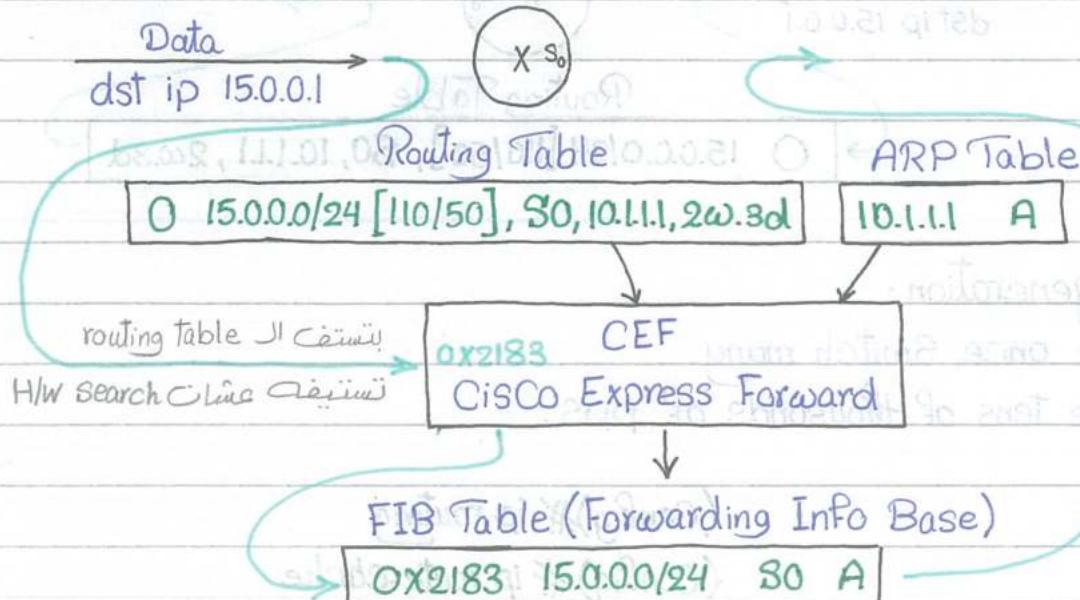
3<sup>rd</sup> generation:

Always switching (Pointer Search).

Route millions of pps.

(Config)#ip routing

(Config)#ip cef

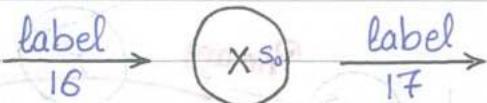


#Show ip cef

4<sup>th</sup> generation:

Route ten of millions of pps.

"MPLS"



LF-IB (Label Forwarding info base) 16 So 17

5<sup>th</sup> generation:

Route hundred of millions of pps.

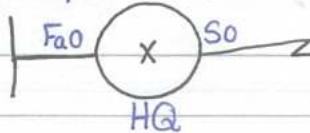
"Segment Routing"

### Session 2 Part 3

#### Static & default static routes

(Config) # ip route network mask vector [Distance] [tag name]  
 exit interface → IP of next hop  
 circuit switching every time  
 only

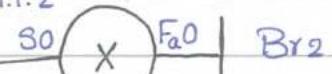
172.16.0.0/16 10.1.1.4



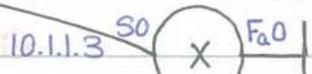
10.1.1.1 Br1 192.168.1.0/24



10.1.1.2 192.168.2.0/24



10.1.1.3 192.168.3.0/24



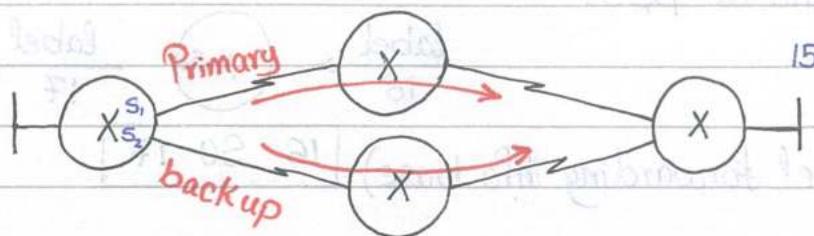
HQ (Config) # ip route 192.168.1.0 255.255.255.0 10.1.1.1

HQ (Config) # ip route 192.168.2.0 255.255.255.0 10.1.1.2

HQ (Config) # ip route 192.168.3.0 255.255.255.0 10.1.1.3

Brx (Config) # ip route 0.0.0.0 0.0.0.0 10.1.1.4

Active / Backup scenario (floating static):



(Config) # ip route 15.0.0.0 255.255.255.0 S, Primary Path

(Config) # ip route 15.0.0.0 255.255.255.0 S<sub>2</sub> 10 backup Path

Equal loadsharing / Loadbalance:

(Config) # ip route 15.0.0.0 255.255.255.0 S,

(Config) # ip route 15.0.0.0 255.255.255.0 S<sub>2</sub>,

Unequal loadsharing / load balance:

(Config) # ip route 15.0.0.0 255.255.255.0 S, (0-255)

(Config) # ip route 15.0.0.0 255.255.255.192 S<sub>2</sub> (0-63) longest bit match

per dst loadsharing

No. of IPs = 256 - mask

$$= (256-255)(256-255)(256-255)(256-192) = 64 \text{ IP}$$

## Session 2 Part 4

### OSPF Operation

@startup: (Config) # router OSPF Process Id

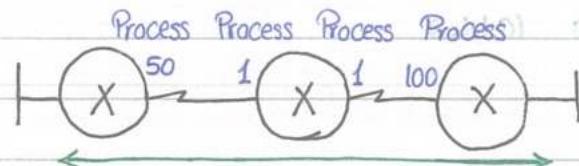
locally significant

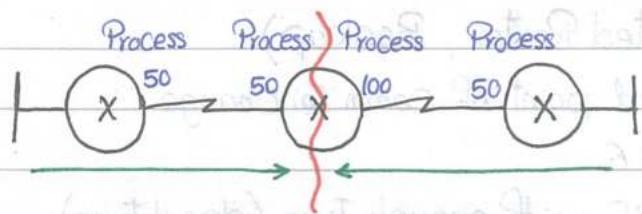
Can never be 0

1- 65535

May not match

between routers





## 0. Create Router ID (32-bit):

0.1 - Manual:

(Config-router) # router-id 10.0.1.1

This is only method on IPv4 ...

0.2 - Highest Loopback interface IP:

0.3 - Highest physical active interface IP:

ethernet ↓ serial ↓ up

★ ★ Can't create OSPF without router id ★ ★

★ ★ Each process id has router id ★ ★

## 1. Neighbor Discovery (hello protocol):



Neighborship Conditions:

- Same subnet.
- Same area id / area type.
- Same Authentication method & key.
- Same hello & dead time.

Build OSPF neighbor table:

Neighbor ID	state	Interface	IP of neighbor

# Show ip ospf neighbor

## 2. DR & BDR Election (Designated Router, Backup):

welcome new comers and focal point of comm @ change.

DR & BDR multicast 224.0.0.6

@ Router that 1<sup>st</sup> boots OSPF with enough time (dead time).

(b) Router having highest priority per interface (by default = 1).

**priority 0** can never be DR or BDR ...

(Config-if) # ip ospf priority 0-255

(c) Router has highest RID.

## 3. Router Discovery :

exchange of LSDB (all LSAs).



← Here is DBD Data Base Description

LSAid	Adv Router	Seq#	CheckSum	Link Count
-------	------------	------	----------	------------

x.x.x.x	x.x.x.x	0x 8000 0001	0x	
---------	---------	--------------	----	--

Cities will change

LSA will

Interfaces will

change will

LSA

Ack

LSR

i need LSAs with id ...

Link State Request

LSU

here are details about  
LSAs you requested it

Link state update

Ack

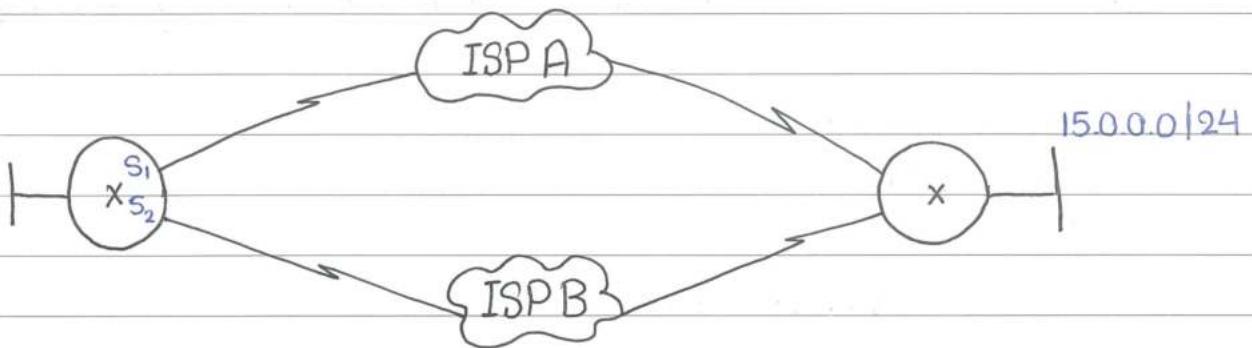
Full state

## Session 3 Part 1

### Case Study

Using static route only satisfy the following:

- Data destined to 15.0.0.0 → 15.0.0.15 should go through ISP A.
- Data destined to 15.0.0.16 → 15.0.0.255 should go through ISP B.
- Data destined to 15.0.0.50 & 15.0.0.100 Should be dropped.



```
(Config) # ip route 15.0.0.0 255.255.255.240 S1      ↗0  
(Config) # ip route 15.0.0.0 255.255.255.0      S2      ↙0  
(Config) # ip route 15.0.0.50 255.255.255.255 null0  
(Config) # ip route 15.0.0.100 255.255.255.255 null0
```

## OSPF in Single Area

@ start up:

Operation :

1. Create RID. manual, highest loopback, highest active physical interface
2. Neighbor discovery. exchange of hello
3. Electing DR/BDR. first to run OSPF, highest priority, highest RID.
4. Routes discovery. exchange of LSDB
5. Choosing best path using Dijkstra (SPF algorithm).
6. Building Forwarding table.

### @Convergence:

-: Periodic hello:-

Point to point : HDLC , PPP

Broadcast Multiple Area (BMA) : Ethernet  $\Rightarrow$  every 10sec/40sec

Non Broadcast Multiple Area (NBMA) : X.25, FR, ATM every 30sec/120 sec

Periodic LSA every 30min for LSDB refreshment.

### @Change:

Router that feels the change will generate new LSA with new sequence no.

→ update LSDB .

→ New Dijkstra calculation .

→ Update routing table .

## Session 3 Part 2 OSPF Multiple Area

Divide main AS into sub ASs called Areas .

### \* Types of Routers :

1. ABR (Area Border Router) .

2. ASBR (AS Boundary Router) .

3. Internal Router .

4. Backbone Router "Can be internal, ABR or ASBR"

router has one interface at least in area Zero .

### \* Types of LSAs :

1. LSA Type 1 (Router LSA) :

It is generated by each router, describing:

1. Interface IP .

2. Neighbor IP (optional) .

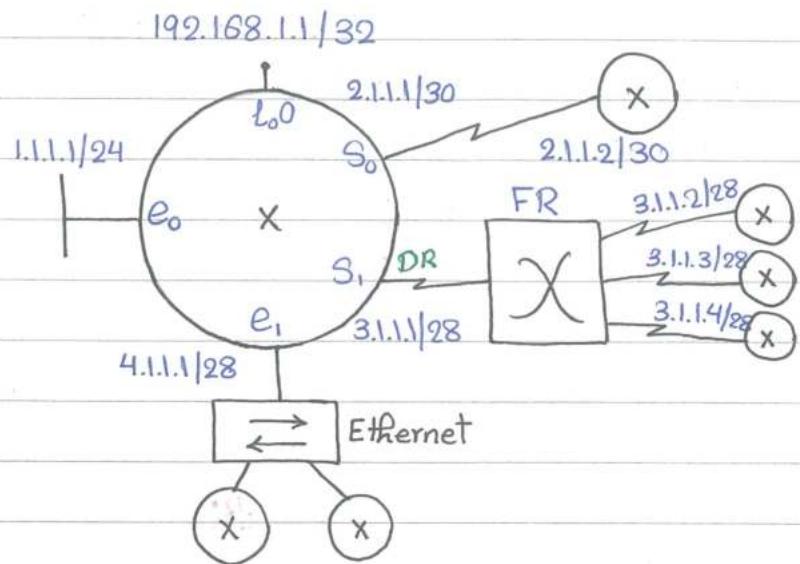
3. Link mask .

4. Link Cost .  $10^8 / \text{BW}$

5. Link Type . P2P, Transit, stub, virtual link

## 6. Router Type.

LSA ID = RID



\* Show ip ospf database

DBD

LSA ID	Adv. RID	Seq#	Link count	checksum
192.168.1.1	192.168.1.1	0x80000001	6	0x21BF

\* Show ip ospf database router

LSU

Interface IP / link mask	Cost	Link Type	Neighbor IP
1.1.1.1 / 24	10	Stub	
2.1.1.1 / 30	64	P2P	2.1.1.2
3.1.1.1 / 28	64	Transit	
4.1.1.1 / 28	64	Transit	
192.168.1.1 / 32	1	Stub	

Router Type : Internal Router

2. LSA Type 2 (Network Link LSA) :

It is generated by DR, describing:

1. DR ID ..

192.168.1.1

2. List of losers ..

3.1.1.2 , 3.1.1.3 , 3.1.1.4

LSA ID = IP of winning Interface

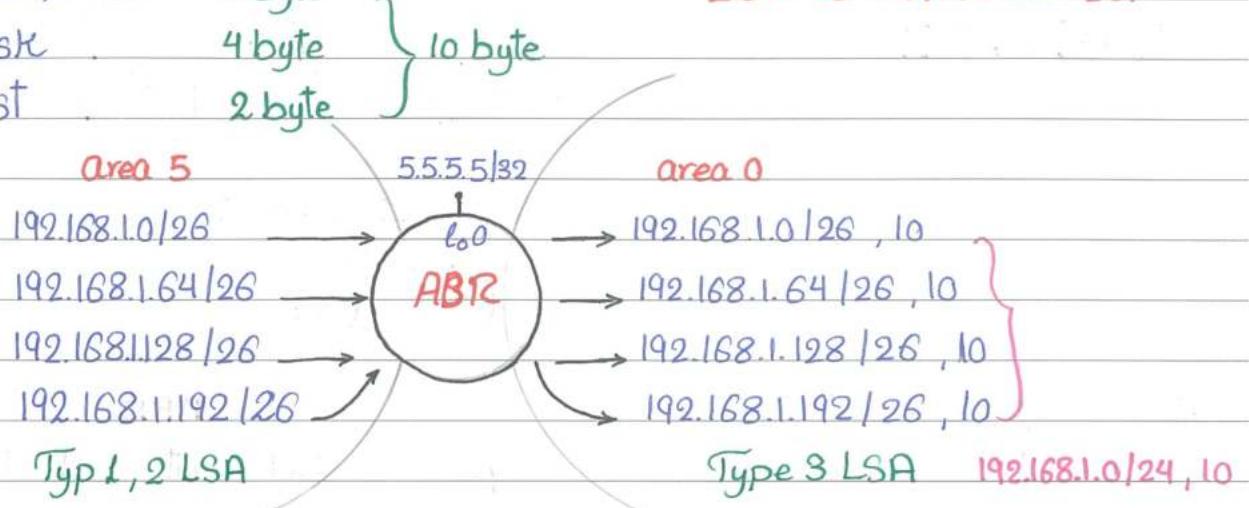
### Session 3 Part 3

#### 3. LSA Type 3 (Summary LSA):

It is generated by ABR, describing:

1. Network Id : 4 byte
2. Mask : 4 byte
3. Cost : 2 byte

LSA Id = Network Id



ABR (Config) # router ospf 1

ABR (Config-router) # area 5 range 192.168.1.0 255.255.255.0

# Show ip ospf database

DBD

LSA Id	Adv. RID	Seq #	Checksum
192.168.1.0	5.5.5.5	0x8000 0001	0x1401

# Show ip ospf database Summary

LSU

Network Id / mask    Cost

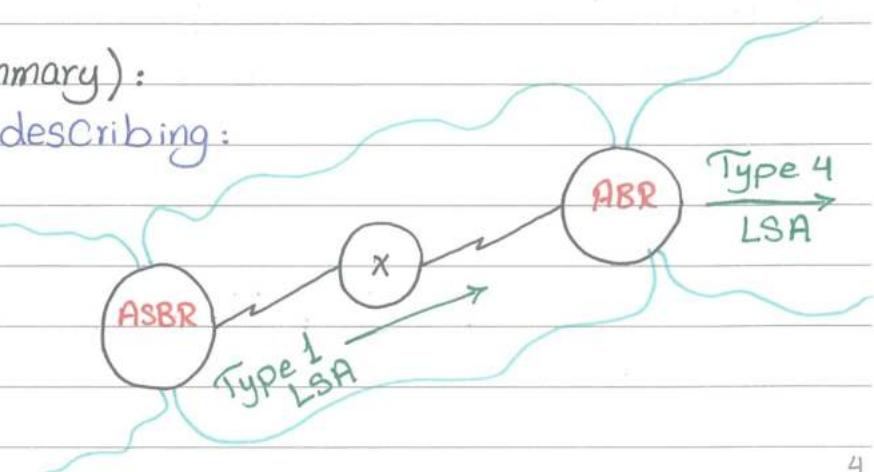
192.168.1.0 / 24    10

#### 4. LSA Type 4 (ASBR Summary):

It is generated by ABR, describing:

1. RID of ASBR
2. Cost to reach ASBR.

LSA Id = ASBR ID

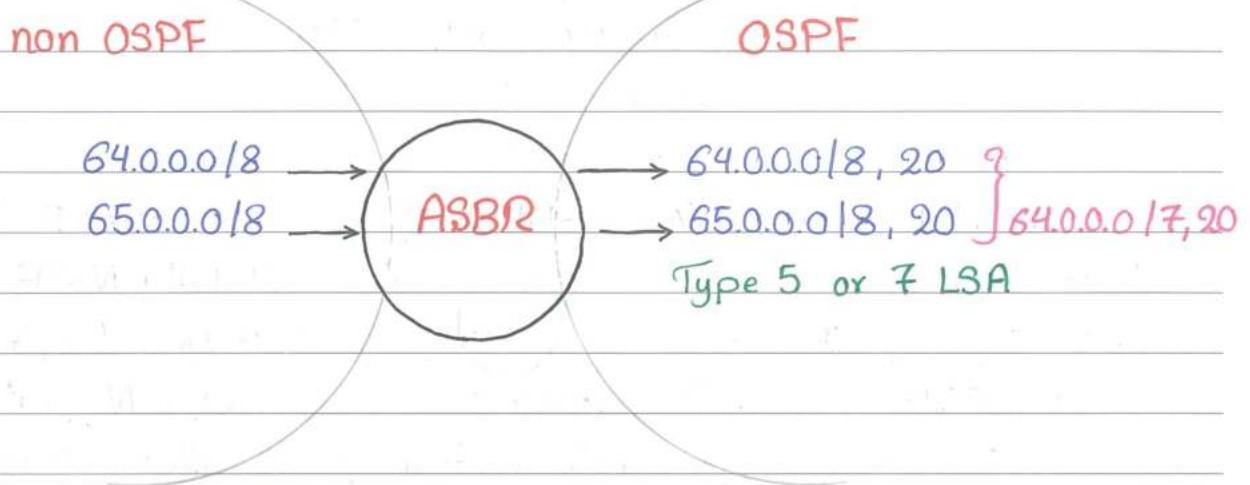


## 5. LSA Type 5 / Type 7 (External / External NSSA):

It is generated by ASBR, describing:

- 1. Network . 4 byte
- 2. Mask . 4 byte } 10 byte
- 3. Cost . 2 byte

LSA Id = Network Id



ASBR(Config) # router ospf 1

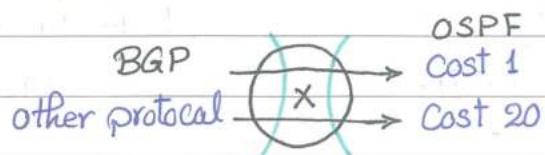
ASBR(Config-router) # summary-address 64.0.0.0 254.0.0.0

→ Redistribution to OSPF:

Metric Type :

Type 1 add internal cost to external cost.

Type 2 (default) ignore internal cost.



## LSDB

Type 1 LSA

Type 2 LSA

Intra area

O

Type 3 LSA

Type 4 LSA

Inter area

OIA

Type 5 LSA

Type 7 LSA

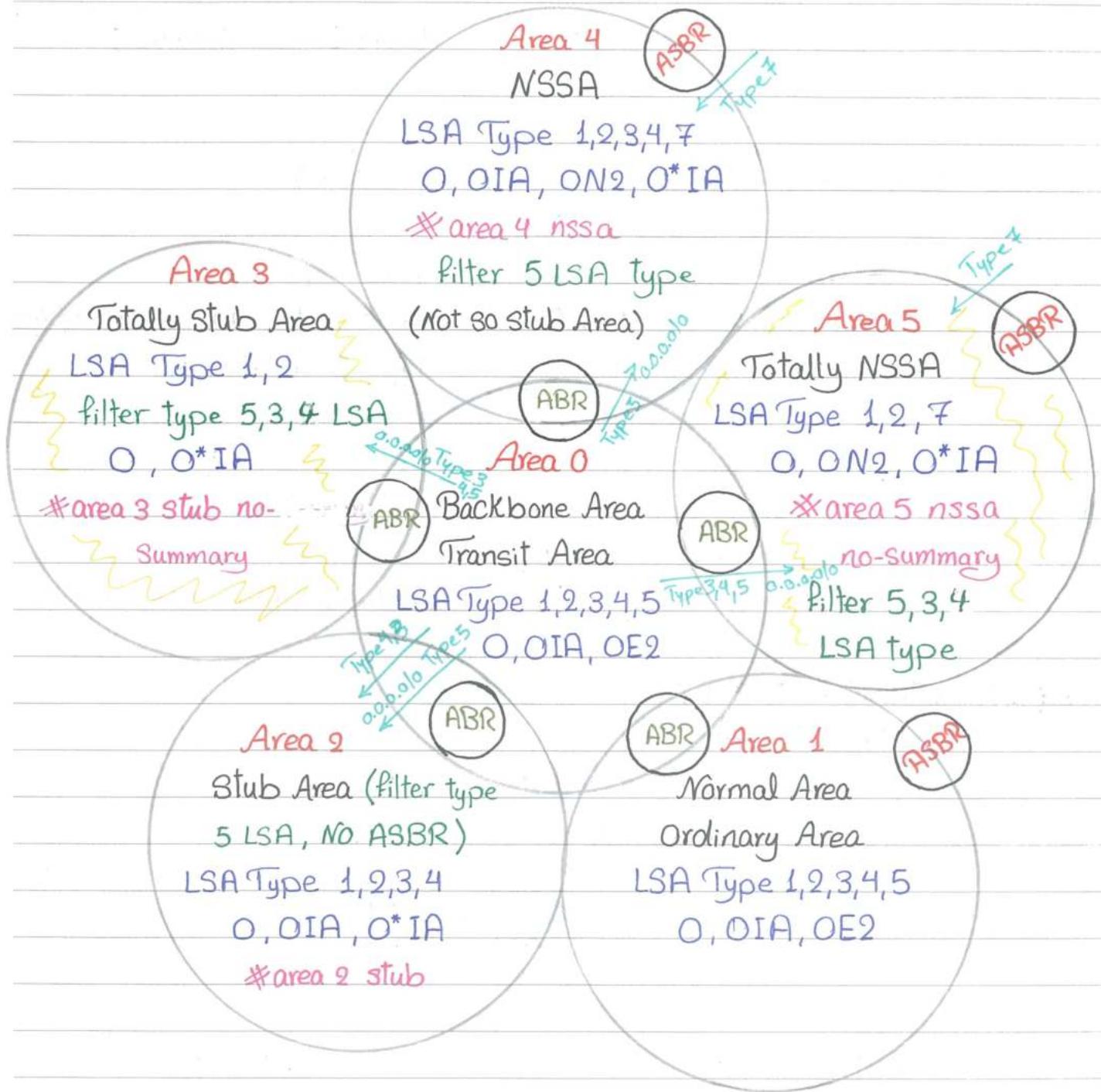
External

OE<sub>2</sub>, ON<sub>2</sub>, OE<sub>1</sub>, ON<sub>1</sub>

## Routing Table

## Session 3 Part 4

### Types of Areas



## Session 4 Part 1

### ISIS overview

why we use ISIS ?

open standard

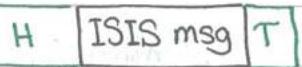
developed before OSPF

1. It is the most robust routing protocol in ISP industry.

AT&T, Flag, TATA, OBS, VIS, Noor ...

2. It is simpler, more stable and more scalable than OSPF.

3. It is well oriented for any new routed protocols (routed protocol independent) because it is layer 2.5 protocol (under layer 3).



why ISIS is not widely spread ?

1. Modern business prefer OSPF because OSPF has more feature (easier troubleshooting) ..

2. Lack of expertise knowing about ISIS (lack of educational resource).

ISIS is OSI model routing protocol:

TCP/IP

OSI

HOST

ES

end system

Router

IS

intermediate system

AS

Domain

ARP

ES-IS

IP protocol

CLNS

connection-less network service

IP header

CLNP

connection-less network protocol

IP address

NSAP

network service access point

Layer 2 address

SNPA

sub network point of attachment

DLCI, VPI/VCI ≡ HAC

## Session 4 Part 2

### OSI Addressing

identifies any system in the OSI model

NSAP address Contains :

1. Domain ID (Private Domain 49) ...
2. Area ID.
3. System ID.
4. NSEL (network selector). Function of device

oo For Routers

} In Hexadecimal.

Domain ID 1 byte	Area ID 1-12 byte	System ID 6 byte	NSEL 1 byte
---------------------	----------------------	---------------------	----------------

NSAP address for router  $\equiv$  NET address (network entity title) ...

(Config) # router isis

(Config-router) # net RID

Rules of IOS addressing:

1. Router should have only one NSAP = NET .

IOS address is assigned to the system, not to the interface ...

2. System id should be unique in same Area (It can be used again in another area). It must have the same length for all ISs & ESs within the domain ...

3. All routers in same area should have same area id.

## Session 4 Part 3

### OSI Routing levels

Level 0 Routing :

Between ES & IS . ex: ES-IS protocol .

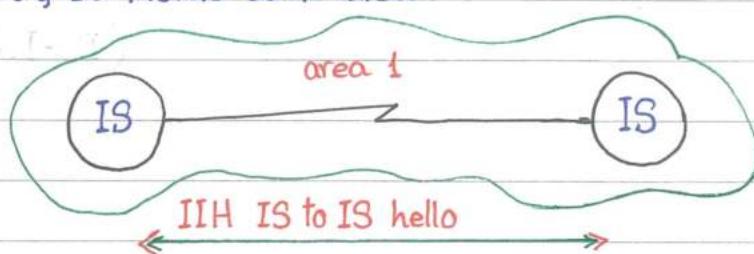


ESH end system hello (who is dg?)

ISH intermediate system hello  
my NET is ... , my SNPA is ...

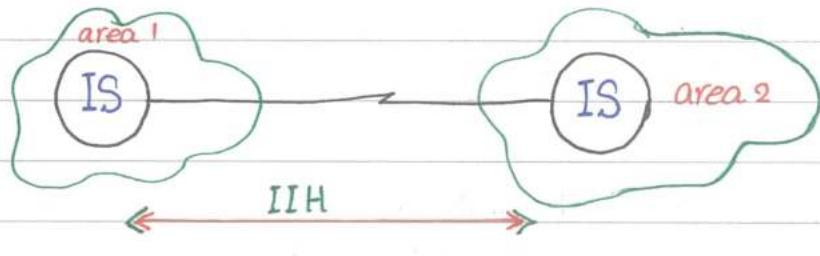
### Level 1 Routing:

Between IS & IS inside same area.



### Level 2 Routing:

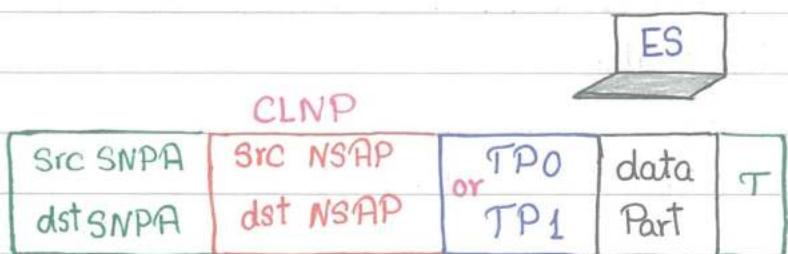
Between IS & IS in different areas.

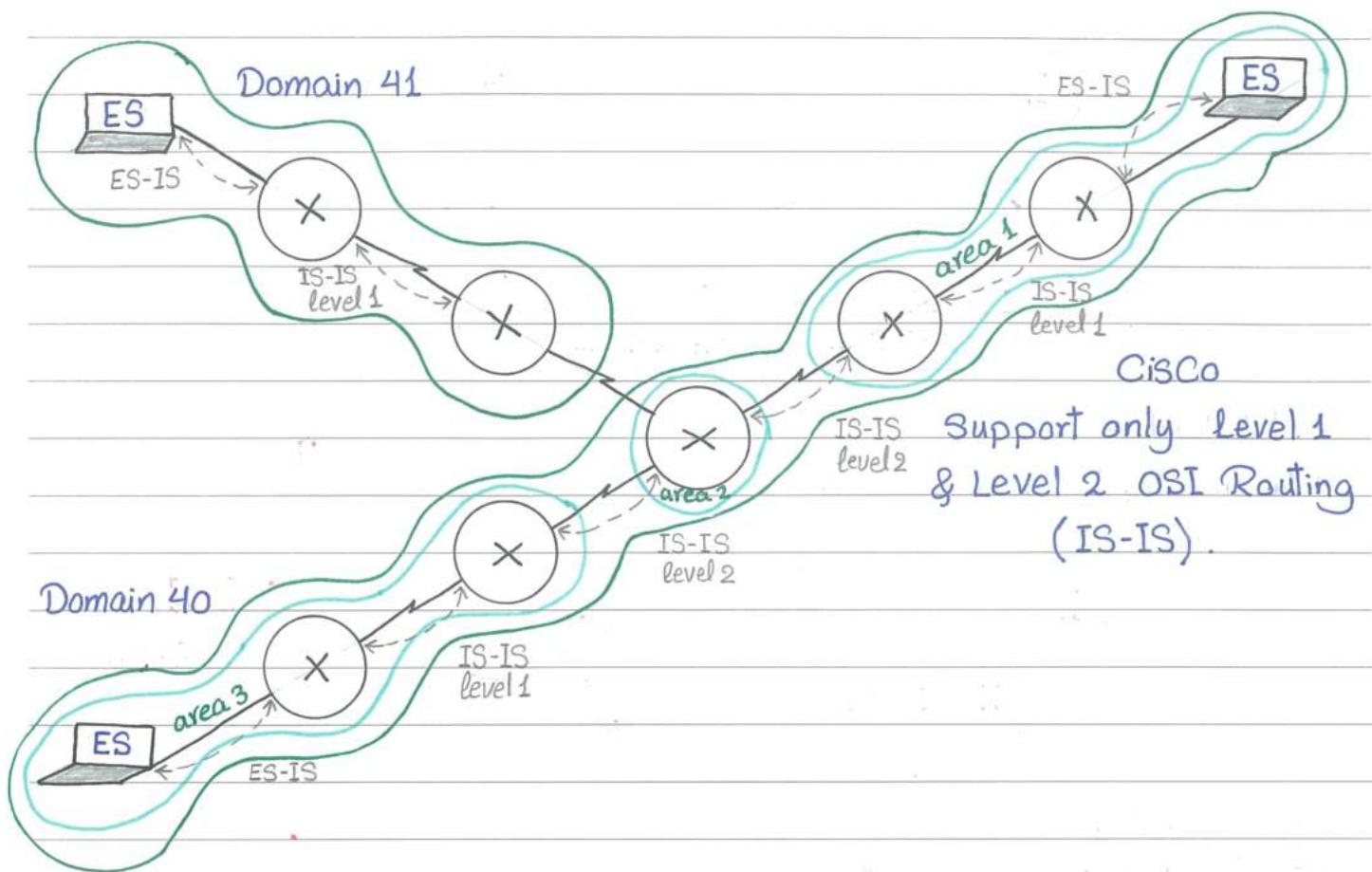


### Level 3 Routing:

Between IS & IS in different domains. It's not supported by Cisco.

ex: IDR (Inter Domain Routing Protocol).





### Session 4 Part 4 ISIS Features

1. It is Link State open standard protocol.
2. It uses Dijkstra Algorithm to calculate best path.
3. It supports Level 1 & Level 2 routing.

intra area      inter area

4. IS-IS router types:

\* Level 1 Router:

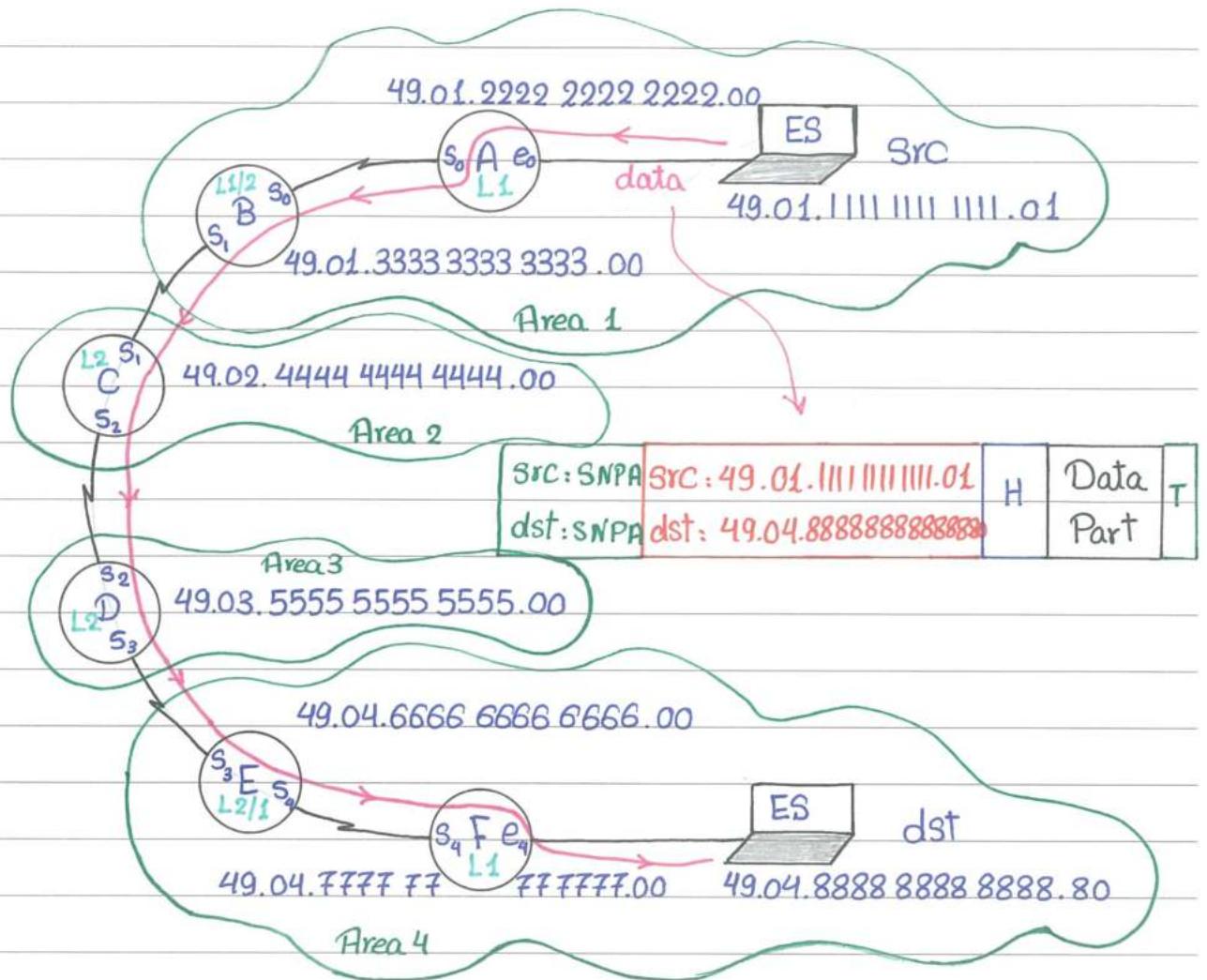
Form only L1 LSDB that contains only system id.

\* Level 2 Router:

Form only L2 LSDB that contains only area id.

\* Level 1 / Level 2 Router:

Form both L1 & L2 LSDB & send default route to internal routers.



- LSDB L1 "A":

1111 1111 1111	e0
2222 2222 2222	s0
3333 3333 3333	-
0.0.0.0	s0

- LSDB L1 "B":

1111 1111 1111	s0
2222 2222 2222	s0
3333 3333 3333	-
0.0.0.0	

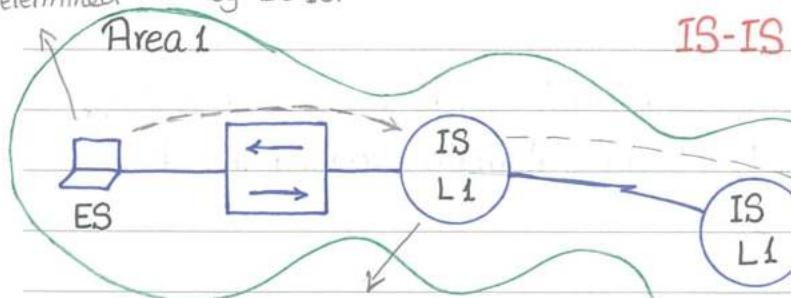
- LSDB L2 "B":

Area 1 -
Area 2 51
Area 3 51
Area 4 51

ISIS Routing Process

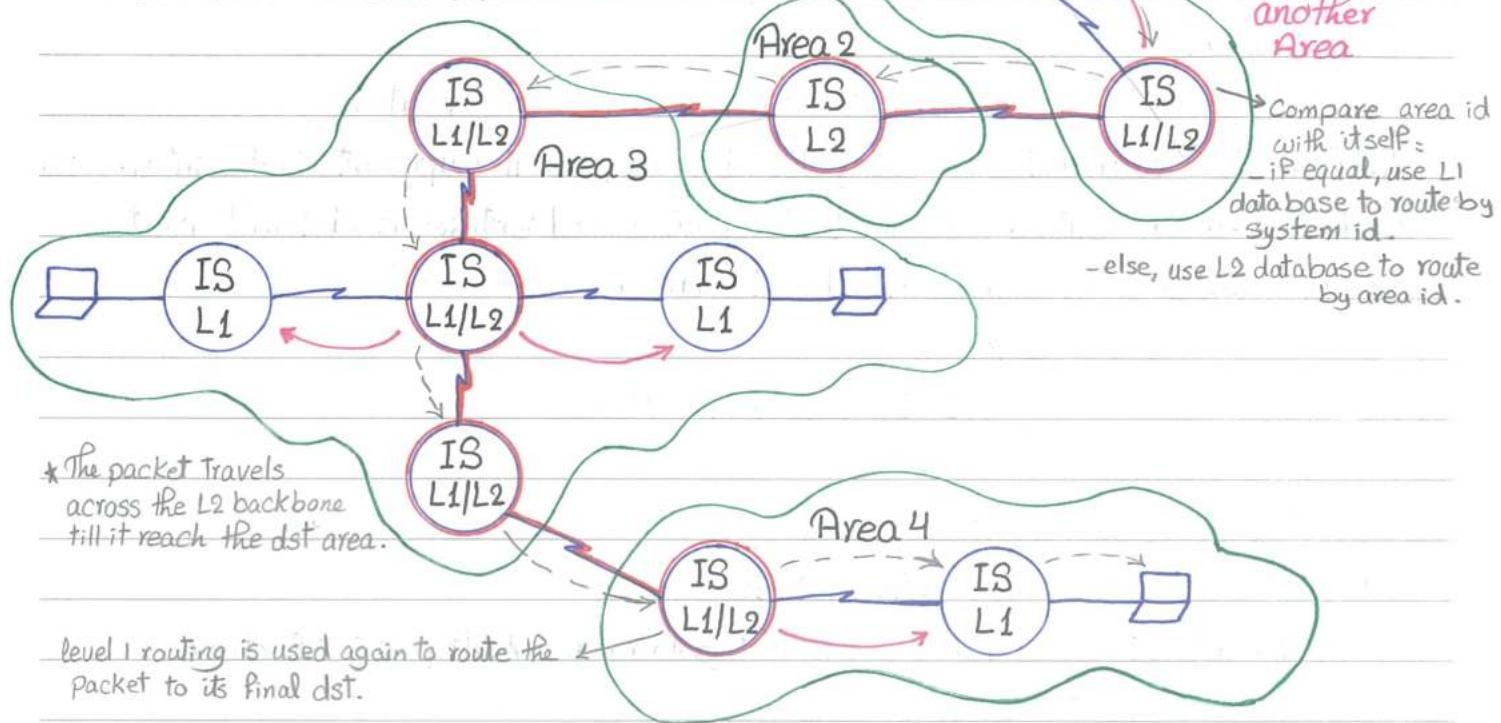
when an ES is required to send a packet to another ES, the packet goes to the nearest L1 router determined by ES-IS.

## Session 5 Part 1



Compare the area id of dst with its area id:

- if equal, it will use its L1 database to route by system ids.
- else route to the nearest L1/L2 router



. L1 LSDB contain: System id ... Routers make routing inside the area only.

. L2 LSDB contain: Area id ... Routers make routing between areas only.

### Notice:

- IS should only exist in one area.
- Each IS need unique NET address.
- IS can have only one NET.
- ISIS backbone is not an area, it is all L2 & L1/L2 routers. Continuous Path

⇒ IS-IS Support two routing levels : level 1 & level 2 routing.

Level 1 router : (like OSPF Internal Routers)

Router that build a L1 LSDB containing system ids only and router interface to reach these system id, because it make routing inside the area only.

Level 2 router : (like OSPF ABR)

Router that build a L2 LSDB containing area ids only and router interface to reach these area id, because it make routing between areas only.

Level 1/Level 2 router : (like OSPF backbone routers)

Router that build both L1 & L2 LSDB, so it support both intra-area and inter-area routing, each L1/2 router advertise a default route to all level 1 routers inside its area.

### IS IS Operation Modes

1. Point to Point mode.
2. BMA (Broadcast Multiple Access).
3. NBMA (Non BMA) : N/A in ISIS
  - 3.1 - Broadcast mode for Full mesh topology for multipoint interfaces (will simulate BMA).
  - 3.2 - Point-to-point mode for Partial mesh point-to-point sub-interface (will simulate point-to-point).

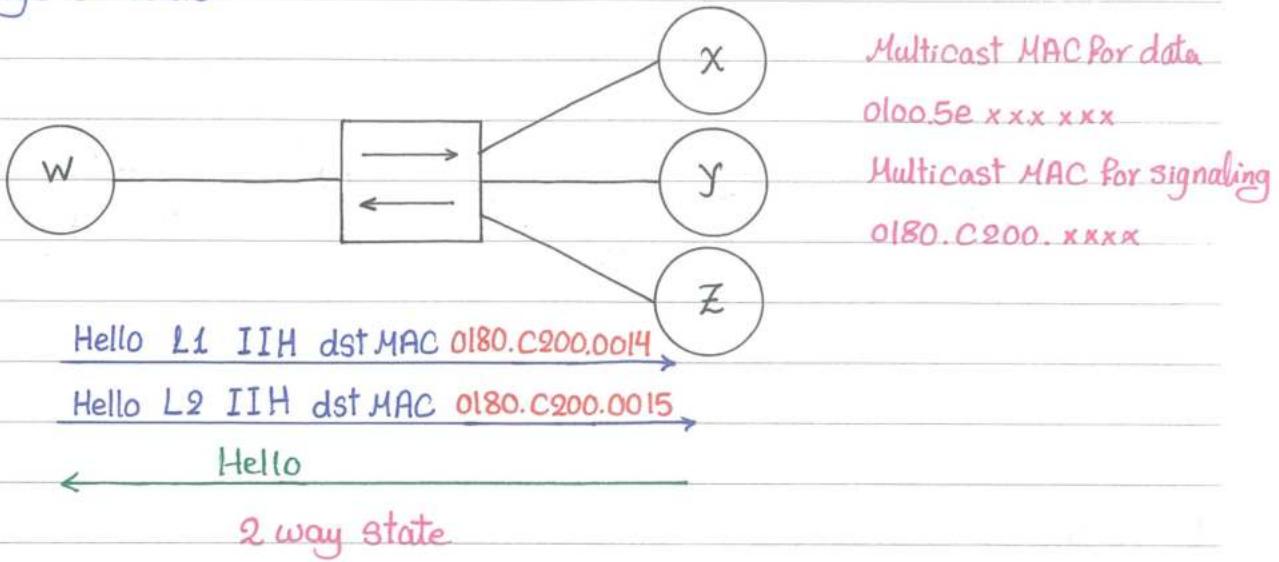
### ISIS Operation

② Create ISIS router ID (NET) :

(Config) # router isis

(Config-router) # net domainid.areaid.Systemid.00

## ① Neighbor Discovery: exchange of hello



⇒ Only one neighborship conditions :

- Same process id.

