

BGP4 workshop scenario

Presented by:

Bruce Morgan Bruce.Morgan@aarnet.edu.au

Terutaka Komorizono teru@ntt.co.th

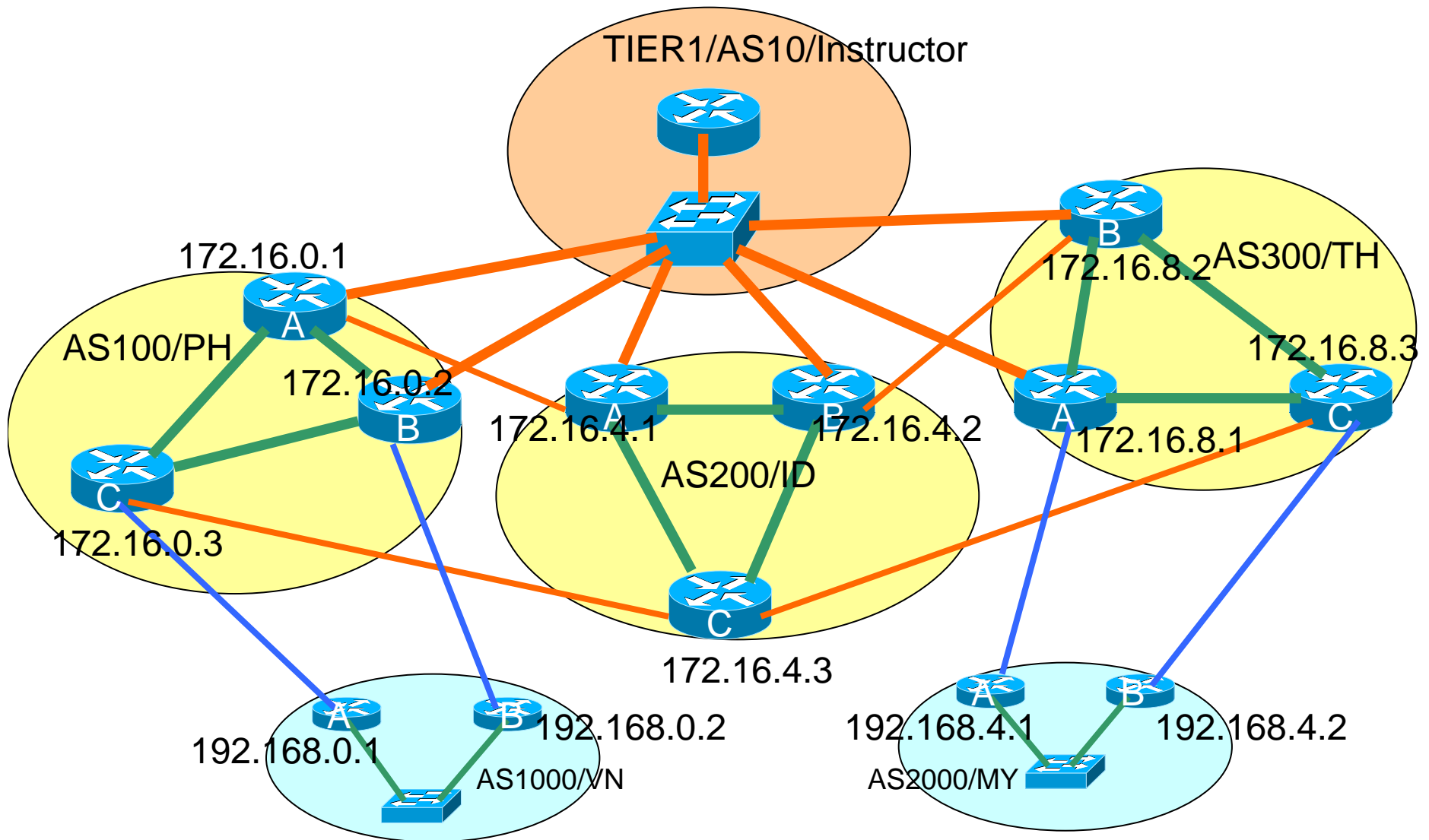
Written by:

Terutaka Komorizono teru@ntt.co.th

Introduction

- Presentation has many configuration examples
- Using Cisco IOS CLI
- Aimed at Service Providers
 - Techniques can be used by many enterprises too
- Feel free to ask questions

Target Network topology



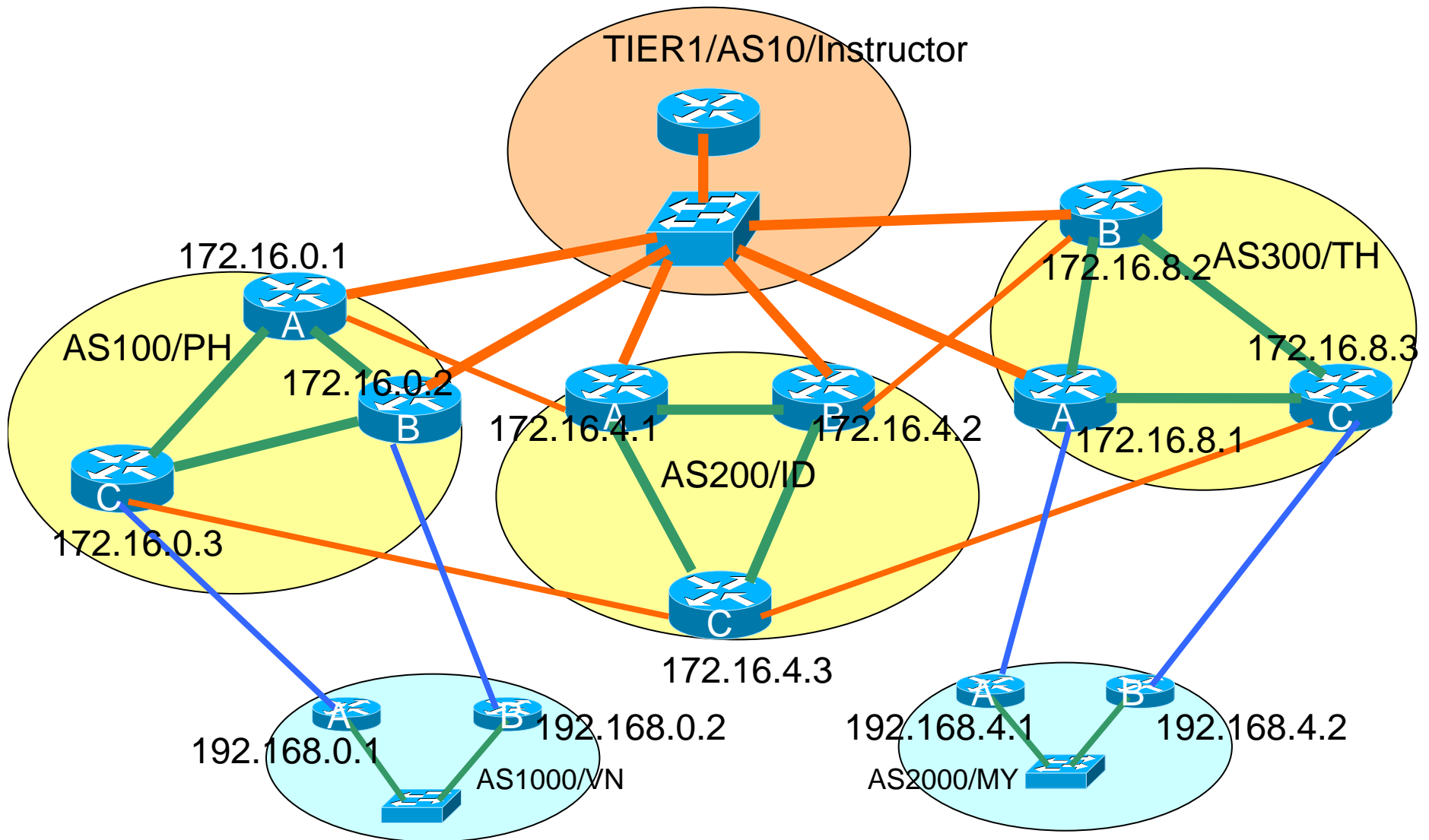
Allocate ASN and IP address

Party	ASN	IP address
TIER1	10	10.0.0.0/22
ISP-A/PH	100	172.16.0.0/22
ISP-B/ID	200	172.16.4.0/22
ISP-C/TH	300	172.16.8.0/22
USER1/VN	1000	192.168.0.0/22
USER2/MY	2000	192.168.4.0/22