⇒ Hello interval :

- Hello every 10 sec except DIS (Designated IS) hello every 3.3 sec.

Session 5 Part 2

② Elect DIS called pseudo node:

data link layer, who

- DIS is IS having highest priority (0-127) default 64.
- Then IS having highest SNPA (MAC) address.

All other routers consider themselves Backup DIS.

valid for OSPF only.

DR welcome newcomer & @ change routes send changes to it before others.

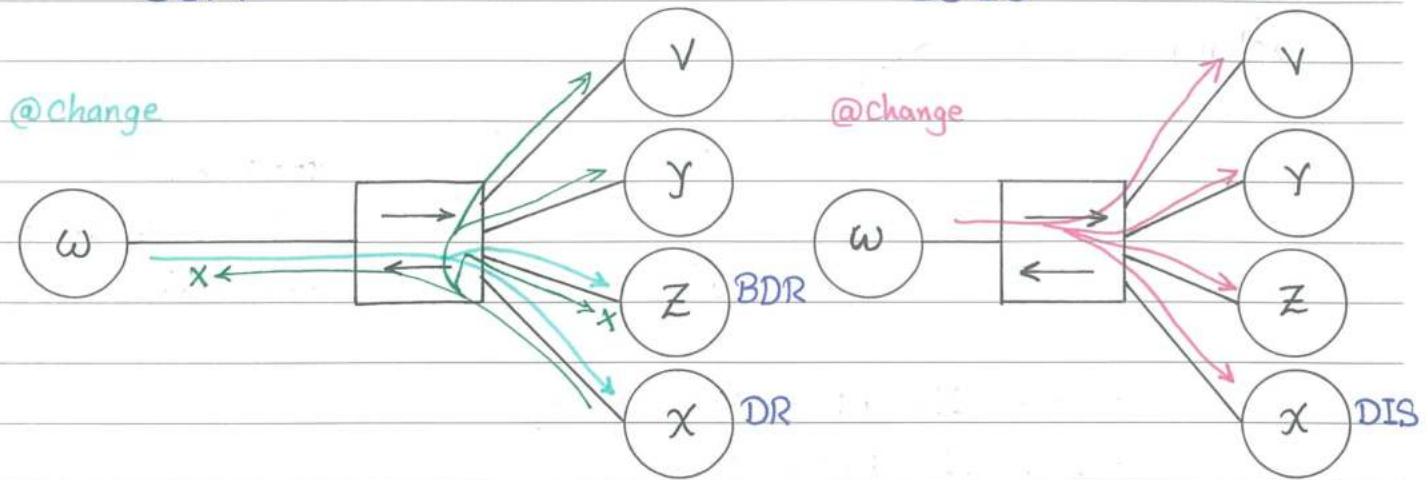
DIS welcome newcomer only & @ change, router that feels the change  
Send to all ISs.

OSPF

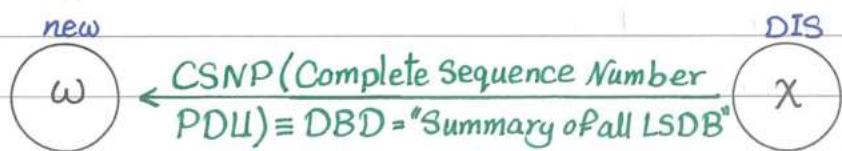
IS-IS

@change

@change



③ Routes Discovery:  
exchange of LSDB



PSNP (Partial Sequence Number PDU)  
≡ LSR "I need details about part"

LSP (Link State PDU)  
≡ LSU "here are all details" Ack

LSP contain full information for certain parts of LSDB described in TLV (Type/Length/Value)

TLV      Type

Area Id      1

IS System Id      2

ES System Id      3

IPv4 Internal network 128

IPv4 External network 130

- Recommended maximum number of routers per area = 1000.
- LSP refreshment is done every 15 minutes, maximum age time for a LSP entry in database is 20 minutes.

### Session 5 Part 4 Configuration

1. Enable TCP/IP & OSI Router Process :

(Config) # ip routing

or (Config) # CLNS routing

2. Create ISIS routing process :

(Config) # router isis [ Tag ]

default 0

(Config-router) # net RID

3. Active ISIS on interface :

(Config) # interface \_\_\_\_\_

(Config-if) # ip router isis

OR

(Config-if) # clns router isis

4. Choose router Level :

(Config-router) # is-type { level-1 / level-1-2 / level-2 only }  
default

5. Choose isis metric : default:10

(Config-if) # isis metric metric [ delay-metric [ expense-metric [ errormetric ] ] ]  
{ level-1 / level-2 } OR

(Config-router) # metric default-value { level-1 / level-2 }

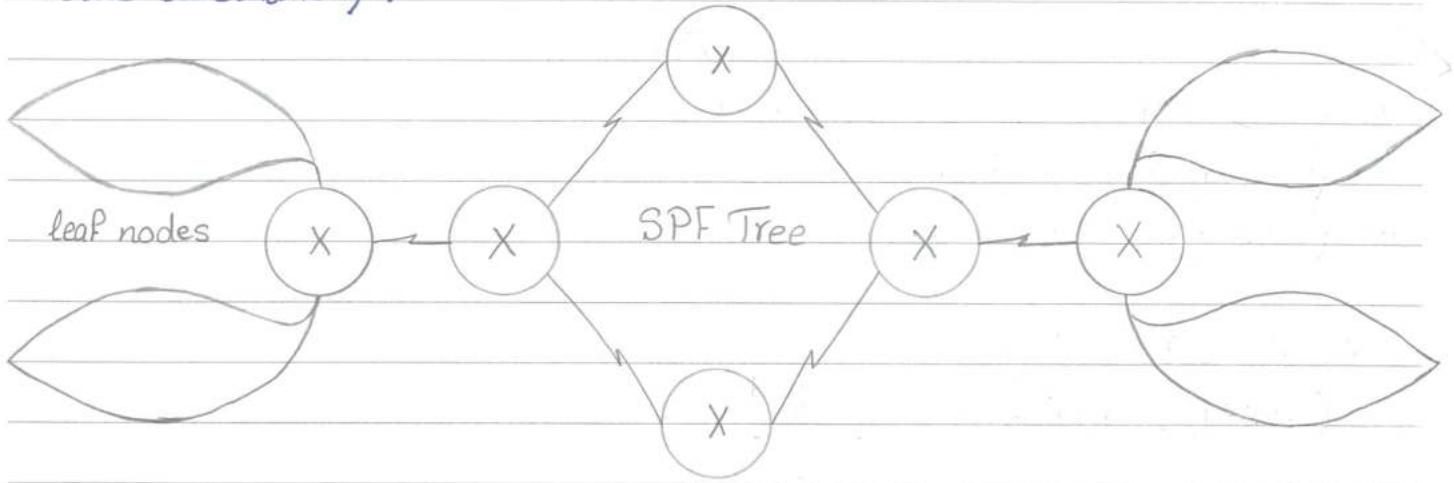
# Show isis route → L1 LSDB

# Show clns route → L2 LSDB

#### ④ Form Routing Table:

Router will run Dijkstra Algorithm to calculate the best path (Best to main nodes = IS).

ISIS consider IP networks a leaf nodes in a tree & use PRC (Partial Route Calculation).



### Session 5 Part 3

#### Integrated IS-IS CLC's

- Link state routing standard protocol. Running on OSI model and now on TCP/IP model.
- Send updates using multicast MAC 0180.C200.0014 and 0180.C200.0015 and using multicast IP 224.0.0.19, 224.0.0.20 and 224.0.0.21.  
L1 update, L2 update and any update
- Classless protocol.
- Support manual summarization only.  
No auto Summary available.
- Symbol in routing table "iL1" & "iL2"  
intra area networks & inter area networks
- Admin. Distance = 115 for TCP/IP model, = 110 for OSI model.
- Metric by default is called default & its default is 10/link.
- IS IS by configuration can add others metric (error, delay, expense).
- More scalable then OSPF (ISIS backbone is chain of L2 & L1/L2 routes).
- ISIS is less CPU intensive then OSPF (use PRC to calculate sub network changes).

## Session 6 Part 1

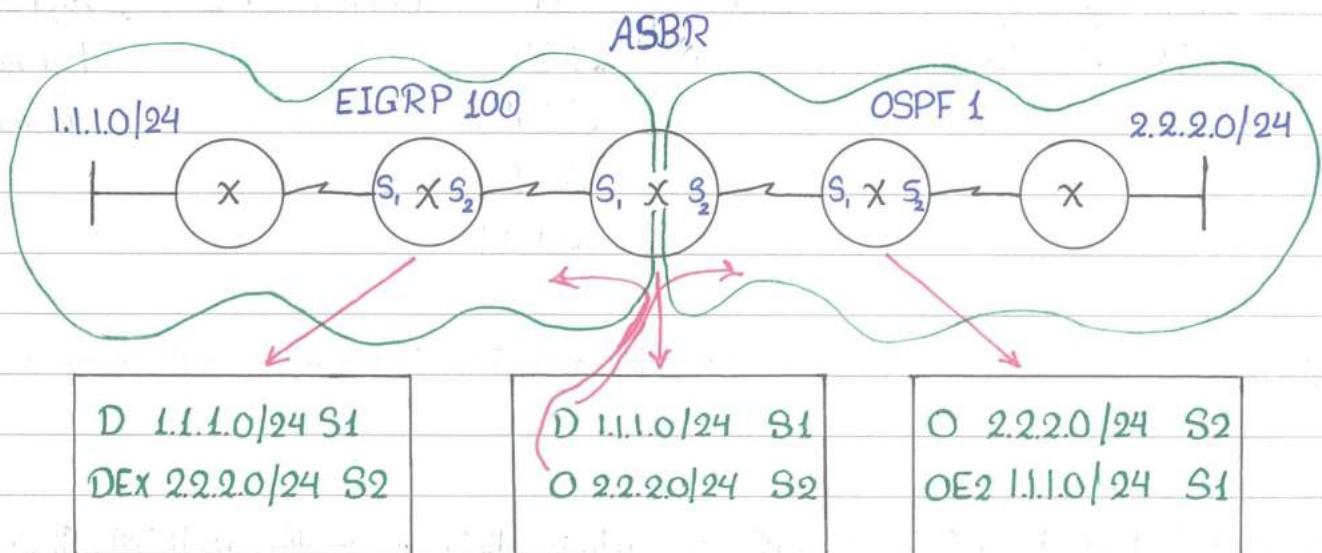
### Manipulating Multiple Routing Protocols (Redistribution)

why we may need multiple routing protocols?

A. Boundary between ASs.

B. Different customer requirements that may differ then provider requirements.

C. Redistribute routing protocols in MP-BGP (L3 VPN=VRF).



ASBR(Config)# router OSPF 1

ASBR(Config-router)# redistribute eigrp 100 [options]

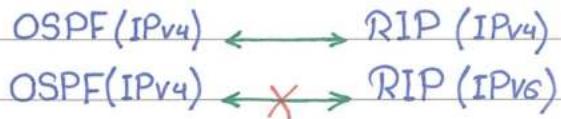
ASBR(Config)# router eigrp 100

ASBR(Config-router)# redistribute ospf 1 [options]

## Session 6 Part 2

### Considerations while redistribution

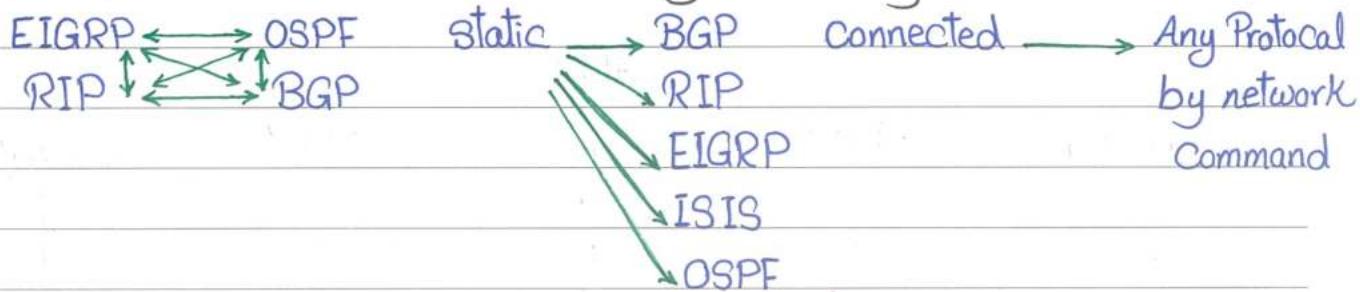
Redistribution can be performed between any two protocols under, same routed protocol.



- Auto redistribution.



Redistribution can be done between any two routing methods.



→ when implementing redistributions the redistributed routes will obey the new domain rules:

- Administrative Distance :

The redistributed protocol inherit the admin. distance of the redistributing protocol.

OSPF:

(Config-router) # distance ospf  
external \_ inter-area \_ intra-area \_

EIGRP:

(Config-router) # distance eigrp \_ in. ex.

	Cisco	Juniper	HP/Huawei
Protocol	Admin Distance	Admin Distance	Admin distance
Connected	0	0	0
static	0, 1	5	
RIP	120	100	
OSPF	110	10, 150	internal, external
EIGRP	90, 170, 5	N/A	internal summary
ISIS	115	15, 18, 160, 165	external
BGP	20, 200	170	255

BGP:

(Config-router) ~~#~~ distance bgp \_\_\_\_\_  
internal external

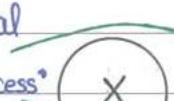
All routing protocols:

(Config-router) ~~#~~ distance \_\_\_\_\_ [acl ~~\*~~]  
Network

### Session 6 Part 3

- Metric:

Any protocol  RIP, EIGRP, IGRP metric  $\infty$

Any protocol  OSPF "other process" OSPF cost 20, OE2, no subnets allowed.  
BGP  Cost 1, OE2, no subnets allowed.

Any protocol  ISIS metric = 0, iL2

(Config-router) ~~#~~ redistribute Protocol 2

metric seed metric initial metric for redistributed routes  
match {internal | external 1 | external 2}  
ospf only {0, OIA or OE1 or OE2  
Optional metric-type {type-1 | type-2}  
subnets consider subnets  
{Level-1 | Level-2} used for isis only  
route-map name use route filter

## Session 6 Part 4 Route-Filters

Distribute List :

It is a filter used to permit or deny

- Incoming Updates.
- Outgoing Updates.
- Redistributed Updates.

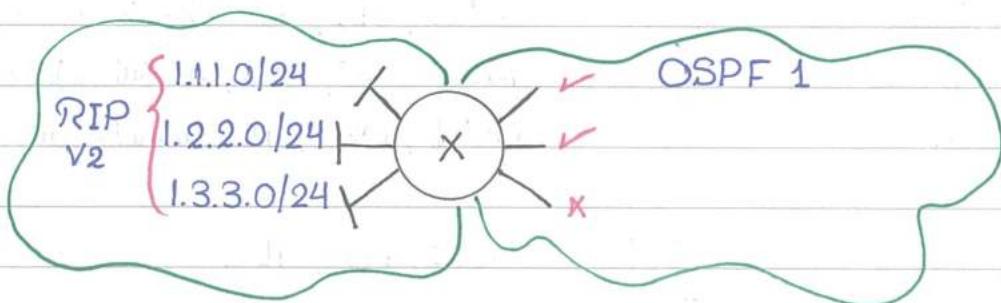
Configuration :-

(Config) # router Protocol 1

(Config-router) # distribute-list ACL# {  
    IN interface  
    Out interface  
    Out Protocol 2

It is not available with  
Link state protocols

ex:



(Config) # access-list 64 deny 1.3.3.0

(Config) # access-list 64 permit any

OR

(Config) # access-list 64 permit 1.1.1.0

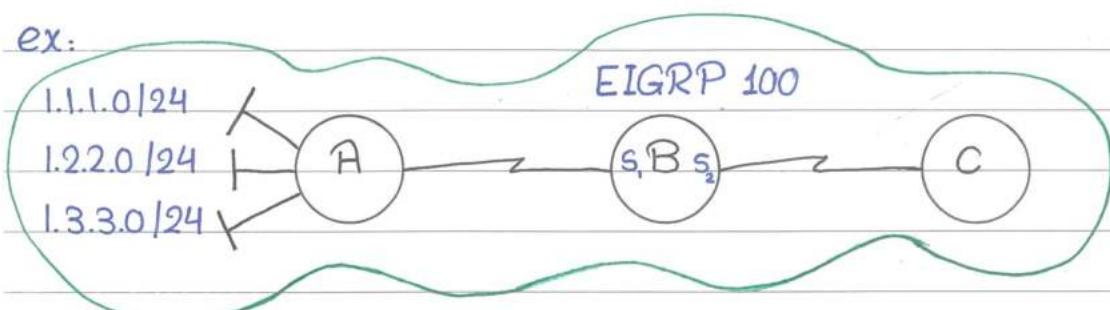
(Config) # access-list 64 permit 1.2.2.0

(Config) # router OSPF 1

(Config-router) # redistribute rip subnets

(Config-router) # distribute 64 out rip

ex:



we need B to get all networks but advertise only 1.1.1.0/24 & 1.2.2.0/24 to router C ...

(Config) # router eigrp 100

(Config-router) # distribute-list 64 out S2

Distribute List:

with ACL:

(Config-router) # distribute-list acl {  
    IN  
    out  
    Out}

with Prefix-list:

(Config-router) # distribute-list prefix-list {  
    IN  
    name {  
        Out  
        Out}}

with route-map:

(Config-router) # distribute-list route-map tag {  
    IN  
    out  
    Out}

## Session 6 Part 5

Prefix-list:

default 5, increment by 5

(Config) # ip prefix-list name [seq \*] {permit|deny} Prefix

Prefix length

[Le — ]

less than or equal

[ge — ]

great than or equal

ex: 192.168.1.0/24

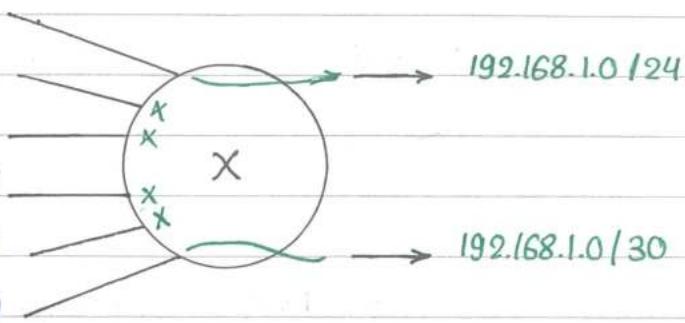
192.168.1.0/25

192.168.1.0/26

192.168.1.0/27

192.168.1.0/29

192.168.1.0/30



(Config) # ip prefix-list answer deny 192.168.1.0/25 le 29

(Config) # ip prefix-list answer permit 0.0.0.0/0 le 32 any

OR

(Config) # ip prefix-list answer permit 192.168.1.0/24

(Config) # ip prefix-list answer permit 192.168.1.0/30

### Session 6 Part 6

- Route-map:

It is a sophisticated ACL used to permit, deny & modify.

(Config) # route-map name {permit/deny} [Seq #] default 10

not increment

or (Config-route-map) # match ip address acl#

(Config-route-map) # match ip address prefix-list name

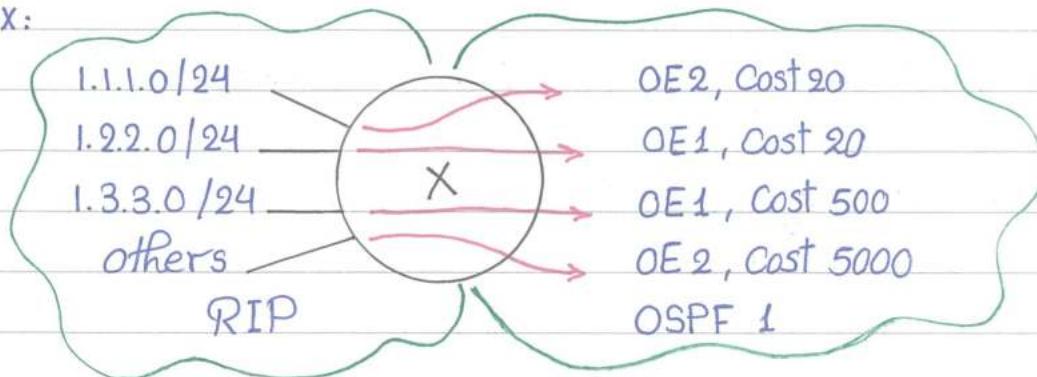
metric, metric-type, interface, ip of next-hop, tag

(Config-route-map) # set metric \*

(Config-route-map) # set metric-type {type-1 | type-2}

(Config-route-map) # set level {1 | 2}

ex:



(Config) # router ospf 1

(Config-router) # redistribute rip subnets route-map nabil

OR

(Config-router) # distribute-list route-map nabil

(Config) # access-list 1 permit 1.1.1.0

(Config) # ip prefix-list 2 permit 1.2.2.0 /24

(Config) # access-list 3 permit 1.3.3.0

(Config) # route-map nabil permit

(Config-route-map) # match ip address 1

(Config) # route-map nabil permit 20

(Config-route-map) # match ip address prefix-list 2

(Config-route-map) # set metric-type type-1

(Config) # route-map nabil permit 30

(Config-route-map) # match ip address 3

(Config-route-map) # set metric 500

(Config-route-map) # set metric-type type-1

(Config) # route-map nabil permit 40

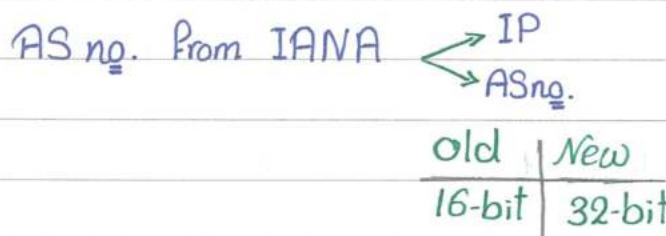
(Config-route-map) # set metric 5000

## Session 7 Part 1

### BGP overview

Border Gateway Protocol is an exterior gateway routing protocol that works between ASs

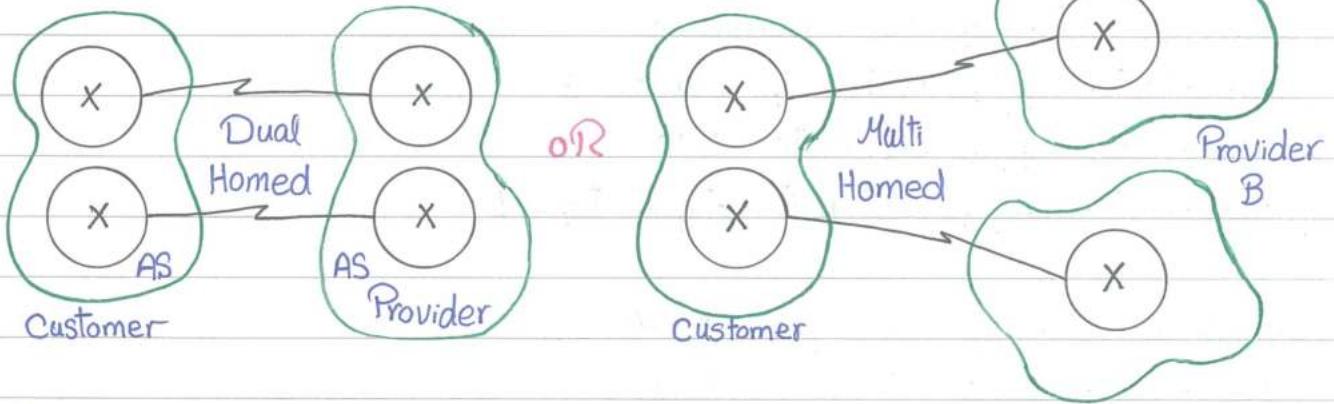
Group of devices under single technical administration (different routing policy)



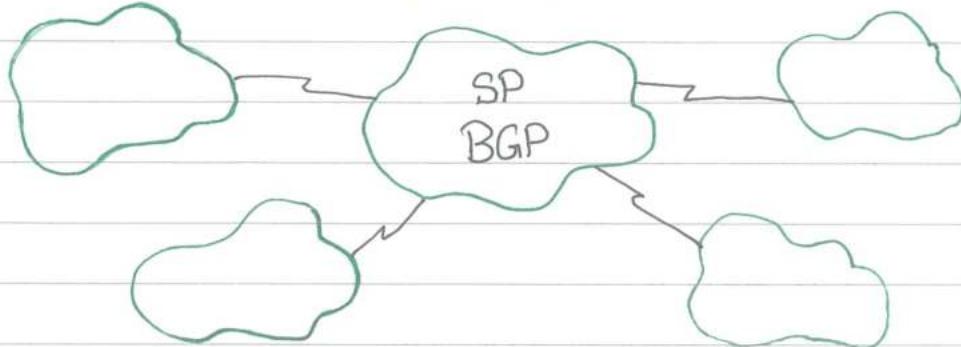
Private AS no. 64512 → 65535

when BGP is appropriate?

1. If many paths exist between ASs.



2. If AS is Transit AS (Service provider).



3. If dynamic routing policies are required.

→ The only acceptable dynamic Routing Protocol between ASs is BGP  
else use static

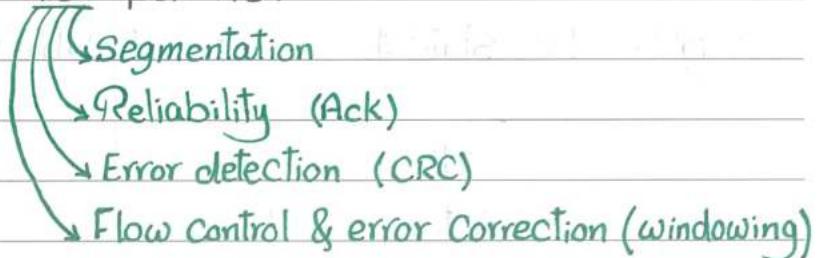
### Session 7 Part 2

#### BGP C/C's

1. It is a path vector EGP standard.

with details (all attributes), Advanced D.V

2. It is an application running over TCP port 179.



3. Classless protocol (FLSM, VLSM, Summarization, CIDR).

4. Optional Support Authentication.

5. Symbol in Routing table "B".

6. Admin. Distance = 20 ebgp , 200 ibgp.

loop prevention mechanism

in case of two ASBR

7. Use rich metric called attributes.

8. Use loop prevention mechanism.

- Normal split horizon: route learnt from neighbor should never be advertised back to same neighbor.

- BGP split horizon: route learnt from iBGP should never be advertised to another iBGP.

- AS path list is used to prevent loops between ASs (router will check its own AS no. in AS path, if my AS no. exists, router should discard update).

9. Support dynamic policy.

## Session 7 Part 3 BGP Tables

1. BGP Neighbor table :

Manual conf. for neighbors .

(Config-router) # neighbor ip of neighbor remote-as neighbor AS\*

# Show ip bgp summary

Neighbor IP	state	PFx rcd	Table ver	msg sent	msg received	AS	version
	idle						
	Active						
	Established						

2. BGP table = BGP topology = BGP database :

All learnt routes

Network	Mask	Attributes

3. Routing table :

Best routes , best attributes .

# Show ip route bgp

B 1.1.1.0/24 [20 / MED], ip of N.H.

## BGP Messages

1. Open msg: (startup @hello)

It Contains :

- RID .
- AS #.

RID BGP:

- Manually .
- Highest loopback ip .
- Highest Active physical interface ip .

2. Keep alive msg (periodic hello) :

Every 60 sec periodic hello .

Deadtime = invalid time = 180 sec .

3. Update msg :

It Contains NLRI (Network Layer Reachability Information) & Attributes .



4. Notification msg :

It exists if problem occurs between routers .

ex: Misconfiguration , Version mismatch .

5. Route refresh :

Used with clear Command .

## Session & Part 5

### BGP Operation

@startup:

(Config)# router bgp AS#

(Config-router) # neighbor ip-of-neighbor remote-as neighbor AS#

(Config-router) # address-family ipv4-unicast

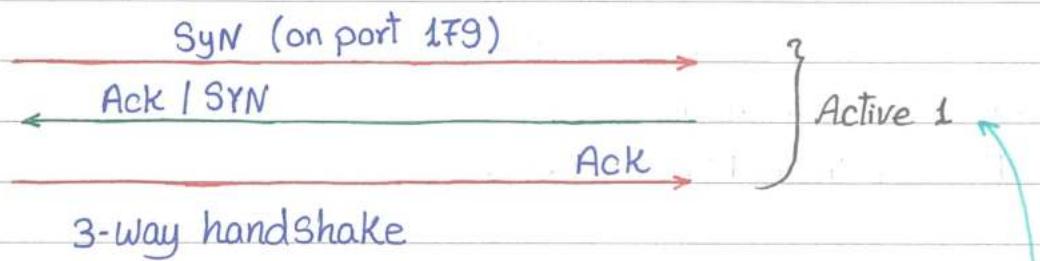
(Config-router-af) # neighbor ip-of-neighbor activate } exist by default for

} ipv4 address family



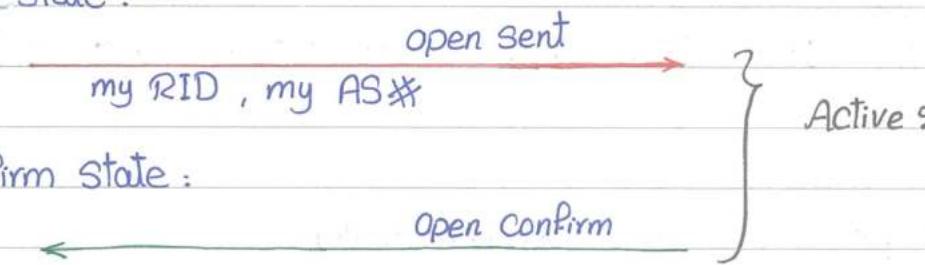
### 1. Neighbor Discovery:

- Idle State: No exchange yet  
No route in routing table for that neighbor.



### - Connect state:

### - Open Sent State:



### - Open Confirm state:

### - Established State:

- Neighbor is not running bgp.
- Neighbor running bgp with wrong peer.

TCP RST msg  
\* debug ip tcp transaction

- eBGP neighbor should be directly connected by default.

neighbor ip enable-connected-check , msgs TTL = 1 .

- iBGP neighbors may not be directly connected by default.

neighbor ip disable-connected-check , msgs TTL = 255 .

- TCP session destroy within 40 sec .

other neighbor doesn't have route back ... or... ACL exist .

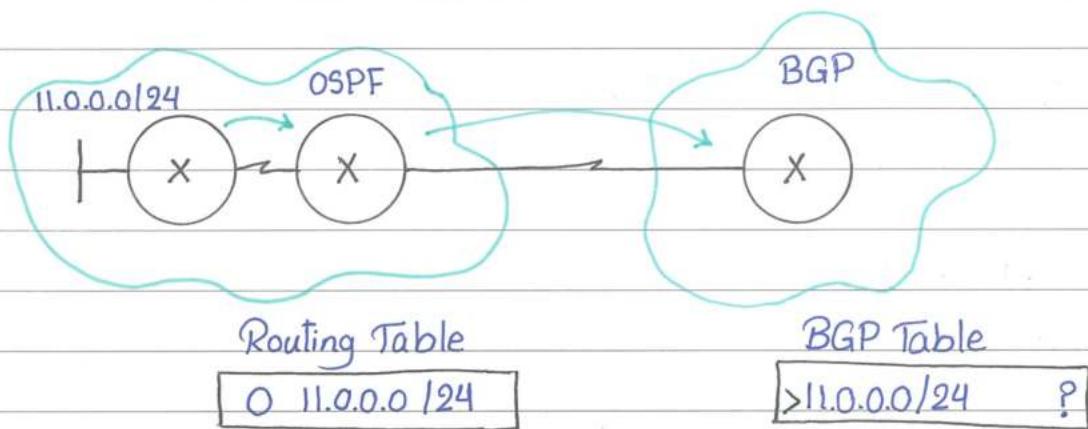
## Session 8 Part 1

### BGP Considerations

1. Advertising Network (Publishing Networks in BGP):  
 "Injecting routes in BGP Table"

1.1. Redistribute IGP into BGP:

Connected static IGP



`router bgp 1`

`redistribute IGP route-map x`

origin (attribute)

(?) incomplete

1.2. Network Command (Recommended):

`(config)* router bgp AS*`

`(Config-router)* network [mask ]`

- If no mask is specified, default masks is assumed.

- Injected network should be exactly matched in routing table by IGP (Synchronized).

origin (attribute)

(i) network (IGP)

⇒ Origin attribute: i(IGP) < e(EGP) < ?(incomplete)  
 least is best...

## Session 8 Part 2

### 2. Advertising Summary :

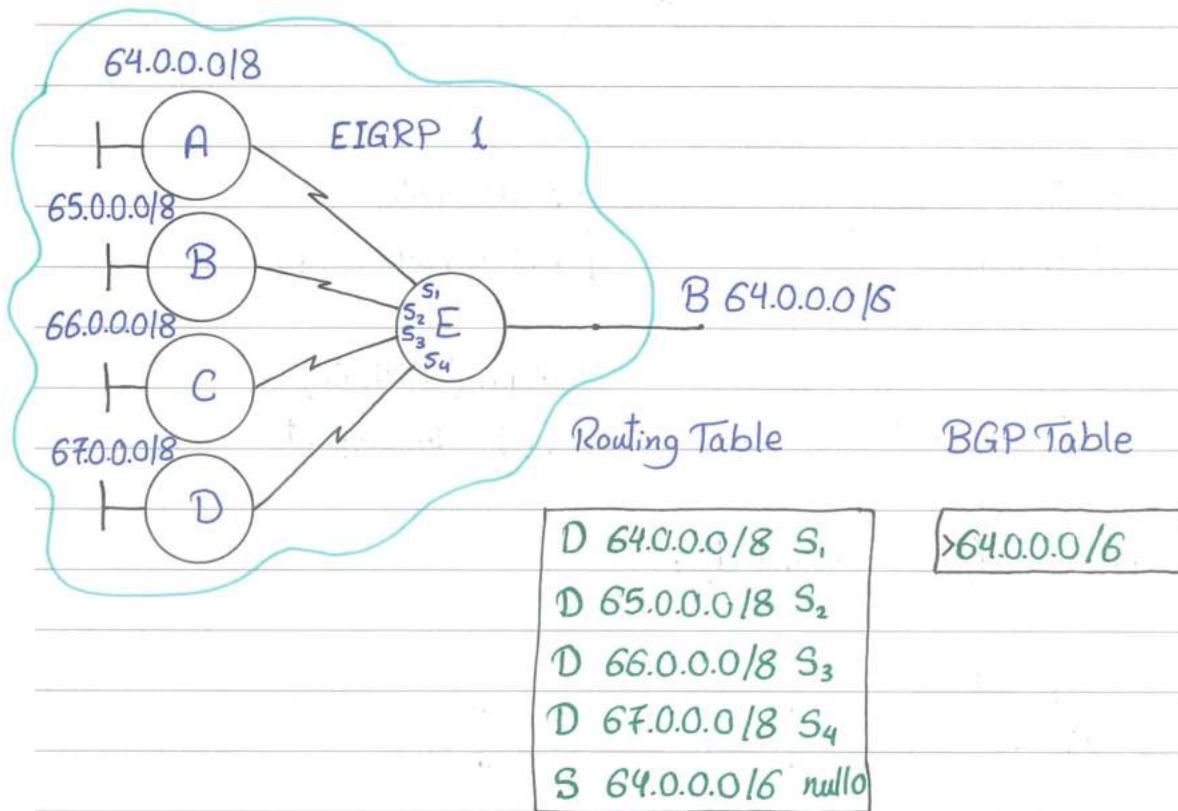
#### 2.1. Network Command + Static route :

(Config)# router bgp AS#

(Config-router)# network 64.0.0.0 mask 252.0.0.0

+

(Config)# ip route 64.0.0.0 252.0.0.0 null 0



#### 2.2. Aggregate address :

(Config)# router bgp AS#

(Config-router)# aggregate-address network mask [as-set] [summary-only]  
[suppress-list \*]

Suppress all

Specific routes

matching aggregate  
address

- Default will advertise aggregate route & specific routes.

- Condition: at least one specific route match the aggregate address should exist in BGP Table.

Routing Table:

D 64.0.0.0/8 S1
D 65.0.0.0/8 S2
D 66.0.0.0/8 S3
D 67.0.0.0/8 S4
B 64.0.0.0/6 null0

BGP Table:

>64.0.0.0/6	----->65.0.0.0/8	(Config-router) aggregate-address 64.0.0.0 252.0.0.0 Summary-only	* > 64.0.0.0/6
S > 65.0.0.0/8			valid Suppress

BGP Table:

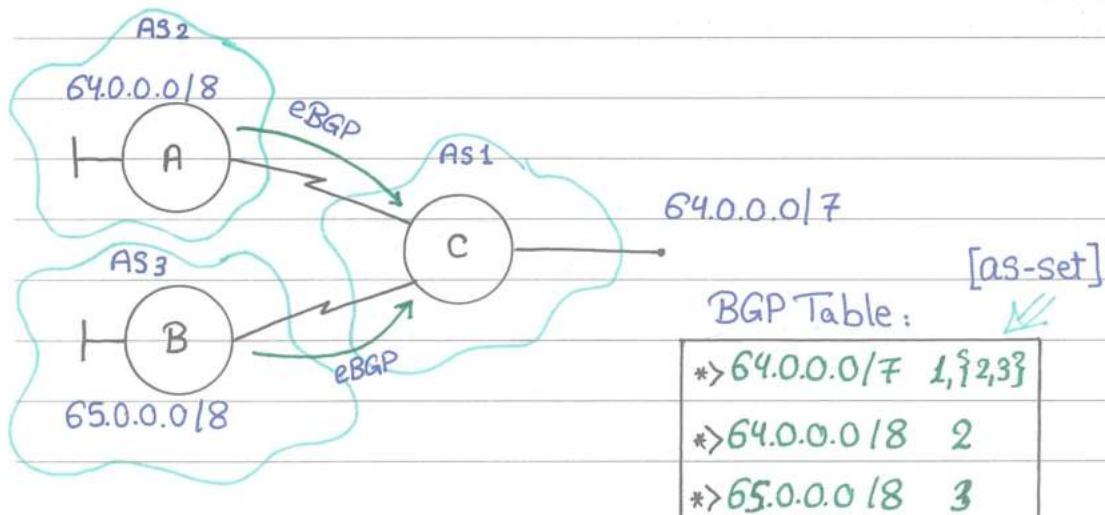
>64.0.0.0/6	(Config-router) aggregate-address 64.0.0.0 252.0.0.0 suppress-list 64	* > 64.0.0.0/6
>65.0.0.0/8		S > 65.0.0.0/8
>66.0.0.0/8		* > 66.0.0.0/8

(config) # access-list 64 permit 65.0.0.0

⇒ Attributes: origin , Aggregator , Atomic aggregate

(i) (AS\*, RID) (0, 1)

specific route aggregated route



## Session 8 Part 3

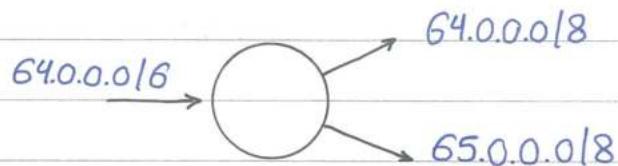
3. Advertising Specific routes under condition:

Conditional route injection

inject-map , exist-map

specific routes if summary exist

(Config-router) # bgp inject-map \_\_\_\_\_ exist-map \_\_\_\_\_  
advertise received



(Config-router) # bgp inject-map anwar exist-map Zico

(Config) # route-map anwar permit

(Config-route-map) # set ip address prefix-list 65

(Config-router-map) # set ip address prefix-list 66

(Config) # ip prefix-list 65 permit 64.0.0.0/8

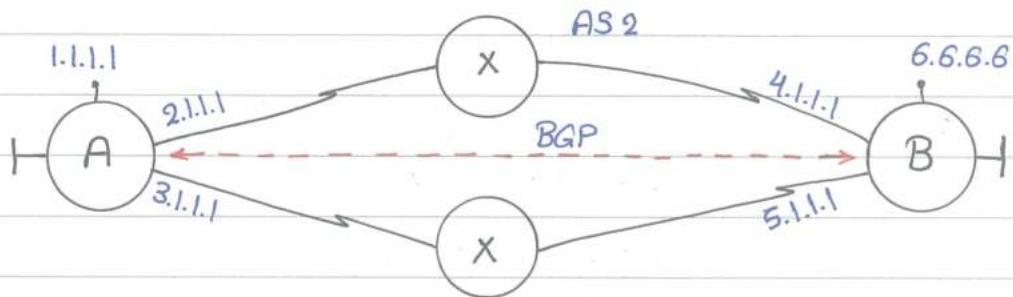
(Config) # ip prefix-list 66 permit 65.0.0.0/8

(Config) # route-map Zico permit

(Config-route-map) # match ip address prefix-list 64

(Config) # ip prefix-list 64 permit 64.0.0.0/16

4. Update Source behaviour:



(Config) # router bgp 2

A(Config-router) # neighbor 6.6.6.6 remote-as 2

A(Config-router) # neighbor 6.6.6.6 update-source loopback0

B(Config-router) # neighbor 1.1.1.1 remote-as 2

B(Config-router) # neighbor 1.1.1.1 update-source loopback0

## 5. Multihop behaviour:

- IBGP neighbor by default may not be directly connected.

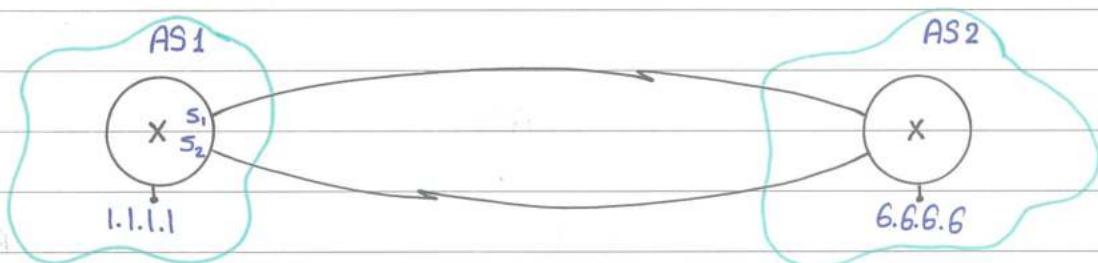
(Config-router) # neighbor \_\_\_\_\_ disable-connected-check

BGP msg TTL = 255 (by default).

- EBGP neighbor by default should be directly connected.

(Config-router) # neighbor \_\_\_\_\_ enable-connected-check

BGP msg TTL = 1 (by default).



(Config) # ip route 6.6.6.6 255.255.255.255 91

(Config) # ip route 6.6.6.6 255.255.255.255 92

(Config) # router bgp 1

(Config-router) # neighbor 6.6.6.6 remote-as 2

(Config-router) # neighbor 6.6.6.6 update-source lo0

(Config-router) # neighbor 6.6.6.6 disable-connected-check TTL = 1

OR

(Config-router) # neighbor 6.6.6.6 ebgp-multihop [ ] TTL = 255

OR

(Config-router) # neighbor 6.6.6.6 ttl-security-check 2 TTL = 255

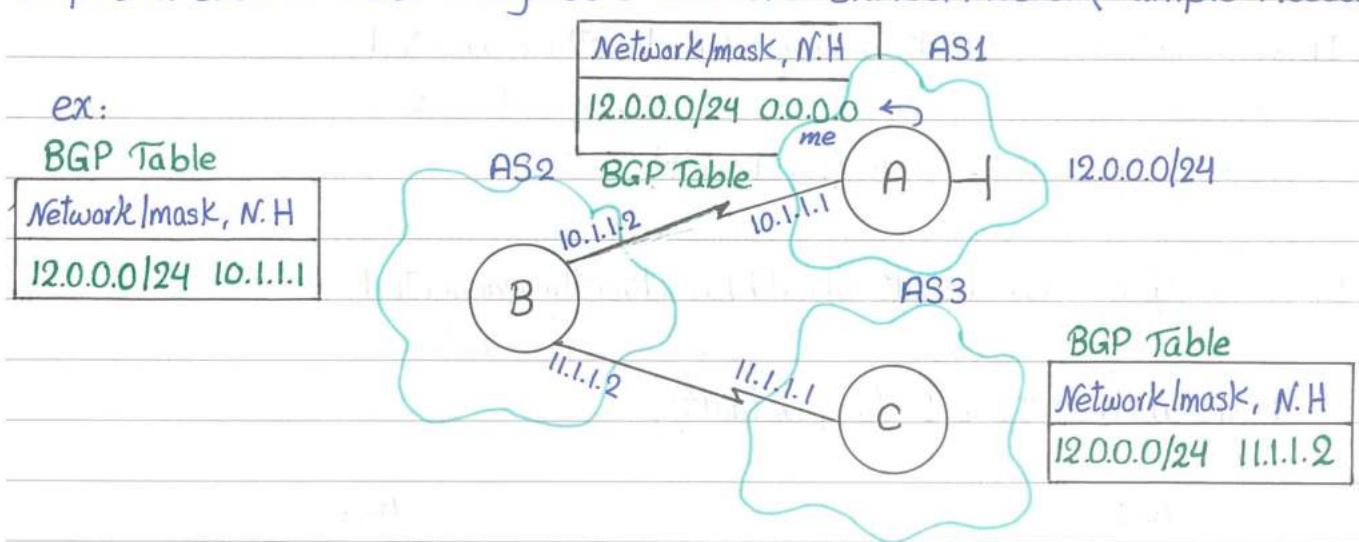
Check TTL = 255 - ( ) = 253 ...

## Session 9 Part 1

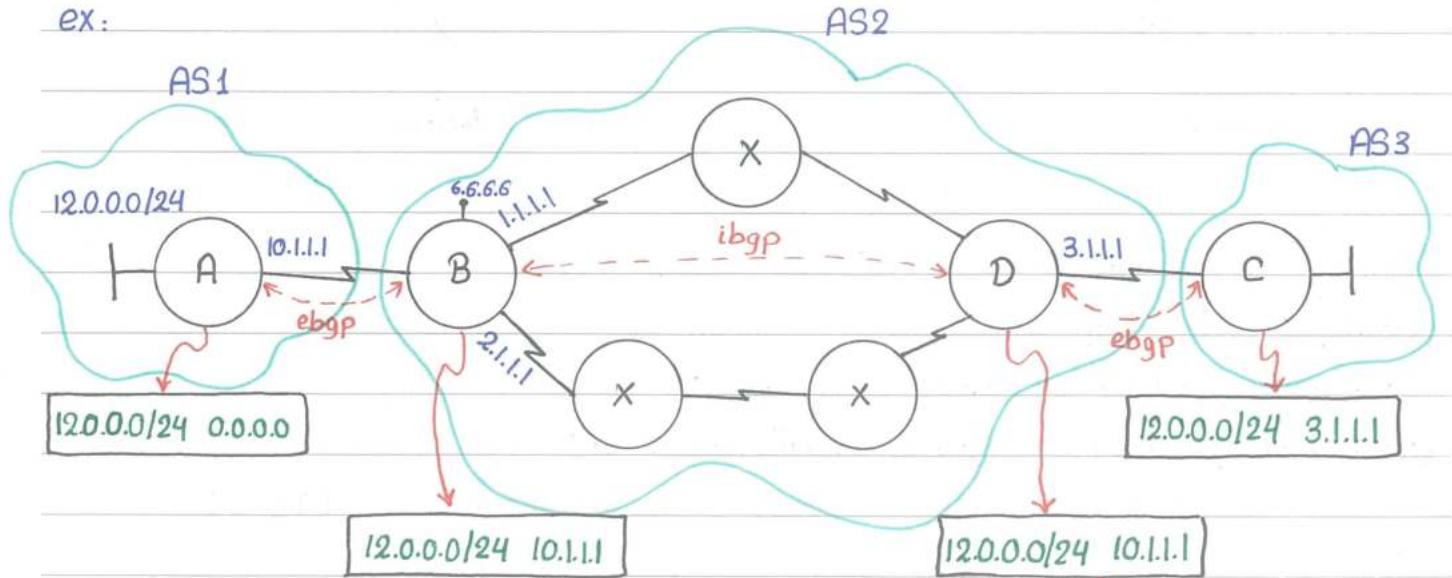
### BGP Consideration

5. Next hop behavior:

- BGP is an AS by AS routing protocol not router by router routing Protocol, so in BGP next hop is router in next AS.
- BGP next hop attribute is changed in eBGP sessions, while Kept Unchanged in iBGP sessions.
- Special case if eBGP neighbors are in a shared media (Multiple Access).



ex:



{ (Config) \*router bgp 2  
 (Config-router) \*neighbor D next-hop-self  
 To override that behaviour. ip of nei.

12.0.0.0/24 6.6.6.6

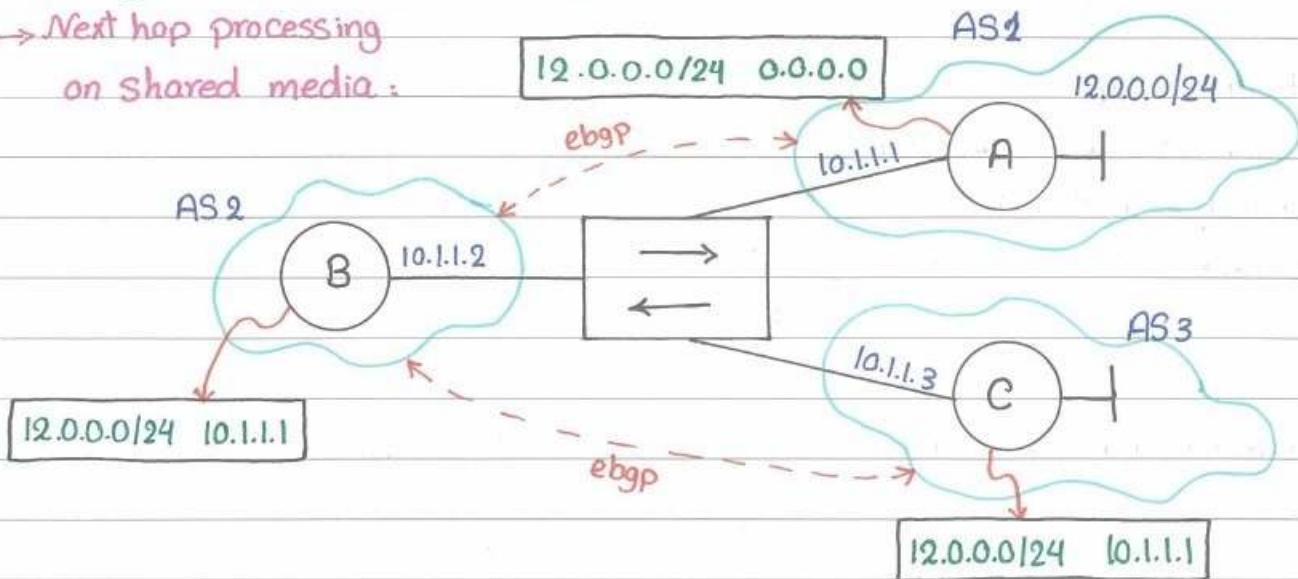
- Redistribute Connected into BGP.

(Config) # router OSPF 1

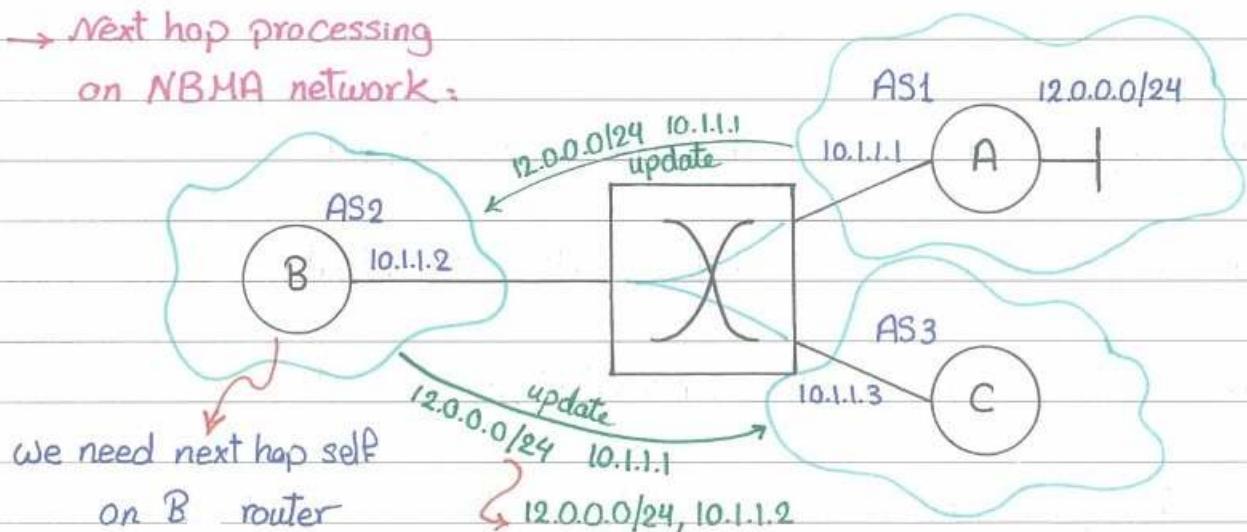
(Config-router) # redistribute connected

→ Next hop processing

on shared media:

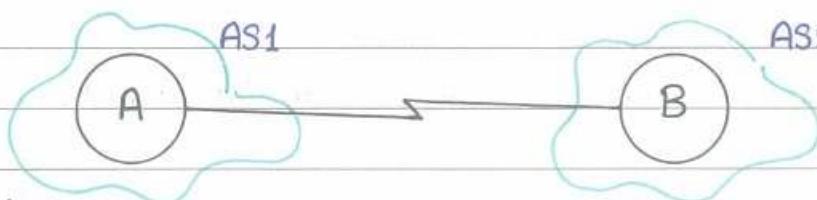


→ Next hop processing  
on NBMA network:



## Session 9 Part 2

6. Authentication:



(Config) # router bgp 1

(Config-router) # neighbor B password Cisco

Using Hashed method (MD5) ...

## 7. Peer Group :

→ Before peer group :

(Config) # router bgp 2

(Config-router) # neighbor ip<sup>1</sup> remote-as AS\*  
: ip 15

(Config-router) # neighbor ip<sup>1</sup> password \_\_\_\_\_  
: ip 15

(Config-router) # neighbor ip<sup>1</sup> update-source lo0  
: ip 15

(Config-router) # neighbor ip<sup>1</sup> ebgp-multihop ~  
: ip 15

(Config-router) # neighbor ip<sup>x</sup> route-map x in/out

(Config-router) # neighbor ip<sup>x</sup> distribute-list x in/out

(Config-router) # neighbor ip<sup>x</sup> prefix-list x in/out

(Config-router) # neighbor ip<sup>x</sup> next-hop-self

120 Command

→ peer group :

- Simpler configuration.

- Dynamic peer update (prepare one update for the peer group).

(Config) # router bgp 2

(Config-router) # neighbor WE peer group name peer-group

(Config-router) # neighbor ip 1  
                          : ip 15      peer-group WE

(Config-router) # neighbor WE remote-as 2

(Config-router) # neighbor WE password cisco

(Config-router) # neighbor WE timers

(Config-router) # neighbor WE update-source lo0

(Config-router) # neighbor WE route-map x in/out

(Config-router) # neighbor WE next-hop-self

(Config-router) # neighbor WE route-reflector-client

23 Command

#### → Peer Group Limitations :

- Neighbor can be only in one peer-group.
- Peer group cannot mix iBGP & eBGP neighbor.
- Neighbor in a peer group cannot have a special filter.

#### Session 9 Part 3

#### → Peer Template :

- Peer Session Template : Commands related to neighborship  
peer group for general commands that is not depending on certain  
address family "(Config-router-stmp)"

- Peer Policy Template : Commands related to updates  
peer group for specific commands that is under a specific address  
family "(Config-router-ptmp)"

(Config) # router bgp 2

(Config-router) # neighbor ip1 inherit peer-session A

(Config-router) # neighbor ip2 inherit peer-session B

(Config-router) # address-family ipv4-unicast

(Config-router-af) # neighbor ip1 inherit peer-policy C

(Config-router) # template peer-session A

(Config-router-stmp) # remote-as 1

(Config-router-stmp) # password cisco

(Config-router-stmp) # update-source lo0

(Config-router-stmp) # inherit peer-session B

(Config-router) # template peer-session B

(Config-router-stmp) # ebgp-multihop

(Config-router) # template peer-policy C

(Config-router-ptmp) # weight \_\_\_\_\_

(Config-router-ptmp) # route-map \_\_\_\_\_ in/out

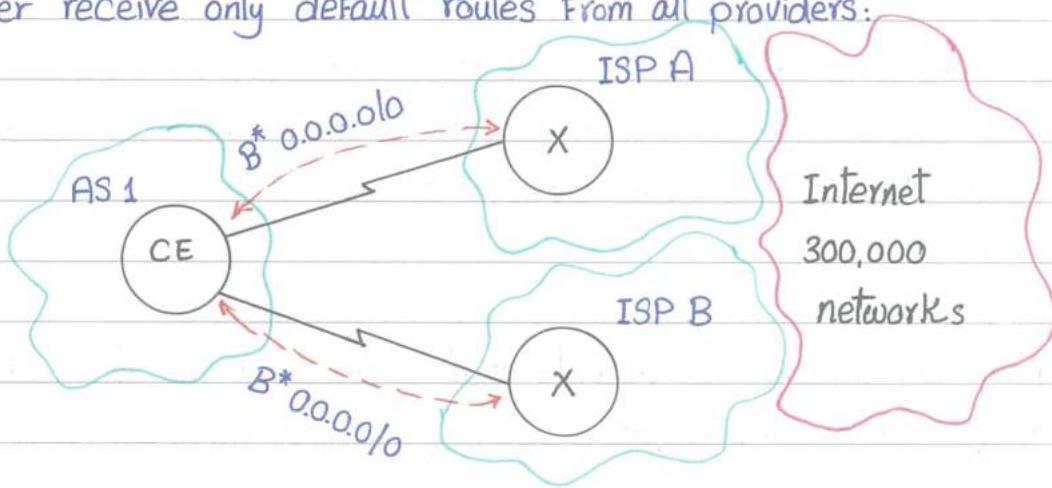
(Config-router-ptmp) # route-reflector-client

(Config-router-ptmp) # next-hop-self

#### Session 9 Part 4

##### 8. Multihoming:

- Customer receive only default routes from all providers:



ISP(Config) # router bgp x

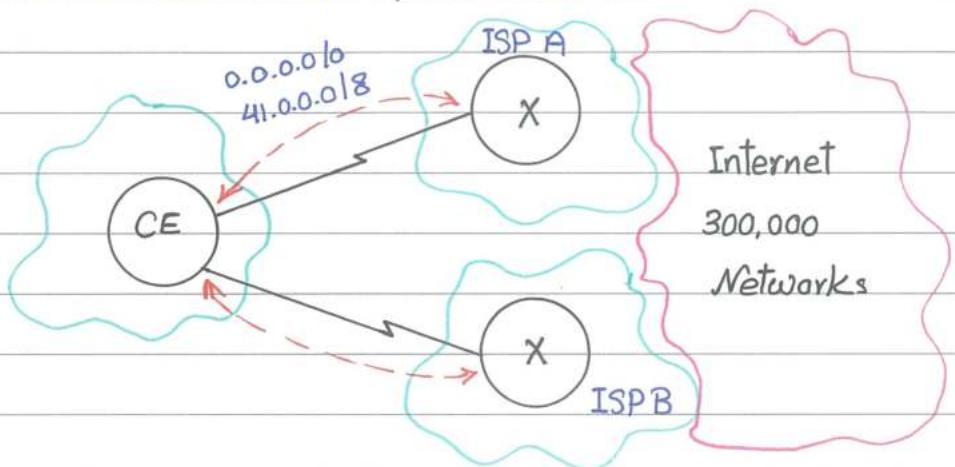
ISP(Config-router) # neighbor CE remote-as y

ISP(Config-router) # neighbor CE default-originate

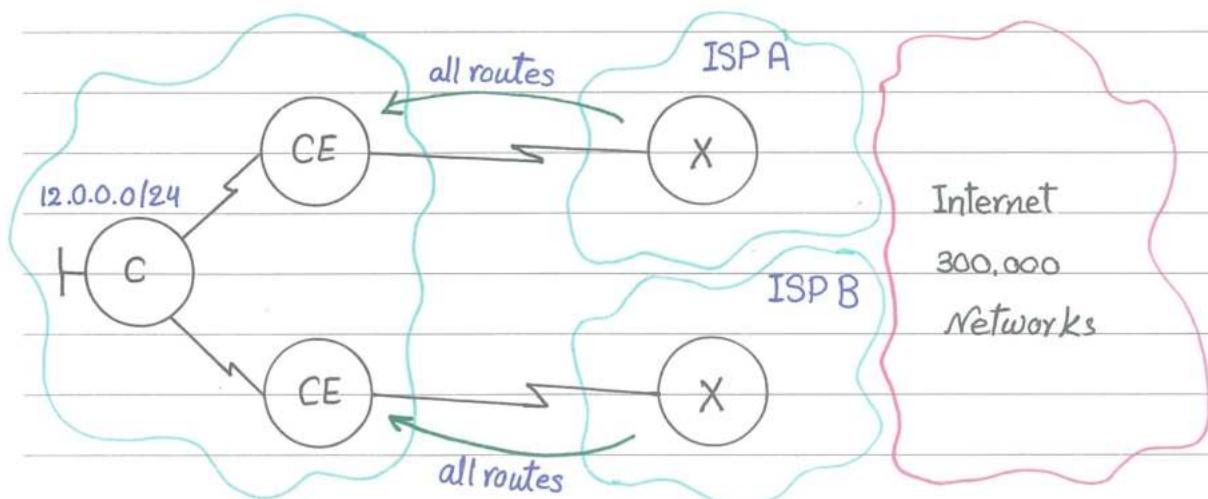
ISP(Config-router) # neighbor CE prefix-list x out

ISP(Config) # ip prefix-list x permit 0.0.0.0/0

- Customer receive default routes + SP specific routes:

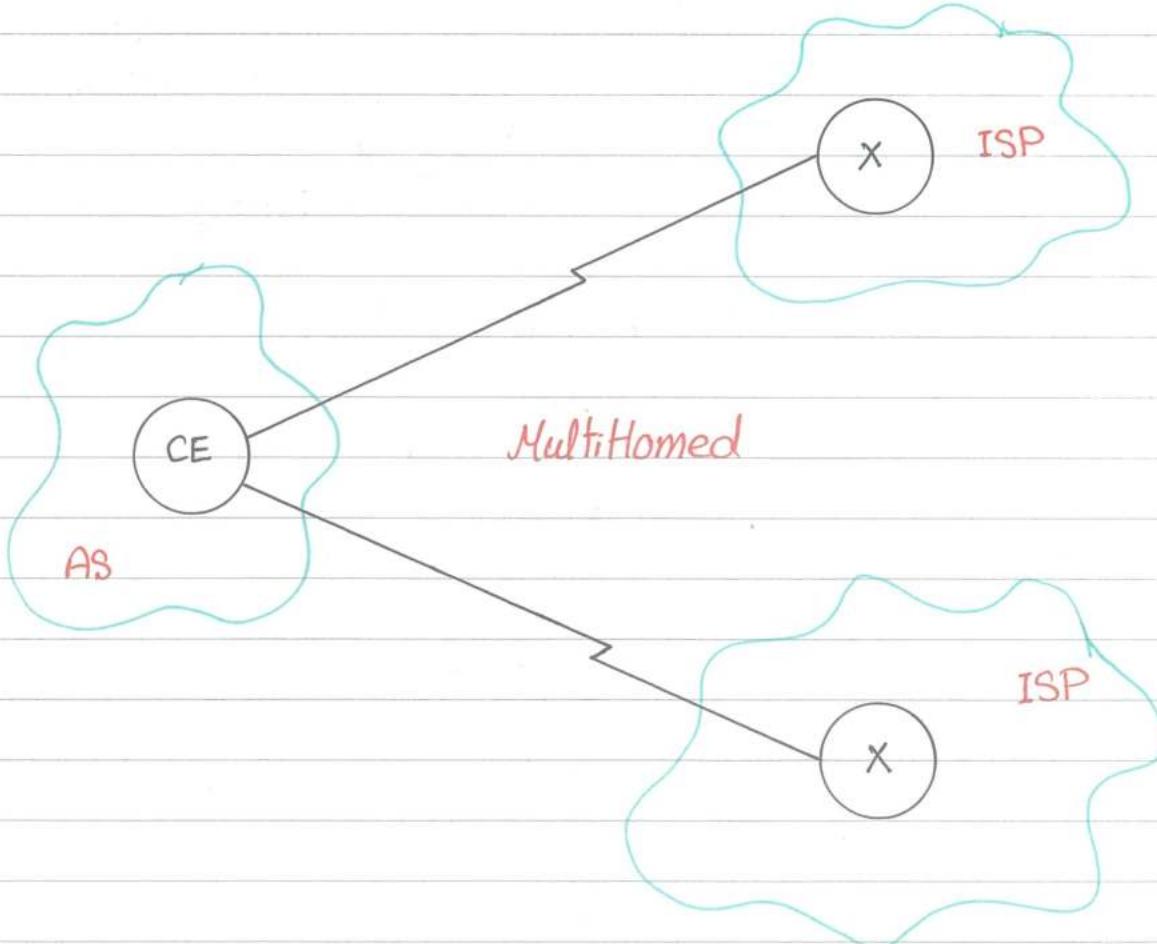
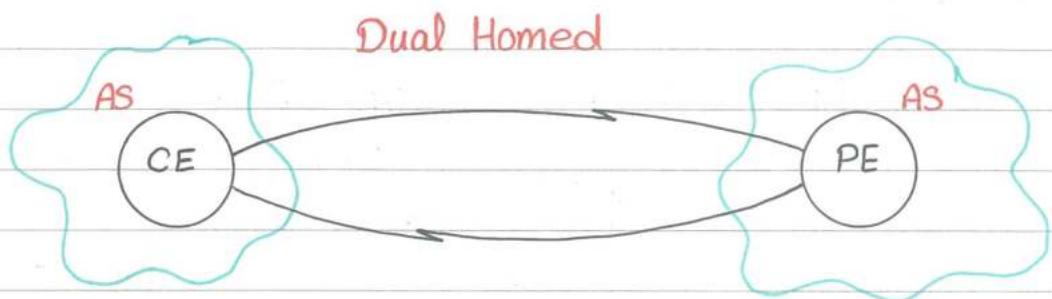
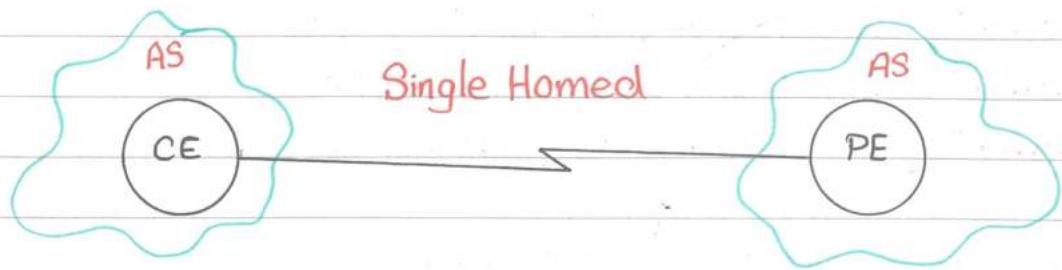


- Customer receive all routes From all providers:



CE(Config) # ip prefix-list 2 permit 12.0.0.0 /24

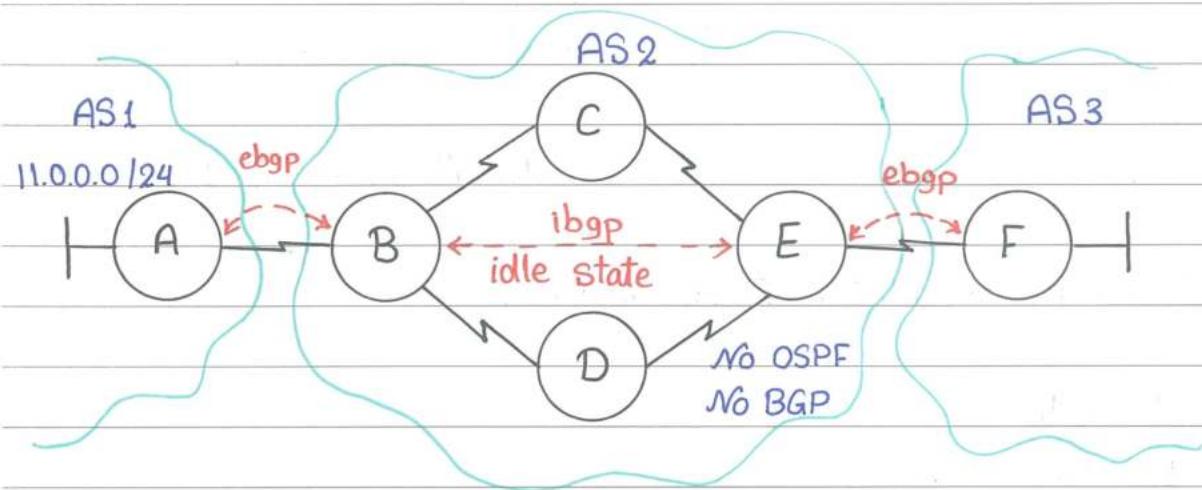
↳ To avoid customer network to become  
transit AS for SP ...



## Session 10 Part 1

### BGP Requirements in Transit AS

1. Run BGP on Border routers & don't run IGP inside transit AS.



## BGP @ change batched updates:

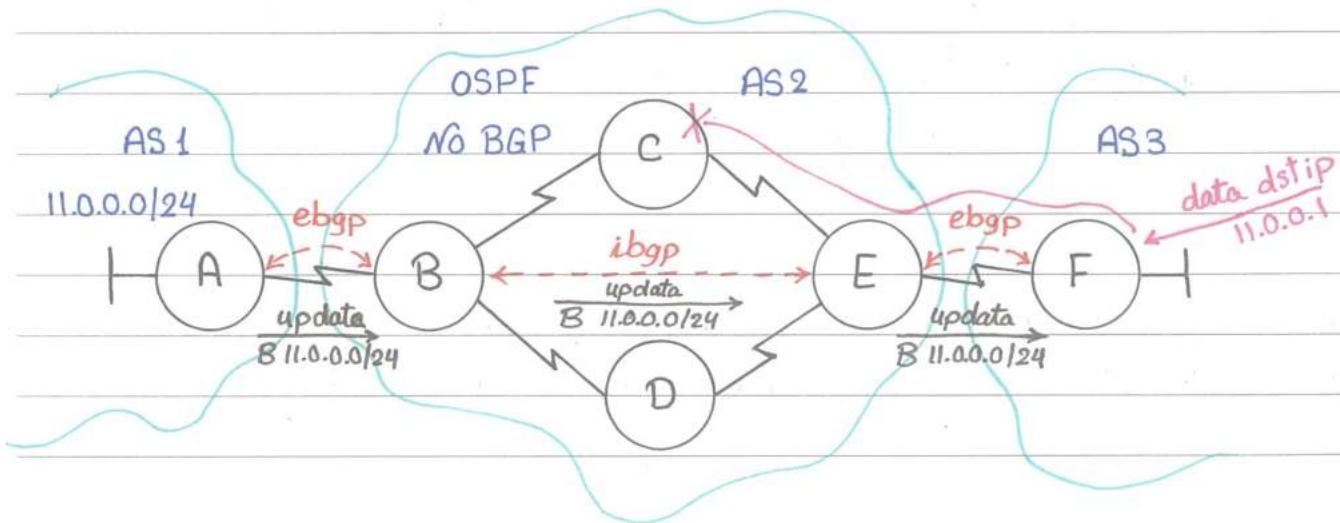
- every 30 sec (eBGP)
  - every 5 sec (iBGP)

We need to run IGP inside transit AS:

- BGP neighbor establishment.
  - Data Forwarding is based on recursive lookup.

## Session 10 Part 2

2. Run BGP on Border routers & run IGP inside transit AS.



## Synchronization Rule:

- Routes learnt by iBGP neighbor should never be advertised to any neighbor (iBGP or eBGP) unless it is synchronized.
  - Know by IGP B should redistribute BGP into IGP
  - To avoid black holes.

(Config) # router bgp AS

(Config-router) # no synchronization ← default on recent IOS

- iBGP should run on all transit AS routers.

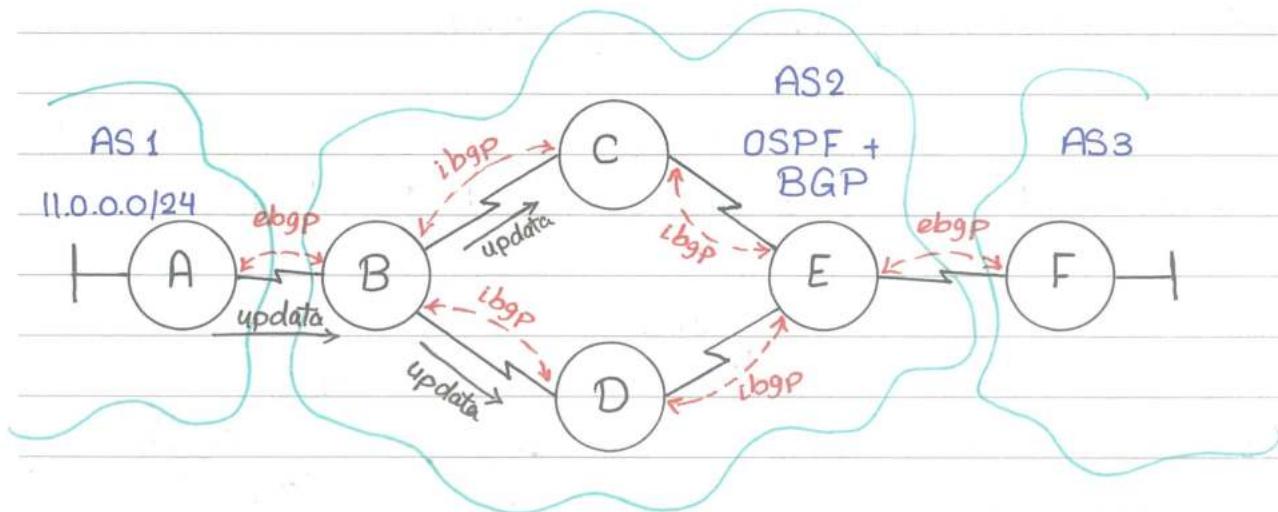
## ⇒ Split Functionality:

### iBGP & IGP Interaction:

- Internal routes are IGP responsibility.
- External routes are iBGP responsibility.

## Session 10 Part 3

3. Run BGP on all transit AS routers & run IGP inside on all transit AS routers.



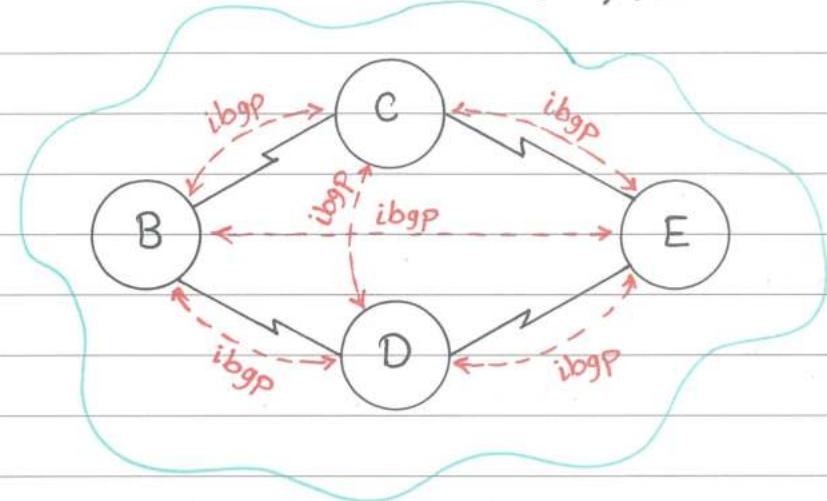
To avoid loops inside AS: BGP obey BGP split horizon rule.

- Routes learnt from iBGP should never be advertised to another iBGP.

- Run BGP on all transit AS routers in Full mesh Fashion.

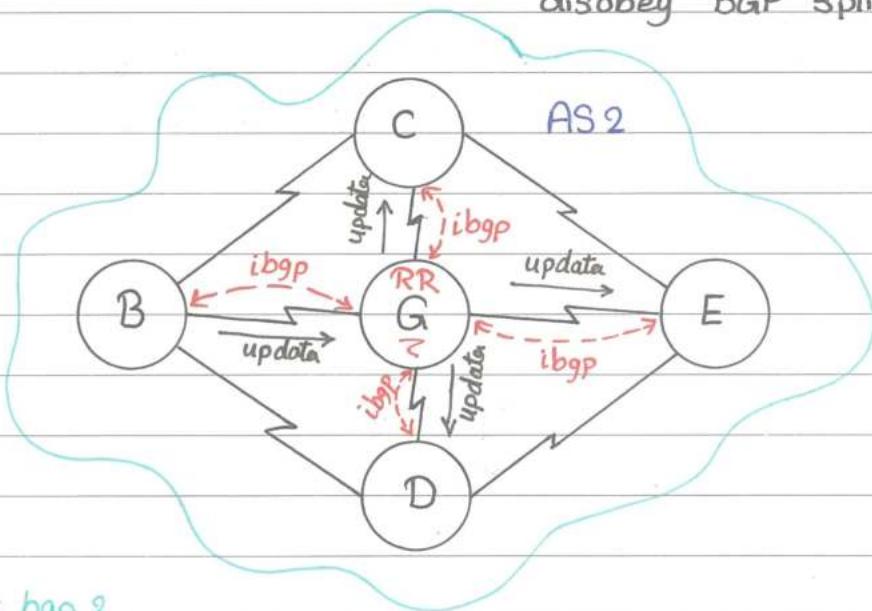
large no. of neighborships

$$n(n-1)/2$$



- Run BGP in transit AS as star topology (Route-Reflector).

disobey BGP split horizon rule



B,C,D,E :

(Config) # router bgp 2

(Config-router) # neighbor G remote-as 2

G :

(Config) # router bgp 2

(Config-router) # neighbor B,C,D,E remote-as 2

(Config-router) # neighbor B,C,D,E route-reflector-client

## Session 10 Part 4

4. Run IGP on all transit AS routers & Run BGP on all transit AS.  
either :

- full mesh BGP neighborship.

- OR - Route-Reflector .

- OR - Confederations .

⇒ BGP Confederation:

- Divide main AS into sub ASs.

Confederation

Private AS (64512-65535)

- Confederation act as :

- \* eBGP in neighborship establishment (between two sub ASs).

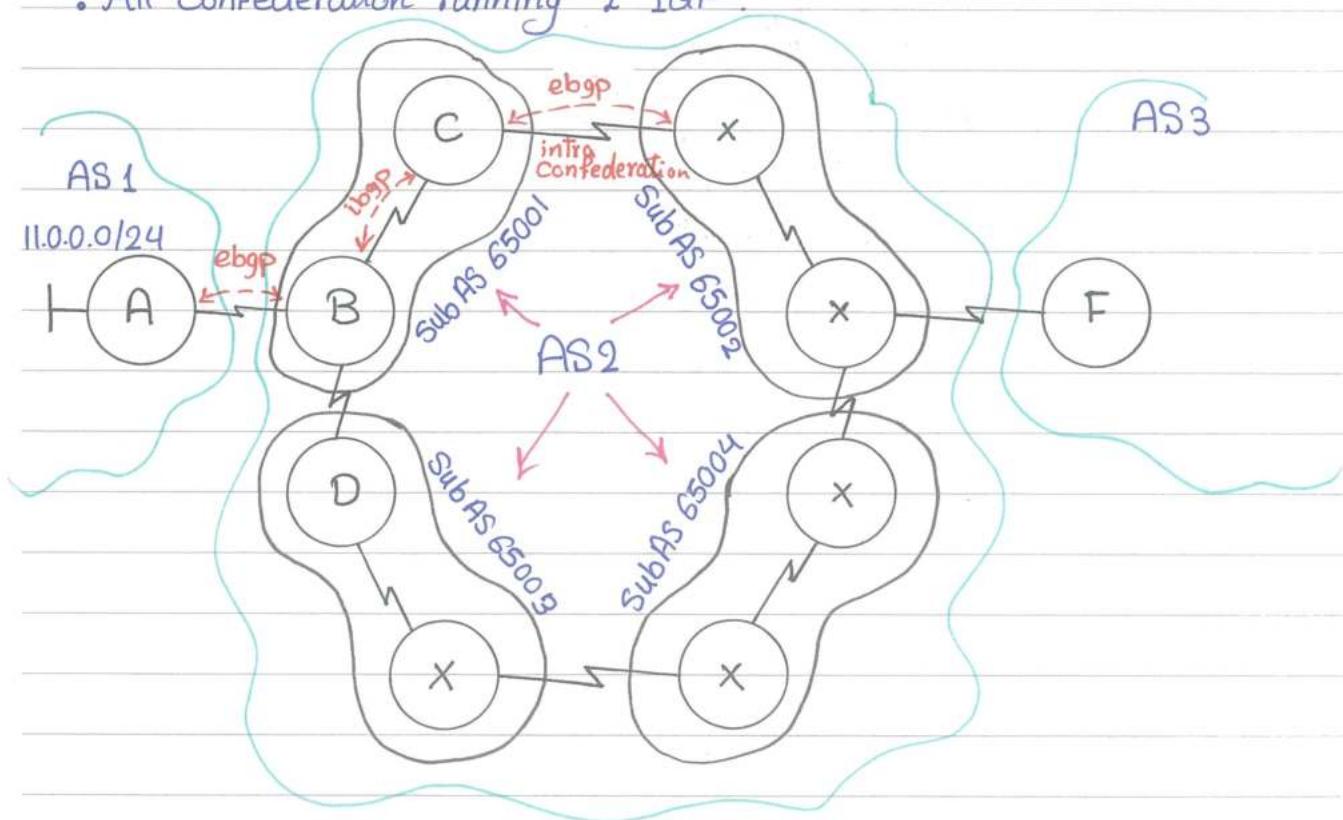
(update - source loopback - ebgp multihop)

- \* iBGP in update transfer.

(local preference - N.H. - MED - ...) except AS Path list

will follow ebgp rules ( ) = intra confederation

- All confederation running 1 IGP.



B: (Config) # router bgp 65001

(Config-router) # bgp confederation-id 2 main AS

(Config-router) # bgp confederation peer 65003

(Config-router) # neighbor D remote-as 65003 ← intra confederation ebgp

(Config-router) # neighbor C remote-as 65001 ← intra confederation ibgp

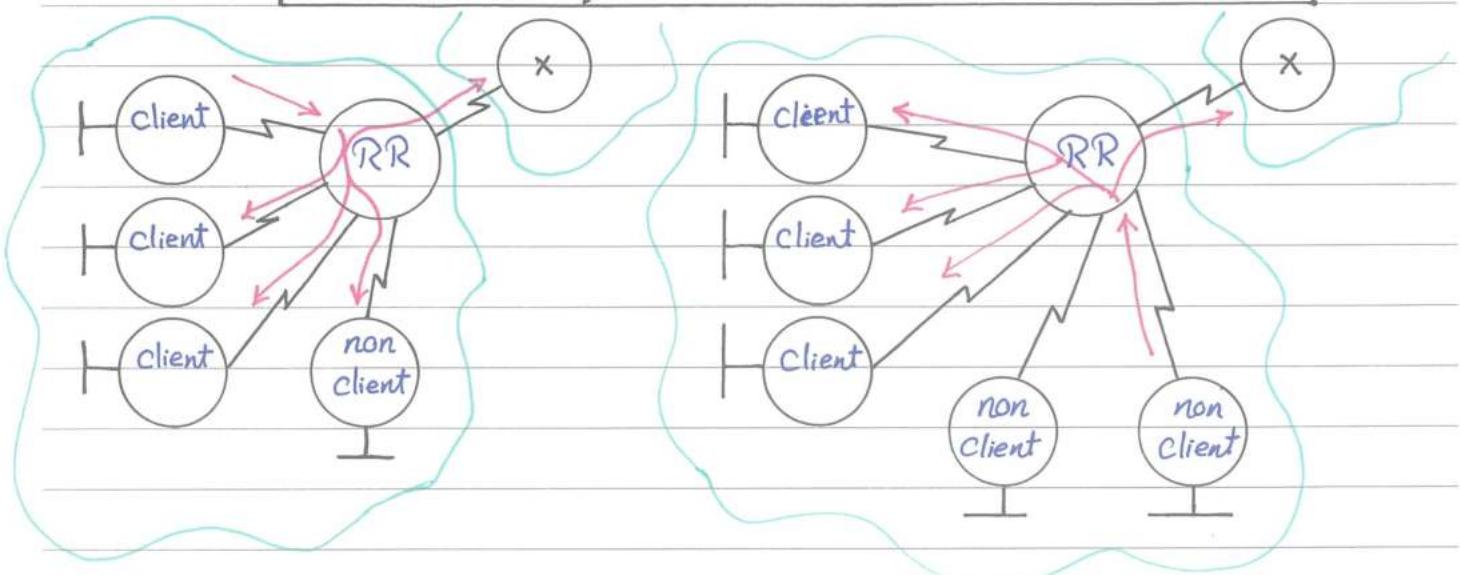
(Config-router) # neighbor A remote-as 1 ← ebgp

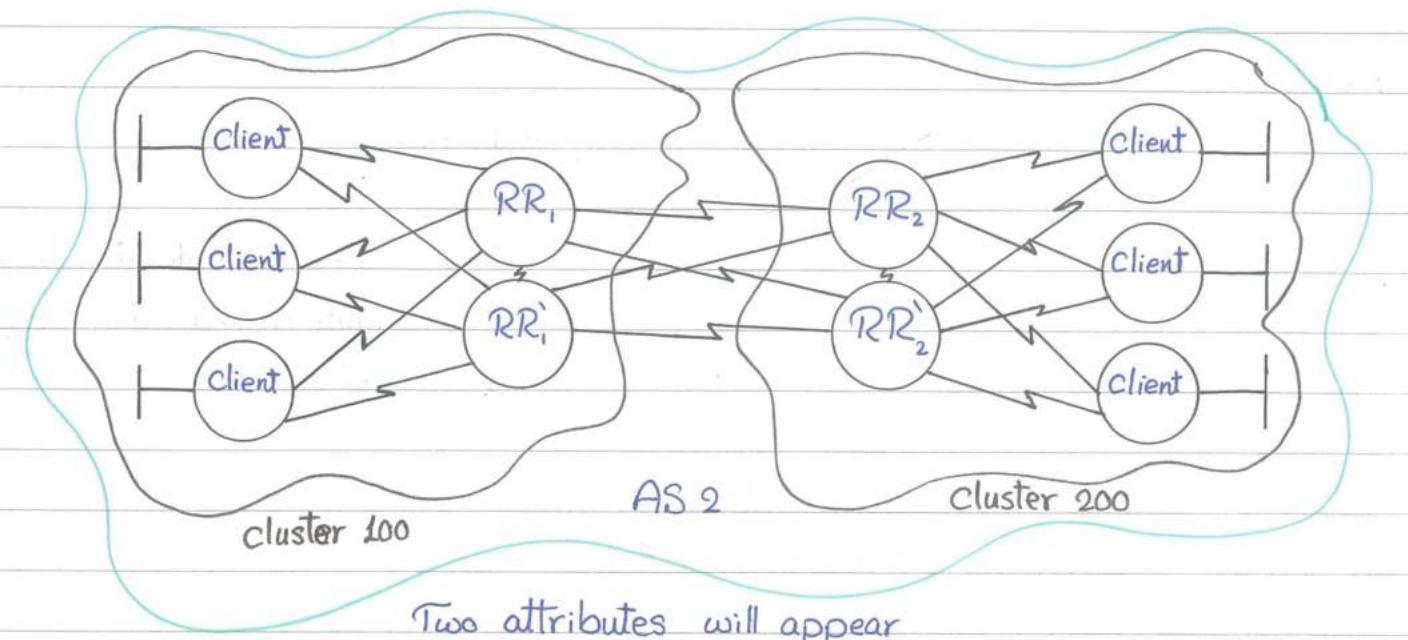
## Session 10 Part 5

### Route-Reflector

Rules:

Routes From	Advertised to
ebgp	Any router
iBGP	eBGP only
client	Any router (eBGP, client, non client)
Non Client	eBGP + Client only





Two attributes will appear

- Cluster list (loop prevention mechanism)  
(100, 200)
- originator Id (RID of RR)

RR:

```
(Config)# router bgp 2
(Config-router) # bgp cluster-id 100
(Config-router) # neighbor _____ remote-as 2
(Config-router) # neighbor _____ route-reflector-client
```

## Session 11 Part 1

### BGP Consideration & Requirements

BGP Neighbor Discovery:

1. Neighbor Command.
2. Neighbor Should be reachable.  
using IGP → Connected, static, RIP, OSPF, ...
3. Source of update.
4. eBGP multihop (eBGP neighbor may not directly connect).