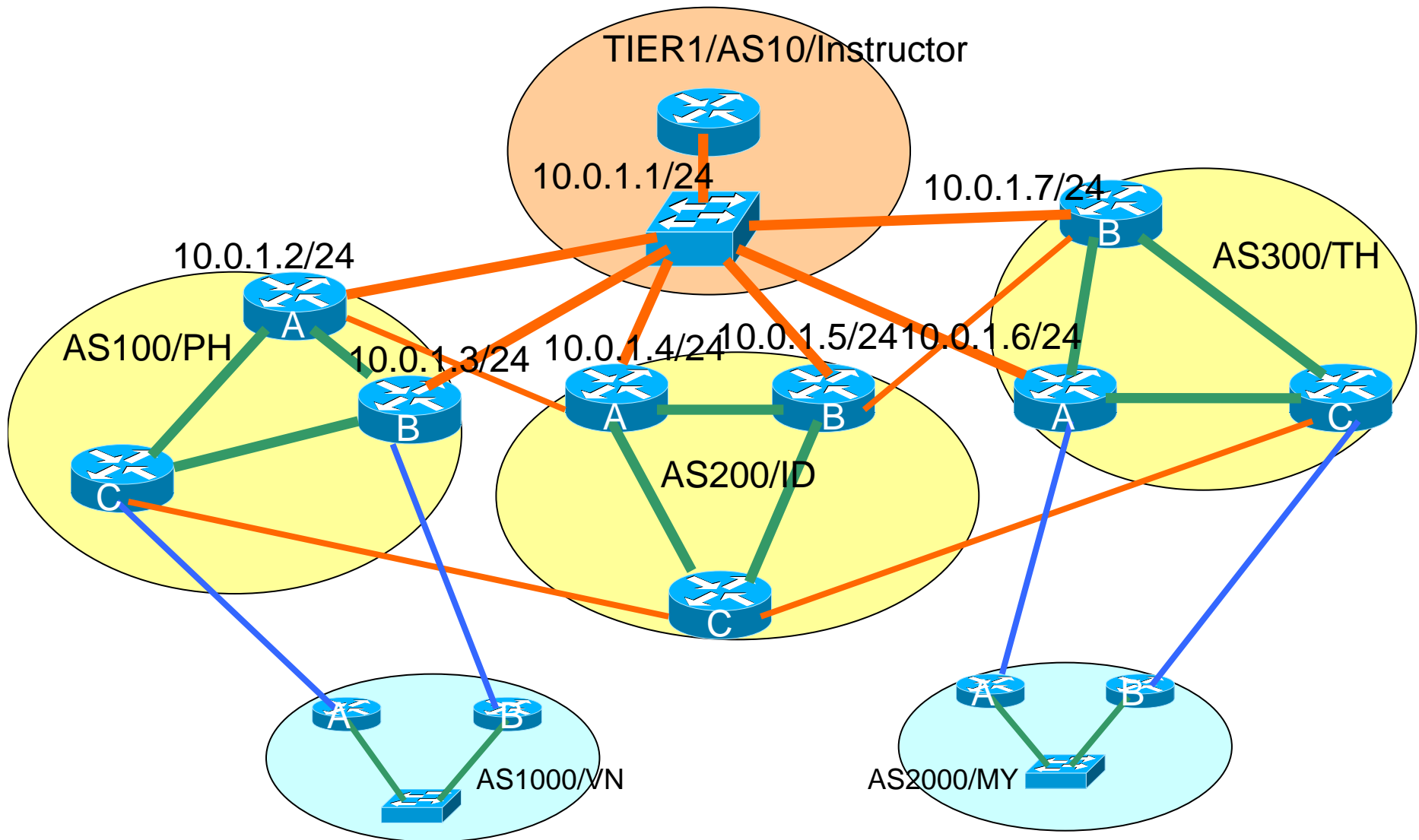
Allocate ASN and IP address

AS	hostname	loopback0 address
TIER1/AS1	tier1	10.0.0.1/32
ISP-A/AS100	my0[012]	172.16.0.[123]/32
ISP-B/AS200	ld0[012]	172.16.4.[123]/32
ISP-C/AS300	th0[012]	172.16.8.[123]/32
Uesr1/AS1000	user1[01]	192.168.0.[12]/32
User2/AS2000	user2[01]	192.168.4.[12]/32

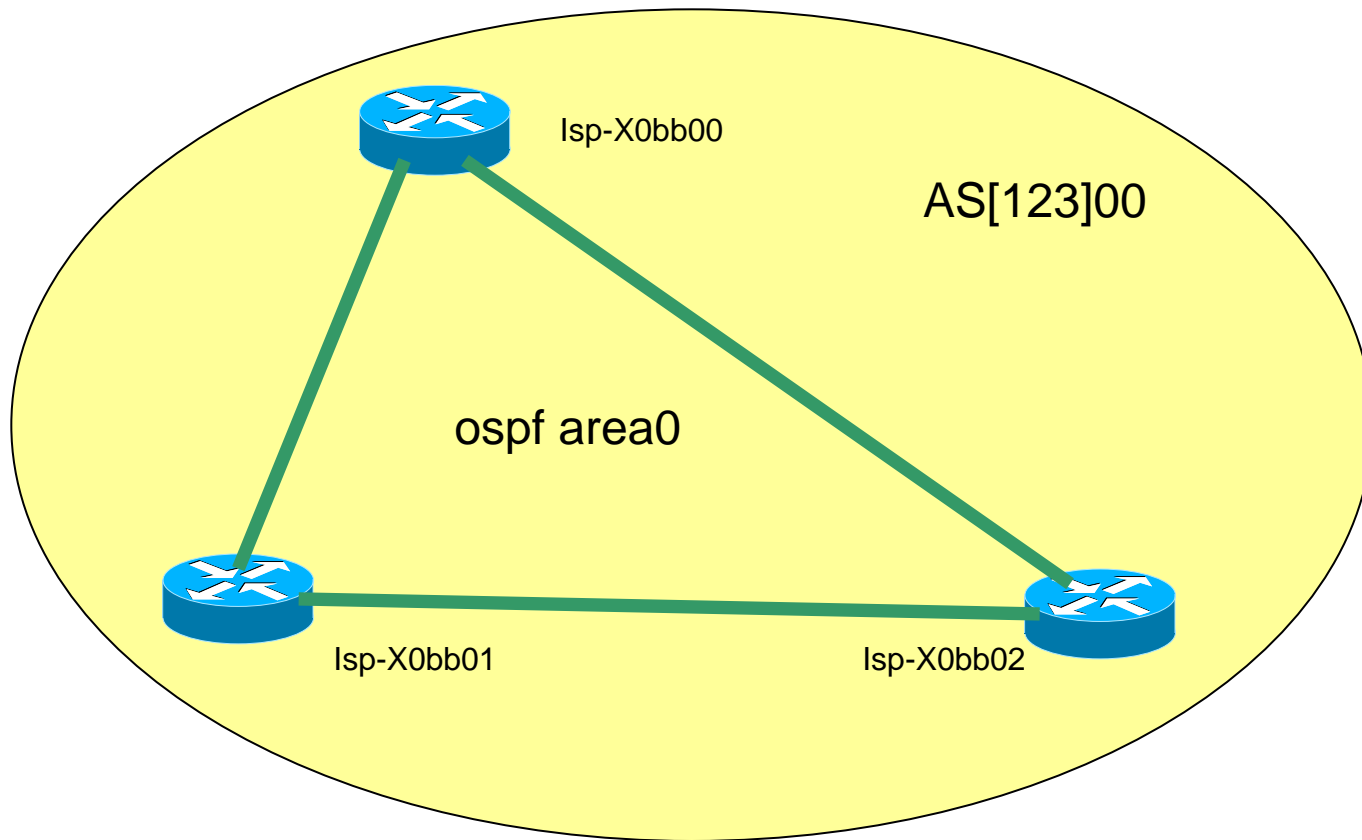
Loopback ip addresses



Physical interface ip addresses



Step1: IGP



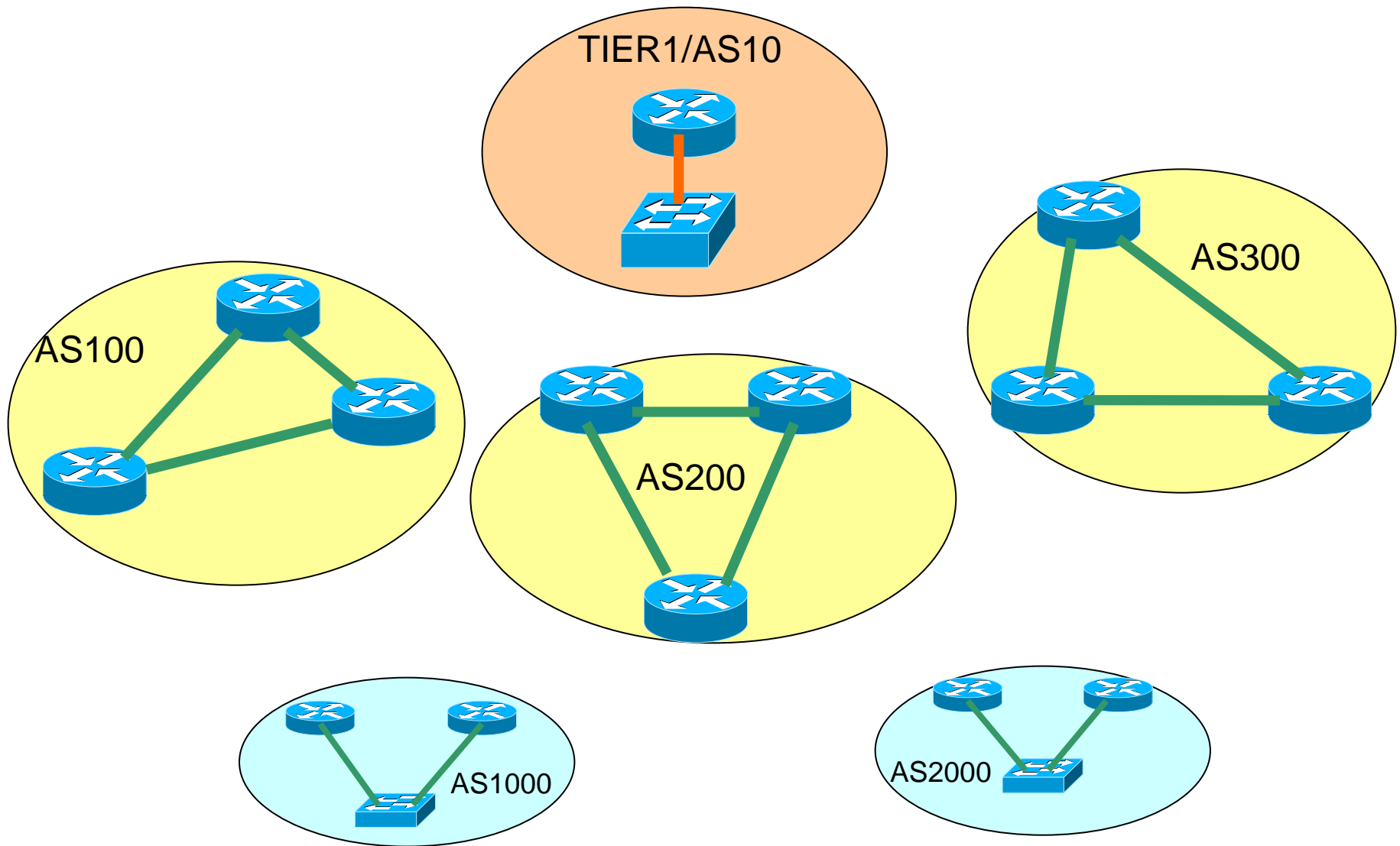
sample ospf config

```
!  
interface serial X  
  ip ospf authentication  
  ip ospf authentication-key *****  
!  
router ospf xxx  
  log-adjacency-changes  
  area 0 authentication message-digest  
  passive-interface default  
  no passive-interface <#i/f>  
  no passive-interface <#i/f>  
  network <#loopback ip> 0.0.0.0 area 0  
  network <#loopback ip> 0.0.0.0 area 0  
  network <#loopback ip> 0.0.0.0 area 0  
  network <#Ptp ip> 0.0.0.3 area 0  
  network <#Ptp ip> 0.0.0.3 area 0  
  network <#Ptp ip> 0.0.0.3 area 0  
!
```

After configure ospf

- show ip route
- show ip route ospf
- show ip ospf neighbor
- debug ip ospf adj

Step2: ibgp



sample ibgp config

```
!AS200 router config
!
router bgp 200
  no synchronization
  bgp log-neighbor-changes
  network X.X.X.X mask 255.255.255.0
  network Y.Y.Y.Y mask 255.255.255.0
  neighbor <#router1 lo0> remote-as 200
  neighbor <#router1 lo0> description <#router-name>
  neighbor <#router1 lo0> update-source Loopback0
  neighbor <#router1 lo0> send-community
  neighbor <#router2 lo0> remote-as 200
  neighbor <#router2 lo0> description <#router-name>
  neighbor <#router2 lo0> update-source Loopback0
  neighbor <#router2 lo0> send-community
  no auto-summary
!
!
ip route X.X.X.X 255.255.255.0 null0
ip route Y.Y.Y.Y 255.255.255.0 null0
!
```

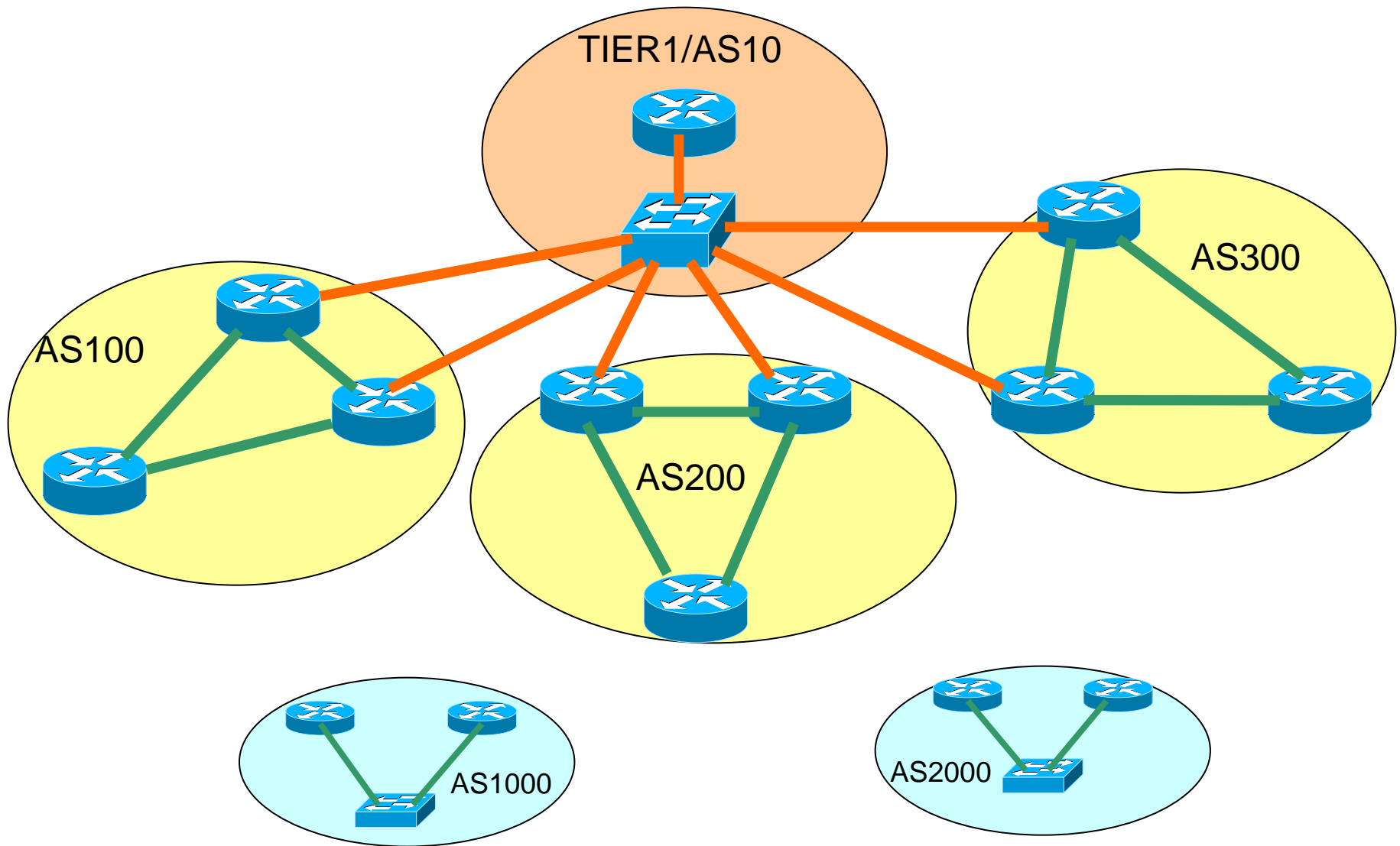
sample peer-group config

```
!AS200 router config
!
router bgp 200
  no synchronization
  bgp log-neighbor-changes
  network X.X.X.X mask 255.255.255.0
  network Y.Y.Y.Y mask 255.255.255.0
  neighbor ISP-N-IBGP peer-group
  neighbor ISP-N-IBGP remote-as 200
  neighbor ISP-N-IBGP update-source Loopback0
  neighbor ISP-N-IBGP send-community
  neighbor <#router1 lo0> peer-group ISP-N-IBGP
  neighbor <#router1 lo0> description <#router-name>
  neighbor <#router2 lo0> peer-group ISP-N-IBGP
  neighbor <#router2 lo0> description <#router-name>
!
!
ip route X.X.X.X 255.255.255.0 null0
ip route Y.Y.Y.Y 255.255.255.0 null0
!
```

After configure iBGP

- show ip route
- show ip bgp
- show ip bgp x.x.x.x/xx
- show ip bgp summary
- show ip bgp neighbor x.x.x.x advertised-route
- show ip bgp neighbor x.x.x.x route
- show ip bgp neighbor x.x.x.x received-route
- debug ip bgp

Step3: ebgp



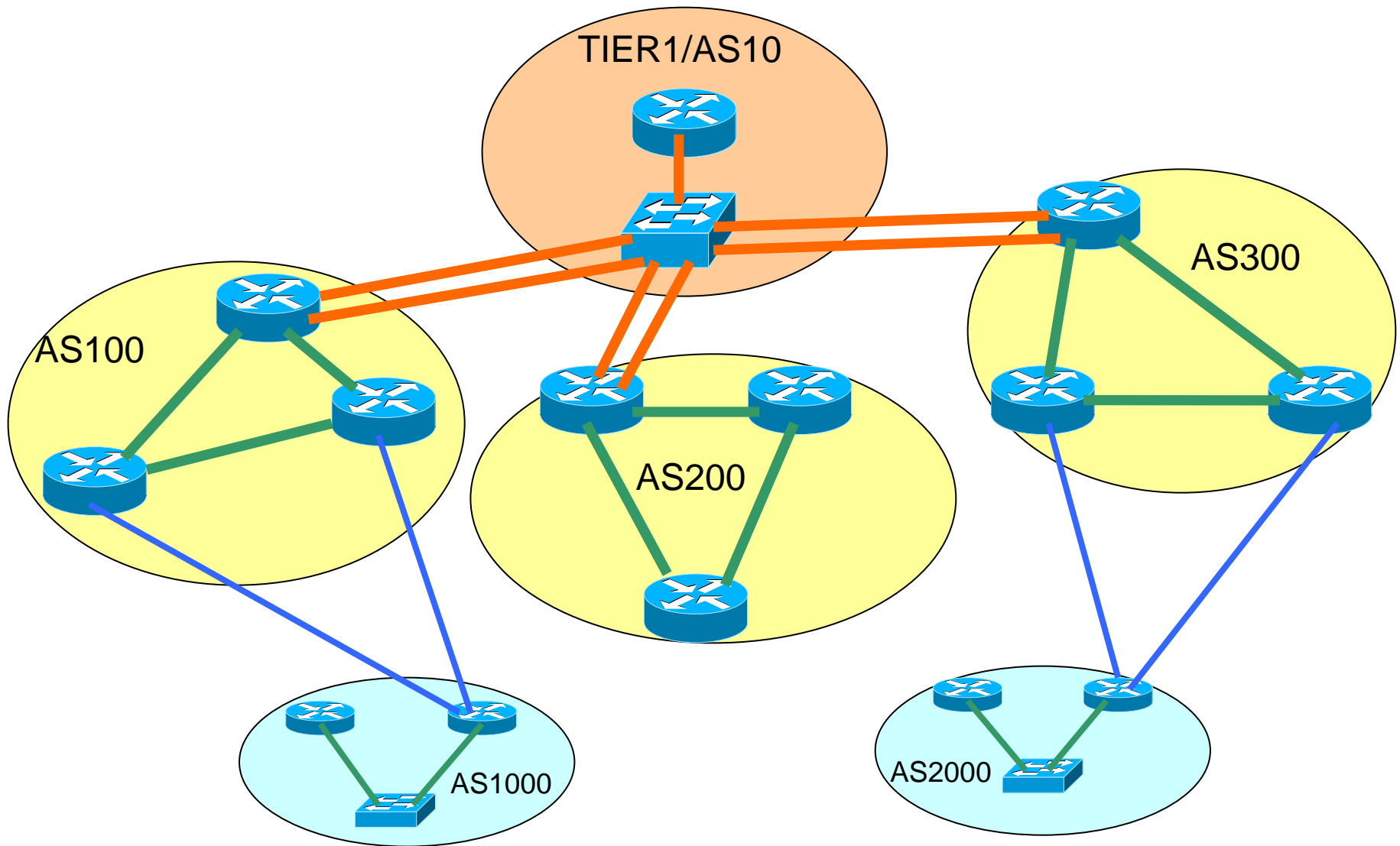
Step3: sample config

```
!Border router in AS200 config
!  
router bgp 200  
  neighbor <#neighbor router ptp> remote-as 1  
  neighbor <#neighbor router ptp> description TIER1 AS1  
  neighbor <#neighbor router ptp> next-hop-self  
  neighbor <#neighbor router ptp> soft-reconfiguration inbound  
!
```


After configure eBGP

- show ip route
- show ip bgp
- show ip bgp x.x.x.x/xx
- show ip bgp summary
- show ip bgp neighbor x.x.x.x advertised-route
- show ip bgp neighbor x.x.x.x route
- show ip bgp neighbor x.x.x.x received-route
- debug ip bgp

Step4: ebgp



Step4: sample config

```
!Aggregation router in AS200 config
!  
router bgp 200  
  neighbor <#neighbor router ptp> remote-as 1000  
  neighbor <#neighbor router ptp> description USER AS1000  
  neighbor <#neighbor router ptp> next-hop-self  
  neighbor <#neighbor router ptp> soft-reconfiguration inbound  
!
```