\* Show ip bgp summary

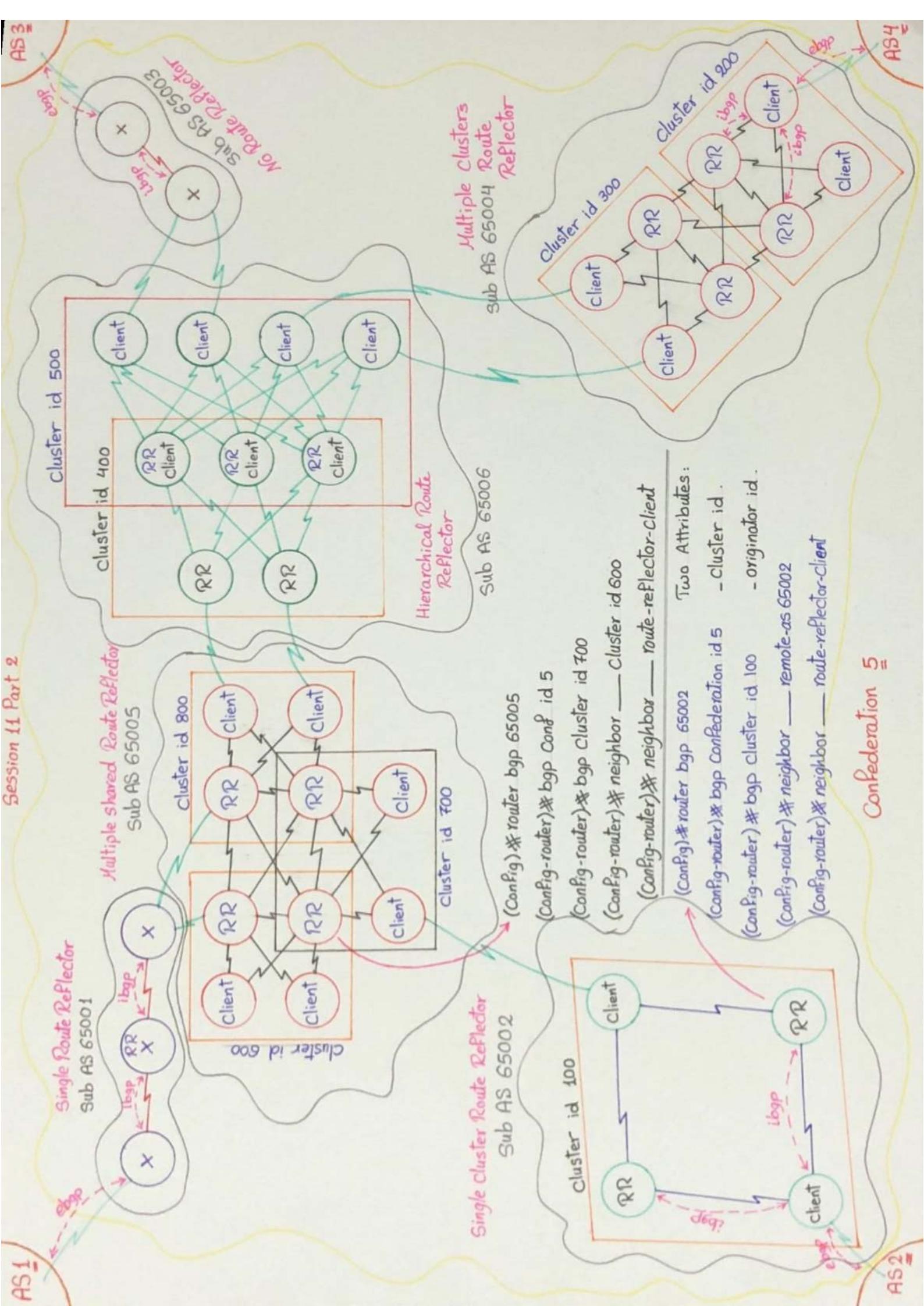
# Show ip bgp neighbor ip of neighbor

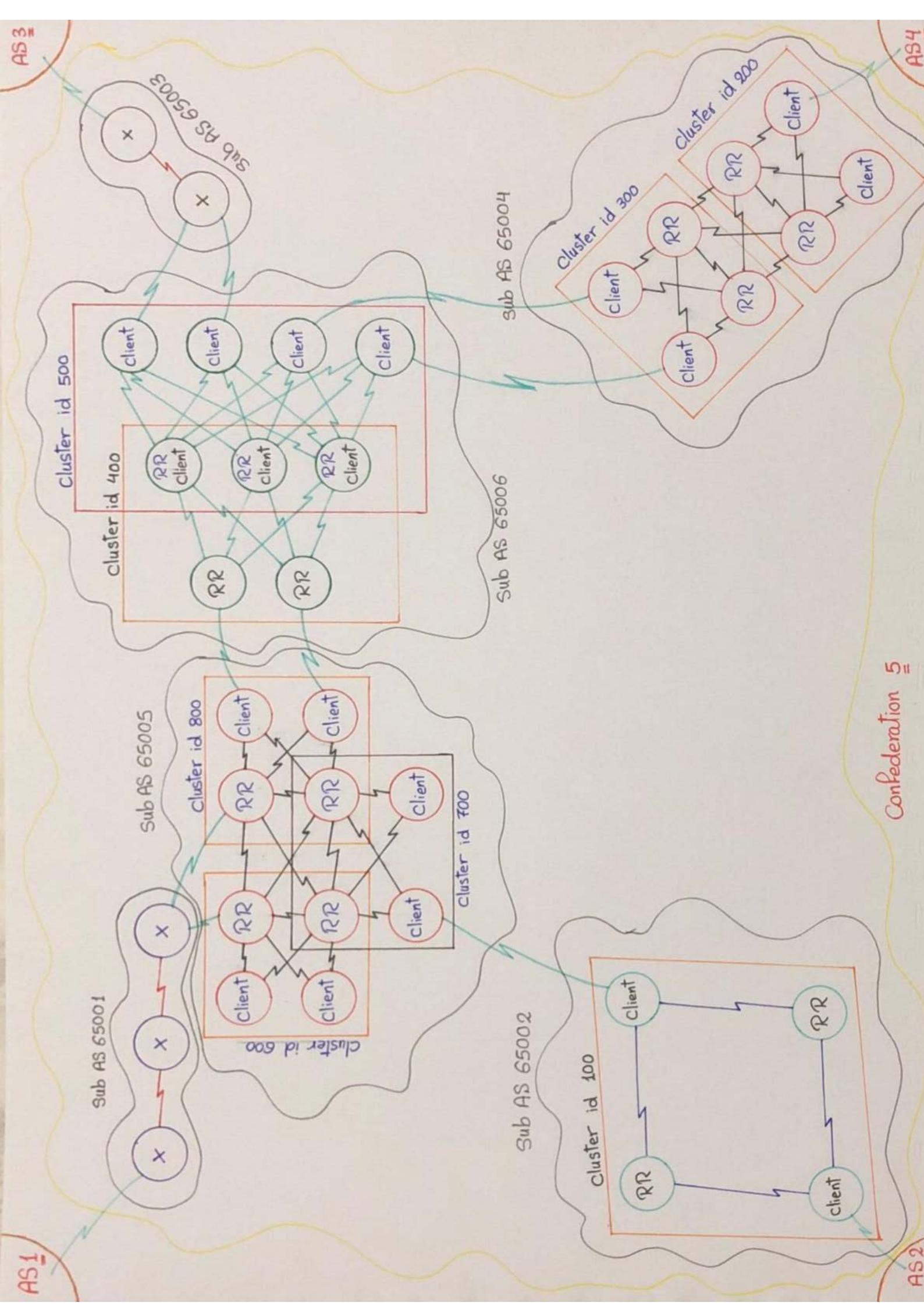
BGP Routes Discovery:

1. Advertise routes into BGP (inject routes in BGP table).  
network aggregate address injectmap/exist map
2. Disable Synchronization.  
(Config-router) # no sync
3. Next hop of update reachable.  
(Config-router) # neighbor ip next-hop-self
4. BGP Split horizon:
  - BGP Full Mesh.
  - BGP Confederation.
  - BGP Route-Reflectors.

→ If neighbor AS is wrongly configured, Flapping state between idle & Active state with notification msg.

Session 14 Part 2





## Session 11 Part 3

### BGP Path Selection

NwLoA OMNi OIII

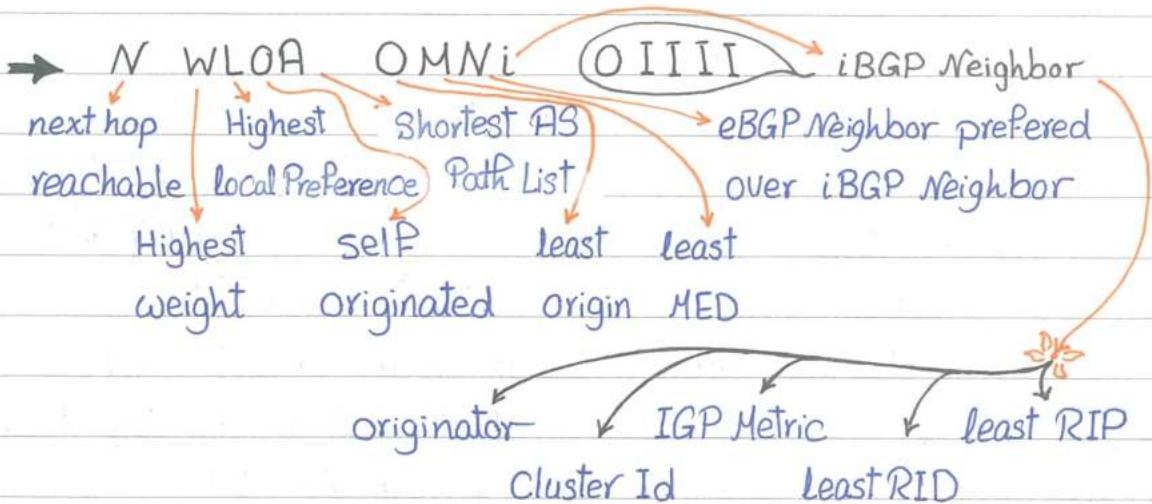
- o. Next hop should be reachable.
- 1. Highest weight by default 0, except if your router is the originator... the network weight = 32 768 (0-65535)...
- 2. Highest local preference by default 100.
- 3. Self originated NH 0.0.0.0 is preferred.
- 4. Least AS path list.
- 5. Least origin ( $i < e < ?$ ).
- 6. Least MED.
- 7. Neighbor eBGP preferred over iBGP.
- 8. For iBGP:
  - Unreflected route preferred over Reflected Route. no originator id
  - Least cluster list is preferred.
  - Least IGP metric.
  - Least Router ID.
  - Least Router IP. (IP of interface received route from it)...

### BGP Attributes

w ano La ac ocm

## Session 12 Part 1

### BGP Algorithm for Best Path Selection



BGP by default choose only one best path ...

(Config-router) # bgp best path Attribute ignore

(Config-router) # maximum-paths 1-6 1 by default eBGP

(Config-router) # maximum-paths iBGP 1-6 1 by default iBGP

Equal Load Sharing:

Same Attributes till IGP Metric "N WLOA OMNI" ...

Unequal Load Sharing:

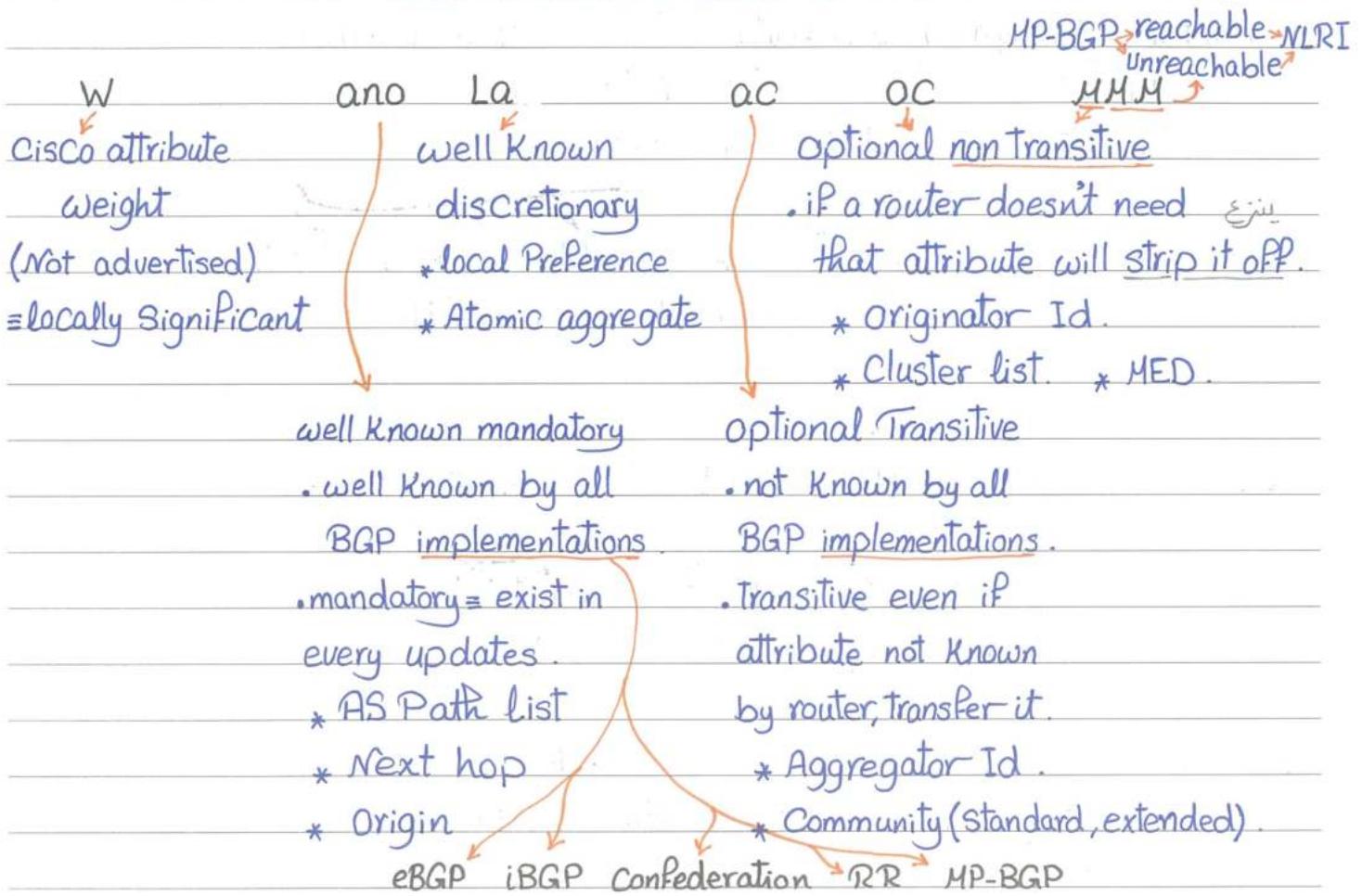
Using BW attribute (extended community attribute)

(Config-router) # bgp dmz-link bw

(Config-if) \* bandwidth  
in Kbps

## Session 12 Part 2

### BGP Attributes Classification



## Session 12 Part 3

### Attributes

#### Origin Attribute :

Describe how update was injected in BGP Table . Least origin is best.

i < e < ?

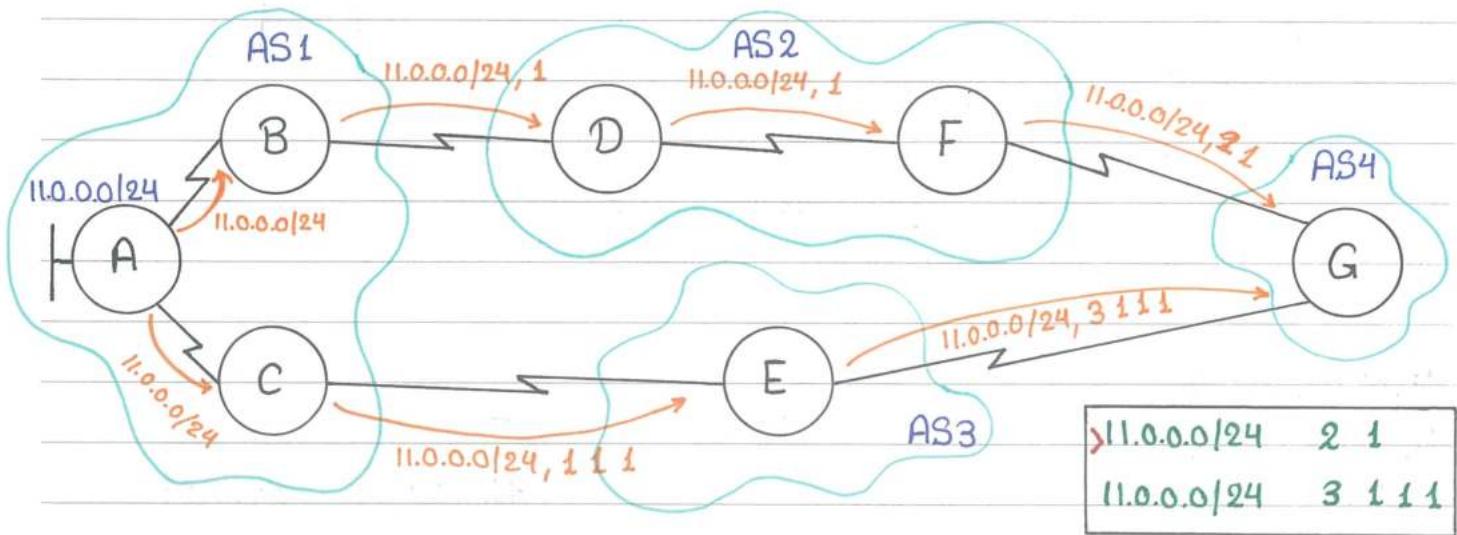
- . network Command.
- . Redistribute .
- . Redistribute .
- . aggregate address . EGP → BGP
- IGP → BGP

### - Next hop Attribute :

Next hop is router in next AS. It is not changed in iBGP, changed only in eBGP sessions unless `next-hop-self` command is used.

### - AS Path:

Least AS path list is preferred AS no. is prepended only in eBGP updated & not prepended in iBGP update.



C(Config) # route-map x permit

C(Config-route-map) # match ip address 11

C(Config-route-map) # set as-path prepend 1 1

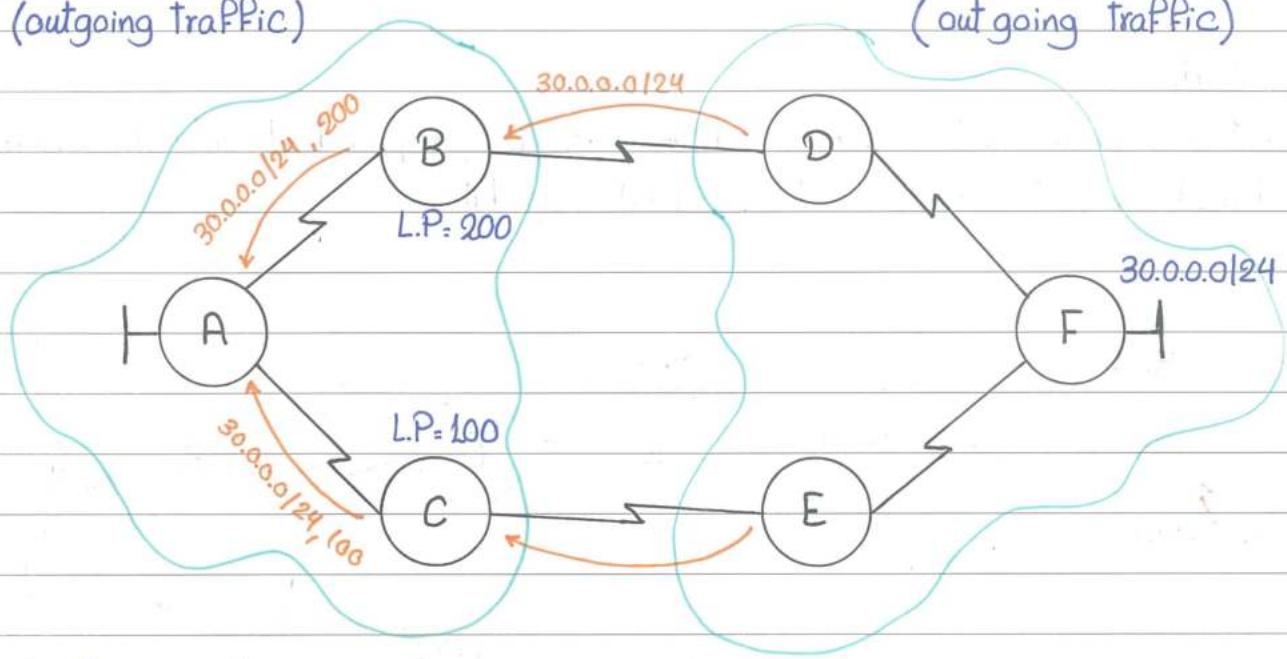
C(Config) # router bgp 1

C(Config-router) # neighbor E route-map x out

## Session 12 Part 4

- weight :

affect incoming update  
(outgoing traffic)



- local Preference :

affect incoming update  
(outgoing traffic)

A(Config) # router bgp 1

A(Config-router) # neighbor B weight 50

OR

A(Config) # route-map x permit

A(config-route-map) # match ip address 30

A(config-route-map) # set weight 50

A(Config) # router bgp 1

A(Config-router) # neighbor B route-map x in

A

30.0.0.0/24	weight
> via B	50
C	0

B(Config) # router bgp 1

B(Config-router) # bgp default local-preference 200

OR

B(Config) # route-map x permit

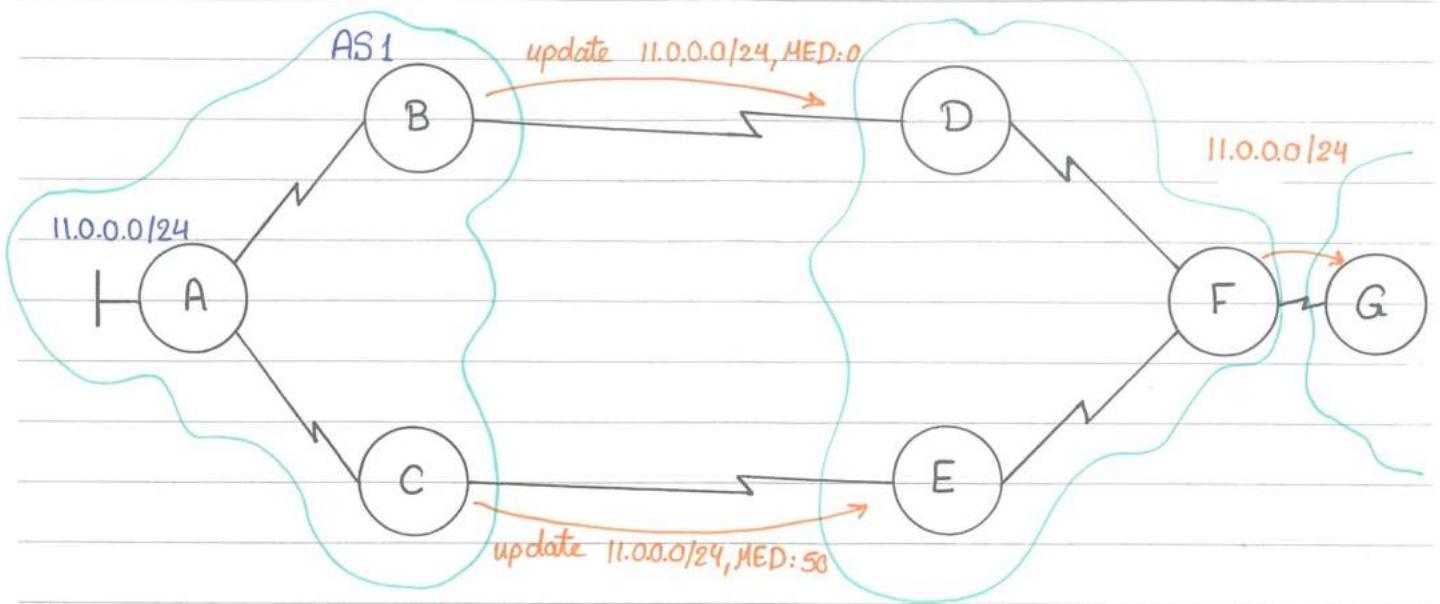
B(config-route-map) # match ip address 30

B(config-route-map) # set local-preference 200

B(Config) # router bgp 1

B(Config-router) # neighbor D route-map x in

30.0.0.0/24	L.P
> via B	200
C	100



- AS Path list:  
outgoing update  
(incoming traffic).

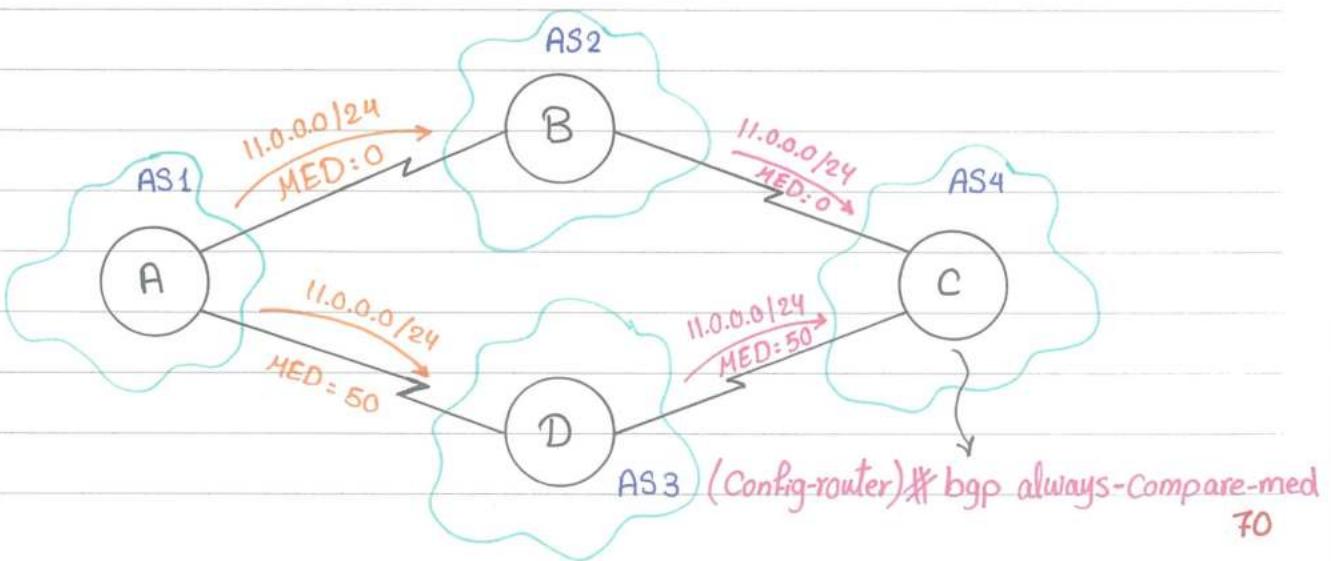
- MED : Multi Exit Discriminator  
by default = 0 , least is best .  
outgoing update (incoming traffic).

C (Config) # router bgp 1  
C (Config-router) # default-metric 50

OR

C (Config) # route-map x permit  
C (Config-route-map) # match ip address 11  
C (Config-route-map) # set metric 50  
C (Config) # router bgp 1  
C (Config-router) # neighbor E route-map x out

11.0.0.0/24	metric
> via D	0
E	50

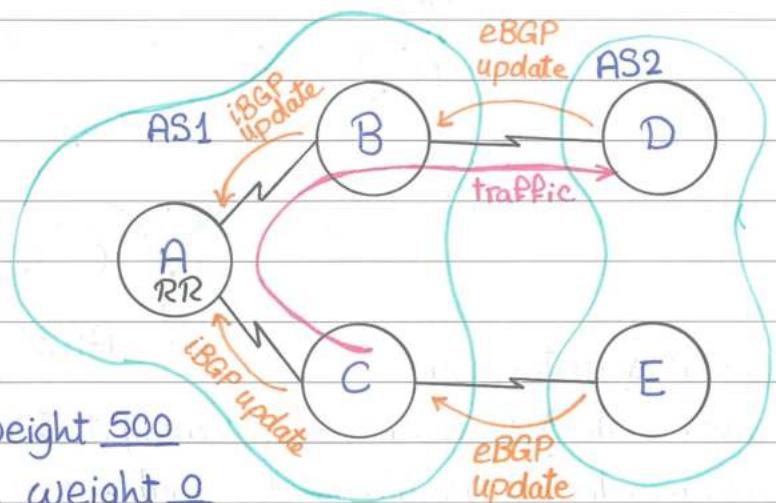


## Session 13

### Advanced Case Studies

Influencing policy using weight : outgoing traffic

Case 1 :



A(config) # router bgp 1

A(config-router) # neighbor B

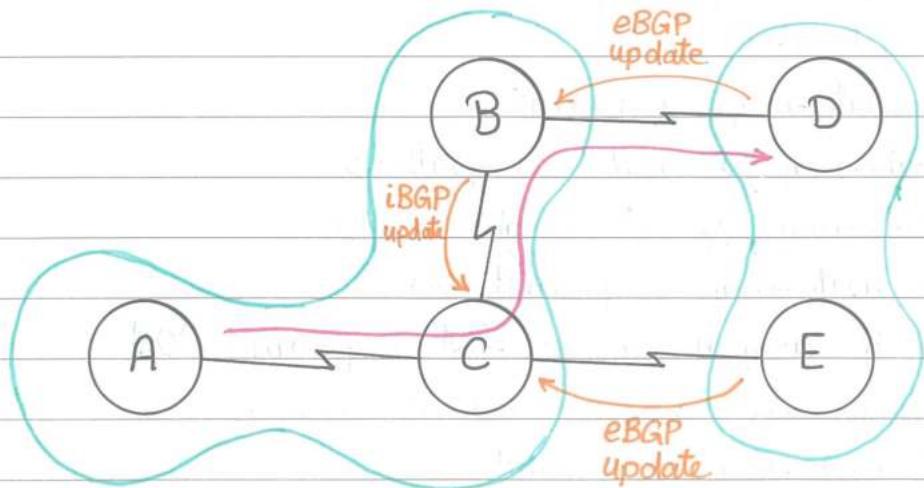
weight 500

A(config-router) # neighbor C weight 0

C(config) # router bgp 1

C(config-router) # neighbor A weight 500

Case 2 :



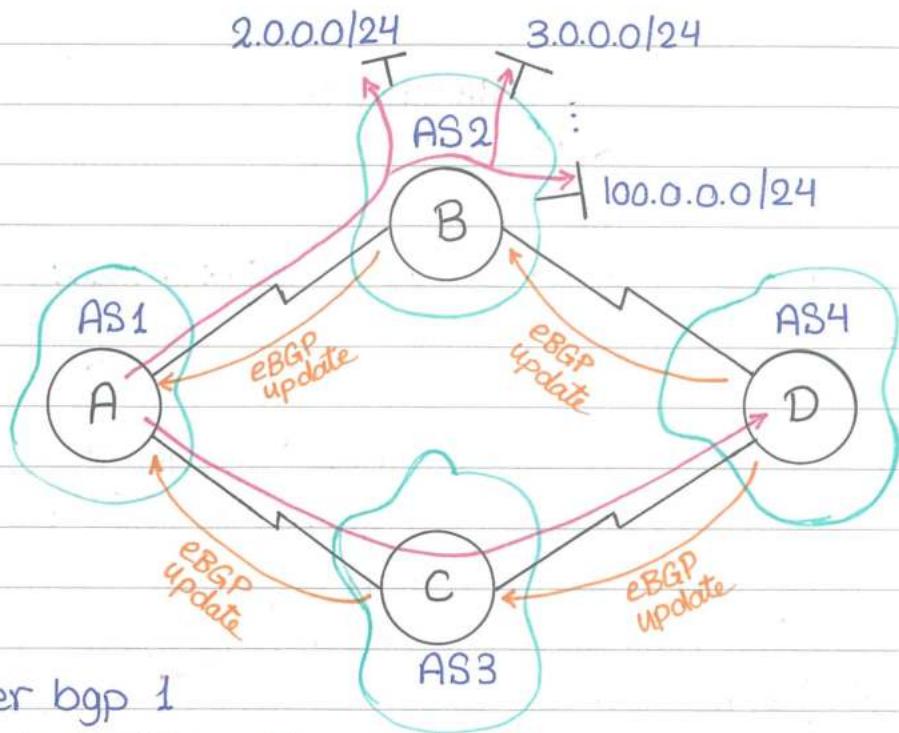
C(config) # router bgp 1

C(config-router) # neighbor B weight 500

B(config) # router bgp 1

B(config-router) # neighbor D weight 500

Case 3 :



A(Config) # router bgp 1

A(Config-router) # neighbor C weight 500

A(Config-router) # neighbor B route-map x in

For 2.0.0.0/24 only

A(Config) # route-map x permit

A(Config-route-map) # match ip address prefix-list 2

A(Config-route-map) # set weight 2000

A(Config) # route-map x permit 20

For AS2 networks

A(Config) # route-map x permit

A(Config-route-map) # match as-path 2

A(Config-route-map) # set weight 2000

A(Config) # route-map x permit 20

A(Config) # ip as-path access-list 2 permit ^2\$

For network 101 from C & another networks from B

A(Config) # route-map x permit

A(Config-route-map) # match ip address 101

A(Config-route-map) # set weight 300

A(Config) # route-map x permit 20

A(Config-route-map) # match as-path 2

A(Config-route-map) # set weight 2000

A(Config) # route-map x permit 30

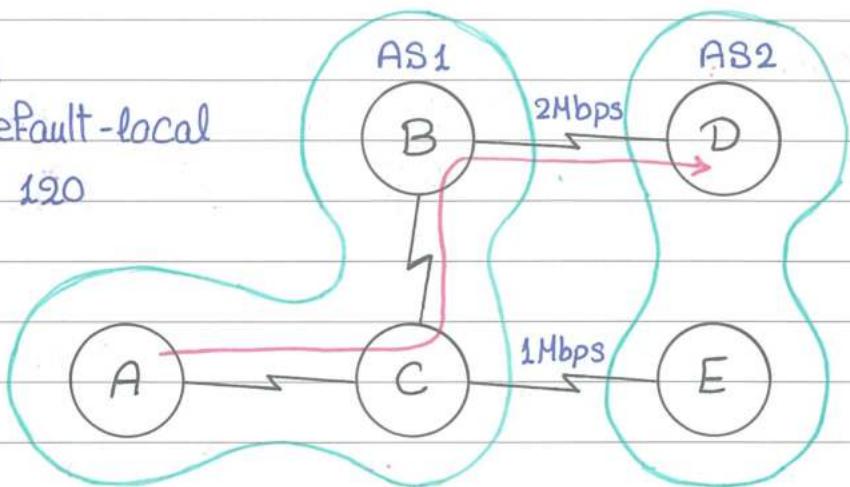
## Session 13 Part 2

- Influencing policy using local preference: outgoing traffic

**Case 1:**

B(Config) # router bgp 1

B(Config-router) # bgp default-local  
preference 120

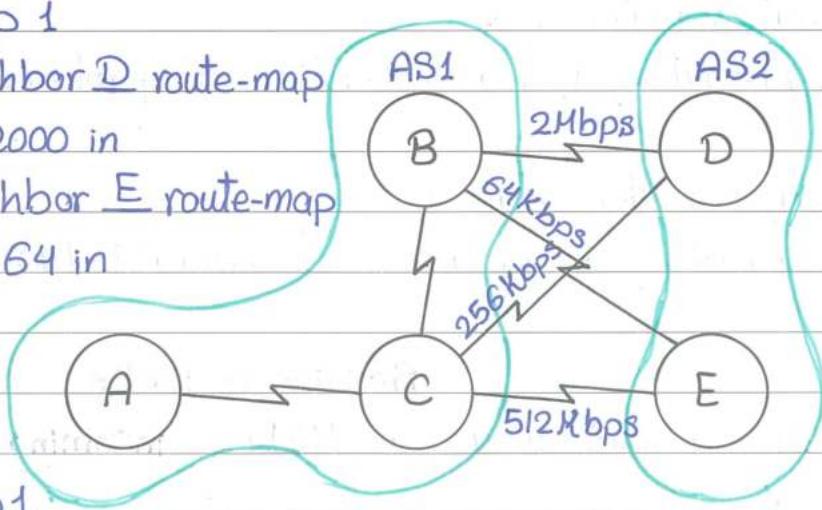


**Case 2:**

B(Config) # router bgp 1

B(Config-router) # neighbor D route-map  
2000 in

B(Config-router) # neighbor E route-map  
64 in



C(Config) # router bgp 1

C(Config-router) # neighbor D route-map 256 in

C(Config-router) # neighbor E route-map 512 in

B(Config) # route-map 2000 permit

B(Config-route-map) # set local preference 2000

B(Config) # route-map 64 permit 20

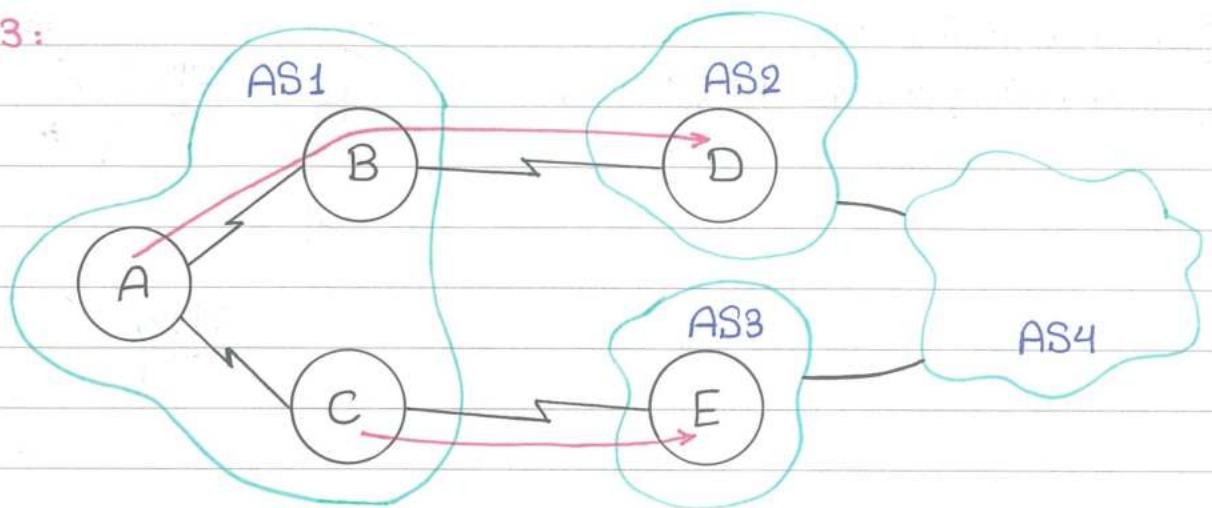
B(Config-route-map) # set local preference 64

C(Config) # route-map 512 permit

C(Config-route-map) # set local preference 512

C(config)# route-map 256 permit 20  
C(config-route-map)# set local preference 256

Case 3:



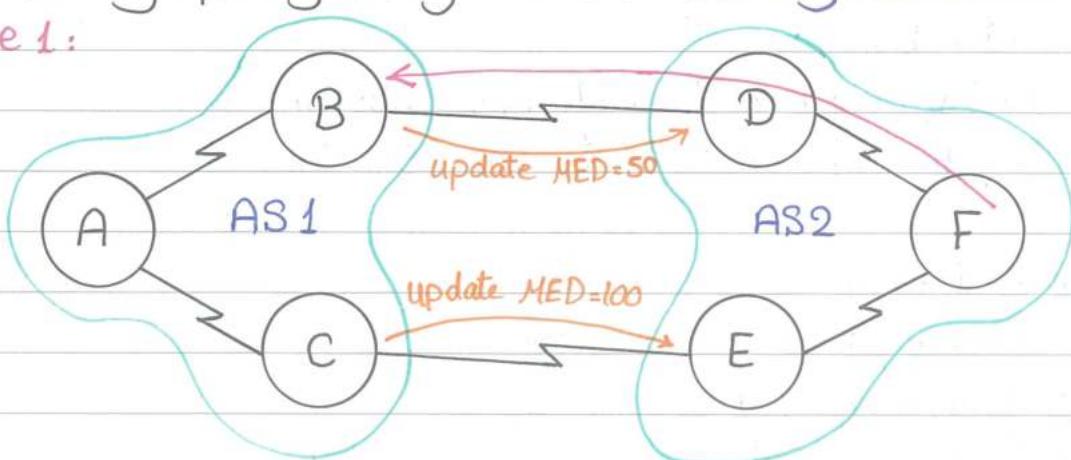
B(Config)# router bgp 1  
B(Config-router)# bgp default-local preference 120

C(Config)# router bgp 1  
c(config-router)# neighbor E weight 100

### Session 13 Part 3

Influencing policy using MED: incoming traffic

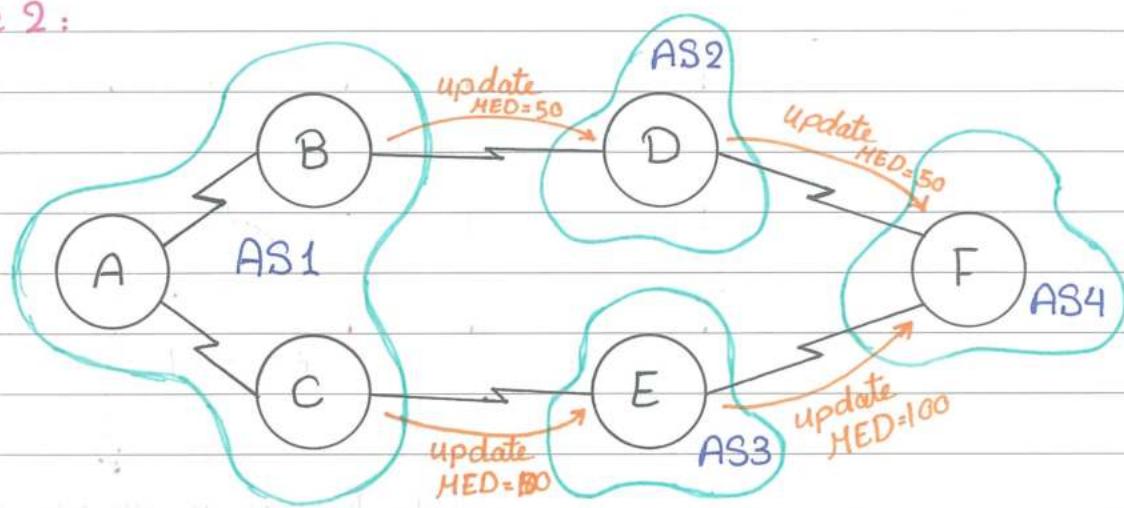
Case 1:



B(Config)# router bgp 1  
B(Config-router)# default-metric 50

```
c(Config)# router bgp 1  
c(Config-router)# default-metric 100
```

Case 2:

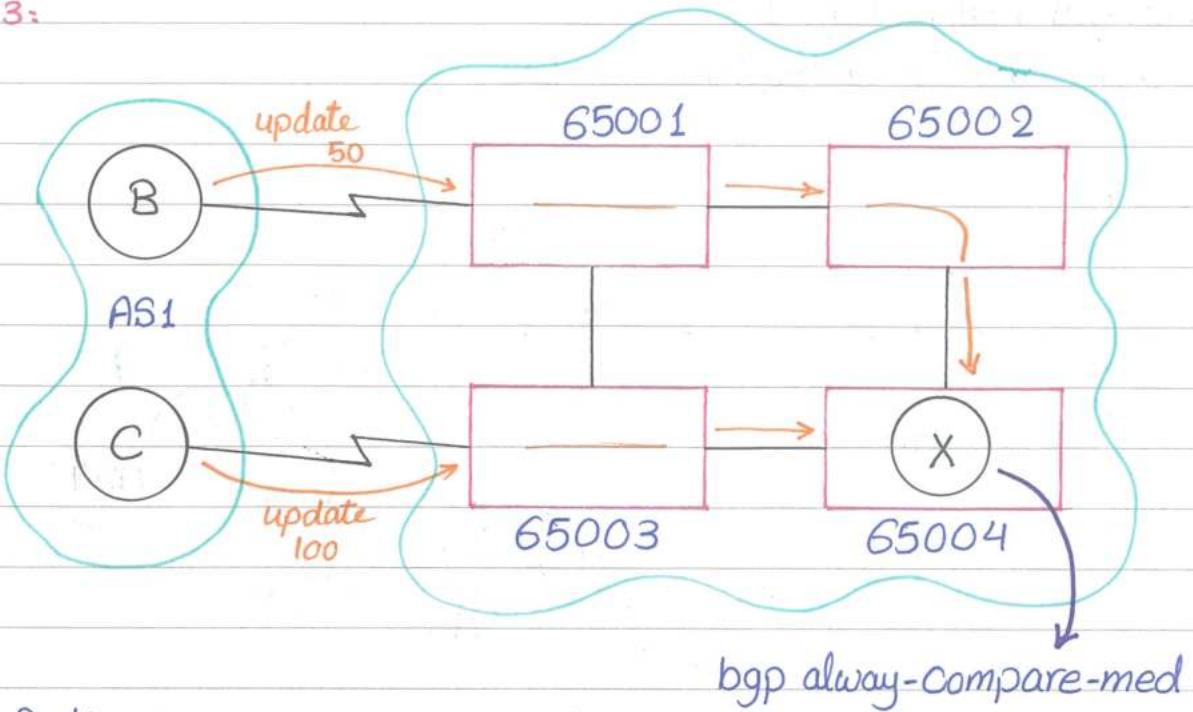


```
D(Config)# router bgp 2  
D(Config-router)# neighbor E route-map x out  
D(Config)# route-map x permit  
D(Config-route-map)# set metric 50
```

```
E(Config)# router bgp 3  
E(Config-router)# neighbor F route-map x out  
E(Config)# route-map x permit  
E(Config-route-map)# set metric 100
```

```
F(Config)# router bgp 4  
F(Config-router)# bgp always-compare-med
```

Case 3:



default MED = 0 in same vendors

standard Missing-Med by default oo

(Config) # router bgp 2

(Config-router) # bgp bestpath med missing-med-worst

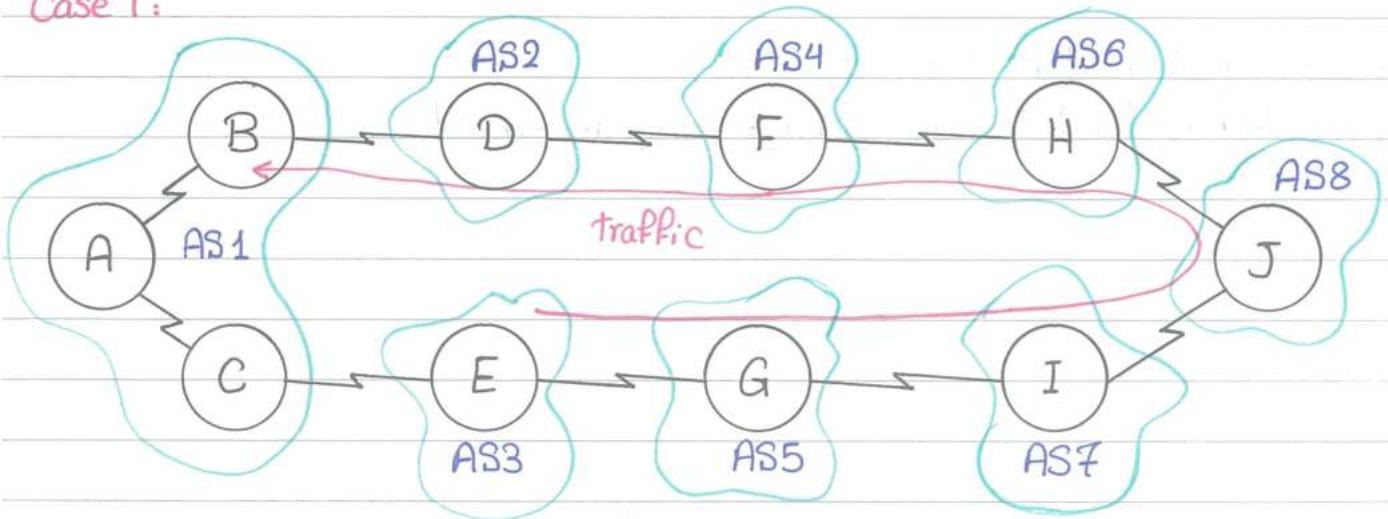
OR

(Config-router) # bgp bestpath med missing-as-worst

### Session 13 Part 4

- Influencing policy using AS Path List : incoming traffic

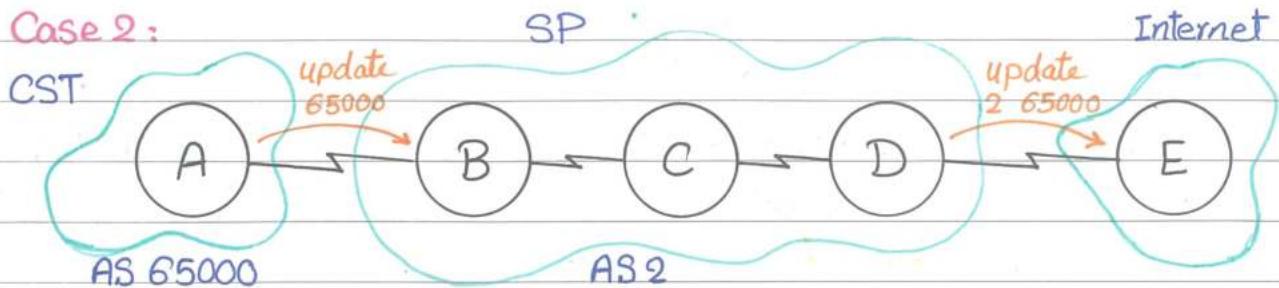
Case 1:



C(Config) # route-map X permit

C(Config-route-map) # set as-path prepend 11111111

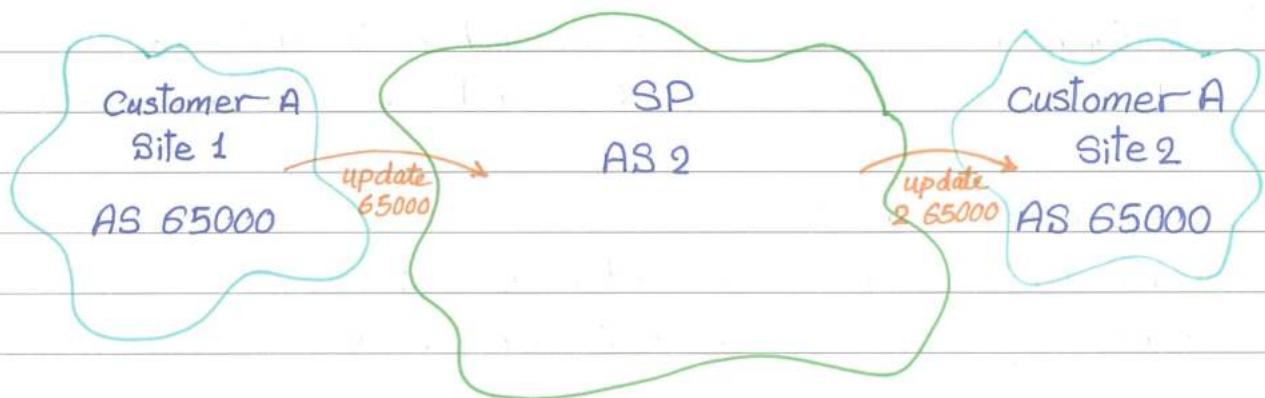
Case 2:



D(Config) # router bgp 2

D(config-router) # neighbor E remove-private-as

Case 3:



E(Config) # router bgp 65000

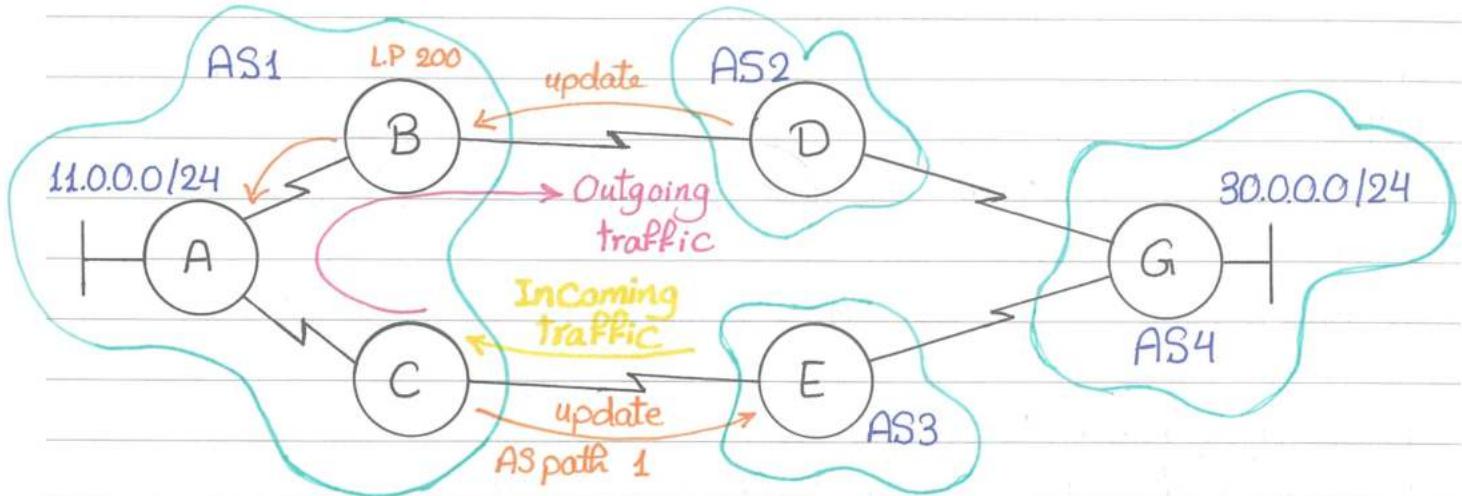
E(Config-router) # neighbor D allowas-in ← neglect AS path loop prevention check  
OR

D(Config) # router bgp 2

D(Config-router) # neighbor E asoverride

## Session 14 Part 1

### Asymmetric Routing Case Study



Required:

- \* Outgoing traffic for AS1 should go through AS2.
- \* Incoming traffic for AS1 should come through AS3.

B(Config)#router bgp 1

B(Config-router)#neighbor D route-map x in

B(Config-router)#neighbor D route-map y out

B(Config)#route-map x permit

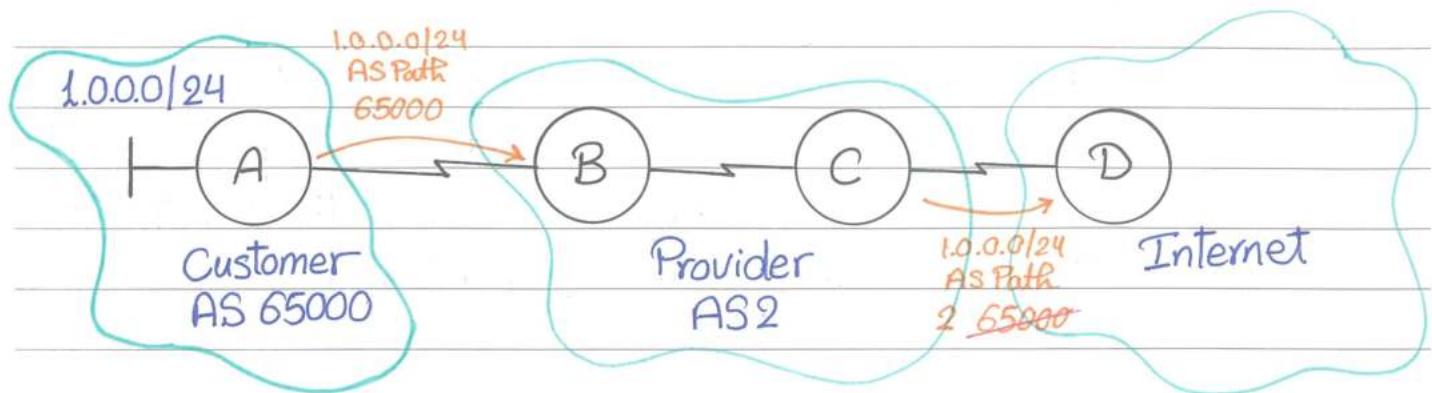
B(Config-route-map)#set local-preference 200

B(Config)#route-map y permit

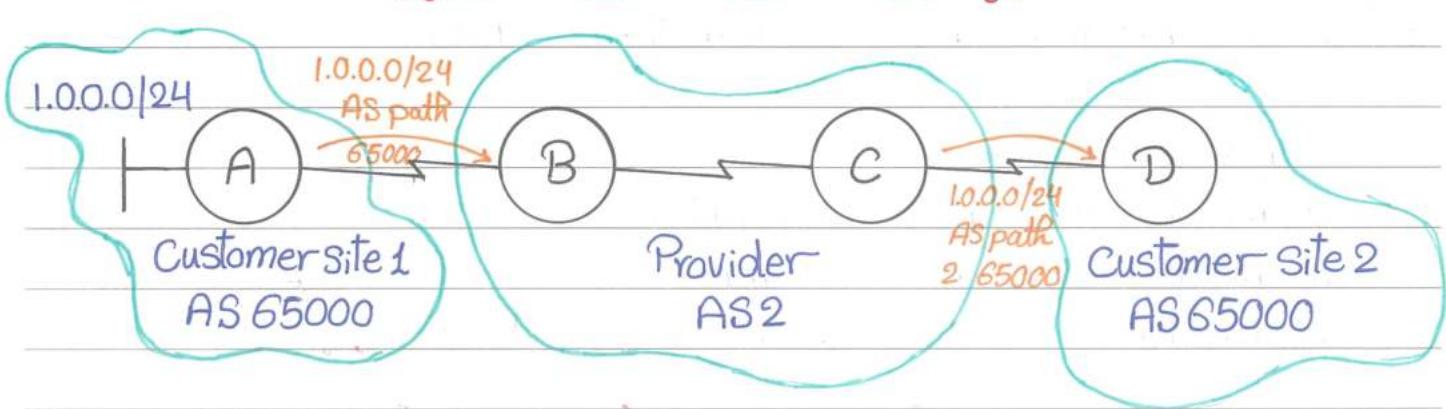
B(Config-route-map)#set as-path prepend 1111

## Session 14 Part 2

### AS Translation



```
C(Config) # router bgp 2
C(Config) # neighbor D remove-private-as
```



```
D(Config) # router bgp 65000
```

```
D(Config-router) # neighbor C allowas-in
```

OR

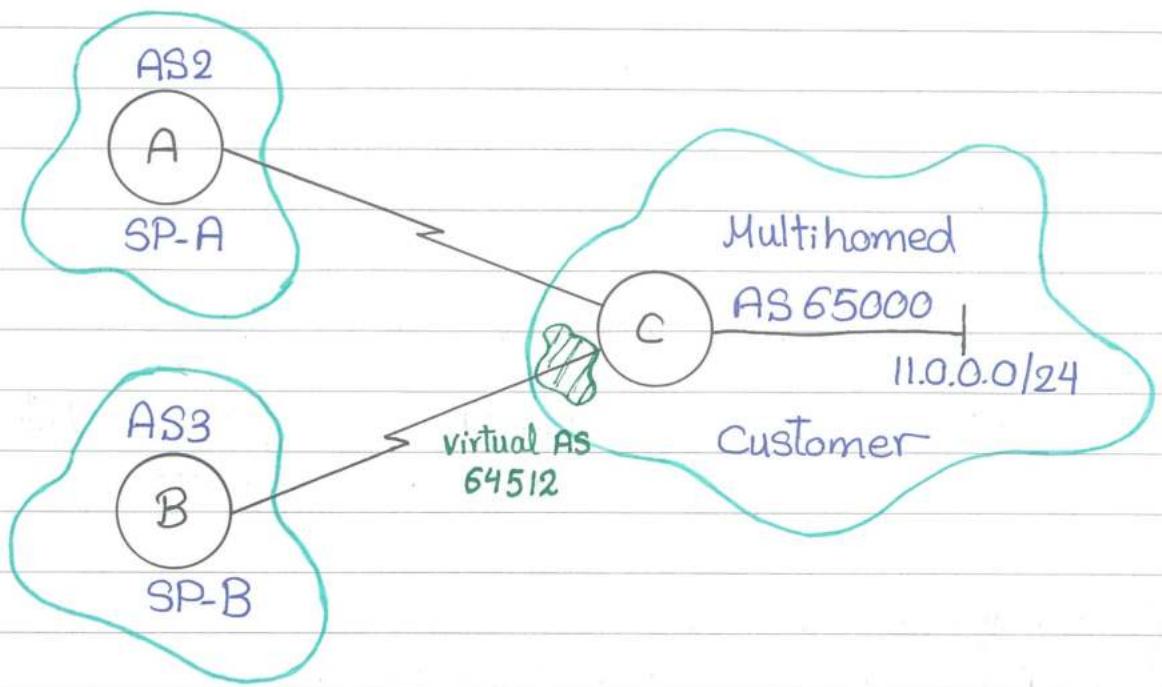
```
C(Config) # router bgp 2
```

```
C(Config-router) # neighbor D as-override
```

stop AS path list loop

Prevention Check

1.0.0.0/24, AS path 2 2



A(Config-router) # neighbor C remote-as 65000

B(Config-router) # neighbor C remote-as 64512

C(config) # router bgp 65000

C(Config-router) # neighbor A remote-as 2

C(Config-router) # neighbor B remote-as 3

C(Config-router) # neighbor B local-as 64512 ...

[no-prepend] [replace-as] [dual-as]

Don't prepend while  
Receiving update

Don't prepend while  
Sending update

Neighborship  
established using  
local as or main as

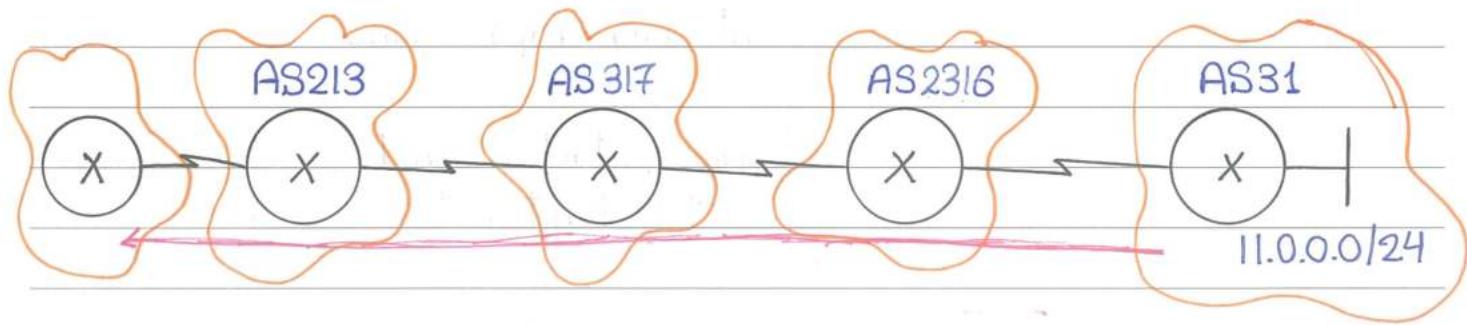
Session 14 Part 3  
**AS-Path Access list**  
**(AS-Path Filter list)**

(Config-router) # neighbor \_\_\_\_\_ Filter-list AS-Path ACL

[weight \*] {in|out}

Create AS path ACL:

(Config) # ip as-path access-list # {Permit|deny} Regular Expression



AS Path list 11.0.0.0/24  
 213 317 2316 31  
 ↓              ↓  
 last AS      1<sup>st</sup> AS "end of string"  
 "beginning of string"

ex: Permit 31      213 317 2316 31  
 1        2        3        4

ex: Permit 21 | 31      213 317 2316 31  
 1        2        3        4

ex: Permit (213 | 218) - 31      213 317 2316 31  
 1                  2

213 317 1218 316 31  
 1                  2

ex: Permit [123].[3-5]      213 317 2316 31  
 1        2

ex: Permit \_2\_

AS 2 calculate

ex: Permit ^21\_31\$ العنوان وال번호 AS21 calculate, AS31 calculate route

Regular Expression	Description
	OR
( )	Parentheses used to include small expression into large expression.
-	Space, start or end "Delimiter"
[ ]	Serial Number
.	Any character include Space
^	Begging of String
\$	End of String
*	Repeat zero or more
?	Repeat zero or one
+	Repeat 1 or more

## Session 14 Part 4

### Community Attribute

- . It is a mean of tagging routes in order to apply policy using attributes or to apply filter.
  - . It is a more generic attribute that can be mapped to other attribute values.
  - . By default all networks belong to Community called Internet.  
more general community
  - . Community is 32-bit, 64-bit & sometimes more.
  - . Community is either standard or extended.  
not send by default      Send by default  
(Config-router) # neighbor \_\_\_\_\_ send-community      ex: dmz-link BW,

every SP have like the below:

Community	meaning	
2:1	L.P = 200	
2:2	AS-Prepend twice	
2:3	AS-Prepend three	
2:4	no-advertise	. don't send update to any router
2:5	no-export	. Filters. don't advertise to real eBGP nei.
2:6	local-as	. Advertise to iBGP only
2:7	Internet	. Any

Use route-map:

(Config) # route-map — permit

(Config-route-map) # match ip address ACL\*  
as-path AS-Path ACL\*  
Community Community ACL\*

(Config-route-map) # set local preference  
as-path prepend  
Community

Community ACL:

(Config) # ip community-list 1-99 {permit|deny} Community  
standard

OR

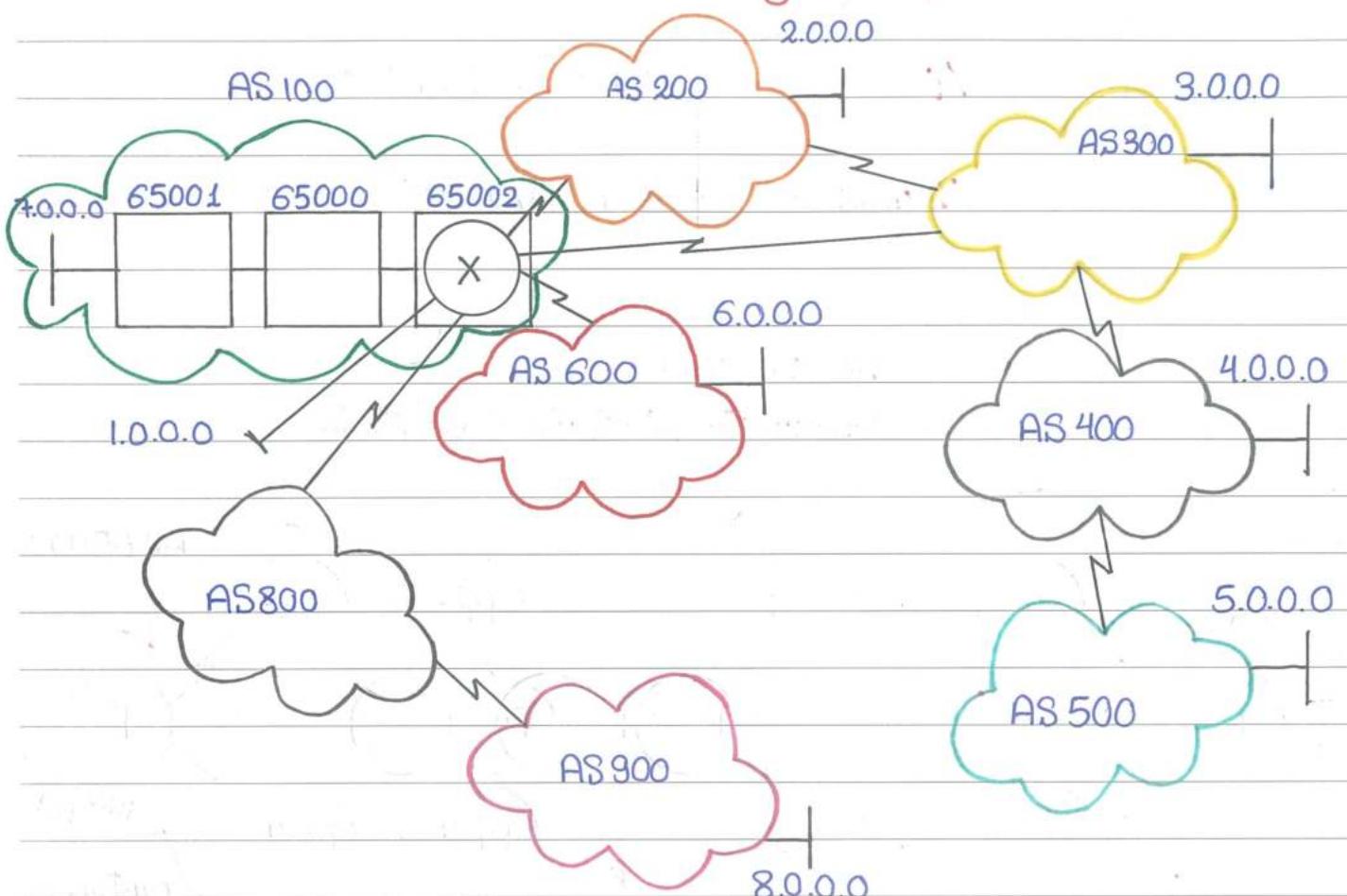
(Config) # ip community-list 100-199 {permit|deny} regular expression  
expanded

ex: 10:103, 10:104, 10: 203, 10:204

10:[1-2]0[3-4]

## Session 15

### — AS Path Regular Expression —



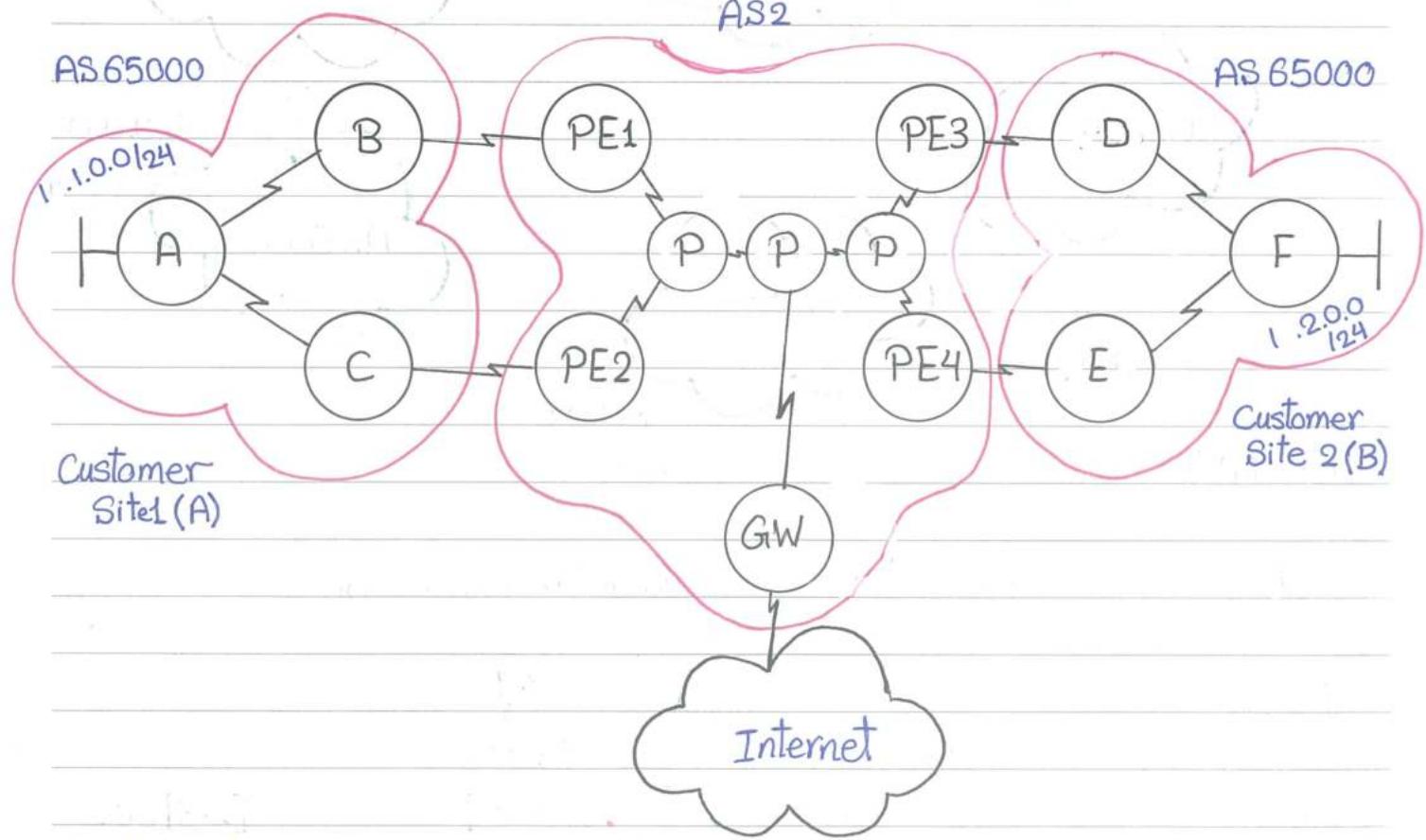
BGP Table : show ip bgp regular expression

Network	AS Path	Regular Expression
1.0.0.0		$^{\$}$
2.0.0.0	200	$^{200\$}$
3.0.0.0	200 300	$..300\$$
4.0.0.0	200 300 400	$..300..$
5.0.0.0	300 400 500	$^300..$
6.0.0.0	600 600 600	$(600..)(/1)^*$
7.0.0.0	(65000 65001)	$/$
8.0.0.0	800 800 900 900	$(800..)(/1)(900..)(/2)$

Regular Expression	Description
/()	Match on bracket
/1 (number)	Copy & Paste 1 <sup>st</sup> bracket

## Session 15 Part 2 Community Attribute Case Study

AS2



PE (Config) # route-map X permit

PE (Config-route-map) # match community Aet "1"

PE (Config-route-map) # set as-path prepend 65000 65000 65000 65000

PE (Config) # ip community-list 1 permit 2:4

C(Config) # route-map y permit

C(Config-route-map) # match ip address 1

C(Config-route-map) # set community 2:4 2:200

C(Config) # router bgp 65000

C(Config-router) # neighbor PE2 send-community

GW(config) # route-map z permit

GW(Config-route-map) # match community 2:200

GW(Config-route-map) # set community no-export

PEs(config) # route-map w permit

PEs(Config-route-map) # match ip address 1

PEs(Config-route-map) # set community 2:200 additive

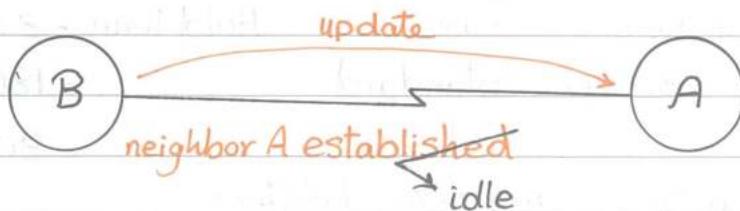
Add new community  
to old community

### Session 15 Part 3

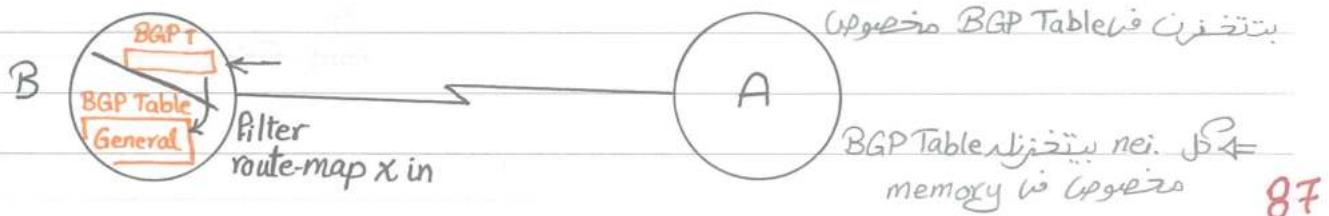
For any BGP Policy to be applied we need to clear bgp session with neighbor.

\* Option 1 : Hard clear

# clear ip bgp {neighbor | \* }  
↓  
All neighbors

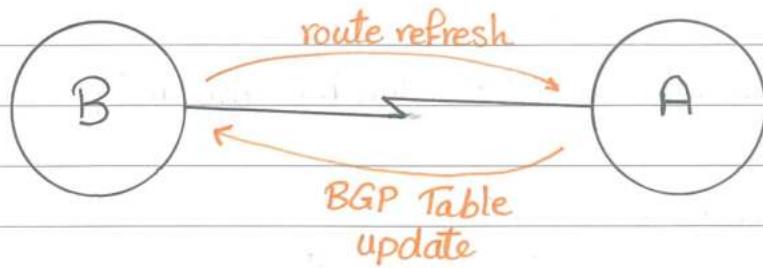


\* Option 2 : soft clear (Config-router) # neighbor A soft-reconfiguration in  
# clear ip bgp neighbor soft in



### \* Option 3 : Route Refresh

# clear ip bgp neighbor soft in



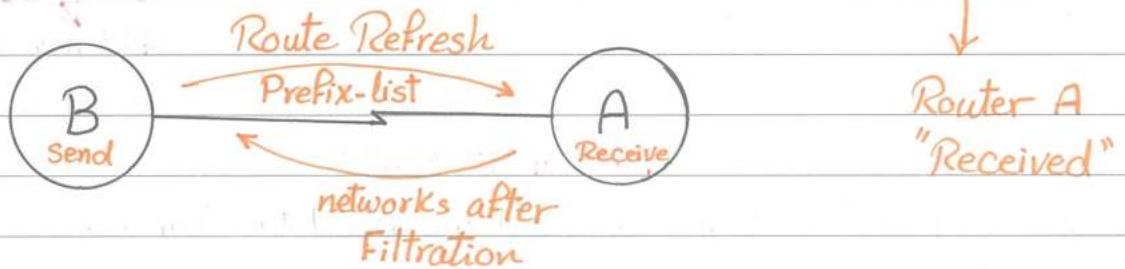
### \* Option 4 : ORF (Outbound Route Filter)

B (Config-router) # neighbor A capability orf Prefix-list

B (Config-router) # neighbor A prefix-list x in

# clear ip bgp neighbor prefix-list in

send
Receive
both



## Session 15 Part 4 BGP Convergence

- Neighborship :

Keepalive every 60 sec Cisco Hold time = 3 x Keepalive  
30 sec standard = 180 sec Cisco

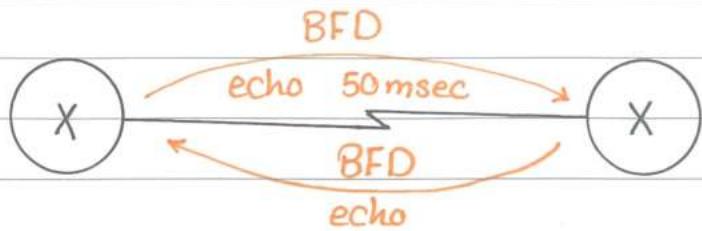
(Config-router) # bgptimer keepalive holdtime

OR

(Config-router) # neighbor timer keepalive holdtime

↓  
min 5sec

## BFD (Bidirectional Forward Detection)



(Config)# router bgp 1

(Config-router) # neighbor B full-over bfd

OR

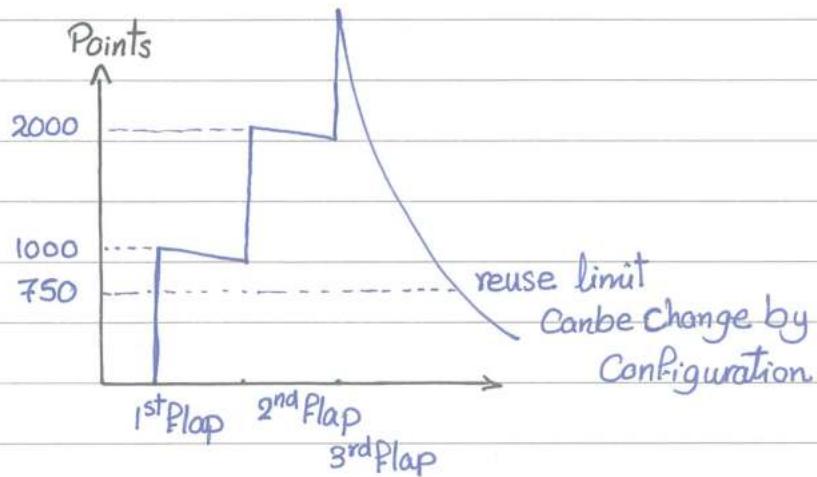
(Config-if) # bfd interval \_\_\_\_\_ min-rx \_\_\_\_\_ multiplier \_\_\_\_\_  
in msec                    msec                    times  
echo send                echo receive            "hold time"

### - Update :

BGP Scan Time: BGP scan next hop in BGP Table once every 60 sec.

## BGP Dampening :

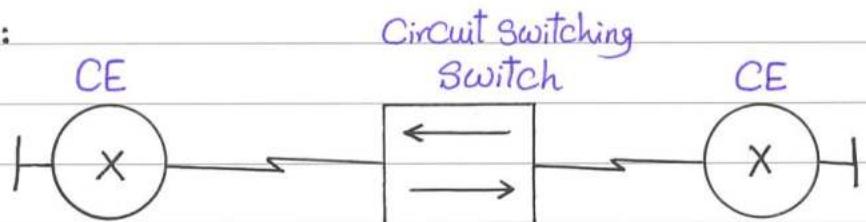
Failed route get 1000 point penalty (Config-router) # bgp dampening  
Suppress limit 2000 ... max Suppress limit 3000 ...



## Session 16

### Private WAN

- Circuit Switching:



ex: HDLC (Cisco)  
PPP (standard)

Point to Point Topology.

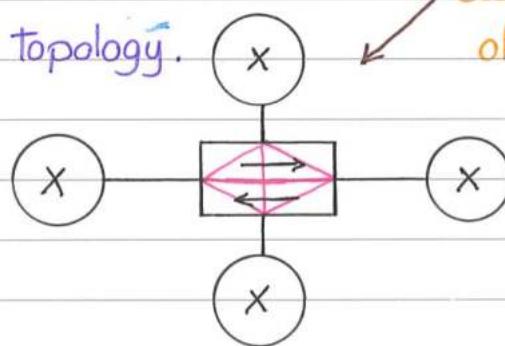
Address: no address.

Flag	Address	Control	Type	Packet	CRC
.....					

- Packet Switching:

Flexible Topology.  
 Point-to-point  
 Point-to-Multipoint  
 Multipoint-to-Multipoint

legacy topology  
 Switch For thousands  
 of Frames per sec



Address: x.25 no. (8bit) → x.25 switch

DLCI (10 bit) → FR switch

VPI/VCI → ATM switch

- Ethernet Switching:

Low latency (low delay).

Switch Million of packets (frames) per second.

Flexible topology.

Address = 48bit (MAC) ex: Metri Ethernet.

- Label switching:

low latency.

Switch tens of Millions of packet per second.

Flexible topology.

Address: Label (20bit).

ex: MPLS (Multi Protocol Label Switching).

## Session 16 Part 2

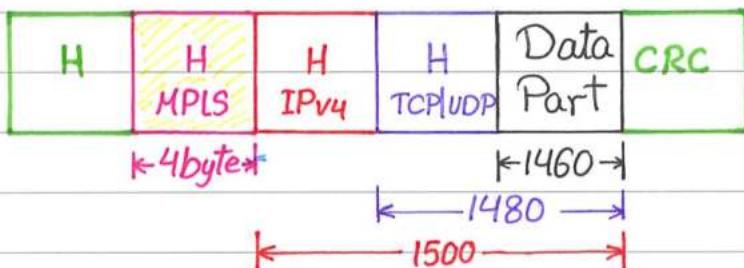
### Multi Protocol Label switching

#### MPLS

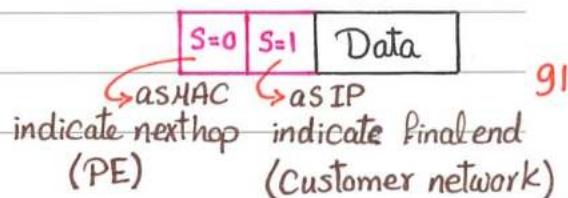
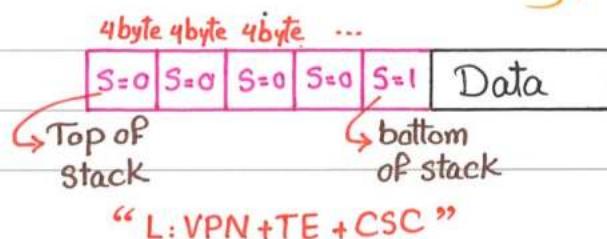
It is a new Forwarding technique based on labels.

MPLS header is imposed between L2 & L3 headers.

L2.5 header (shim header)

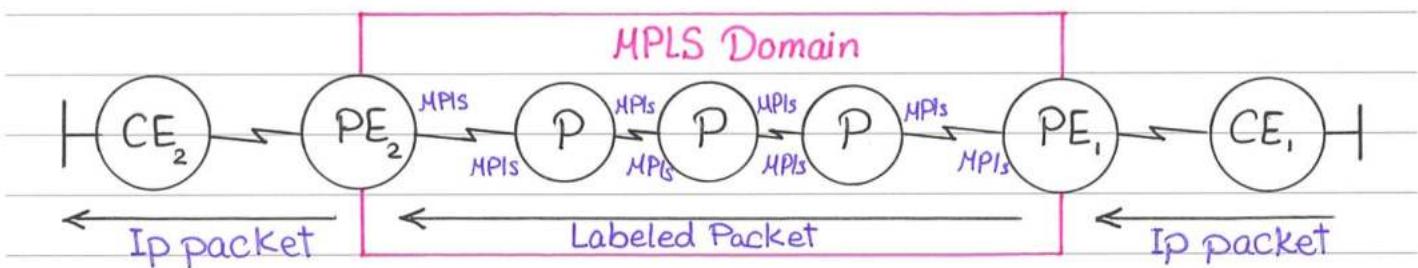


Label 20-bit	Experimental bit (Traffic class) 3-bits	Bottom of stack 1-bit	hop limit (TTL) 8-bits
0-15 Reserved	Quality of service (QoS) 0,1,2,3 Data 4 video, 5 voice 6 Routing, 7 Switching	1 ≡ 1 label 0 ≡ many labels label	hop = any device can understand MPLS (R,S). avoid loops



## Session 16 Part 3

### MPLS Terminologies



Edge LSR  $\equiv$  LER  $\equiv$  label edge router

ex: PE<sub>1</sub>, PE<sub>2</sub>

(Config)# ip cef

(Config)# interface \_\_\_\_\_

(Config-if)# mpls ip

Router that receives IP packet & transfer it as a labeled packet & Vice Versa.

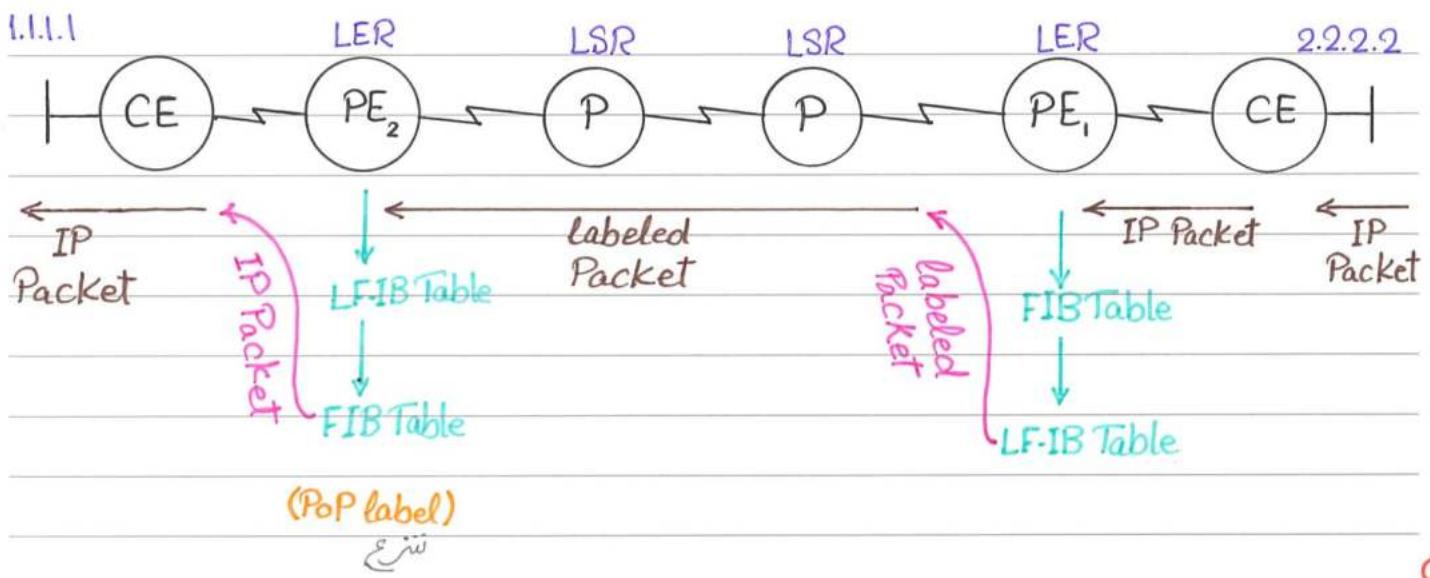
LSR = label switch router

Receive labeled Packet & send labeled packet. ex: P

LSP  $\equiv$  label switch path

It is the best path that data will use using labels.

- Packet Forwarding in MPLS domain:



## Session 16 Part 4

### MPLS Tables needs a Signaling protocols

<b>LDP</b> Label Distribution Protocols (standard) Port 646, UDP & TCP	<b>TDP</b> Tag Distribution Protocols (Cisco) Port 711, UDP & TCP
---	--

- LDP operation:

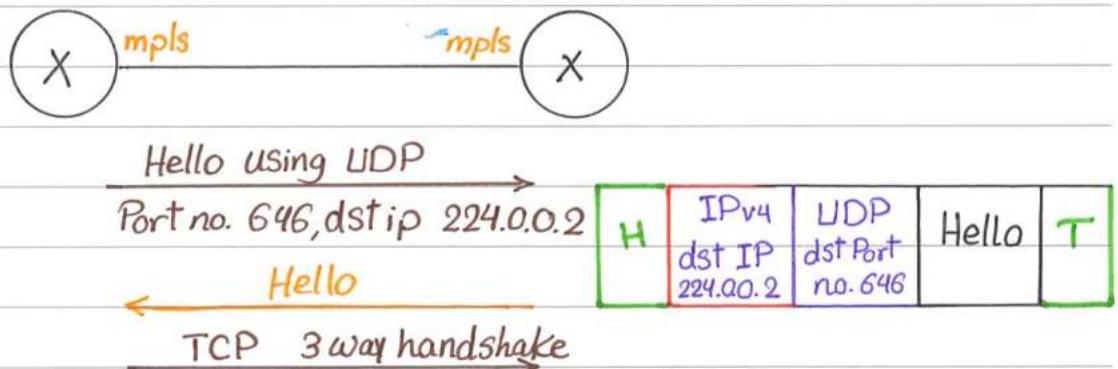
☒ Router ID:

- Manual.