Applying Policy

Applying Policy

- **Prefix filter**
Applying all received route from User at User connected router
- **AS-PATH filter**
Applying AS-PATH filter to received route from User and at User connected router
Applying AS-PATH filter to received route from PEER and at PEER connected router. Due to normally we haven't prefix-filter for PEERs link because difficult to follow PEERs announced prefixes.
- **MED**
Set in/out metric at border router with UPSTREAM and PEERS
- **Local Preference**
Set local-preference as below
 - Upstream route 90
 - User route 110

sample as-path prepend config

```
!Border router in AS200 config
!
router bgp 200
  neighbor <#neighbor router ptp> remote-as 1000
  neighbor <#neighbor router ptp> description USER AS1000
  neighbor <#neighbor router ptp> next-hop-self
  neighbor <#neighbor router ptp> soft-reconfiguration inbound
  neighbor <#neighbor router ptp> prefix-list toUPSTREAM out
!
ip community-list standard PREPEND2 permit 200:10002
!
route-map toUPSTREAM permit 10
  ! set as-path prepend twice
  match community 200:10002
  set as-path prepend 200 200
!
```

sample prefix-filter config

```
!aggregation router in AS200 config
!
router bgp 200
  neighbor <#neighbor router ptp> remote-as 1000
  neighbor <#neighbor router ptp> description USER AS1000
  neighbor <#neighbor router ptp> next-hop-self
  neighbor <#neighbor router ptp> soft-reconfiguration inbound
  neighbor <#neighbor router ptp> prefix-list USER in
!
ip prefix-list USER description USER AS1000
ip prefix-list USER seq 5 permit 192.168.0.0/24
ip prefix-list USER seq 10 permit 192.168.1.0/24
ip prefix-list USER seq 15 permit 192.168.2.0/24
ip prefix-list USER seq 20 permit 192.168.3.0/24
ip prefix-list USER seq 25 permit 192.168.4.0/24
ip prefix-list USER seq 30 permit 192.168.5.0/24
ip prefix-list USER seq 35 permit 192.168.6.0/24
ip prefix-list USER seq 40 permit 192.168.7.0/24
!
```

sample AS-PATH filter config

```
!border router in AS200 config
!
router bgp 200
  neighbor <#neighbor router ptp> remote-as 1000
  neighbor <#neighbor router ptp> description USER AS1000
  neighbor <#neighbor router ptp> next-hop-self
  neighbor <#neighbor router ptp> soft-reconfiguration inbound
  neighbor <#neighbor router ptp> filter-list 50 in
!
ip as-path access-list 50 permit ^(1000_)+$
!
```


Regular explanation

- \wedge = begin of the line

sample: $\wedge(100)$

100 200 -> OK

200 100 -> NG

- $\$$ = end of the line

sample: $(100)+\$$

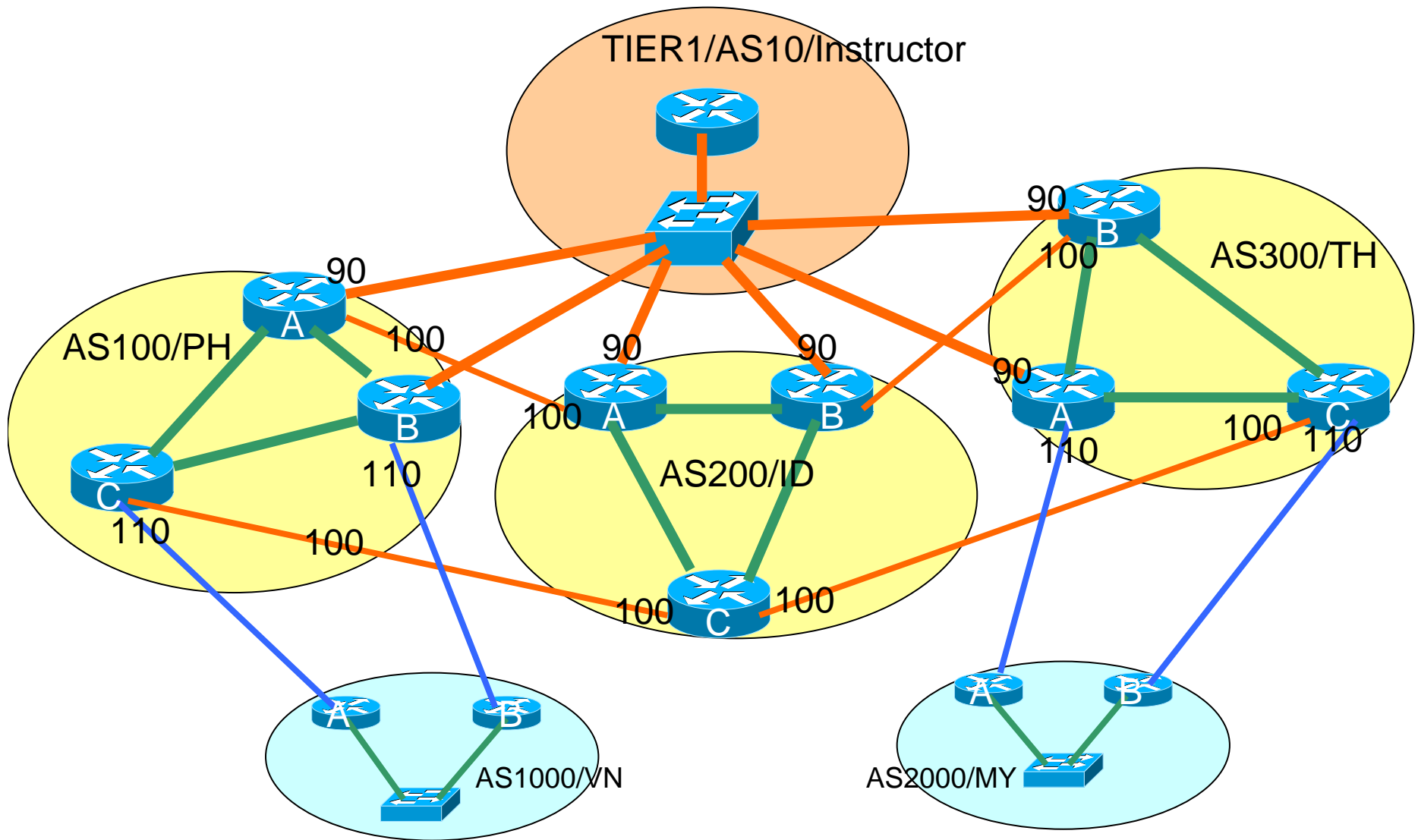
100 -> OK

200 100 -> OK

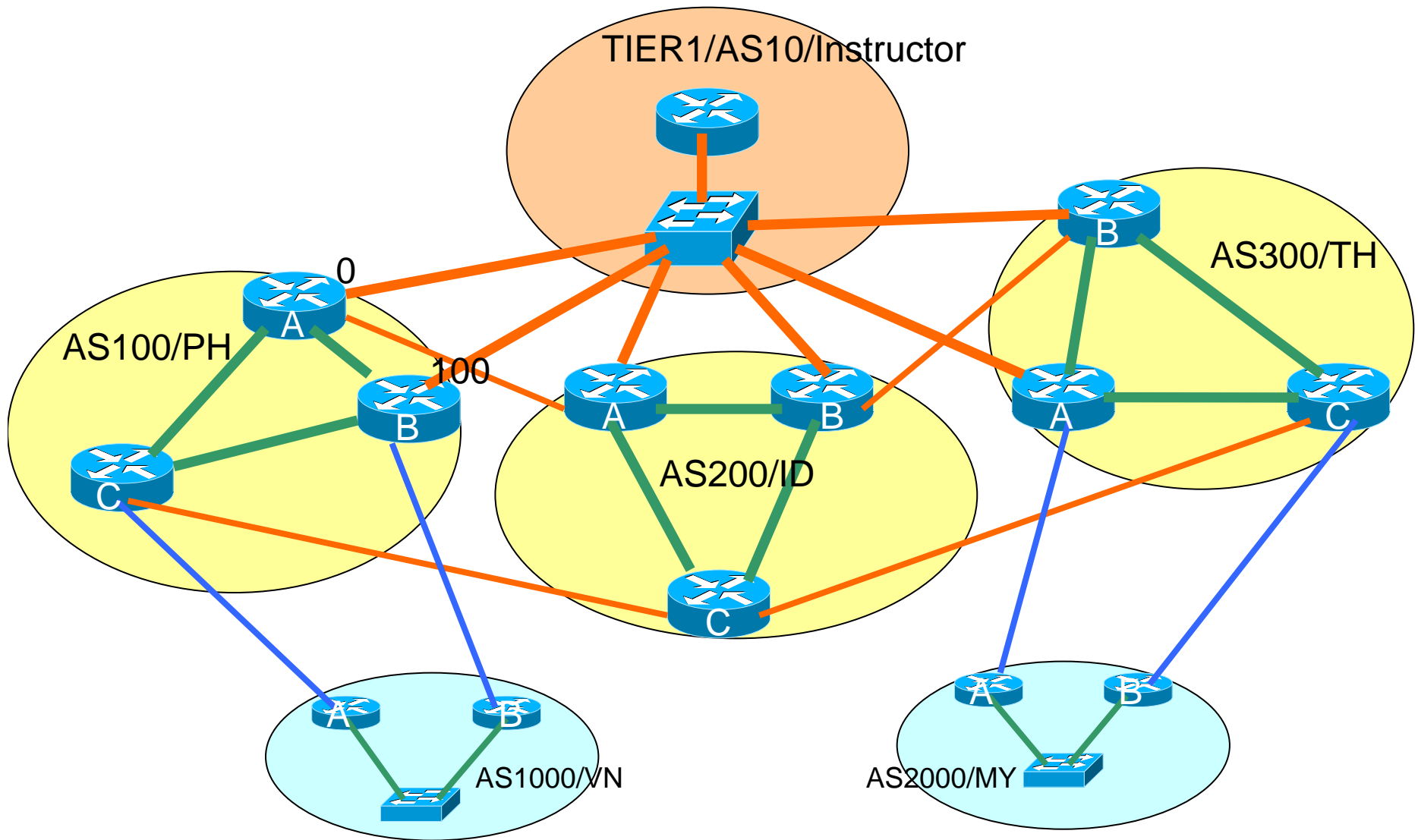
sample metric and local-preference and med config

```
!Border router in AS200 config
!
router bgp 200
  neighbor <#neighbor router ptp> remote-as 1000
  neighbor <#neighbor router ptp> description USER AS1000
  neighbor <#neighbor router ptp> next-hop-self
  neighbor <#neighbor router ptp> soft-reconfiguration inbound
  neighbor 10.0.1.5 route-map fromUPSTREAM in
  neighbor 10.0.1.5 route-map toUPSTREAM out
!
!
route-map toUPSTREAM permit 10
  set metric 0
!
route-map fromUPSTREAM permit 10
  set local-preference 90
  set metric 1000
!
```