- Highest Loopback IP.

- Highest physical active interface IP.

☒ Neighbor Discovery:



Initialization msg (using TCP)

← Initialization

☒ Routes Discovery (Label Discovery):

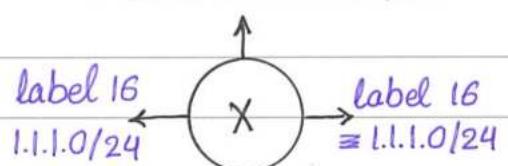
label 16 = 1.1.1.0/24

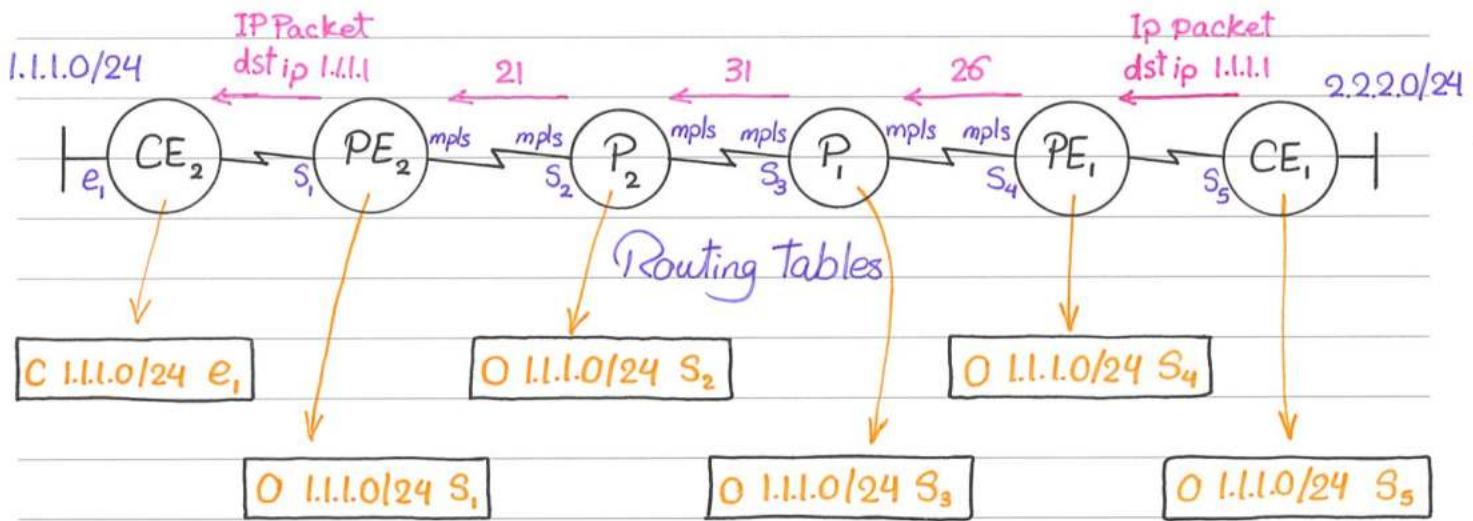
routes to label mapping.

using per-platform label distribution.

all interfaces advertise same

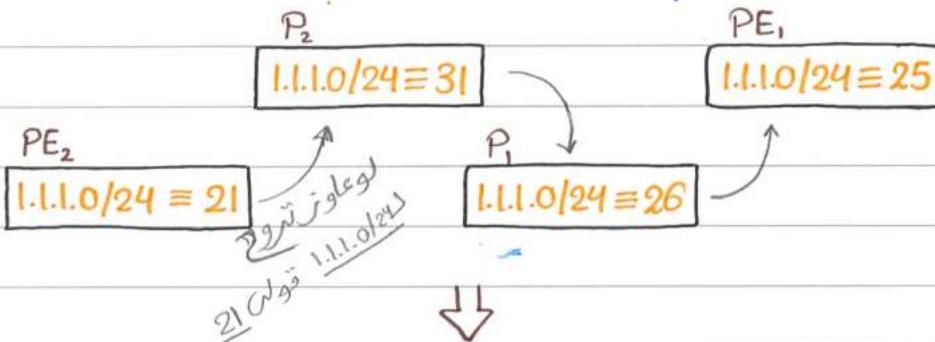
IP to label mapping...





LIB Table

(label Info. Base)



LF-IB Table

(label Forward Info Base)

PE <sub>2</sub>		
In label	Outlabel	vector
21	untagged	$S_1$

P <sub>1</sub>		
In label	outlabel	vector
26	31	$S_3$

P <sub>2</sub>		
In label	outlabel	vector
31	21	$S_2$

PE <sub>1</sub>		
In label	out label	vector
untagged	26	$S_4$

update



Data

POP → PE<sub>2</sub> → P<sub>2</sub> → 3 (1.1.1.0/24) ← لواعز تردد → label 3 ≡ untagged ...

## Session 17

### MPLS

LDP Protocol: @startup

## LDP Router ID:

## 0.1 Manual:

(Config) # mpls ldp router-id interface name

## 0.2 Automatic:

- Highest Loopback IP .
  - Highest Physical\*Interface (Active) IP .

## ① LDP Neighbor Discovery:

exchange of hello (dst IP: 224.0.0.2, dst Port: 646).

## ② LDP Label Discovery:

exchange of labels.

⇒ Label Conditional advertisement:

(Config) # mpls ldp advertise-label [for \_\_\_\_\_ to \_\_\_\_\_]  
ACL Prefixes ↕ ACL neighbor

## Routing Table

1

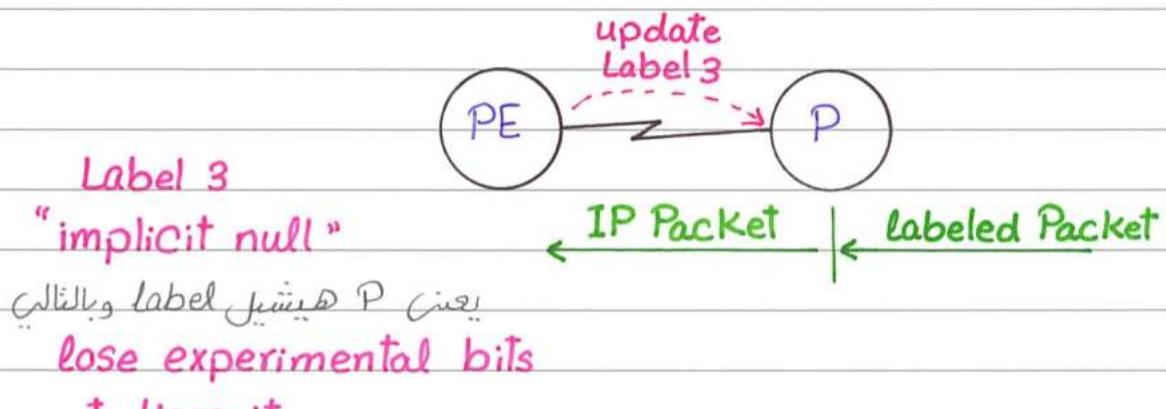
## LIB Table (label Info Base)

↓

## LFIB Table (Label Forward Info Base)

## Best path using labels

## ⇒ Penultimate Hop Popping (PHP) :

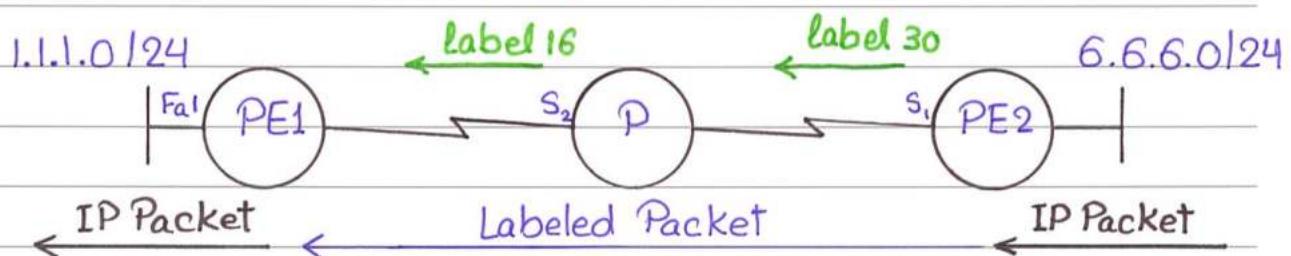


(Config) ~~mpls ldp~~ explicit-null  
Label 0

## — MPLS Application —

### 1. Unicast Routing :

FEC (Forward Equivalent Class) = DST IP unicast = label  
only concern label 11



PE1 1.1.1.0/24

PE2 1.1.1.0/24

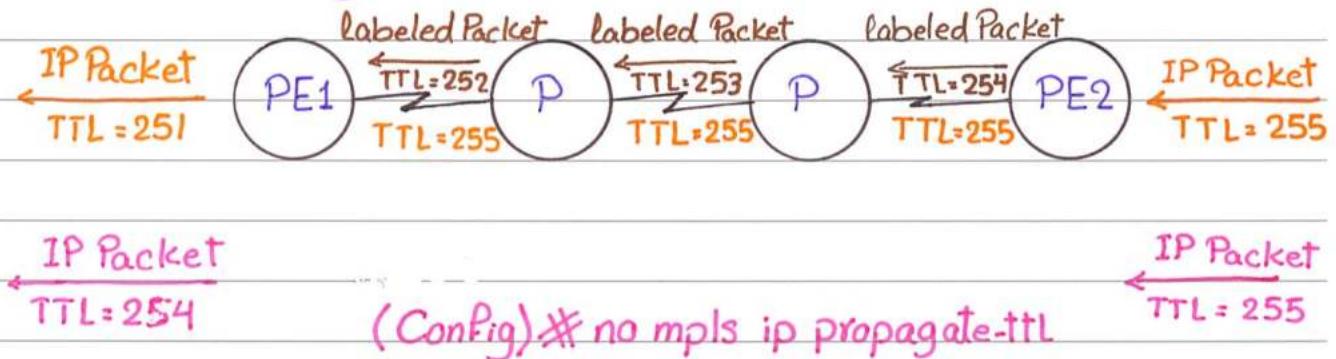
In label	Outlabel	Interface
16	Untagged	Fa 1

In label	Outlabel	Interface
untagged	30	S <sub>1</sub>

P 1.1.1.0/24

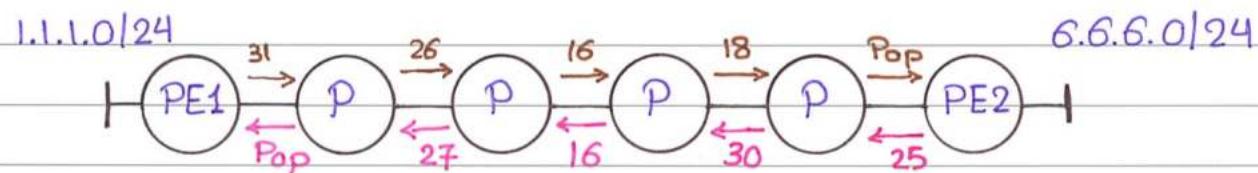
In label	Outlabel	Interface
30	16	S <sub>2</sub>

## ⇒ TTL Propagation:



## Session 17 Part 2

### ⇒ LSP Label Switch Path:



LSP = 25, 30, 16, 27 then pop

LSP = 31, 26, 16, 18 then pop

— Build by LDP

*(Config-iP) # mpls ip*

### 2. Multicast IP Routing:

FEC = DST IP Multicast

41.1.1.1



Class D

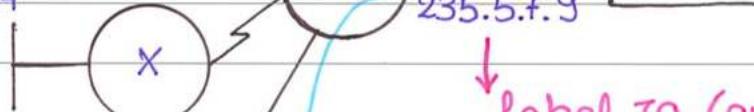
239.x.x.x

224.x.x.x

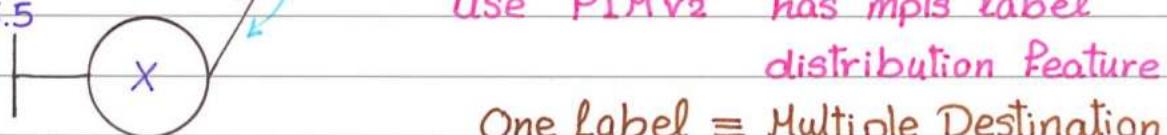
57.3.1.1



173.2.5.4



68.2.5.5



data

dst ip

235.5.7.9

label 70 (one send & many received)

use PIMv2 has mpls label

distribution feature

One Label = Multiple Destination

## Routing Table

41.1.1.0	S1
57.3.1.0	S2
173.2.5.4	S3
68.2.5.0	S4

PIHv9

multicast routing table

238.5.7.9 S1,S2,S4

# Show ip mroute



## 3. QoS Quality of Service :

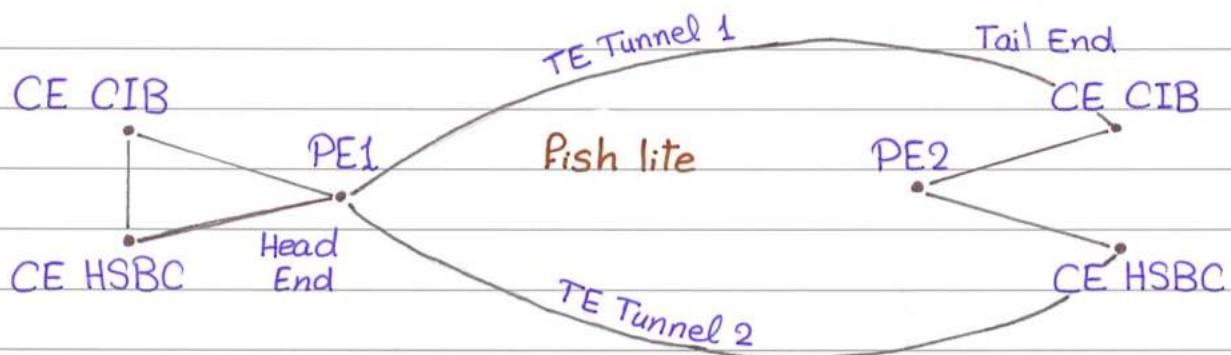
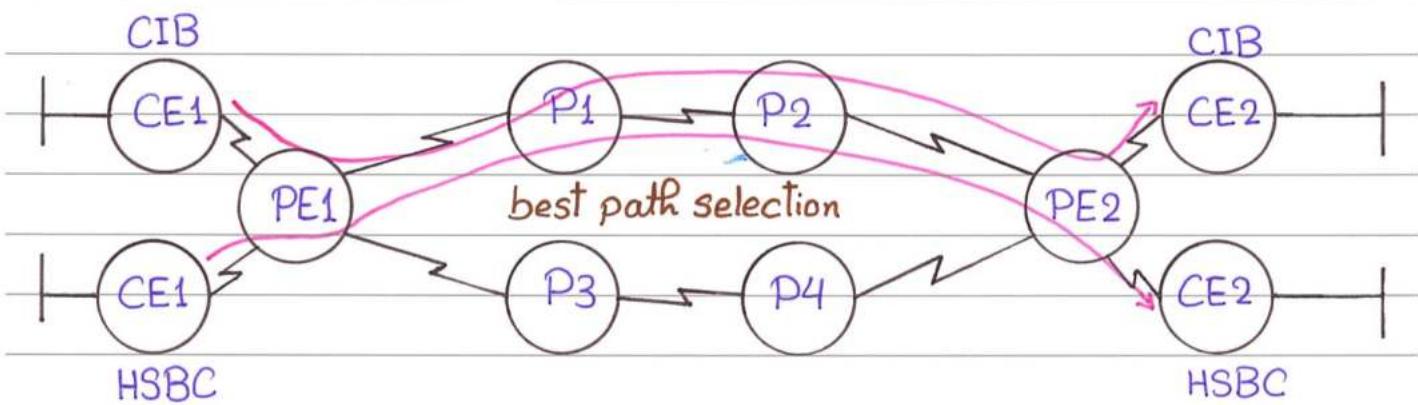
FEC  $\equiv$  Class of Service .

Label reflect traffic class .

## Session 17 Part 3

### 4. MPLS Traffic Engineering :

Using all available resources (dynamic TE)



MPLS need 2 labels & sometimes 3 labels

Use LDP  $\rightarrow$  head end / Tail end

RSVP  $\equiv$  Resource Reservation Protocol  $\longrightarrow$  Build Tunnels .

OSPF or ISIS (as must use one of them)

to can detect the below:

- Maximum BW
- Available BW
- Used BW
- Reserved BW
- Priority

⇒ OSPF v2 :

- Type 9 LSA , Type 10 LSA & Type 11 LSA  
is called .. Opaque LSA .. used with  
MPLS TE Contains info above .

9. Link local    10. Intra Area    11. AS-wide  
between 2 Routers    Area local

- Type 8 LSA + Type 5 LSA  
when using BGP

8. hold BGP attributes.    5. NLRI .

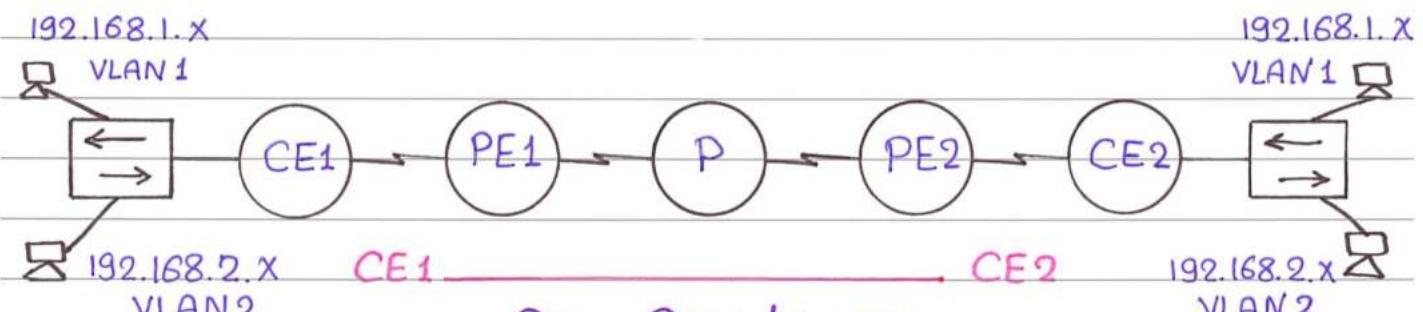
} not used

⇒ ISIS :

TLV 22, TLV 135

### Session.17 Part 4

#### 5. Layer 2 VPN :



• Pw = Pseudo wire

• VPLS = Virtual Private LAN Service

• EVN = Easy Virtual Network

## 6. Layer 3 VPN :

.. First we need to classify VPN ..

### Session 17 Part 5 VPN Classification

#### - Intranet VPN :

VPN between different branches of same company.

#### - Extranet VPN :

VPN between different sites of different companies.

#### - Access VPN :

VPN using dial up or broadband or 4G network.  
VPDN ≡ Virtual Private Dial up Network .

#### - Overlay VPN:

Service provider doesn't participate in customer routing.  
Service provider participates in customer routing.

#### - Tunneling :

• L2 tunneling:  
L2TP, L2F, PPPP  
• L3 tunneling:  
GRE, IPsec.

#### - Simple VPN :

All customer site share same VPN.

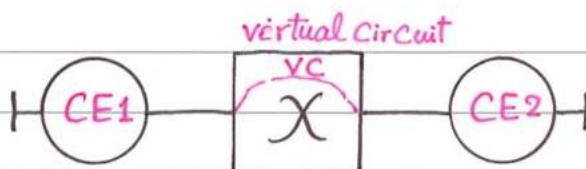
#### - Complex VPN :

Customer sites can share many different VPN.  
• Overlapping VPN.  
• Centralised Service VPN.  
• Management VPN.

#### - Hierarchical VPN :

• Inter AS VPN.  
• CSC ≡ Carrier Supporting Carrier .

#### → Overlay VPN :

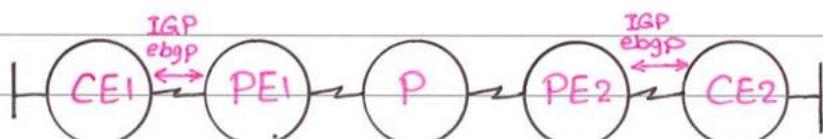


disadvantage .

- Control .

- Security .

#### → Peer to Peer VPN :



need alot of packets Filter & route Filter ...

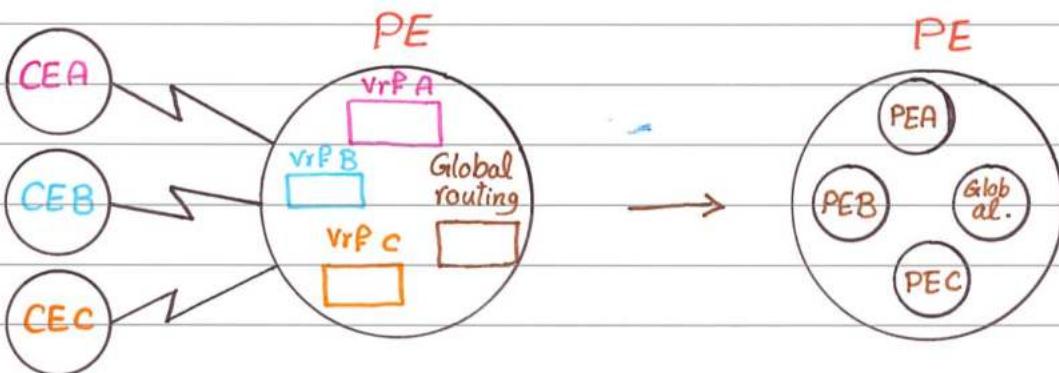
## Session 18

### MPLS L3 VPNs

We can implement :

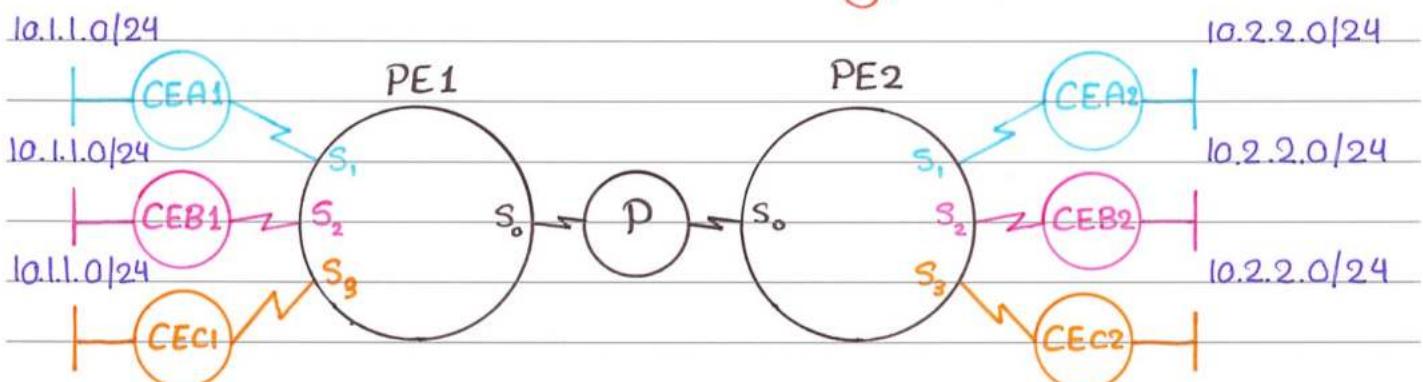
- Intranet VPN (Sites of same enterprise).
- Extranet VPN (Sites of different companies).
- Overlay VPN (IP address overlapping).
- Peer-to-peer VPN.
- Simple VPN.
- Complex VPN.
- Hierarchical VPN.
- Tunneling.

VRF (Virtual Routing & Forwarding) :



## Session 18 Part 2

### MPLS L3 VPNs using VRFs



On PE1:

- Create VRF :

(Config) # ip vrf A

optional

(Config-vrf) # vpn id OUI : VPN Index

RD = Route Distinguisher

used to separate IP network

overlapping, will create

new address-family VPNv4

RD : IP Address = 96 bits

(Config-vrf) # rd 2:1

(Config-vrf) # route-target 2:100

- Activate VRF :

(Config) # interface S1

(Config-if) # ip vrf forwarding A

(Config-if) # ip address \_\_\_\_\_

(Config) # ip vrf B

(Config-vrf) # vpn id \_\_\_\_\_

(Config-vrf) # rd 2:2

(Config-vrf) # route-target 2:200

(Config) # interface S2

(Config-if) # ip vrf forwarding B

(Config-if) # ip address \_\_\_\_\_

Route-Target Option :

receive

send

(Config-vrf) # route-target [import] export | both ] \_\_\_\_\_

by default ↴

. Selective route target

On PE2:

- Import RT :

(Config) # ip vrf A

(Config-vrf) # rd 2:1

(Config-vrf) # route-target 2:100

(Config-vrf) # import map x

(Config) # route-map x permit

(Config-route-map) # match ip address 1

(Config) # access-list 1 permit 10.1.1.0

- Export RT :

(Config) # ip vrf B

(Config-vrf) # rd 2:2

(Config-vrf) # export map y

(Config-vrf) # route-target 2:1000

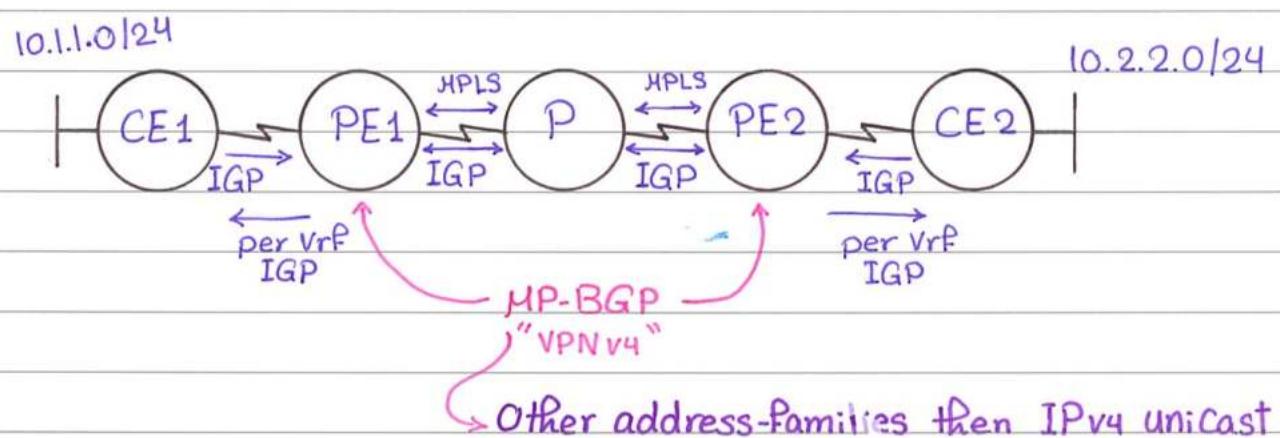
(Config) # route-map y permit

(Config-route-map) # match ip address 2

(Config-route-map) # set extcommunity rt 2:200 additive

(Config) # access-list 2 permit 10.2.2.0

### Session 18 Part 3 MPLS L3VPN routing Scheme



VPNv4 Update:

NLRI + Attributes

RD : Network/mask, RT & VPN Label &  
SoO Site of Origin (optional)

ex: 2:1 : 10.1.1.0/24 , RT = 2:100 , VPN label = 16

2:1 : 10.3.3.0/24 , RT = 2:100 , VPN label = 17

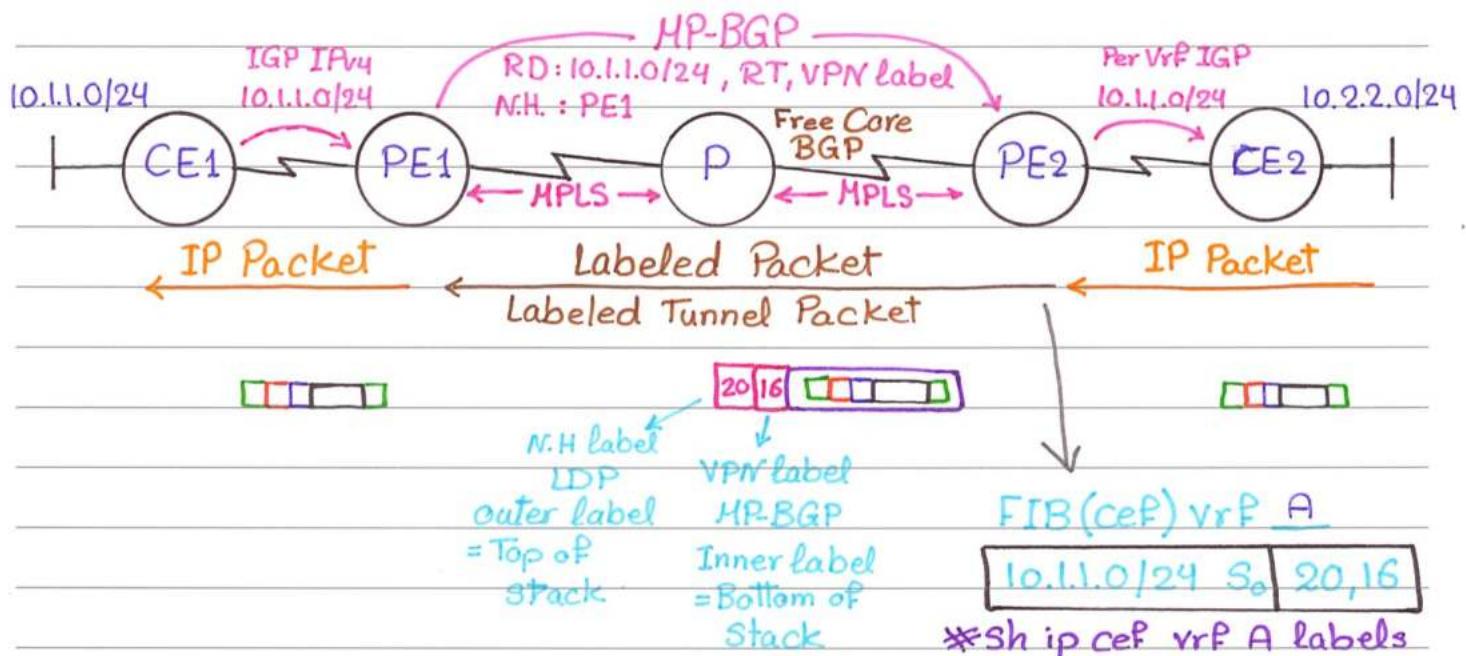
CE1 Routing Table:

C 10.1.1.0 /24 e <sub>1</sub>
R 10.2.2.0 /24 S <sub>1</sub>

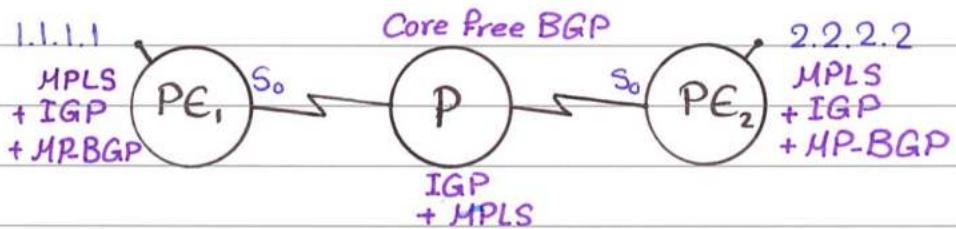
CE2 Routing Table:

C 10.2.2.0 /24 e <sub>2</sub>
R 10.1.1.0 /24 S <sub>0</sub>

L3VPN Target



## Session 18 Part 4 PE to PE routing

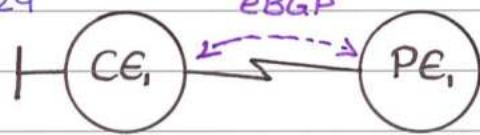


PE (Config) # interface S0  
 PE (Config-if) # mpls ip  
 PE (Config-if) # mpls label protocol ldp

PE (Config) # router bgp 2  
 PE (Config-router) # neighbor 2.2.2.2 remote-as 2  
 PE (Config-router) # neighbor 2.2.2.2 update-source loopback 0  
 PE (Config-router) # address-family VPNv4  
 PE (Config-router-af) # neighbor 2.2.2.2 activate  
 PE (Config-router-af) # neighbor 2.2.2.2 next-hop-self  
 PE (Config-router-af) # neighbor 2.2.2.2 send-community extended

## PE to CE routing

10.1.1.0/24



PE (Config) # router bgp 2

PE, (Config-router) # neighbor CE1 remote-as 65000

PE (Config-router) # address-family ipv4 vrf A

PE, (Config-router-af) # neighbor CE1 activate

PE (Config-router-af) # exit

PE (Config-router) # no bgp default ipv4-unicast

PE (Config-router) # address-family ipv4 vrf A

PE, (Config-router-af) # neighbor CE1 as\_override

The same Configuration on PE<sub>2</sub>, just change

IP From 2.2.2.2 → 1.1.1.1 & CE<sub>1</sub> → CE<sub>2</sub>

CE, (Config) # router bgp 65000 -

CE, (Config-router) # neighbor 1.1.1.1 remote-as 2

CE, (Config-router) # network 10.1.1.0 mask 255.255.255.0

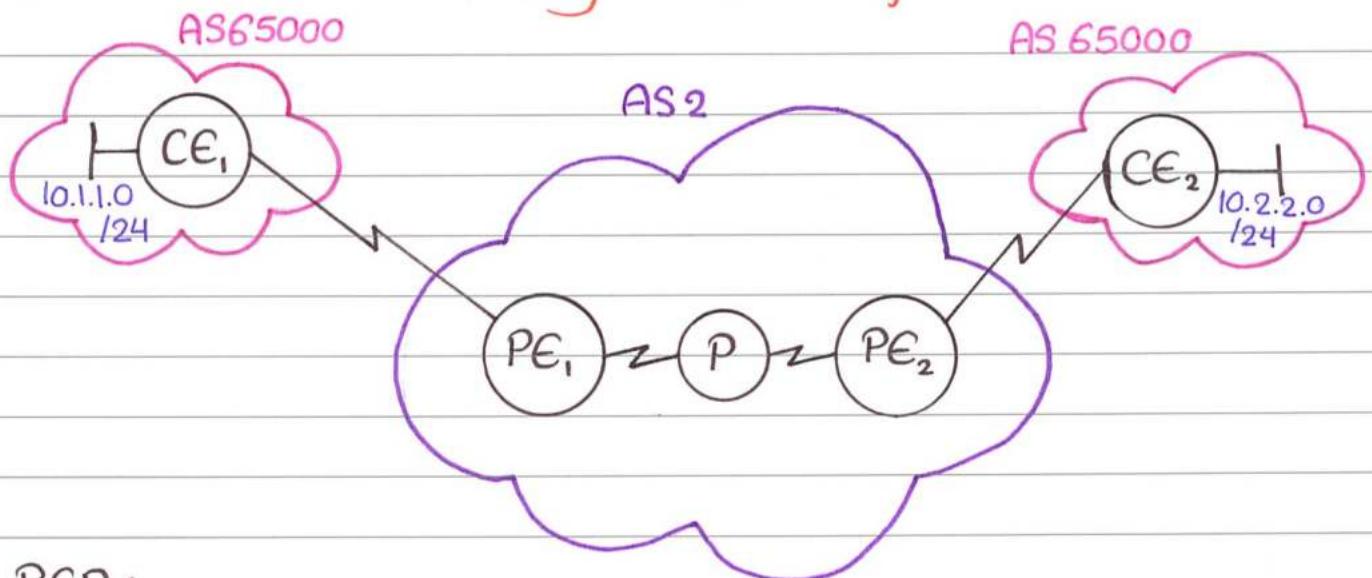
CE<sub>2</sub> (Config) # router bgp 65000

CE<sub>2</sub> (Config-router) # neighbor 2.2.2.2 remote-as 2

CE<sub>2</sub> (Config-router) # network 10.2.2.0 mask 255.255.255.0

## Session 19

### Routing between CE & PE



- BGP :

```
CE1 (Config) # router bgp 65000
```

```
CE1 (Config-router) # neighbor PE1 remote-as 2
```

```
CE1 (Config-router) # network 10.1.1.0 mask 255.255.255.0
```

```
PE1 (Config) # router bgp 2
```

```
PE1 (Config-router) # address-family ipv4 vrf A
```

```
PE1 (Config-router-af) # neighbor CE1 remote-as 65000
```

```
PE1 (Config-router-af) # neighbor CE1 as_override
```

```
PE1 (Config-router-af) # exit
```

```
PE1 (Config-router) # neighbor PE2 remote-as 2
```

```
PE1 (Config-router) # address-family vrf v4
```

```
PE1 (Config-router-af) # neighbor PE2 activate
```

```
PE2 (Config) # router bgp 2
```

```
PE2 (Config-router) # address-family ipv4 vrf A
```

```
PE2 (Config-router-af) # neighbor CE2 remote-as 65000
```

```
PE2 (Config-router-af) # neighbor CE2 as_override
```

```
PE2 (Config-router-af) # exit
```

```
PE2 (Config-router) # neighbor PE1 remote-as 2
```

PE<sub>2</sub> (Config-router) # address-family Vpn v4

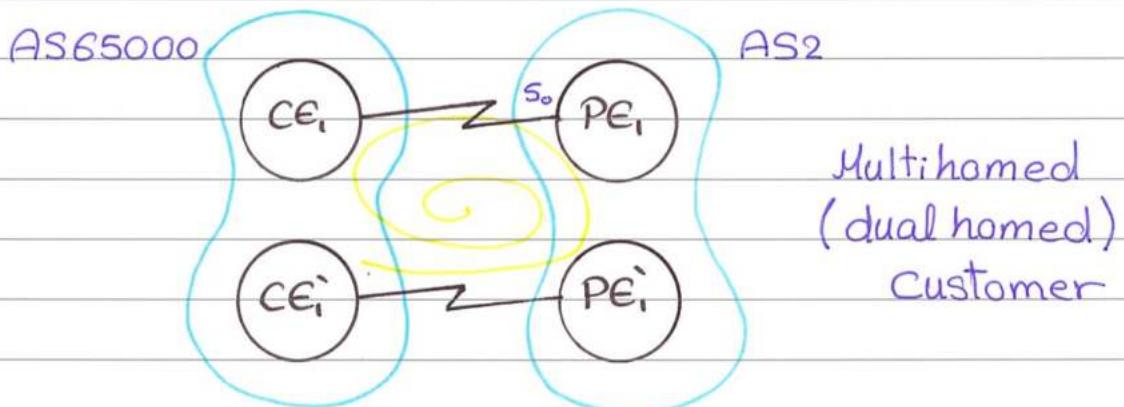
PE<sub>2</sub> (Config-router-af) # neighbor PE<sub>2</sub> activate

CE<sub>2</sub> (Config) # router bgp 65000

CE<sub>2</sub> (Config-router) # neighbor PE<sub>2</sub> remote-as 2

CE<sub>2</sub> (Config-router) # network 10.2.2.0 mask 255.255.255.0

### Site of Origin (SOO)



To avoid loops:

PE<sub>1</sub> (Config) # router bgp 2

PE<sub>1</sub> (Config-router) # address-family ipv4 vrf A

PE<sub>1</sub> (Config-router-af) # neighbor CE<sub>1</sub> sooo 2:100

OR

PE<sub>1</sub> (Config) # route-map x permit

PE<sub>1</sub> (Config-route-map) # match ip address 1

PE<sub>1</sub> (Config-route-map) # set extcommunity sooo 2:100

PE<sub>1</sub> (Config) # access-list 1 permit 10.1.1.0

PE<sub>1</sub> (Config) # interface S0

PE<sub>1</sub> (Config-if) # ip vrf Forwarding A

PE<sub>1</sub> (Config-if) # ip address \_\_\_\_\_

PE<sub>1</sub> (Config-if) # ip vrf sitemap x

## Session 19 Part 2

- Static :

CE, (Config) # ip route 0.0.0.0 0.0.0.0 S.

PE, (Config) # ip route vrf A 10.1.1.0 255.255.255.0 10.5.5.1

PE, (Config) # router bgp 2

PE, (Config-router) # address-family ipv4 vrf A

PE, (Config-router-af) # redistribute static

- EIGRP :

CE, (Config) # router eigrp 65000

CE, (Config-router) # network 10.1.1.0

PE, (Config) # router eigrp 2

PE, (Config-router) # address-family ipv4 vrf A

PE, (Config-router-af) # autonomous-system 65000

PE, (Config-router-af) # redistribute bgp 2 metric BDRLH

PE, (Config-router-af) # network .

PE, (Config) # router bgp 2

PE, (Config-router) # address-family ipv4 vrf A

PE, (Config-router-af) # redistribute eigrp 65000

## Session 19 Part 3

- RIPv2 :

CE, (Config) # router rip

CE, (Config-router) # version 2

CE, (Config-router) # network 10.1.1.0

PE, (Config) # router rip

PE, (Config-router) # version 2

PE, (Config-router) # address-family ipv4 vrf A

PE, (Config-router) # network \_\_\_\_\_

PE, (Config-router) # redistribute bgp 2 metric transparent

PE, (Config) # router bgp 2

PE, (Config-router) # address-family ipv4 vrf A

PE, (Config-router) # redistribute rip

- OSPF v2 :

CE, (Config) # router ospf 1

CE, (Config-router) # network \_\_\_\_\_ area 1

PE, (Config) # router ospf 1 vrf A

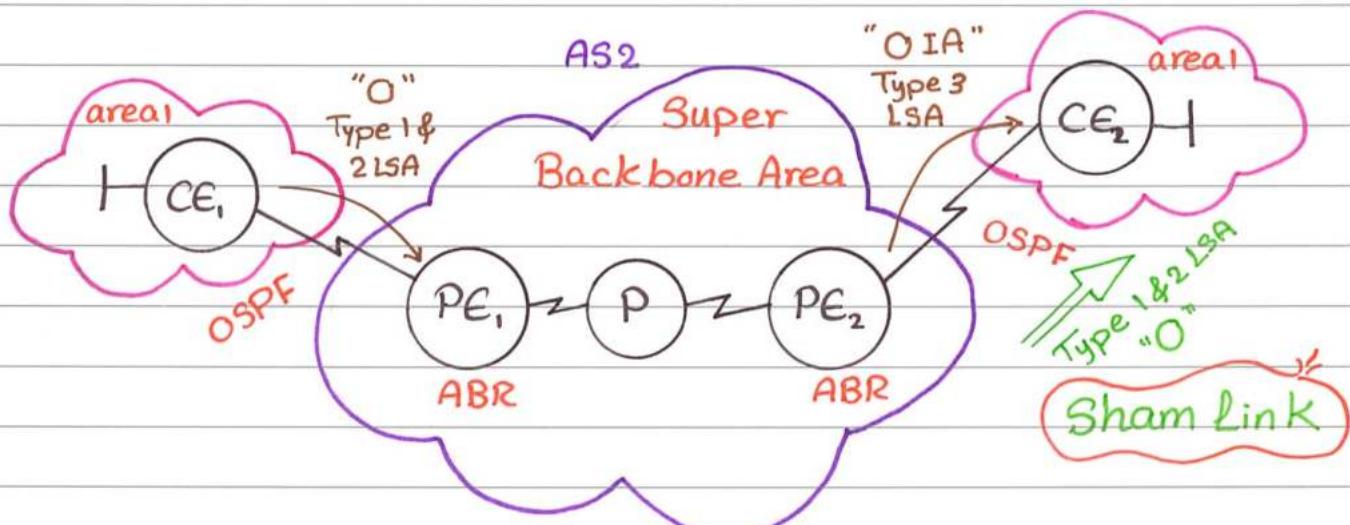
PE, (Config-router) # network 10.5.5.2 0.0.0.0 area 1

PE, (Config-router) # redistribute bgp 2 subnets

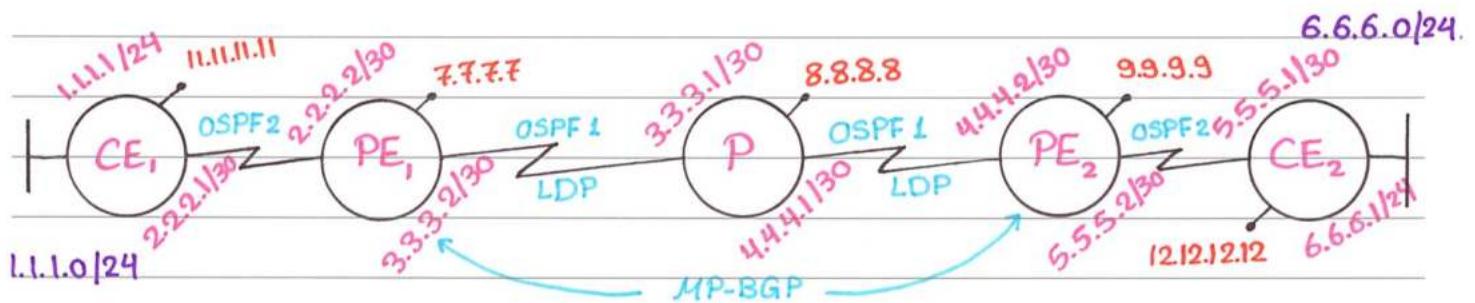
PE, (Config) # router bgp 2

PE, (Config-router) # address-family ipv4 vrf A

PE, (Config-router-af) # redistribute ospf 1 vrf A



## Session 20



P # Show ip route

O 1.1.1.0 /24	110/3	3.3.3.2	Fast Ethernet 0/0
O 2.2.2.0 /30	110/2	3.3.3.2	Fast Ethernet 0/0
C 3.3.3.0 /30	directly connected		Fast Ethernet 0/0
C 4.4.4.0 /30	directly Connected		Fast Ethernet 0/1
O 5.5.5.0 /30	110/11	4.4.4.2	Fast Ethernet 0/1
O 6.6.6.0 /24	110/21	4.4.4.2	Fast Ethernet 0/1
O 7.7.7.7 /32	110/2	3.3.3.2	Fast Ethernet 0/0
C 8.8.8.8 /32	directly Connected		Loopback 0
O 9.9.9.9 /32	110/2	4.4.4.2	Fast Ethernet 0/1
O 11.11.11.11 /32	110/3	3.3.3.2	Fast Ethernet 0/0
O 12.12.12.12 /32	110/12	4.4.4.2	Fast Ethernet 0/1

P # Conf t

P(Config) # int F0/0

P(Config-if) # mpls ip

P(Config-if) # mpls label protocol ?

both use LDP or TDP

ldp use LDP (default)

tdp use TDP

P(Config-if) # int F0/1

P(Config-if) # mpls ip

P# 7.7.7.7

Trying 7.7.7.7 ... open

PE1 # Conf t

PE1(Config) # int F0/0

PE1(Config-if) # mpls ip

PE1 # 9.9.9.9

Trying 9.9.9.9 ... open

PE2 # Conf t

PE2(Config) # int F0/1

PE2(Config-if) # mpls ip

P# Show mpls interfaces

Interface	IP	Tunnel	Operational
FastEthernet 0/0	Yes (ldp)	No	Yes
FastEthernet 0/1	Yes (ldp)	No	Yes

P# show mpls ldp neighbor

Peer LDP Ident	Local LDP Ident	TCP Connection
7.7.7.7 : 0	8.8.8.8 : 0	7.7.7.7.646 - 8.8.8.8.24997
9.9.9.9 : 0	8.8.8.8 : 0	9.9.9.9.11726 - 8.8.8.8.646

P# show mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Outgoing interface	Next Hop
20	Pop tag	7.7.7.7/32	Fa0/0	3.3.3.2
21	Pop tag PHP	9.9.9.9/32	Fa0/1	4.4.4.2

P \* 7.7.7.7

Trying 7.7.7.7 ... open

PE1 # Conf +

PE1 (Config) # ip vrf a

PE1 (Config-vrf) # rd 1:1

PE1 (Config-vrf) # route-target ?

both import & export

export export

import import

PE1 (Config-vrf) # route-target import 2:1

PE1 (Config-vrf) # import map x

PE1 (Config-vrf) # export map y

PE1 (Config-vrf) # exit

PE1 (Config) # access-list 6 permit 6.6.6.0

PE1 (Config) # access-list 1 permit 1.1.1.0

PE1 (Config) # route-map x

PE1 (Config-route-map) # match ip address 6

PE1 (Config-route-map) # exit

PE1 (Config) # route-map y

PE1 (Config-route-map) # match ip address 1

PE1 (Config-route-map) # set extcommunity ?

rt Route Target extended Community

soo Site-of-Origin extended Community

PE1 (Config-route-map) # set extcommunity rt 2:2 additive

PE1 (Config-route-map) # do show run

ip vrf a

rd 1:1

import map x

export map y

route-target import 2:1

PE1 (Config-route-map) # do telnet 9.9.9.9

Trying 9.9.9.9 ... Open

PE2 # Config t

PE2 (Config) # ip vrf a

PE2 (Config-vrf) # rd 1:1

PE2 (Config-vrf) # route-target export 2:1

PE2 (Config-vrf) # route-target import 2:2

PE2 (Config-vrf) # exit

PE2 (Config) # exit

PE2 # exit

PE1 (Config-route-map) # do show route-map

route-map X, permit, Sequence 10

Match clauses :

ip address (access-list) : 6

route-map Y, permit, Sequence 10

Match clauses :

ip address (access-list) : 1 -

Set clauses :

extended community RT: 2:2 additive

PE1 (Config-route-map) # exit

PE1 (Config) # int F0/1

PE1 (Config-if) # ip vrf forwarding a

PE1 (Config-if) # ip address 2.2.2.2 255.255.255.252

PE1 (Config-if) # router bgp 2

PE1 (Config-router) # neighbor 9.9.9.9 remote-as 2

PE1 (Config-router) # neighbor 9.9.9.9 update-source Lo

PE1 (Config-router) # address-family vpnv4

PE1 (Config-router-af) # neighbor 9.9.9.9 activate

PE1 (Config-router-af) # neighbor 9.9.9.9 next-hop-?

next-hop-self

next-hop-unchanged

PE1 (Config-router-af) # neighbor 9.9.9.9 next-hop-self

PE1 (Config-router-af) # ^Z

PE1 # 9.9.9.9

Trying 9.9.9.9 ... open

PE2 # Config +

PE2 (Config) # router bgp 2

PE2 (Config-router) # neighbor 7.7.7.7 remote-as 2

PE2 (Config-router) # neighbor 7.7.7.7 update-source Lo

PE2 (Config-router) # address-family vpnv4

PE2 (Config-router-af) # neighbor 7.7.7.7 activate

PE2 (Config-router-af) # neighbor 7.7.7.7 next-hop-self

PE2 (Config-router-af) # ^Z

PE2 # Show ip bgp summary

Neighbor AS PfxRcd

7.7.7.7 2 0

PE2 # exit

PE1 # Config +

PE1 (Config) # router ospf 2 vrf a

PE1 (Config-router) # network 2.2.2.2 0.0.0.0 area 0

PE1 (Config-router) # do telnet 2.2.2.1 /vrf a

Trying 2.2.2.1 ... open

CE1 # Show run

router ospf 1

log-adjacency-changes

network 0.0.0.0 255.255.255.255 area 0

CE1 # exit

PE1 (Config-router) # redistribute bgp 2 subnets

PE1 (Config-router) # exit

PE1 (Config) # router bgp 2

PE1 (Config-router) # address-family ipv4 vrf a

PE1 (Config-router-af) # redistribute ospf 2 vrf a

PE1 (Config-router-af) # do telnet 9.9.9.9

Trying 9.9.9.9 ... open

PE2 # Config +

PE2 (Config) # do show run

ip vrf a

rd 1:1

route-target export 2:1

route-target import 2:2

PE2 (Config) # int F0/0

PE2 (Config-if) # ip vrf forwarding a

PE2 (Config-if) # ip address 5.5.5.2 255.255.255.252

PE2 (Config-if) # router ospf vrf a

PE2 (Config-router) # network 5.5.5.2 0.0.0.0 area 0

PE2 (Config-router) # redistribute bgp 2 subnets

PE2 (Config-router) # exit

PE2 (Config) # router bgp 2

PE2 (Config-router) # address-family ipv4 vrf a

PE2 (Config-router-af) # redistribute ospf 2 vrf a

PE2 (Config-router-af) # ^Z

PE2 #

PE2 # show ip route

O	3.3.3.0 /30	110/2	4.4.4.1	Fast Ethernet 0/1
---	-------------	-------	---------	-------------------

C	4.4.4.0 /30	directly connected		Fast Ethernet 0/1
---	-------------	--------------------	--	-------------------

O	7.7.7.7 /32	110/3	4.4.4.1	Fast Ethernet 0/1
---	-------------	-------	---------	-------------------

O	8.8.8.8 /32	110/2	4.4.4.1	Fast Ethernet 0/1
---	-------------	-------	---------	-------------------

C 9.9.9.9

directly connected

Loopback 0

PE2 # show ip route vrf a

B 1.1.1.0 /24 200/2 7.7.7.7

C 5.5.5.0 /30 directly connected Fast Ethernet 0/0

O 6.6.6.0 /24 110/20 5.5.5.1 Fast Ethernet 0/0

O 12.12.12.12 /32 110/11 5.5.5.1 Fast Ethernet 0/0

PE2 \* 12.12.12.12 /vrf a

Trying 12.12.12.12 ... Open

CE2 # show ip route ospf

OIA 1.1.1.0 /24 110/12 5.5.5.2 Ethernet 3/0

CE2 \* exit

PE2 \* show ip cef vrf a 1.1.1.0

1.1.1.0 /24 outer label inner label  
next hop 4.4.4.1 Fast Ethernet 0/1 label 20 16

PE2 \* show mpls forwarding-table 7.7.7.7

Local Label	Outgoing Label or VC	Prefix or Tunnel Id	Outgoing Interface	Next Hop
20	20	7.7.7.7/32	Fa0/1	4.4.4.1

PE2 \* show ip bgp vpnv4 vrf a labels

Network Next Hop In label / Out label

1.1.1.0 /24 7.7.7.7 no label / 16

5.5.5.0 /30 0.0.0.0 16 / no label (a)

6.6.6.0 /24 5.5.5.1 17 / no label

12.12.12.12 /32 5.5.5.1 22 / no label

PE2 # 12.12.12.12 /vrf a  
Trying 12.12.12.12 ... open

CE2 # Show ip route

OIA	1.1.1.0 /24	110/12	5.5.5.2	Ethernet 3/0
C	5.5.5.0 /30	directly Connected		Ethernet 3/0
C	6.6.6.0 /24	directly Connected		Ethernet 3/1
C	12.12.12.12 /32	directly Connected		Loopback0

CE2 # Show ip ospf database

OSPF Router with ID (12.12.12.12)(Process ID 1)

Router Link States (Area 0)

Link ID	ADV Router	Seq #
5.5.5.2	5.5.5.2	0x80000002
7.7.7.7	7.7.7.7	0x80000005
8.8.8.8	8.8.8.8	0x80000004
9.9.9.9	9.9.9.9	0x80000004
11.11.11.11	11.11.11.11	0x80000003
12.12.12.12	12.12.12.12	0x80000006

Net Link States (Area 0)

Link ID	ADV Router	Seq #
2.2.2.1	11.11.11.11	0x80000002
3.3.3.1	8.8.8.8	0x80000002
4.4.4.1	8.8.8.8	0x80000002
5.5.5.1	12.12.12.12	0x80000001

Summary Net Link States (Area 0)

Link ID	ADV Router	Seq #
1.1.1.0	5.5.5.2	0x80000001

CE2 #exit

PE2 #exit

P > 7.7.7.7

Trying 7.7.7.7 ... open

PE1 #Config t

PE1 (Config) # int lo 2

PE1 (Config-if) # ip vrf Forwarding a

PE1 (Config-if) # ip address 13.13.13.13 255.255.255.255

PE1 (Config-if) # router bgp 2

PE1 (Config-router) # address-family ipv4 vrf a

PE1 (Config-router-af) # network 13.13.13.13 mask 255.255.255.255

PE1 (Config-router-af) # exit

PE1 (Config) # router ospf 2 vrf a

PE1 (Config-router) # area 0 sham-link 13.13.13.13 14.14.14.14

PE1 (Config-router) # ^Z

PE1 # 9.9.9.9

Trying 9.9.9.9 ... open

PE2 #Conf t

PE2 (Config) # int lo 2

PE2 (Config-if) # ip vrf Forwarding a

PE2 (Config-if) # ip address 14.14.14.14 255.255.255.255

PE2 (Config-if) # router bgp 2

PE2 (Config-router) # address-family ipv4 vrf a

PE2 (Config-router-af) # network 14.14.14.14 mask 255.255.255.255

PE2 (Config-router-af) # router ospf 2 vrf a

PE2 (Config-router) # area 0 sham-link 14.14.14.14 13.13.13.13

PE2 (Config-router) # ^Z

PE2 # 7.7.7.7

Trying 7.7.7.7 ... open

PE1 # Config t

PE(Config) # ip vrf a

PE1(Config-vrf) # route-target export 2:2

PE1(Config-vrf) # no export map y

PE1(Config-vrf) # exit

PE1(Config) # access-list 6 permit 14.14.14.14

PE1(Config) # ^Z

PE1 # clear ip bgp \* soft

PE1 # 9.9.9.9

Trying 9.9.9.9 ... open

PE2 # Show ip bgp vpnv4 vrf a

Network

Next Hop

r>i 1.1.1.0/24

7.7.7.7

r>i 2.2.2.0/30

7.7.7.7

\*> 5.5.5.0/30

0.0.0.0

\*> 6.6.6.0/24

5.5.5.1

r>i 11.11.11.11/32

7.7.7.7

\*> 12.12.12.12/32

5.5.5.1

\*>i 13.13.13.13/32

7.7.7.7

\*> 14.14.14.14/32

0.0.0.0

PE2 # 12.12.12.12

Trying 12.12.12.12 ... open

CE2 # Show ip route

O	1.1.1.0/24	110/13	5.5.5.2	Ethernet 3/0
O	2.2.2.0/30	110/12	5.5.5.2	Ethernet 3/0
C	5.5.5.0/30	directly connected		Ethernet 3/0
C	6.6.6.0/24	directly connected		Ethernet 3/1
O	11.11.11.11/32	110/13	5.5.5.2	Ethernet 3/0

C 12.12.12.12/32 directly connected Loopback0  
OE2 13.13.13.13/32 110/1 5.5.5.2 Ethernet 3/0

CE2 \*exit

P> 9.9.9.9

Trying 9.9.9.9 ... open

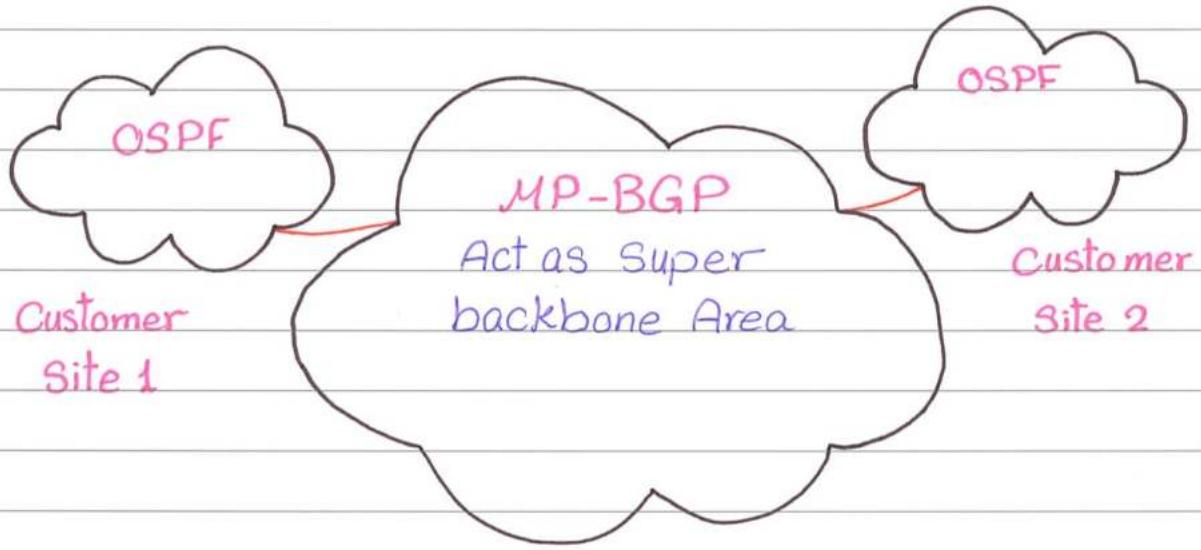
PE2 \* Show ip ospf sham-links

Sham Link OSPF\_SLO to address 13.13.13.13 is up

Area 0 Source address 14.14.14.14

## Session 21

- Run OSPF between PE & CE:



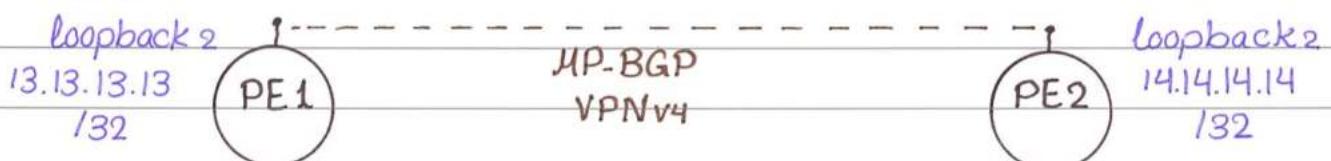
Type 1,2

Type 3

\* OSPF internal routes should be kept internal. we need sham link (Type 1,2).

. Sham Link :

- Virtual Link between PE & PE used to tunnel CE Type 1 & 2 LSAs.
- Shamlink need source & destination stable interfaces to run the tunnel.



- Sham link source & destination interfaces should be member in Customer Vrf .

on PE1 & PE2 :

(Config) # interface lo 2

(Config-if) # ip vrf forwarding a

- Those interfaces should be advertised using MP-BGP.

On PE1 "repeat on PE2 with update IPs":

(Config)# router bgp 2

(Config-router)# address-family ipv4 vrf a

(Config-router-af)# network 13.13.13.13 mask 255.255.255.255

(Config)# router ospf 2 vrf a

(Config-router)# area 0 sham-link 13.13.13.13 14.14.14.14

Type 1,2 →

Type 1,2 →

\* OSPF external routes should be kept external.

Type 5,7 →

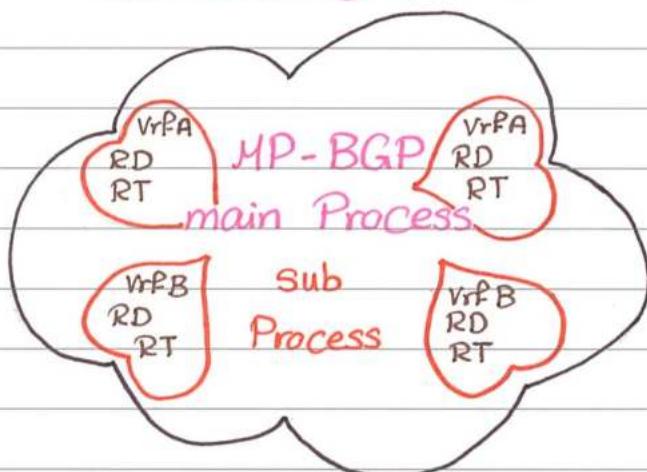
Type 5 →

\* Metric should be kept transferred.

Metric x →

Metric x →

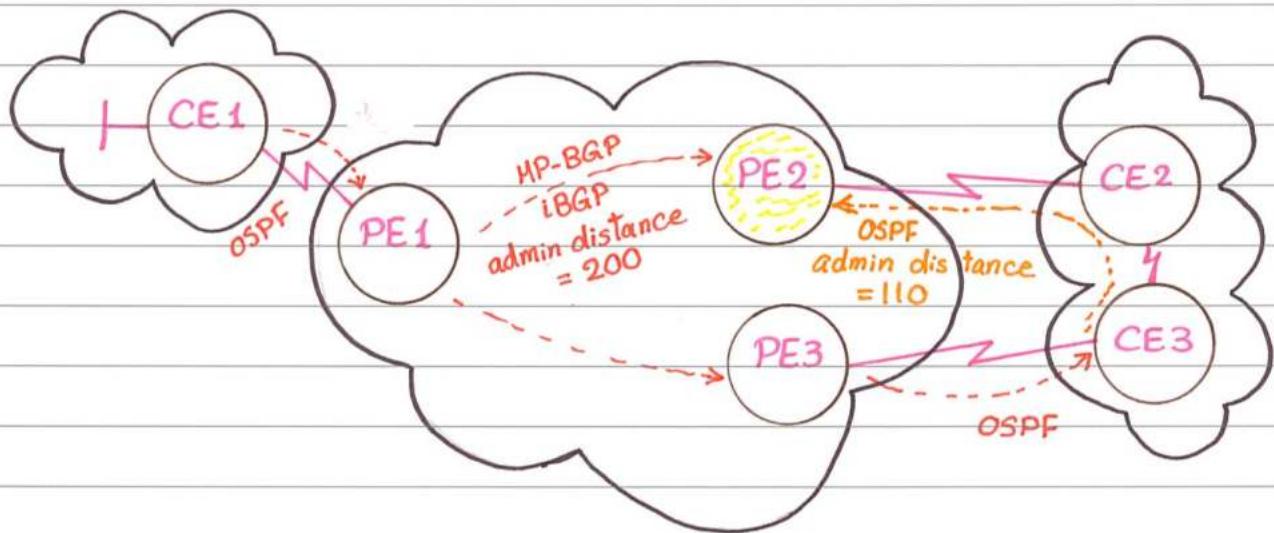
address-Family VPN v4



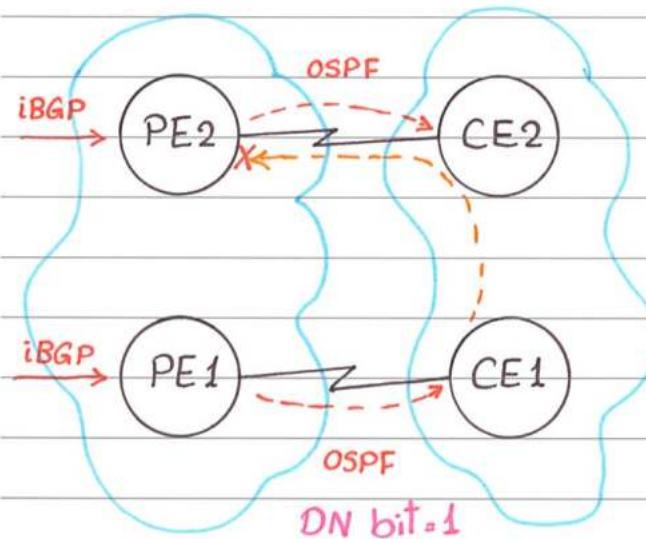
address-Family ipv4 vrf —

## Session 21 Part 2

. Avoid loops :

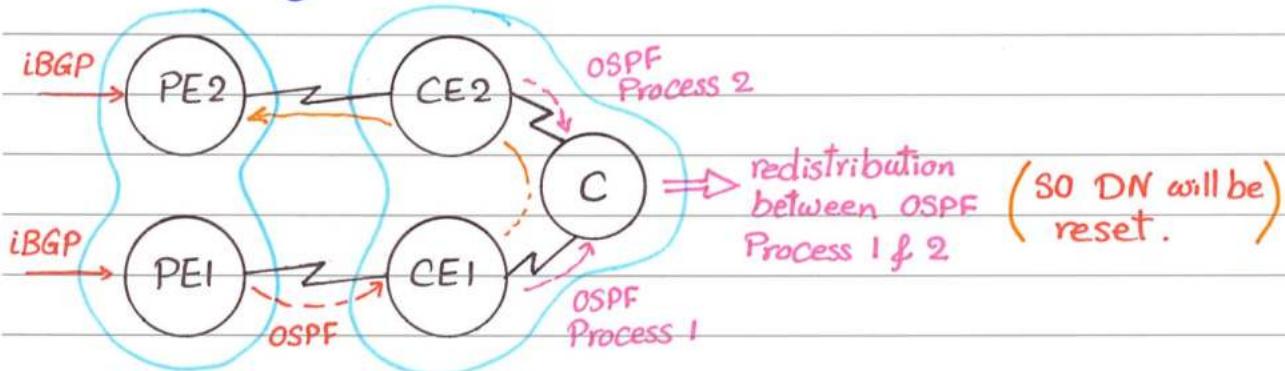


- Down Bit : used type 3, type 5,7 LSAs .



- DN by default = 0
- PE2 Add update "DN=1"
- in database only but didn't add on routing table .
- Routes has DN=1 Can not be used by dijkstra algorithm .
- Can not be set manually .

- Domain Tag : used type 5,7 LSAs .



(Config) # router ospf vrf A

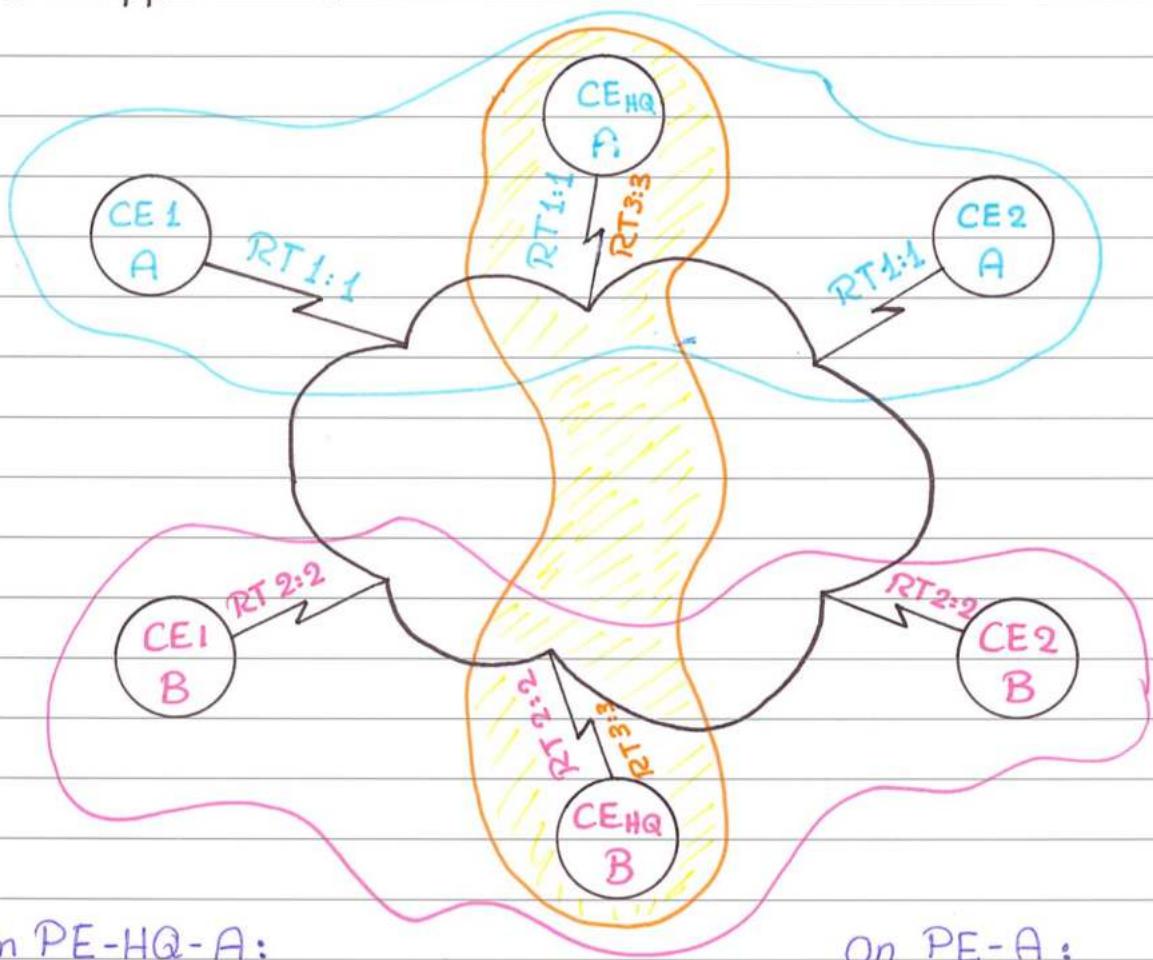
(Config-router) # domain-tag x  
↓

Auto domain tag = 32 bit

1 101 0000 0000 0000 xx  
1 bit    3bit    12 bit    AS# (16 bit)

### Session 12 Part 3 Complex VPNs

- Overlapped VPN:



On PE-HQ-A:

(Config) # ip vrf A

(Config-vrf) # RD 1:1

(Config-vrf) # route-target both 1:1

(Config-vrf) # route-target both 3:3

On PE-A:

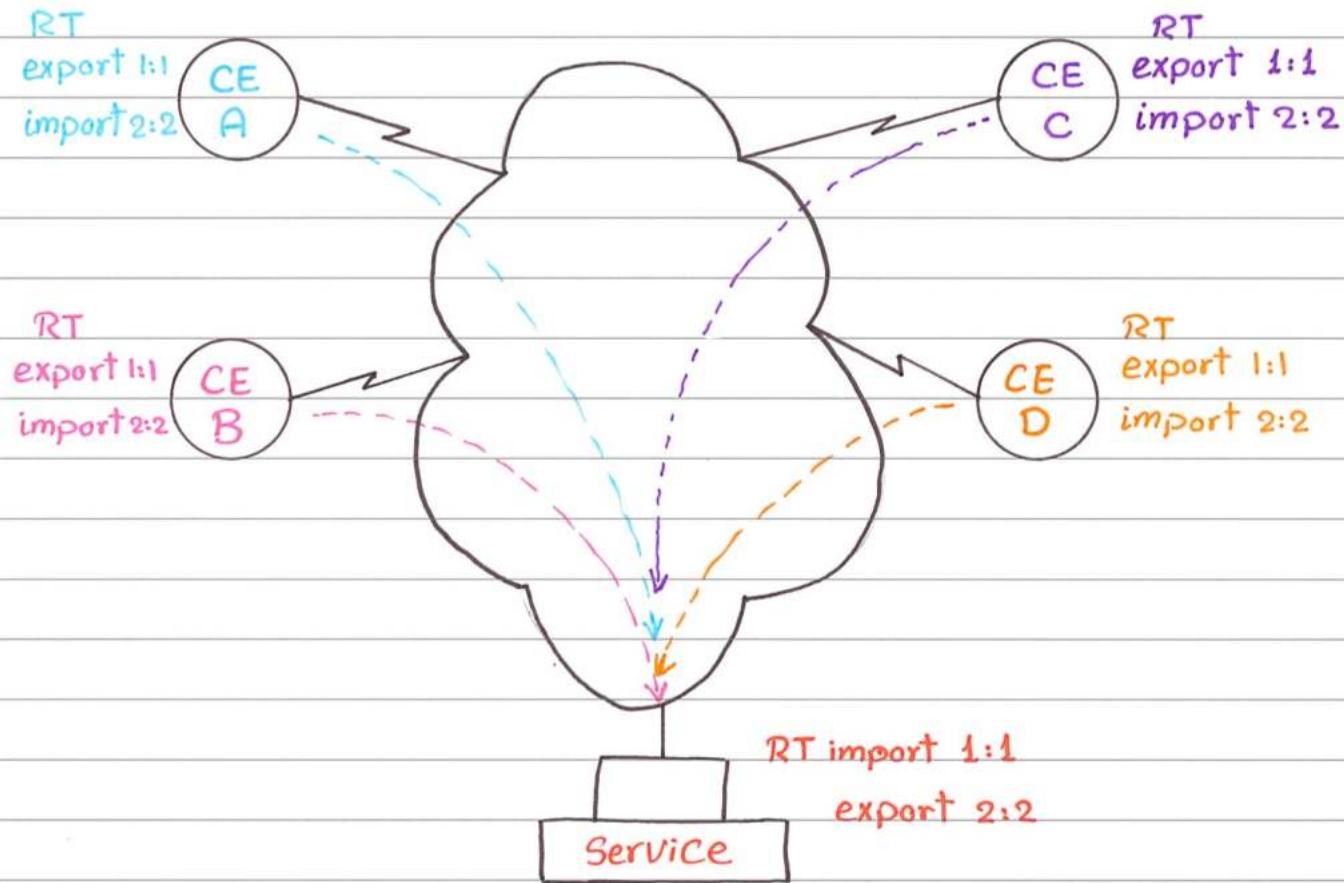
(Config) # ip vrf A

(Config-vrf) # RD 1:1

(Config-vrf) # route-target both 1:1

124

## Centralised Service VPN:



Customers in different subnets...

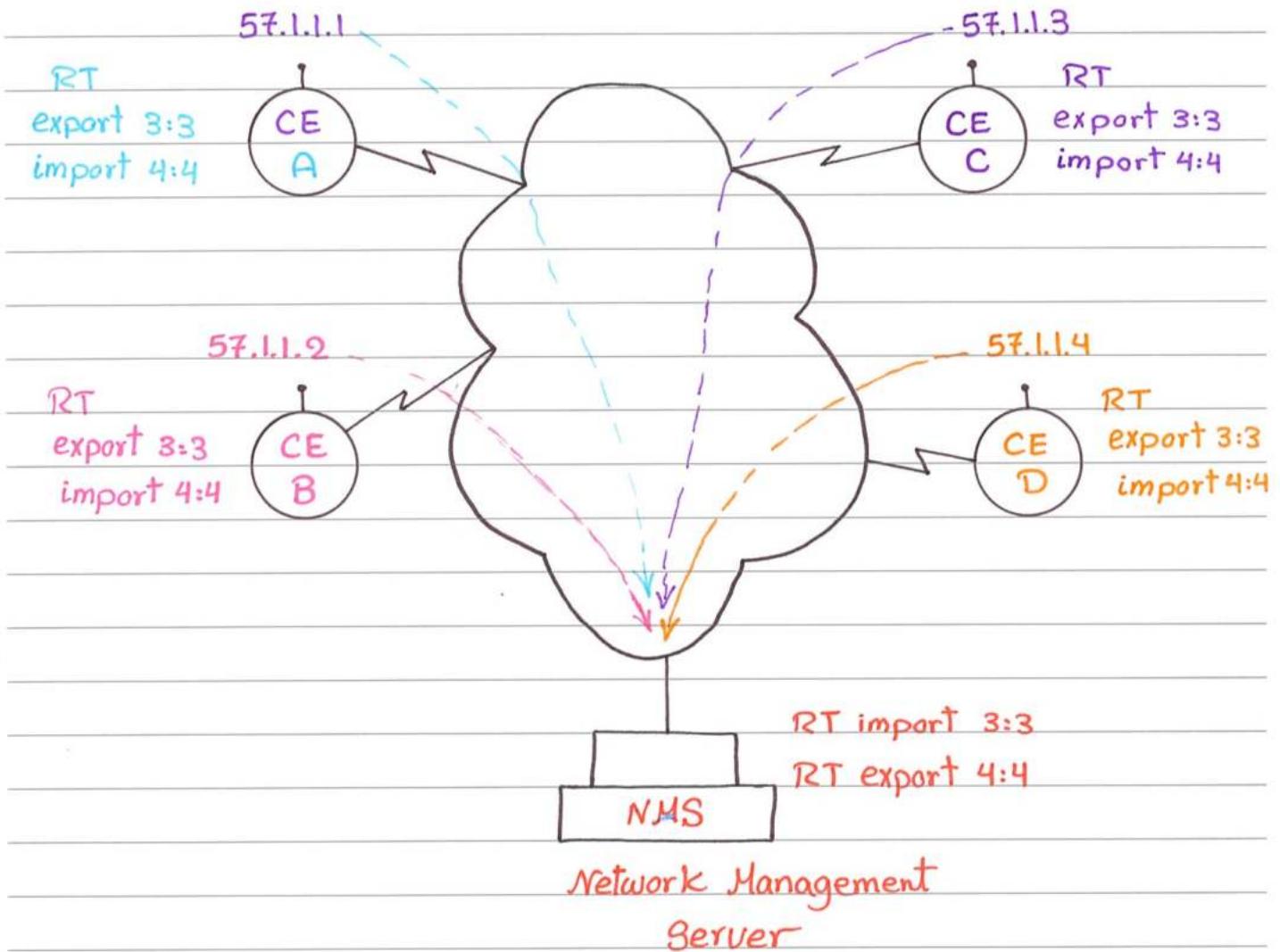
Any customer send → Server only will be received.

Server send → Customer will be received  
depend on (IP)

⇒ in Same Subnet: mismatch

PE هي المفترض Cloud يكون جزو من "VRF, RD, RT" لربت  
in same subnet هو src IP يخرج server والباقي يدخل للCustomer  
!!! Customer ليس في same server !! لأن...

## Management VPN:



On PEs :

```
(Config)# ip vrf A  
(Config-vrf)# export map y
```

```
(Config)# access-list 1 permit 57.1.1.0 0.0.0.255  
(Config)# route-map y  
(Config-route-map)# match ip address 1  
(Config-route-map)# set ext-community rt 3:3
```

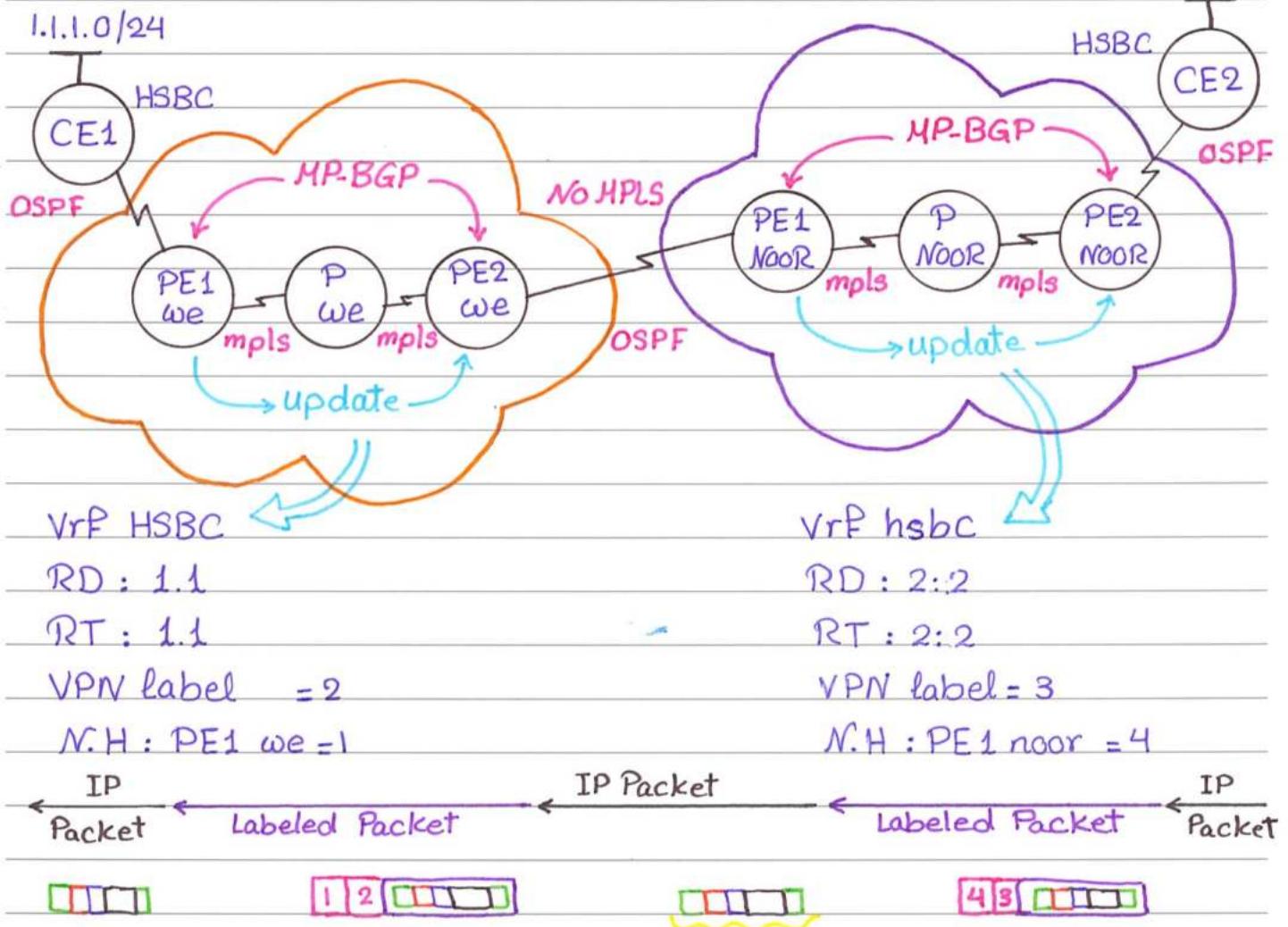
## Session 22

### Hierarchical VPNs

#### Inter AS VPN: "Option A"

Two Different Customer sites belonging to two different providers.