Set local preference

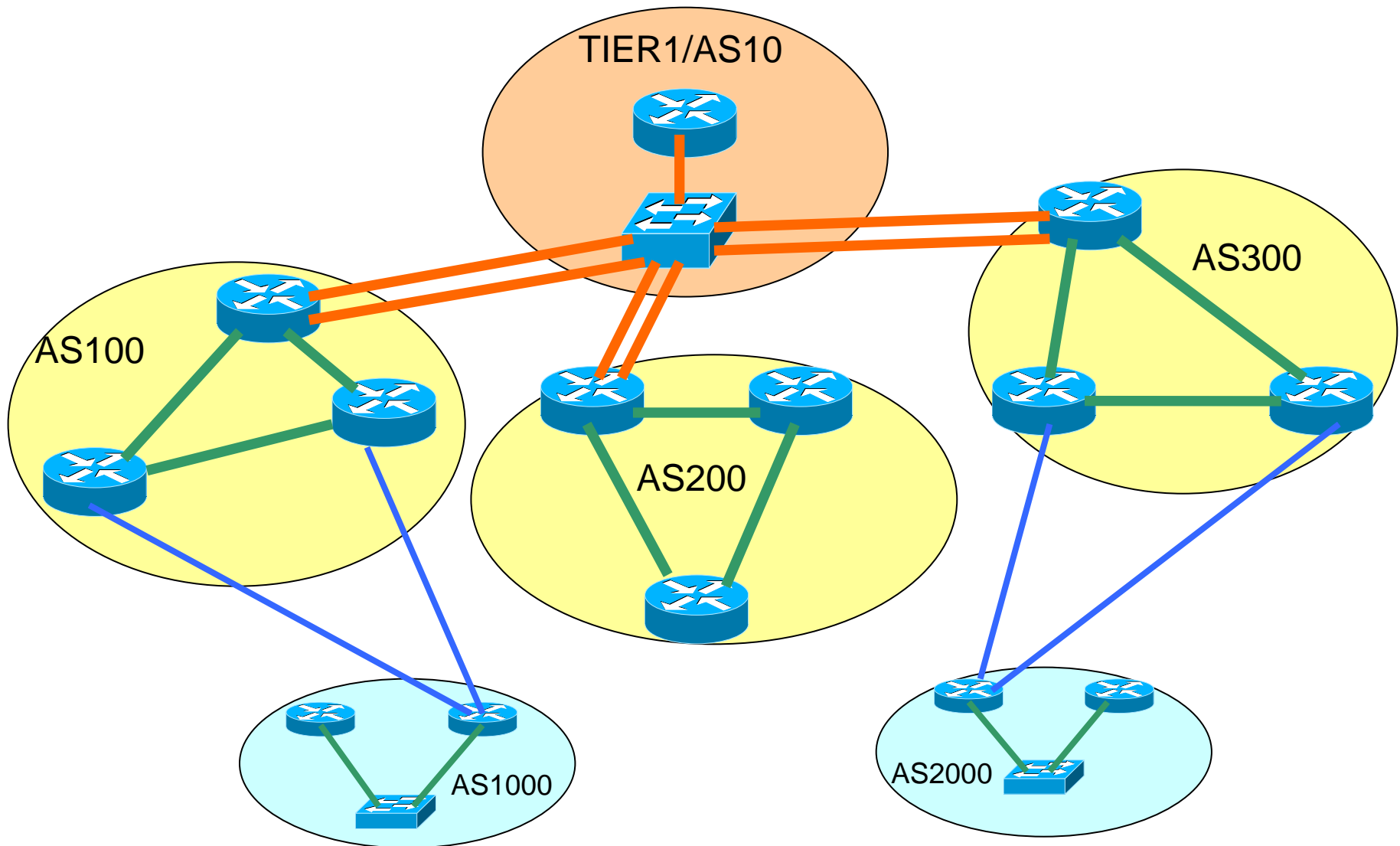


Set MED

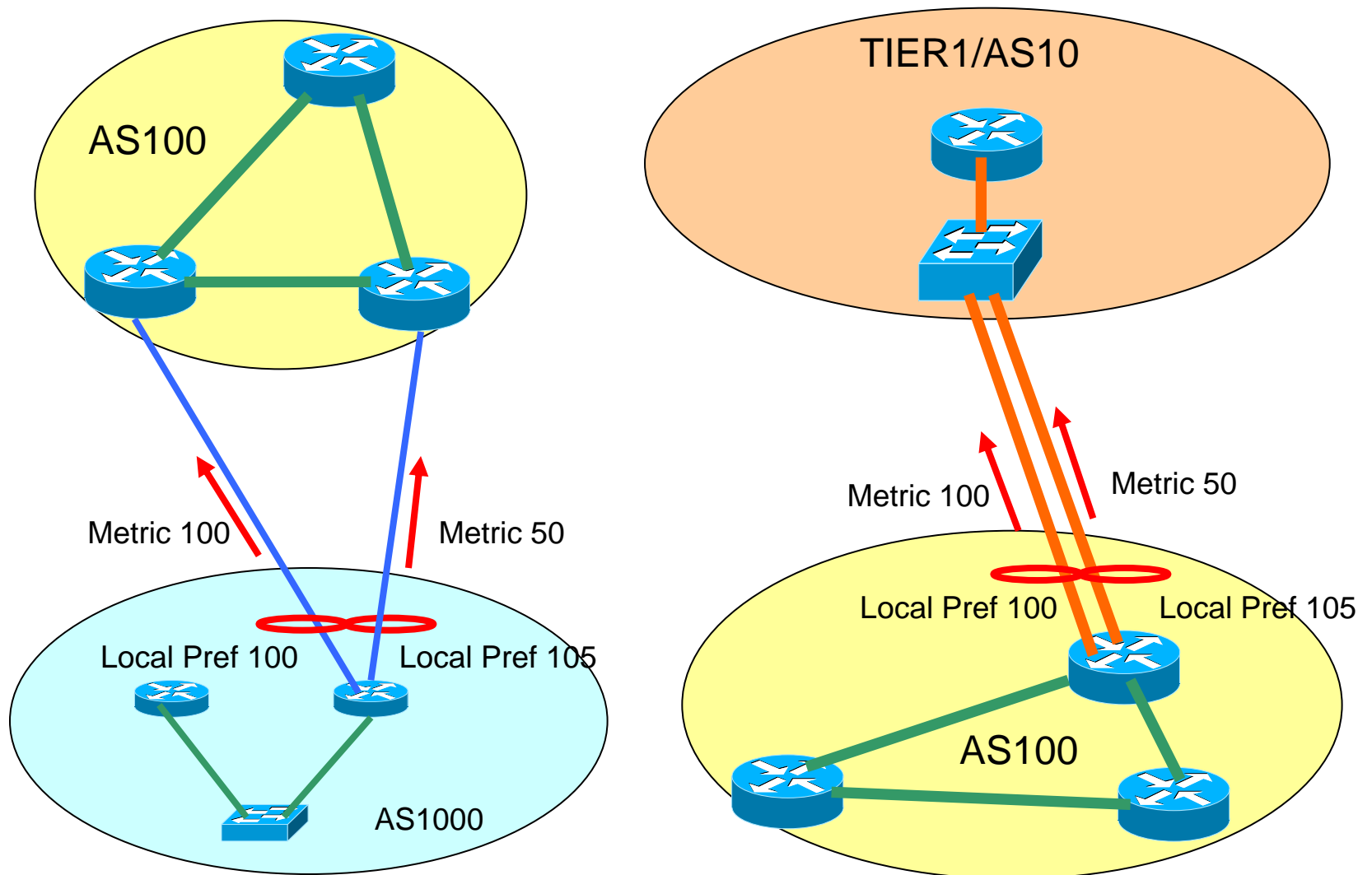


multihoming

Step5: multihoming – single router and multi path



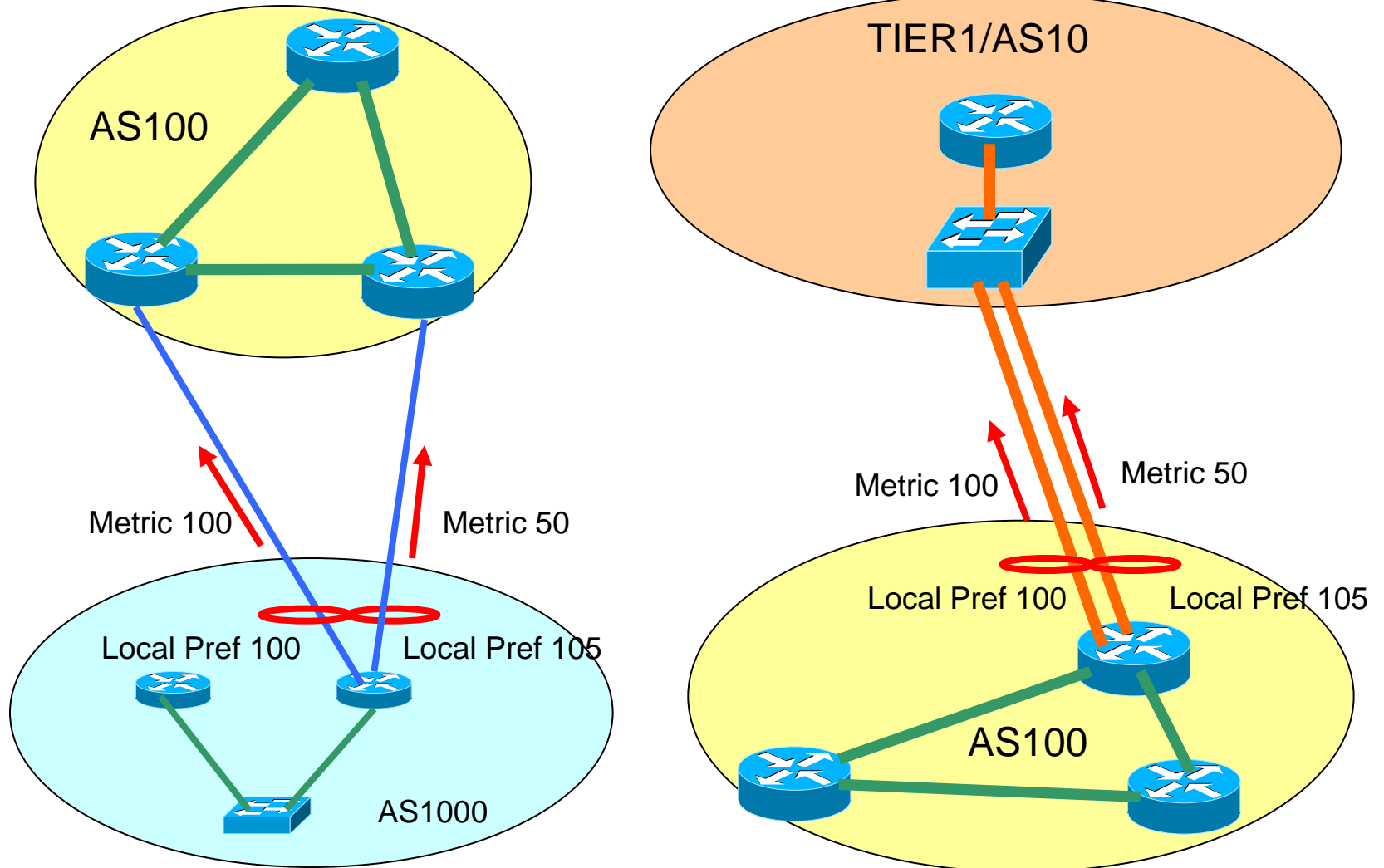
Step6: multihoming - set primary/secondary by MED and local-preference



Step6: sample config

```
!user router config
!
router bgp 1000
  neighbor A.B.C.D route-map fromUPSTREAM0 in
  neighbor A.B.C.D route-map toUPSTREAM0 out
  neighbor E.F.G.H route-map fromUPSTREAM1 in
  neighbor E.F.G.H route-map toUPSTREAM1 out
!
route-map fromUPSTREAM0 permit 10
  set local-preference 105
route-map toUPSTREAM0 permit 10
  set metric 50
!
route-map fromUPSTREAM1 permit 10
  set local-preference 100
route-map toUPSTREAM1 permit 10
  set metric 100
!
```


Step7: multihoming – load-sharing



Step8: sample config

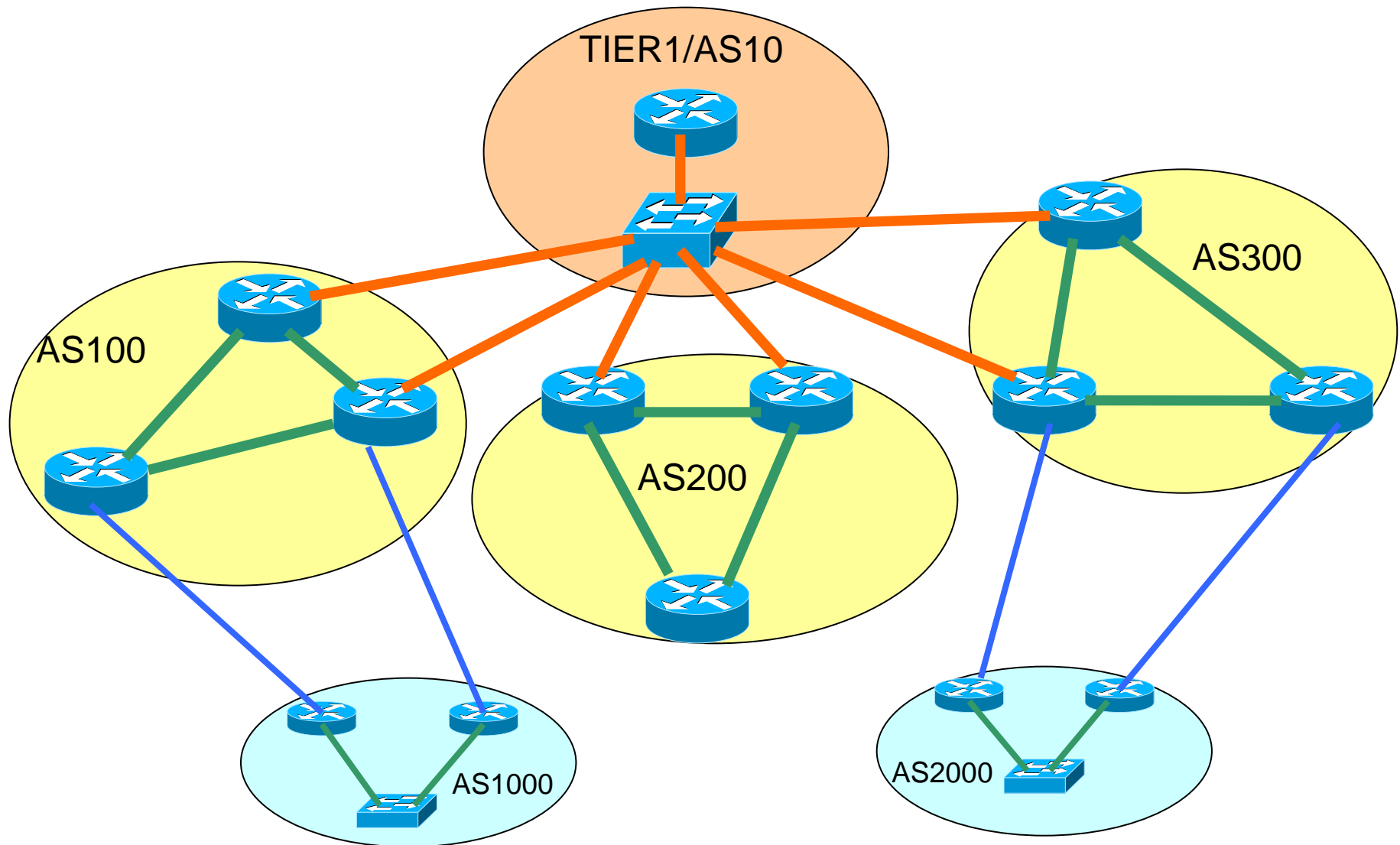
```
!user router config
!  
router bgp 1000  
  neighbor A.B.C.D route-map fromUPSTREAM0 in  
  neighbor A.B.C.D route-map toUPSTREAM0 out  
  neighbor E.F.G.H route-map fromUPSTREAM1 in  
  neighbor E.F.G.H route-map toUPSTREAM1 out  
  maximum-paths 2  
!
```

Step8: sample config

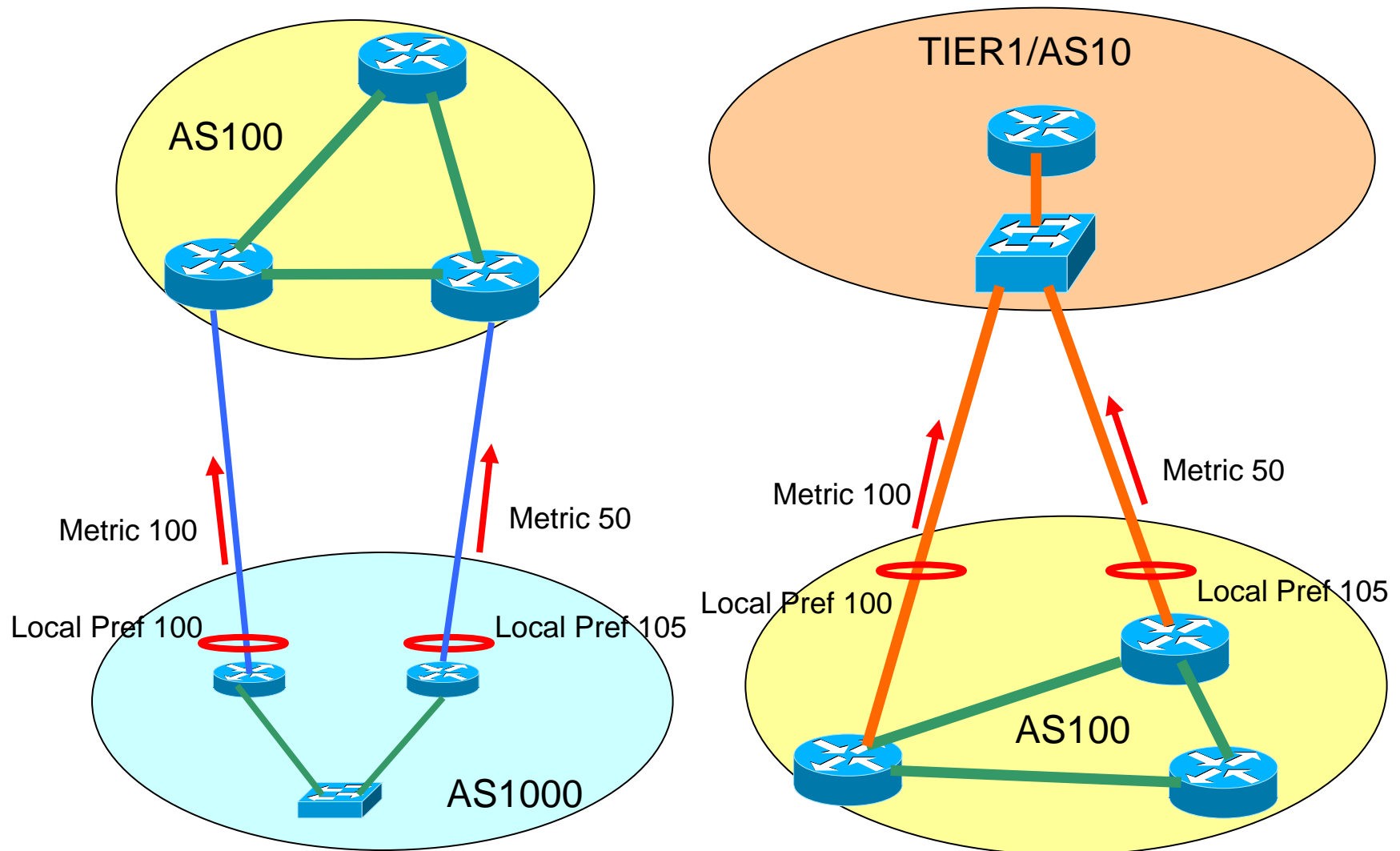
```
!border router in as1 config
!
router bgp 1
  neighbor A.B.C.D remote-as 100
  neighbor A.B.C.D ebgp-multihop 2
  neighbor A.B.C.D update-source Loopback0
!
ip route A.B.C.D 255.255.255.255 <#ptp ip1>
ip route A.B.C.D 255.255.255.255 <#ptp ip2>
!

!border router in as100 config
!
router bgp 100
  neighbor E.F.G.H remote-as 1
  neighbor E.F.G.H ebgp-multihop 2
  neighbor E.F.G.H update-source Loopback0
!
ip route E.F.G.H 255.255.255.255 <#ptp ip1>
ip route E.F.G.H 255.255.255.255 <#ptp ip2>
!
```