6.6.6.0/24



On PE1:

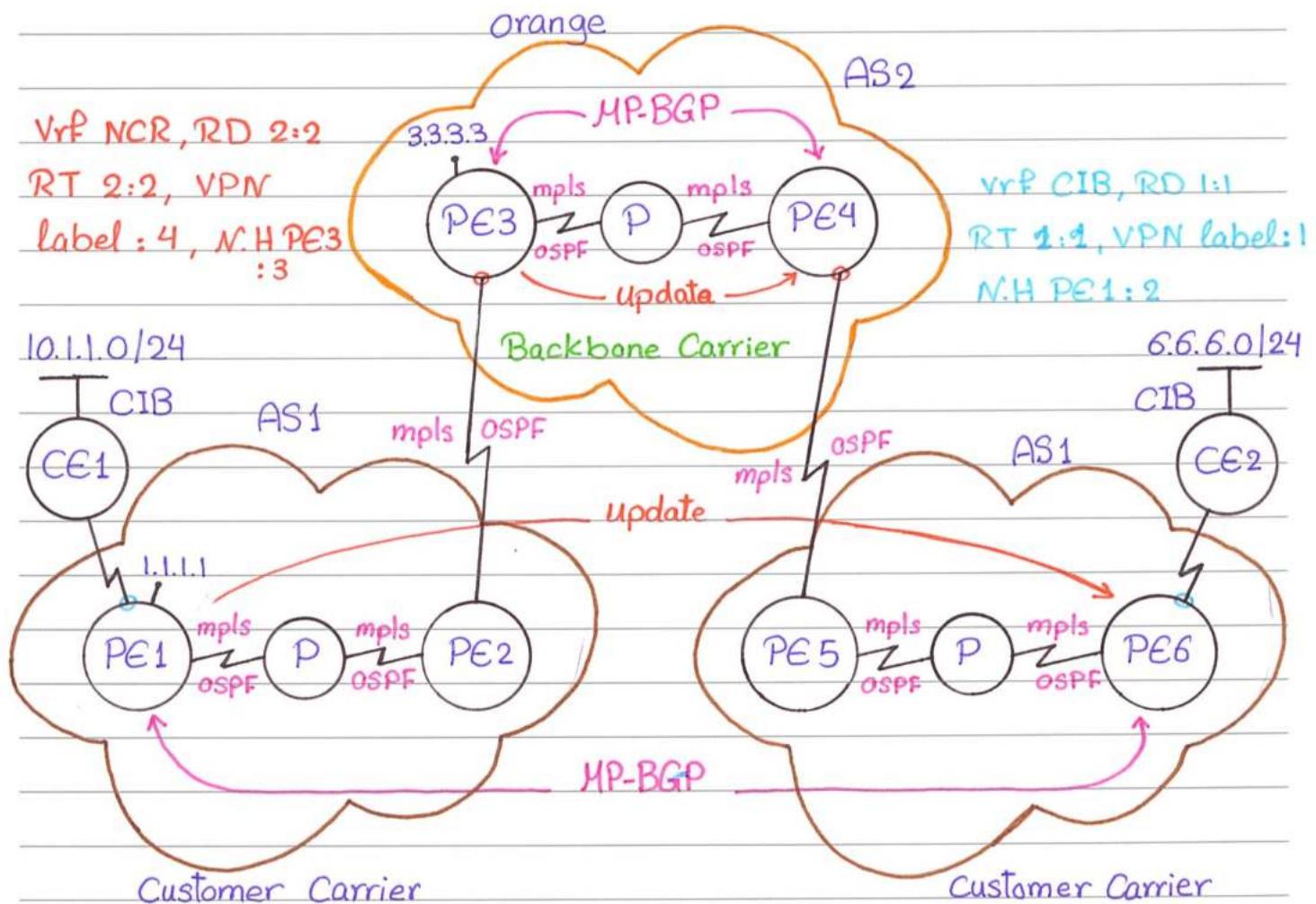
```
(Config)# router bgp 1
(Config-router)# address-family ipv4 vrf HSBC
(Config-router-af)# redistribute OSPF 2
```

On PE2:

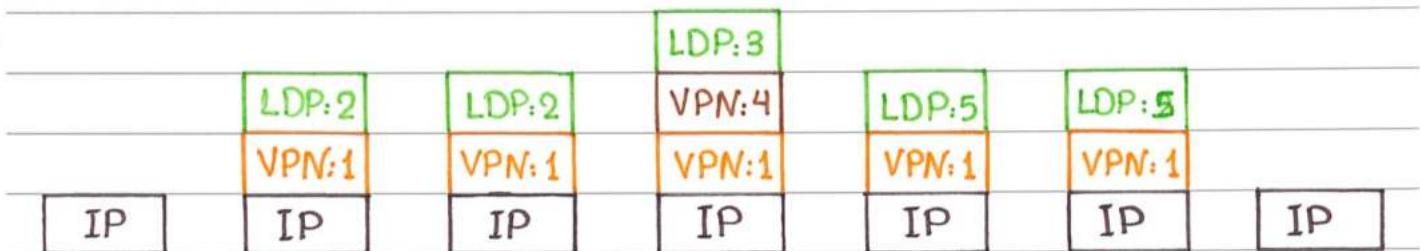
```
(Config)# router ospf 2 vrf HSBC
(Config-router)# redistribute bgp 1 subnets
```

## Session 22 Part 2

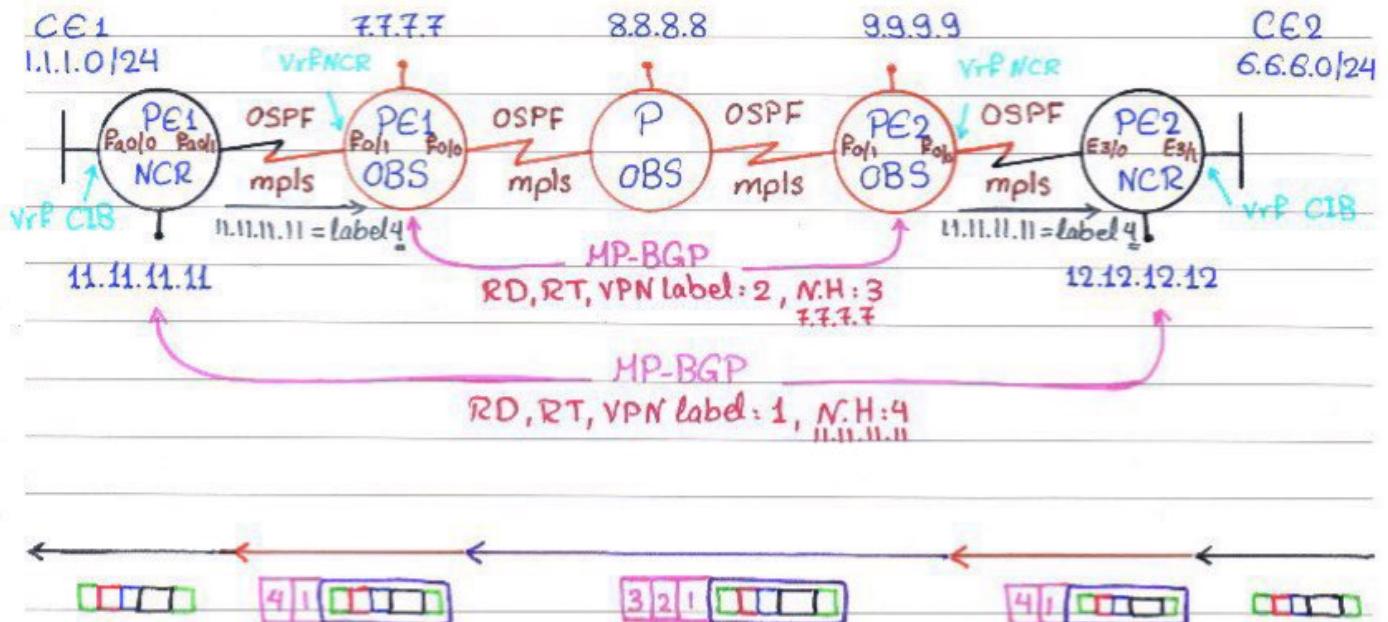
- CSC (Carrier Supporting Carriers) : "option B"



CE1 ← PE1 ← PE2 ← PE3 ← PE4 ← PE5 ← PE6 ← CE2



Session 23  
— CSC —



P\_OBS > Show ip route

O	1.1.1.0/24	110/3	3.3.3.2	Fast Ethernet 0/0
O	2.2.2.0/30	110/2	3.3.3.2	Fast Ethernet 0/0
C	3.3.3.0/30	directly Connected		Fast Ethernet 0/0
C	4.4.4.0/30	directly Connected		Fast Ethernet 0/1
O	5.5.5.0/30	110/11	4.4.4.2	Fast Ethernet 0/1
O	6.6.6.0/30	110/21	4.4.4.2	Fast Ethernet 0/1
O	7.7.7.7/32	110/2	3.3.3.2	Fast Ethernet 0/0
C	8.8.8.8/32	directly Connected		Loopback 0
O	9.9.9.9/32	110/2	4.4.4.2	Fast Ethernet 0/1
O	11.11.11.11/32	110/3	3.3.3.2	Fast Ethernet 0/0
O	12.12.12.12/32	110/12	4.4.4.2	Fast Ethernet 0/1

P\_OBS > 11.11.11.11

Trying 11.11.11.11 ... Open

PE1\_NCR # Config t

PE1\_NCR(Config) # int Po11

PE1\_NCR(Config-if) # mpls ip

PE1\_NCR(Config-if) # ^Z

PE1\_NCR # exit

P\_OBS > 7.7.7.7

Trying 7.7.7.7 ... open

PE1\_OBS # Config t

PE1\_OBS(Config) # int Po11

PE1\_OBS(Config-if) # mpls ip

PE1\_OBS(Config-if) # int Po10

PE1\_OBS(Config-if) # mpls ip

PE1\_OBS(Config-if) # ^Z

PE1\_OBS # exit

P\_OBS > 8.8.8.8

Trying 8.8.8.8 ... open

PE2\_OBS # Config t

PE2\_OBS(Config) # int Po11

PE2\_OBS(Config-if) # mpls ip

PE2\_OBS(Config-if) # int Po10

PE2\_OBS(Config-if) # mpls ip

PE2\_OBS(Config-if) # ^Z

PE2\_OBS # exit

P\_OBS > 12.12.12.12

Trying 12.12.12.12 ... open

PE2\_NCR \* Config t  
PE2\_NCR(Config) # int e310  
PE2\_NCR(Config-if) # mpls ip  
PE2\_NCR(Config-if) # ^Z  
PE2\_NCR \* exit

P\_OBS > en  
P\_OBS \* Conft  
P\_OBS(Config) # int Po10  
P\_OBS(Config-if) # mpls ip  
P\_OBS(Config-if) # int Po11  
P\_OBS(Config-if) # mpls ip  
P\_OBS(Config-if) # ^Z  
P\_OBS # 11.11.11.11  
Trying 11.11.11.11 ... Open

PE1\_NCR \* Conf t  
PE1\_NCR(Config) # ip vrf cib  
PE1\_NCR(Config-vrf) # rd 1:1  
PE1\_NCR(Config-vrf) # route-target 1:1  
PE1\_NCR(Config-vrf) # maximum routes 4 50  
PE1\_NCR(Config-vrf) # exit  
PE1\_NCR(Config) # int Po10  
PE1\_NCR(Config-if) # ip vrf forwarding cib  
PE1\_NCR(Config-if) # ip add 1.1.1.1 255.255.255.0  
PE1\_NCR(Config-if) # ip vrf sitemap x  
PE1\_NCR(Config-if) # exit  
PE1\_NCR(Config) # access-list 1 permit 1.1.1.0  
PE1\_NCR(Config) # route-map x  
PE1\_NCR(Config-route-map) # match ip address 1  
PE1\_NCR(Config-route-map) # set extcommunity soo 5:5  
PE1\_NCR(Config-route-map) # exit

```
PE1_NCR(Config) # router bgp 1
PE1_NCR(Config-router) # neighbor 12.12.12.12 remote-as 1
PE1_NCR(Config-router) # neighbor 12.12.12.12 update-source lo0
PE1_NCR(Config-router) # address-family vpnv4
PE1_NCR(Config-router-af) # neighbor 12.12.12.12 activate
PE1_NCR(Config-router-af) # neighbor 12.12.12.12 next-hop-self
PE1_NCR(Config-router-af) # address-family ipv4 vrf cib
PE1_NCR(Config-router-af) # redistribute connected
PE1_NCR(Config-router-af) # ^z
PE1_NCR # exit
```

P\_OBS > 12.12.12.12

Trying 12.12.12.12 ... open

```
PE2_NCR # Config
PE2_NCR(Config) # ip vrf cib
PE2_NCR(Config-vrf) # rd 1:1
PE2_NCR(Config-vrf) # route-target 1:1
PE2_NCR(Config-vrf) # exit
PE2_NCR(Config) # int e3/1
PE2_NCR(Config-ip) # ip vrf Forwarding cib
PE2_NCR(Config-ip) # ip add 6.6.6.1 255.255.255.0
PE2_NCR(Config-ip) # exit
PE2_NCR(Config) # router bgp 1
PE2_NCR(Config-router) # neighbor 11.11.11.11 remote-as 1
PE2_NCR(Config-router) # neighbor 11.11.11.11 update-source lo0
PE2_NCR(Config-router) # address-family vpnv4
PE2_NCR(Config-router-af) # neighbor 11.11.11.11 activate
PE2_NCR(Config-router-af) # neighbor 11.11.11.11 next-hop-self
PE2_NCR(Config-router-af) # address-family ipv4 vrf cib
PE2_NCR(Config-router-af) # redistribute connected
```

PE2\_NCR (Config-router-af) # ^Z  
PE2\_NCR # exit

P\_OBS > 7.7.7  
Trying 7.7.7 ... open

PE1\_OBS # Config t  
PE1\_OBS (Config) # ip vrf NCR  
PE1\_OBS (Config-vrf) # rd 2:2  
PE1\_OBS (Config-vrf) # route-target 2:2  
PE1\_OBS (Config-vrf) # exit  
PE1\_OBS (Config) # int Po11  
PE1\_OBS (Config-if) # ip vrf Forwarding NCR  
PE1\_OBS (Config-if) # ip add 2.2.2.2 255.255.255.252  
PE1\_OBS (Config-if) # exit  
PE1\_OBS (Config) # router bgp 2  
PE1\_OBS (Config-router) # neighbor 9.9.9.9 remote-as 2  
PE1\_OBS (Config-router) # neighbor 9.9.9.9 update-source lo0  
PE1\_OBS (Config-router) # address-family vpnv4  
PE1\_OBS (Config-router-af) # neighbor 9.9.9.9 activate  
PE1\_OBS (Config-router-af) # neighbor 9.9.9.9 next-hop-self  
PE1\_OBS (Config-router-af) # address-family ipv4 vrf NCR  
PE1\_OBS (Config-router-af) # redistribute ospf 2  
PE1\_OBS (Config-router-af) # exit  
PE1\_OBS (Config-router) # exit  
PE1\_OBS (Config) # router ospf 2 vrf NCR  
PE1\_OBS (Config-router) # network 2.2.2.2 0.0.0.0 area 0  
PE1\_OBS (Config-router) # redistribute bgp 2 subnets  
PE1\_OBS (Config-router) # ^Z  
PE1\_OBS # exit

P\_OBS > 9.9.9.9

Trying 9.9.9.9 ... Open

PE2\_OBS # Config t

PE2\_OBS (Config) # ip vrf NCR

PE2\_OBS (Config-vrf) # rd 2:2

PE2\_OBS (Config-vrf) # route-target 2:2

PE2\_OBS (Config-vrf) # exit

PE2\_OBS (Config) # int Polo

PE2\_OBS (Config-if) # ip vrf Forwarding NCR

PE2\_OBS (Config-if) # ip add 5.5.5.2 255.255.255.252

PE2\_OBS (Config-if) # exit

PE2\_OBS (Config) # router bgp 2

PE2\_OBS (Config-router) # neighbor 7.7.7.7 remote-as 2

PE2\_OBS (Config-router) # neighbor 7.7.7.7 update-source lo0

PE2\_OBS (Config-router) # address-family vpnv4

PE2\_OBS (Config-router-af) # neighbor 7.7.7.7 activate

PE2\_OBS (Config-router-af) # neighbor 7.7.7.7 next-hop-self

PE2\_OBS (Config-router-af) # address-family ipv4 vrf NCR

PE2\_OBS (Config-router-af) # redistribute ospf 2

PE2\_OBS (Config-router-af) # exit

PE2\_OBS (Config-router) # exit

PE2\_OBS (Config) # router ospf 2 vrf NCR

PE2\_OBS (Config-router) # network 5.5.5.2 0.0.0.0 area 0

PE2\_OBS (Config-router) # redistribute bgp 2 subnets

PE2\_OBS (Config-router) # exit

PE2\_OBS (Config) # mpls label range 50 150

PE2\_OBS (Config) # ^Z

PE2\_OBS #

PE2\_OBS # reload

P\_OBS > 9.9.9.9

Trying 9.9.9.9 ... open

PE2\_OBS ✘ show ip bgp summary

Neighbor	AS
7.7.7.7	2

PE2\_OBS ✘ show ip route vrf NCR

B	2.2.2.0/30	200/0	7.7.7.7	
C	5.5.5.0/30	directly connected		Fast Ethernet 0/0
B	11.11.11.11/32	200/2	7.7.7.7	
O	12.12.12.12/32	110/11	5.5.5.1	Fast Ethernet 0/0

PE2\_OBS # 12.12.12.12 /vrf NCR

Trying 12.12.12.12 ... open

PE2\_NCR ✘ show ip route

OIA	2.2.2.0/30	110/11	5.5.5.2	Ethernet 3/0
C	5.5.5.0/30	directly connected		Ethernet 3/0
OIA	11.11.11.11/32	110/12	5.5.5.2	Ethernet 3/0
C	12.12.12.12/32	directly connected		Loopback0

PE2\_NCR ✘ show ip ospf database summary

OSPF Router with ID (12.12.12.12) (Process ID 1)

Summary Net Link States (Area 0)

Options: (No TOS-Capability, DC, Downward)

PE2\_NCR ✘ exit

PE2\_OBS # Config t

PE2\_OBS (Config) # int lo 2

PE2\_OBS (Config-if) # ip vrf for NCR

PE2\_OBS (Config-if) # ip add 14.14.14.14 255.255.255.255

PE2\_OBS (Config-if) # router bgp 2

PE2\_OBS (Config-router) # address-family ipv4 vrf NCR

PE2\_OBS (Config-router-af) # network 14.14.14.14 mask 255.255.255.255

PE2\_OBS (Config-router-af) # ^Z

PE2\_OBS # exit

P\_OBS > 7.7.7.7

Trying 7.7.7.7 ... open

PE1\_OBS # Config t

PE1\_OBS (Config) # int lo 2

PE1\_OBS (Config-if) # ip vrf for NCR

PE1\_OBS (Config-if) # ip add 13.13.13.13 255.255.255.255

PE1\_OBS (Config-if) # router bgp 2

PE1\_OBS (Config-router) # address-family ipv4 vrf NCR

PE1\_OBS (Config-router-af) # network 13.13.13.13 mask 255.255.255.255

PE1\_OBS (Config-router-af) # ^Z

PE1\_OBS # show ip bgp vpnv4 vrf NCR

Network

Next Hop

\*> 2.2.2.0/30 0.0.0.0

\*> i 5.5.5.0/30 9.9.9.9

\*> 11.11.11.11/32 2.2.2.1

\*> i 12.12.12.12 /32 9.9.9.9

\*> 13.13.13.13 /32 0.0.0.0

\*> i 14.14.14.14 /32 9.9.9.9

PE1\_OBS # Config t

PE1\_OBS(Config) # router ospf 2 vrf NCR  
PE1\_OBS(Config-router) # area 0 Sham-link 13.13.13.13  
14.14.14.14

PE1\_OBS(Config-router) # ^Z  
PE1\_OBS # exit

P\_OBS > 9.9.9.9

Trying 9.9.9.9 ... open

PE2\_OBS # Conf t

PE2\_OBS(Config) # router ospf 2 vrf NCR

PE2\_OBS(Config-router) # area 0 Sham-link 14.14.14.14  
13.13.13.13

PE2\_OBS(Config-router) # ^Z

PE2\_OBS # show ip ospf sham-link

Sham Link OSPF\_SLO to address 13.13.13.13 is up area 0

Source address 14.14.14.14.

PE2\_OBS # 12.12.12.12 /vrf NCR

Trying 12.12.12.12 ... Open

PE2\_NCR #

PE2\_NCR # show ip route

O	2.2.2.0/30	110/12	5.5.5.2	Ethernet 3/0
C	5.5.5.0/30	directly connected		Ethernet 3/0
O	11.11.11.11/32	110/10	5.5.5.2	Ethernet 3/0
C	12.12.12.12/32	directly connected		Loopback 0
OE2	13.13.13.13/32	110/1	5.5.5.2	Ethernet 3/0
OE2	14.14.14.14/32	110/1	5.5.5.2	Ethernet 3/0

PE2-NCR # show ip ospf database external  
OSPF Router with ID (12.12.12.12) (Process ID 1)  
Type 5 AS External Link States

External Route Tag : 3489660930

PE2-NCR # show ip bgp vpngv4 cib 1.1.1.0  
Extended Community: S00:5:5 RT: 1:1

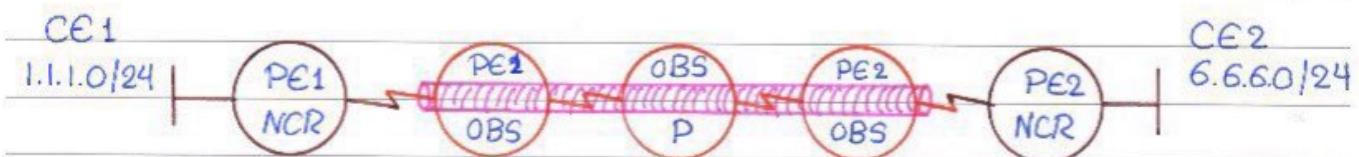
PE2-NCR # traceroute vrf cib 1.1.1.1 source 6.6.6.1

- 1 5.5.5.2 MPLS labels 55/24
- 2 4.4.4.1 MPLS labels 16/23/24
- 3 2.2.2.2 MPLS labels 23/24

PE2-NCR # exit

PE2-OBS # show ip cef vrf NCR detail  
11.11.11.11 /32, epoch 0  
local label info : other/55  
recursive via 7.7.7.7 label 23  
nexthop 4.4.4.1 Fast Ethernet 0/1 label 16

### — L2 VPN —



P\_OBS > 9.9.9.9

Trying 9.9.9.9 ... open

PE2\_OBS \* Conf t

PE2\_OBS(Config) \* no cdp run

PE2\_OBS(Config) \* router ospf 2 vrf NCR

PE2\_OBS(Config-router) \* no area 0 sham-link 14.14.14.14

13.13.13.13

PE2\_OBS(Config-router) \* exit

PE2\_OBS(Config) \* int F0/0

PE2\_OBS(Config-if) \* no ip address

PE2\_OBS(Config-if) \* xconnect 7.7.7.7 3 encapsulation mpls

PE2\_OBS(Config-if-xconn) \* ^Z

PE2\_OBS \* 7.7.7.7

Trying 7.7.7.7 ... Open

PE1\_OBS \* Conf t

PE1\_OBS(Config) \* int F0/1

PE1\_OBS(Config-if) \* no ip address

PE1\_OBS(Config-if) \* xconnect 9.9.9.9 3 encapsulation mpls

PE1\_OBS(Config-if-xconn) \* ^Z

PE1\_OBS \* 11.11.11.11

Trying 11.11.11.11 ... Open

PE1\_NCR \* Conf t

PE1\_NCR(Config) \* int F0/1

PE1\_NCR(Config-if) \* ip address 5.5.5.2 255.255.255.252

PE1\_NCR(Config-if) \* router ospf 1

PE1\_NCR(Config-router) \* net 5.5.5.2 0.0.0.0 area 0

PE1\_NCR(Config-router) \* ^Z

PE1-NCR \* show ip ospf int br

Interface	Area	IP / mask	State
Lo 0	0	11.11.11.11/32	LOOP
Fa0/1	0	5.5.5.2/30	BDR

PE1-NCR \* show ip ospf neighbor

Neighbor ID	State	Address	Interface
12.12.12.12	FULL / DR	5.5.5.1	FastEthernet 0/1

PE1-NCR \* 12.12.12.12

Trying 12.12.12.12 ...open

PE2-NCR \* traceroute vrf cib 1.1.1.1 source 6.6.6.1

1 1.1.1.1

PE2-NCR \* show ip route vrf cib

B	1.1.1.0/24	200/0	11.11.11.11
C	6.6.6.0/24	directly connected	Ethernet 3/1

PE2-NCR \* 7.7.7.7

Trying 7.7.7.7 ...open

PE1-OBS \* show mpls forwarding table

17	Pop tag	8.8.8.8/32	Fa0/0	3.3.3.1
18	Pop tag	4.4.4.0/30	Fa0/0	3.3.3.1
29	19	9.9.9.9/32	Fa0/0	3.3.3.1
22		12ckt (3)	none	point2point

PE1-OBS \* show mpls l2transport vc detail

Destination address : 9.9.9.9 , VC ID : 3 , VC status : up

## Session 24

### L2 MPLS VPN

#### - VPWS (Virtual Private Wire Service)

Pseudo wire = ATOM (Any Transport Over MPLS)  
standard Cisco Point-to-Point

#### - VPLS (Virtual Private LAN Service)

TLS, DVS, QinQ, EoMPLS

Point-to-Point, Point-to-Multipoint or Multipoint-to-Multipoint (Full Mesh topology).

#### - EVPN (Extended VPN or Ethernet VPN)

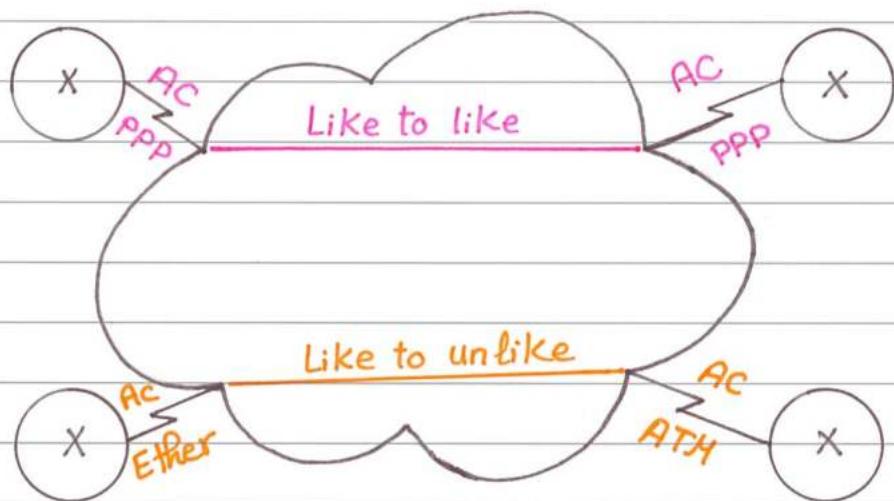
Support Full Mesh + Dual homed, Redundancy.

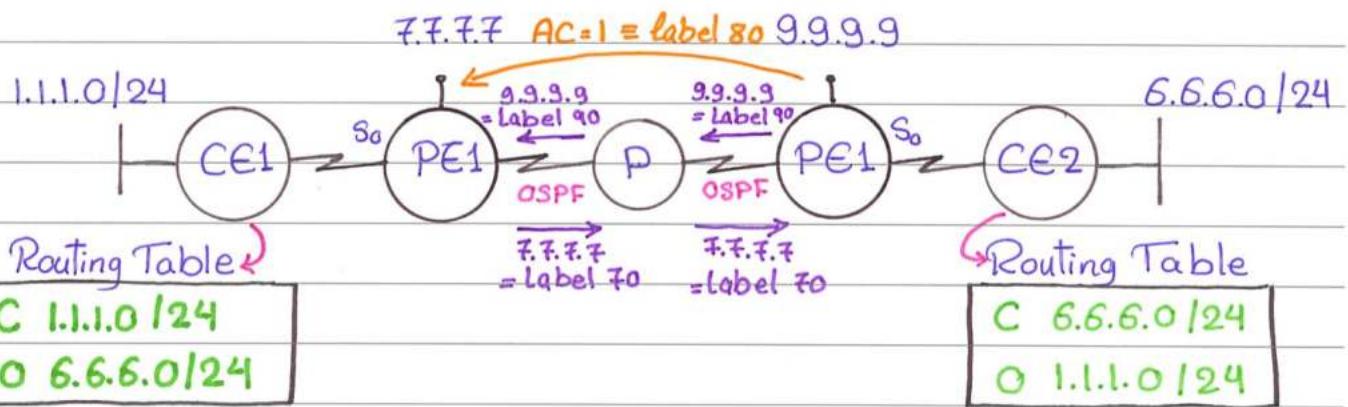
## Session 24 Part 2

### VPWS

VPWS Can Support both :

- Like to like attached circuit.
- Interworking attached circuit.  
↓  
different Topology





PE1 (Config) # interface S0

PE1 (Config-if) # no ip address

PE1 (Config-if) # xconnect 9.9.9.9 1 encapsulation mpls  
IP of next PE vc id using LDP

PE2 (Config) # interface S0

PE2 (Config-if) # no ip address

PE2 (Config-if) # xconnect 7.7.7.7 1 enc mpls

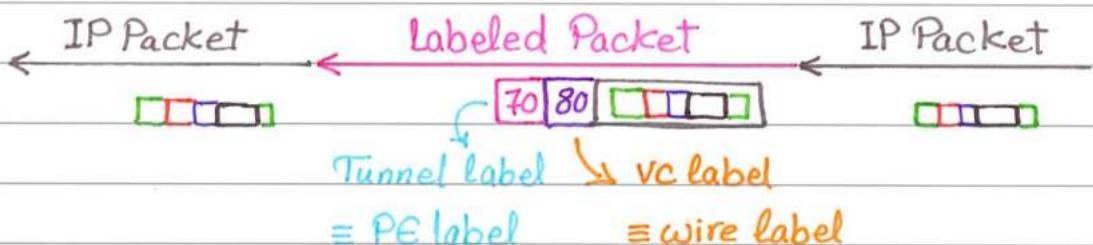
→ PW operation :-

① Auto Discovery :

PE try to discovery other PE using  
normal LDP  $\Rightarrow$  discovery tunnel Label

② Signaling :

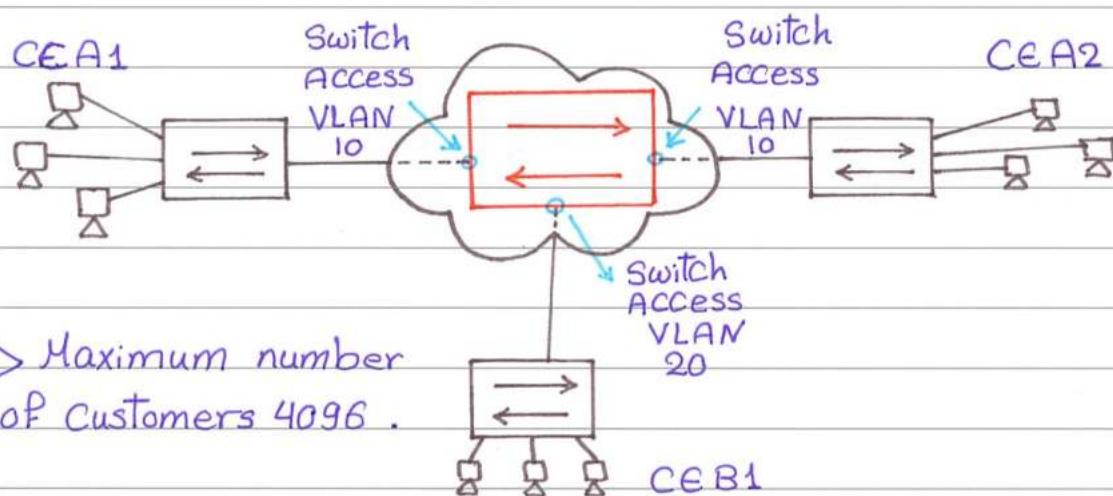
Each PE will map the AC=VC with  
label using Targeted LDP  $\Rightarrow$  discovery VPN label  
 $\equiv$  discovery VC label



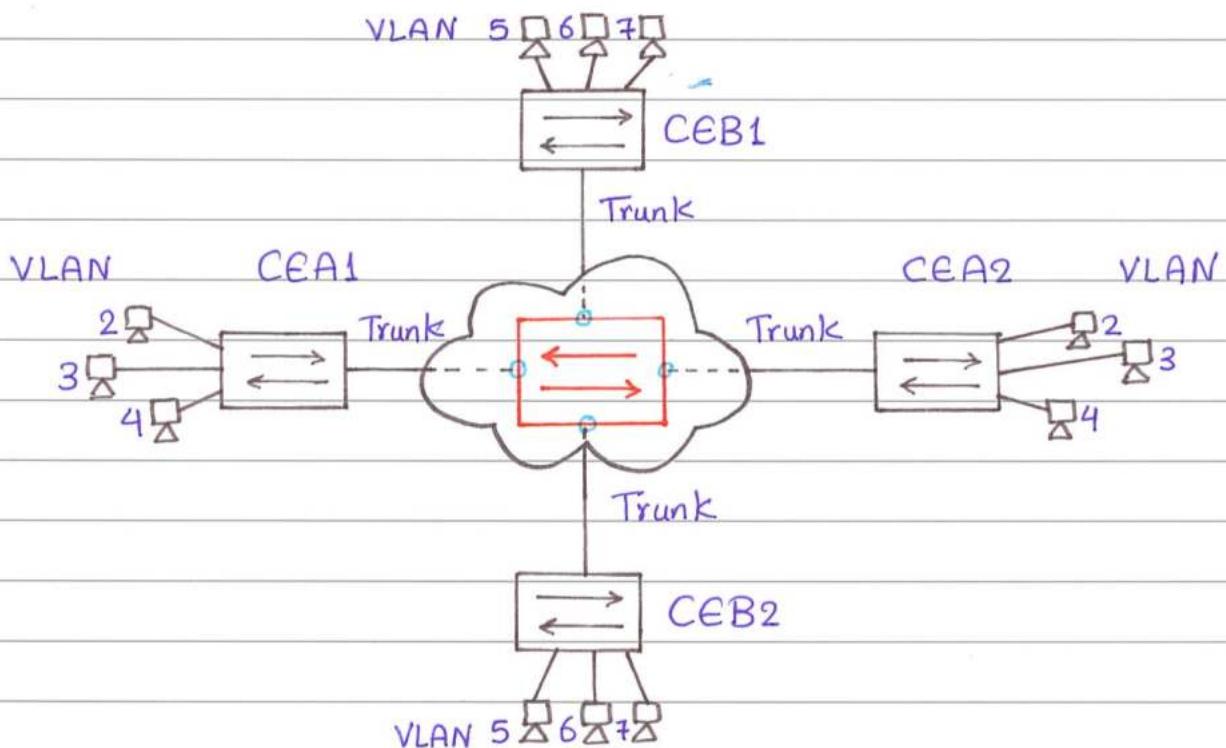
## Session 24 Part 3

### VPLS

#### ① TLS Transparent LAN Service :



#### ② DVS Directed VLAN Services :



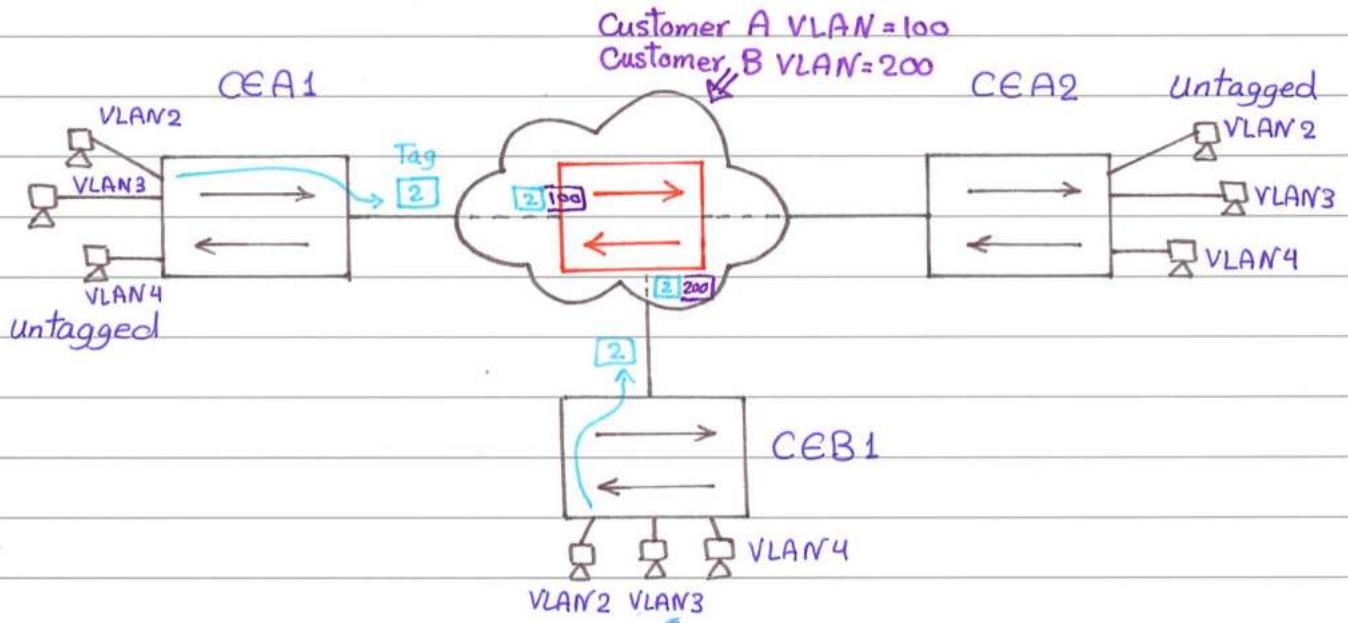
(Config-if) # switchport mode trunk  
 (Config-if) # switchport trunk allowed VLAN 2-4

(Config-if) # switchport trunk allowed VLAN [add | remove] \*

⇒ If each customer has 10 VLAN,  
Service Provider can serve 409 customer only.

### Session 24 Part 4

#### ③ Q-in-Q Double Tagging :



DST MAC	SRC MAC	dot1q tag	Type	Packet	CRC	normal
---------	---------	-----------	------	--------	-----	--------

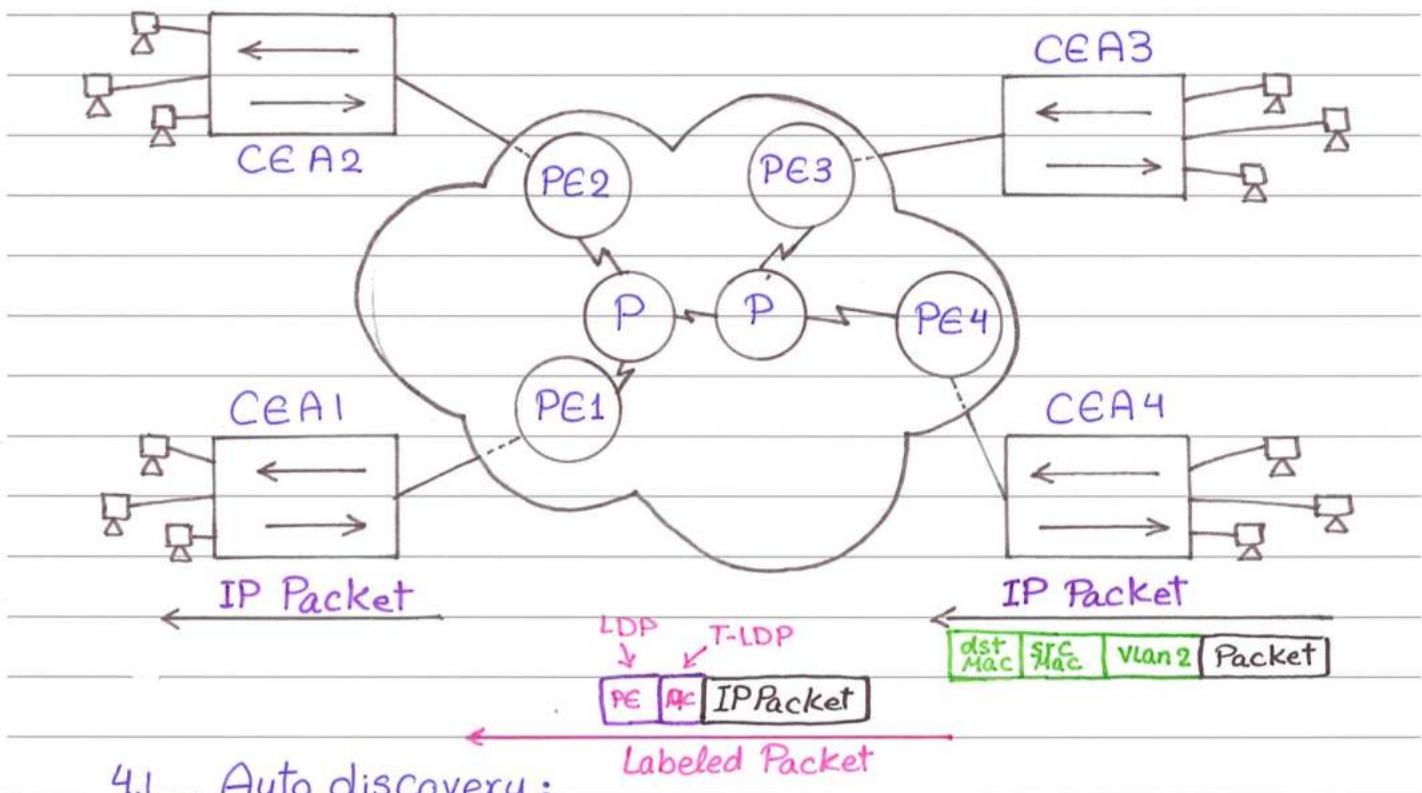
⇒ SP Can Serve  
4096 Customers.

DST MAC	SRC MAC	dot1q tag	dot1q tag	Type	Packet	CRC	Q-in-Q
↑ Customer ID = 100 VLAN 2							

(Config-if) # switchport mode dot1q\_tunnel

(Config-if) # switchport access vlan 100

## ④ EoMPLS Ethernet over MPLS:



### 4.1 - Auto discovery:

PE discover other PEs using LDP or MP-BGP  
(new address family called L2 VPN VPLS).

### 4.2 - Signaling:

Each AC should be mapped with label using Targeted-LDP or MP-BGP .

On PE1 using LDP & T-LDP :

(Config) # L2 vfi Context CIB Creation  
 (Config-vfi) # vpn id 100  
 (Config-vfi) # neighbor IP of PE2|PE3|PE4 encapsulation mpls

(Config) # interface \_\_\_\_\_

Activation

(Config-if) # xconnect vfi CIB

PE1 using MP-BGP:

(Config) # L2VPN vfi CIB  
(Config-vfi) # vpn id 1       $\Rightarrow$  RD = 2:1      RT = 2:1  
(Config-vfi) # rd \_\_\_\_\_  
(Config-vfi) # route-target \_\_\_\_\_

AS\* ↓      Vpn id ↓      AS\* ↓      Vpn id ↓  
} OR {

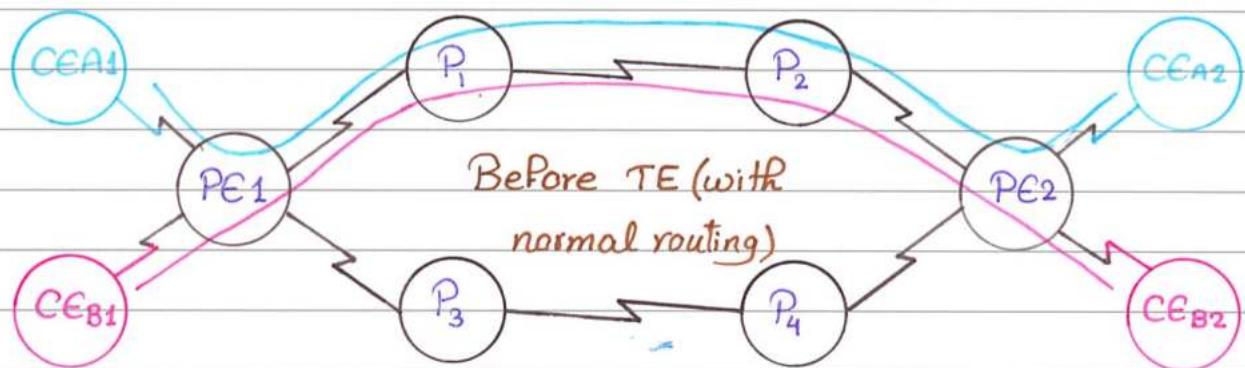
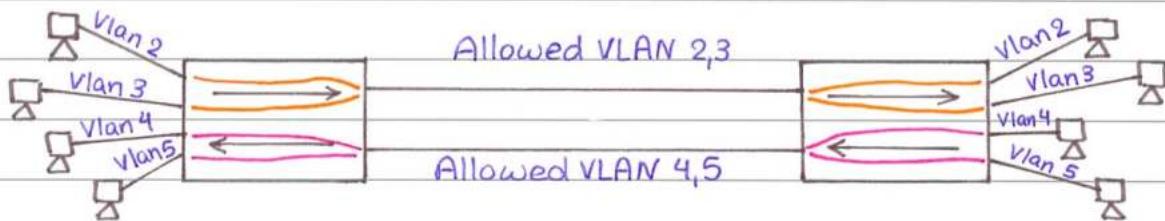
(Config) # router bgp 2

(Config-router) # neighbor PE2|3|4 remote-as 2  
(Config-router) # neighbor PE2|3|4 update-source fa0  
(Config-router) # address-family L2VPN VPLS  
(Config-router-af) # neighbor PE2|3|4 activate  
(Config-router-af) # neighbor PE2|3|4 next-hop-self

## Session 25

### What is Traffic engineering?

It is manipulate traffic to fit network resources.



After TE



After TE



TE Implementation :

1. Static Routing.
2. Policy map.
3. Adjust IGP metric.
4. Using MPLS.

## Session 25 Part 2

### MPLS TE

TE using MPLS is formed using traffic engineering tunnels.

- what is TE Tunnel ?

TE Tunnel is a path that contains Head End, Tail End  
& Mid Points.

Tunnel Source    Tunnel dst

- To Create TE Tunnel we need :

- OSPF | ISIS with TE extensions.
- Calculation method of best path using CSPF = Constrained Shortest Path First. ( $P_{\text{calc}} \equiv P_{\text{path calc}}$ )
- RSVP with MPLS TE extension.

- Configuration on Head End :

(Config) # interface Tunnel 0

(Config-if) # ip unnumbered loopback 0

(Config-if) # tunnel destination Tail End

(Config-if) # tunnel mode mpls traffic-eng

(Config-if) # tunnel mpls traffic-eng bandwidth \_\_\_\_\_ kbps

(Config-if) # tunnel mpls traffic-eng path-option \*

{dynamic | explicit} name Anwar

(Config-if) # tunnel mpls traffic-eng auto route announce

(Config) # ip path-option explicit name Anwar

(Config-exp-path) # next-address PE 2

(Config-exp-path) # next-address P1

(Config-exp-path) # next-address P2

(Config-if) # tunnel mpls traffic-eng path-option 1 explicit name Anwar

(Config-if) # tunnel mpls traffic-eng path-option 2 dynamic

## Session 25 Part 3

### MPLS TE Operation

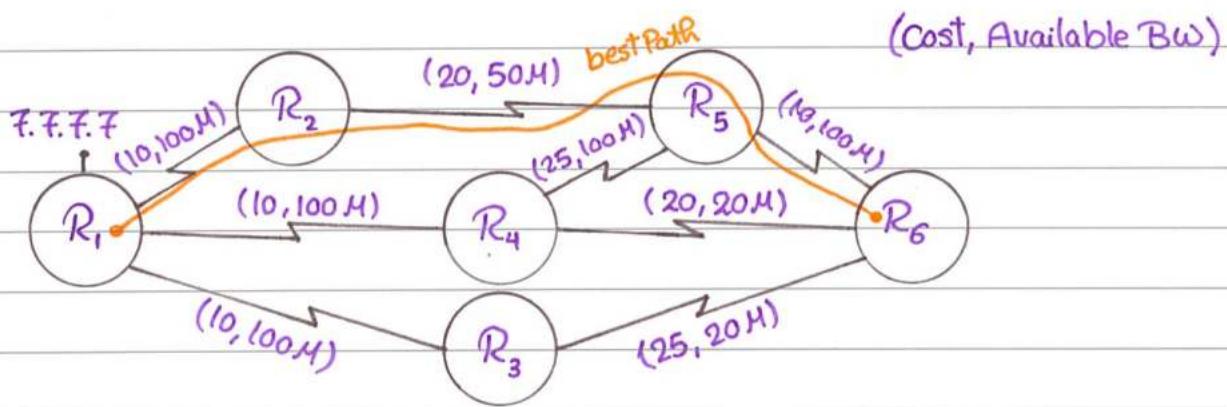
**Phase 1: Topology Discovery**

using OSPF or ISIS with TE extension

**Advertise (max Bw, reserved Bw & available Bw)**

using opaque LSAs (Type 9, 10, 11)

link local      area local      Intra AS



(Config) # router ospf 1

(Config-router) # mpls traffic-eng router-id 60

(Config-router) # mpls traffic-eng area 0

⇒ Build TE LSDB

**Phase 2: Choosing Constraint best path**

using CSPF(PCalc) :

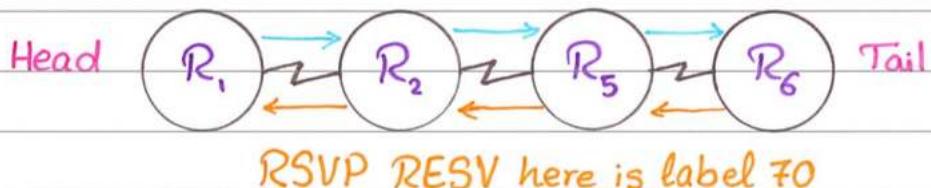
- Available BW.
- Least Cost.
- More Available BW.
- Least hop Count.

⇒ Best Path From Head End to Tail End.

Phase 3 : Build LSP (label switch path)

use RSVP = Resource Reservation Protocol

RSVP Path, i need label for Tunnel 0 (BW=30Mbps)



RSVP RESV here is label 70

→ every router :

(Config) # ip cef

(Config) # mpls traffic-eng

(Config-if) # mpls ip

(Config-if) # ip rsvp bandwidth [—]

→  $R_i$  :

(Config) # ip route ——————— Two

OR PBR

OR autoroute announce → on all routers

→ in this scenario :

use only one label = TE label

another scenario :

use 3 labels = TE label, LDP(next PE), & MP-BGP(vrf).