Step9: multihoming – two router and multi path



Step9: multihoming - set primary/secondary by MED and local-preference

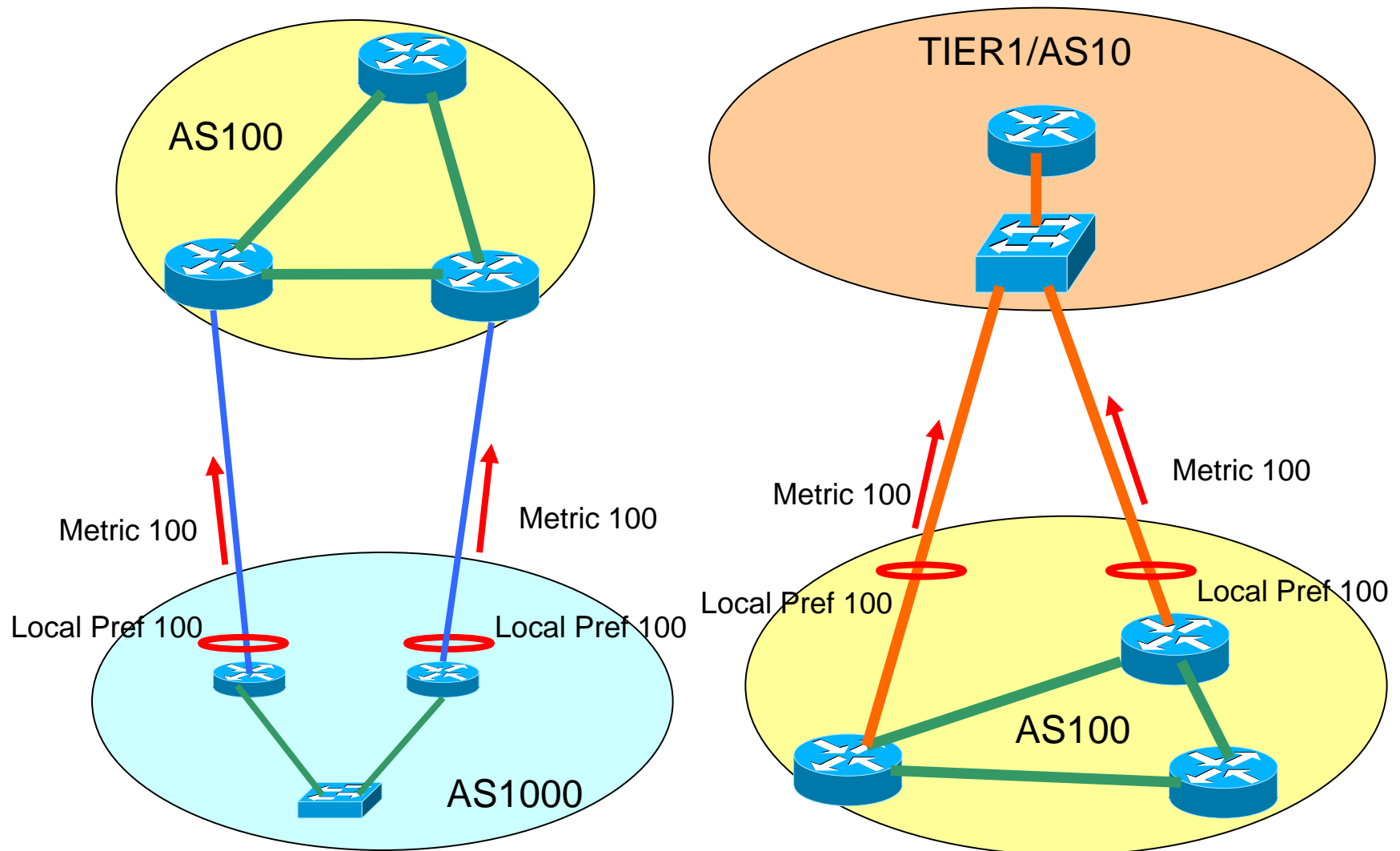


Step9: sample config

```
!Border router1 in AS200 config
!
router bgp 200
  neighbor A.B.C.D route-map fromUPSTREAM in
  neighbor A.B.C.D route-map toUPSTREAM out
!
route-map fromUPSTREAM permit 10
  set local-preference 100
route-map toUPSTREAM permit 10
  set metric 100
!

!Border router2 in AS200 config
!
router bgp 200
  neighbor A.B.C.D route-map fromUPSTREAM in
  neighbor A.B.C.D route-map toUPSTREAM out
!
route-map fromUPSTREAM permit 10
  set local-preference 105
route-map toUPSTREAM permit 10
  set metric 50
!
```

Step10: multihoming - set primary/secondary by MED and local-preference with load-sharing



Step10: sample config

```
!Border router1 in as200 config
!
router bgp 200
  neighbor A.B.C.D route-map fromUPSTREAM in
  neighbor A.B.C.D route-map toUPSTREAM out
!
route-map fromUPSTREAM permit 10
  set local-preference 100
route-map toUPSTREAM permit 10
  set metric 100
!

!Border router2 in as200 config
!
router bgp 200
  neighbor A.B.C.D route-map fromUPSTREAM in
  neighbor A.B.C.D route-map toUPSTREAM out
!
route-map fromUPSTREAM permit 10
  set local-preference 100
route-map toUPSTREAM permit 10
  set metric 100
!
```


Step 11: BGP communities

This chapter provides how ISP make a policy by bgp communities

Step11: example: private peering - three types peers

- This examples how communities might be used at the peering edge of an ISP network
- ISP has three types BGP peers:
 - Customer
 - IXP and Private Peers
 - Upstream
- The prefixes received from each can be classified using communities
- Customer can opt to receive any or all of the above
- Basic usage:
 - When receiving routes, tagging communities to the received routes at the received router
 - When announcing routes, do action along tag at the place announced route

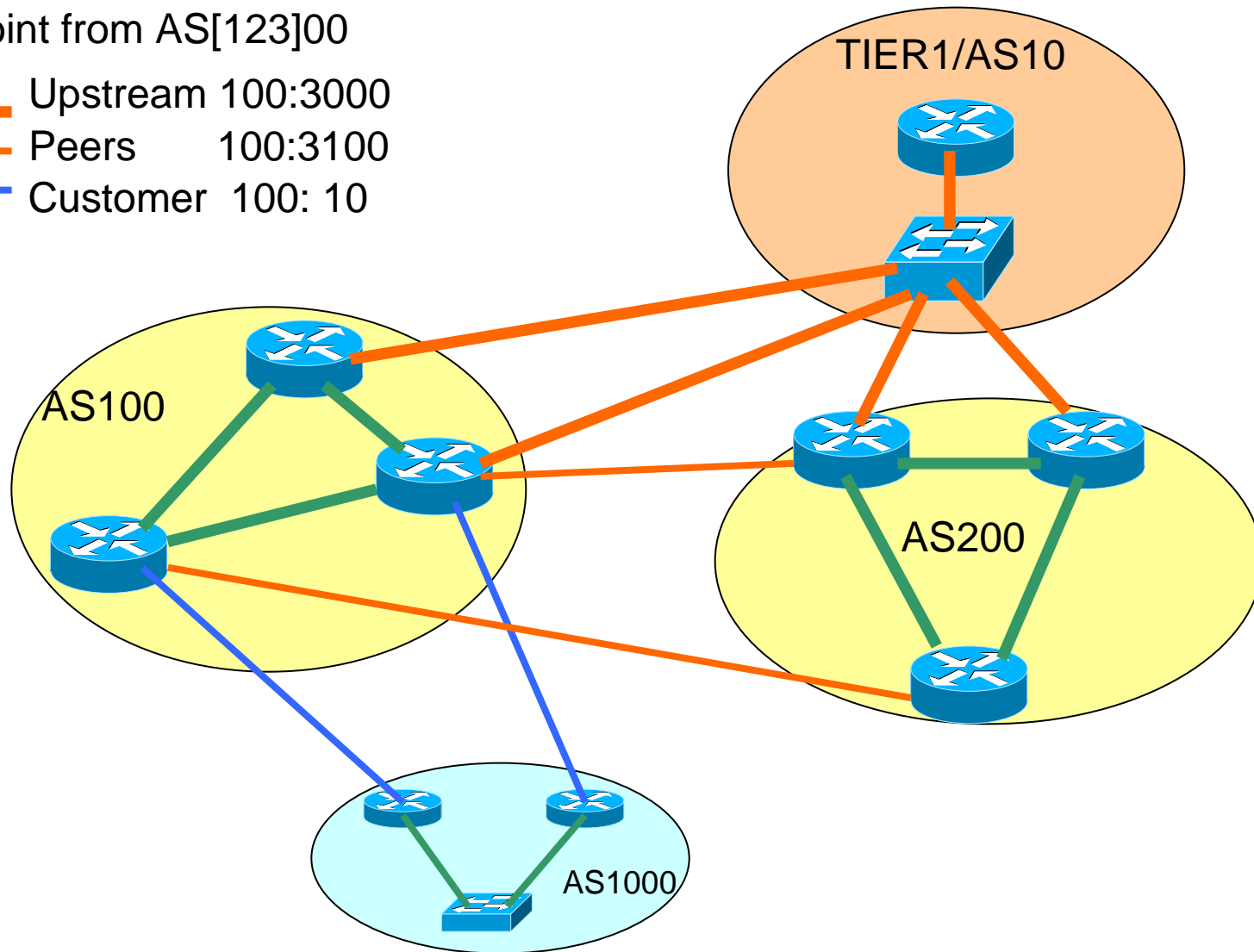
Step11: example: private peering – Internet Edge

- Community assignments:
 - Customer prefix: community 100:3000
 - Peers prefix: community 100:3100
- BGP customer who buys local connectivity gets 100:3000
- BGP customer who buys peers connectivity receives community 100:300 and 100:3100
- Customer who wants “the Internet” gets everything
 - Gets default route via “default-originate”
 - Or pays money to get all 160k prefixes

Step11: example: private peering

View point from AS[123]00

- Upstream 100:3000
- Peers 100:3100
- Customer 100: 10



Step11: example: private peering

```
!Border router in AS100 config
```

```
!
```

```
router bgp 100
```

```
neighbor TRANSIT-CUSTOMER peer-group
```

```
neighbor TRANSIT-CUSTOMER route-map TRANSIT-CUSTOMER in
```

```
neighbor PEERS peer-group
```

```
neighbor PEERS route-map PEERS in
```

```
neighbor UPSTREAM peer-group
```

```
neighbor UPSTREAM ...
```

```
!
```

```
route-map TRANSIT-CUSTOMER permit 10
```

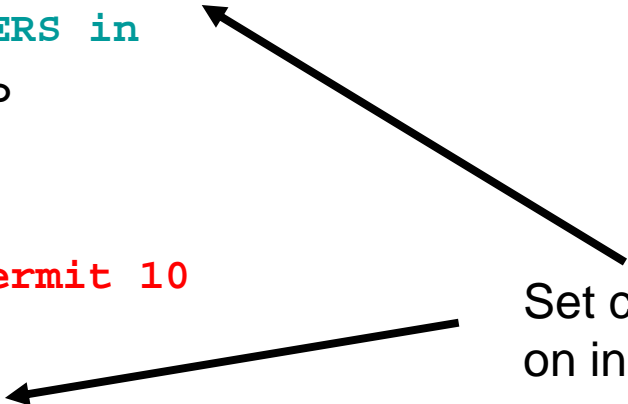
```
set community 100:3000
```

```
route-map PEERS permit 10
```

```
set community 100:3100
```

```
!
```

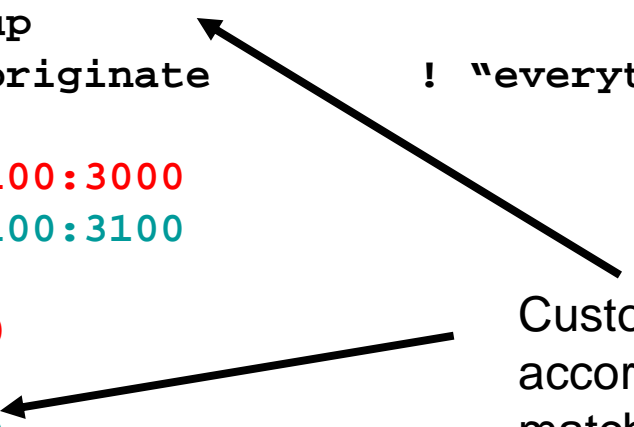
Set communities
on inbound
announcements



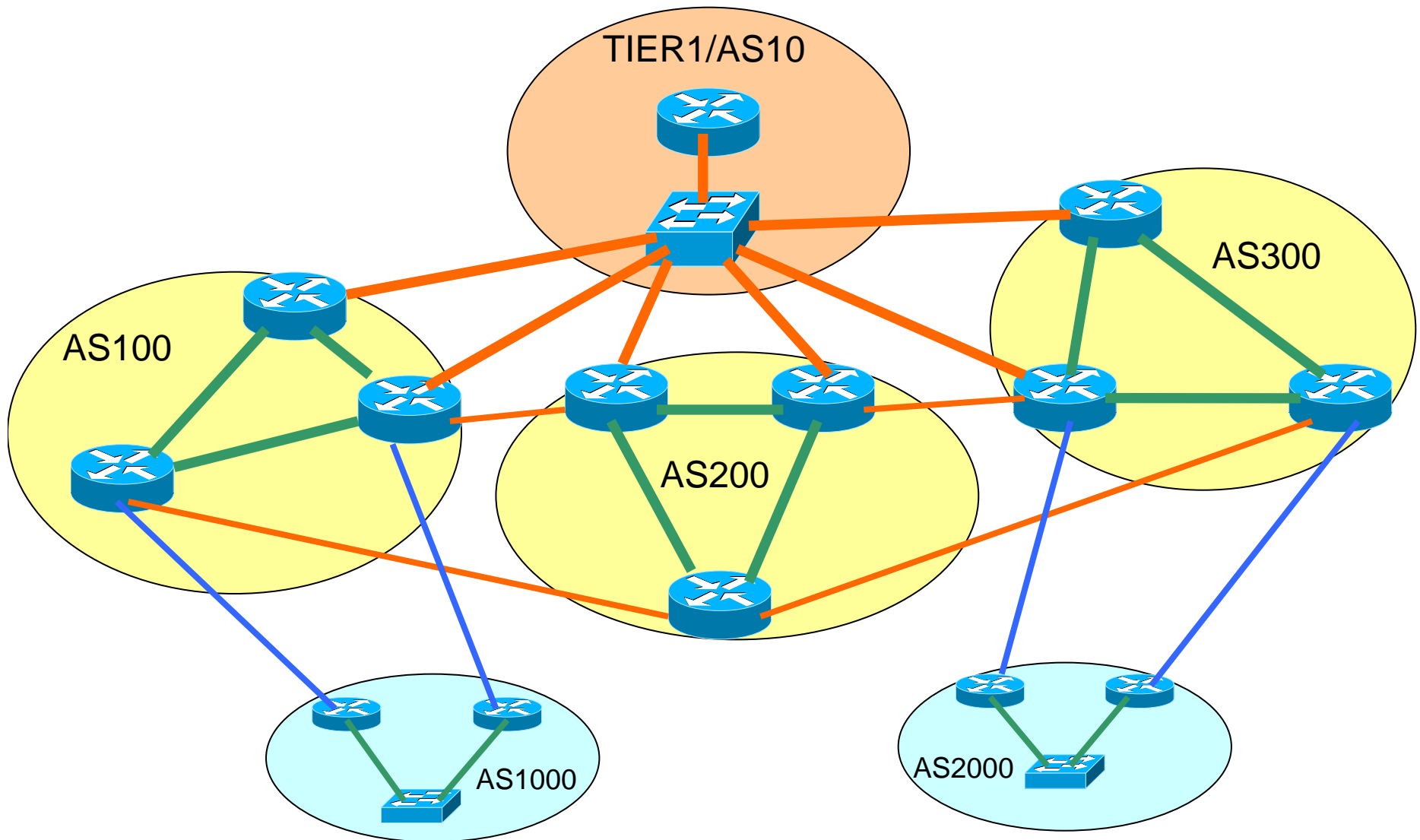
Step11: example: private peering

```
!aggregation (customer) router in AS100 config
!  
router bgp 100  
  neighbor CUSTOMER1 peer-group  
  neighbor CUSTOMER1 route-map CUSTOMER1 out  ! Local routes  
  neighbor CUSTOMER2 peer-group  
  neighbor CUSTOMER2 route-map CUSTOMER2 out  ! Local+Peers routes  
  neighbor CUSTOMER3 peer-group  
  neighbor CUSTOMER3 default-originate      ! "everything"  
!  
ip community-list 23 permit 100:3000  
ip community-list 24 permit 100:3100  
!  
route-map CUSTOMER1 permit 10  
  match community 23  
route-map CUSTOMER2 permit 10  
  match community 24  
!
```

Customer gets prefixes
according to community
matches



TRY! Step11: private peering



TRY! Step11: private peering

- Policies:
 - To upstream: announce local routes
 - From upstream: receive full routes
 - To peers: announce local and customer routes
 - From peers: receive peers routes
 - To customers: announce full routes
 - From customers: receive customers routes
- Community assignments:
 - Full routes: community <#ASN>:3000
 - Customer prefix: community <#ASN>:10
 - Peers prefix: community <#ASN>:3100

Step 11: sample config

```
!Border router in AS200 config
!
```

```
router bgp 200
```

```
neighbor A.B.C.D remote-as 1
```

```
neighbor A.B.C.D route-map fromUPSTREAM in
```

```
neighbor A.B.C.D route-map toUPSTREAM out
```

```
neighbor E.F.G.H remote-as 1000
```

```
neighbor E.F.G.H default-originate
```

```
neighbor E.F.G.H route-map fromTRANSIT-CUSTOMER in
```

```
neighbor E.F.G.H route-map toTRANSIT-CUSTOMER out
```

```
neighbor J.K.L.M prefix-list PEER in
```

```
neighbor J.K.L.M route-map fromPEER in
```

```
neighbor J.K.L.M route-map toPEER out
```

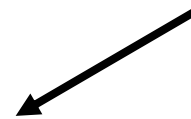
```
!
```

```
ip bgp-community new-format
```

```
!
```

```
!continue next slide...
```

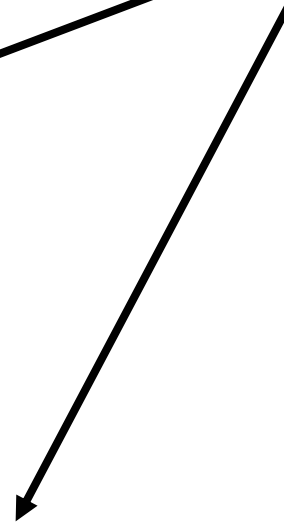
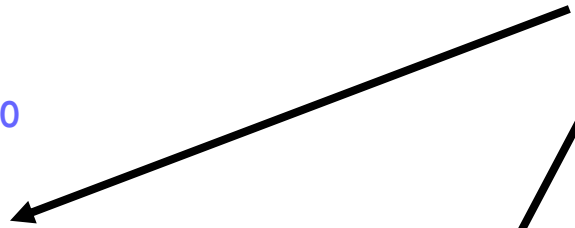
Set communities
on inbound
announcements



Step11: sample config

```
!  
route-map fromUPSTREAM permit 10  
  set local-preference 90  
  set community 200:3000  
!  
route-map toUPSTREAM permit 10  
  set community none  
!  
route-map fromPEER permit 10  
  set local-preference 100  
  set community 200:3100  
!  
route-map toPEER permit 10  
  set community none  
!  
route-map toTRANSIT-CUSTOMER permit 10  
  set community none  
!  
route-map fromTRANSIT-CUSTOMER permit 10  
  set community 200:10 additive  
  set local-preference 110  
!
```

Set communities
on inbound
announcements



Step11: sample config (Cont.)

```
!aggregation (customer) router in AS200 config
!
router bgp 200
  neighbor UPSTREAM peer-group
  neighbor UPSTREAM route-map toUPSTREAM out          ! Local routes
  neighbor PEERS peer-group
  neighbor PEERS route-map toPEERS out              ! Local routes
  neighbor TRANSIT-CUSTOMER peer-group
  neighbor TRANSIT-CUSTOMER route-map toTRANSIT-CUSTOMER out ! Full routes
!
ip community-list 21 permit 200:10                 ! Customer routes
ip community-list 23 permit 200:3000               ! Upstream routes
ip community-list 24 permit 200:3100               ! PEERS routes
!
route-map toUPSTREAM permit 10
  match community 23
  set community none
route-map toPEERS permit 10
  match community 23
  set community none
!
```



PEER and UPSTREAM
gets prefixes
according to community
matches

QUIZ: Provide traffic control methods

AS-PATH PREFENDS

String	Resulting AS-PATH to ASXXX
200:1001	Default
200:1002	200 200
200:1003	200 200 200
200:1009	Do not advertise to ASXXX

- ISP allows customers to use AS-path prepending to adjust route preference on the network.

BGP workshop scenario

End of Tutorial

Presented by:

Bruce Morgan Bruce.Morgan@aarnet.edu.au

Terutaka Komorizono teru@ntt.co.th

Written by:

Terutaka Komorizono teru@ntt.co.th

My test network :)

