

BGP 102: Scaling the Network

Avi Freedman

Akamai

Introduction

- BGP is relatively easy to get configured and basically announcing and using routes.
- It is difficult to scale to the tens-to-hundreds of routers scale with full iBGP mesh, AS-Path filters, and AS-Path padding as the only tools.
- We present Communities, Confederations, and local-pref use, and some other features, and show them used in context.

Topics (1)

- Review basic BGP concepts
- Simple BGP Scaling concepts
 - Inserting BGP Routes
 - Stable Routing and Scaling w/ Loopbacks
 - Save CPU and Typing w/ Peer-Groups

Topics (2)

- Scalable Advertisements with Communities
- Scalable Route-Selection with local-prefs
- iBGP Scaling Issues
- BGP Confederations
- BGP Scaling with Confederations

Topics (3)

- Supporting Multi-Homed Customers
- Backup Transit

- Sample Network - Topology
- Sample Network - Design Goals
- Sample Network - Implementation
- Review Router Configuration

BGP Concept Review

BGP Intro

- BGP4 is the protocol used on the Internet to exchange routing information between providers, and to propagate external routing information through networks.
- Each autonomous network is called an Autonomous System.
- ASs which inject routing information on their own behalf have ASNs.

BGP Peering

- BGP-speaking routers peer with each other over TCP sessions, and exchange routes through the peering sessions.
- Providers typically try to peer at multiple places. Either by peering with the same AS multiple times, or because some ASs are multi-homed, a typical network will have many candidate paths to a given prefix.

The BGP Route

- The BGP route is, conceptually, a “promise” to carry data to a section of IP space. The route is a “bag” of attributes.
- The section of IP space is called the “prefix” attribute of the route.
- As a BGP route travels from AS to AS, the ASN of each AS is stamped on it when it leaves that AS. Called the AS_PATH attribute, or “as-path” in Cisco-speak.

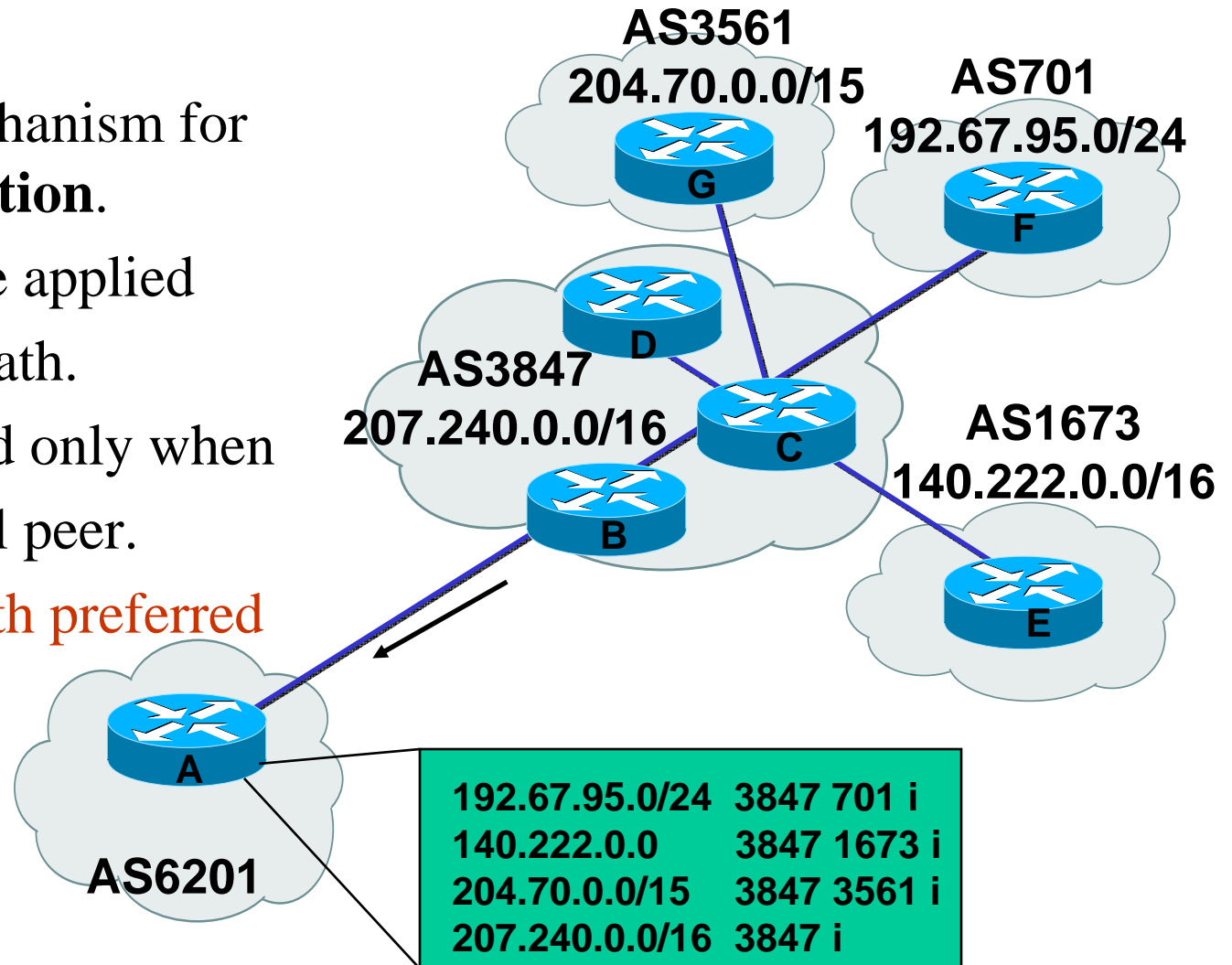
BGP Route Attributes

- In addition to the prefix, the as-path, and the next-hop, the BGP route has other attributes, affectionately known as “knobs and twiddles” -
 - weight, rarely used - “sledgehammer”
 - local-pref, sometimes used - “hammer”
 - origin code, rarely used
 - MED (“metric”) - a gentle nudge

AS Path

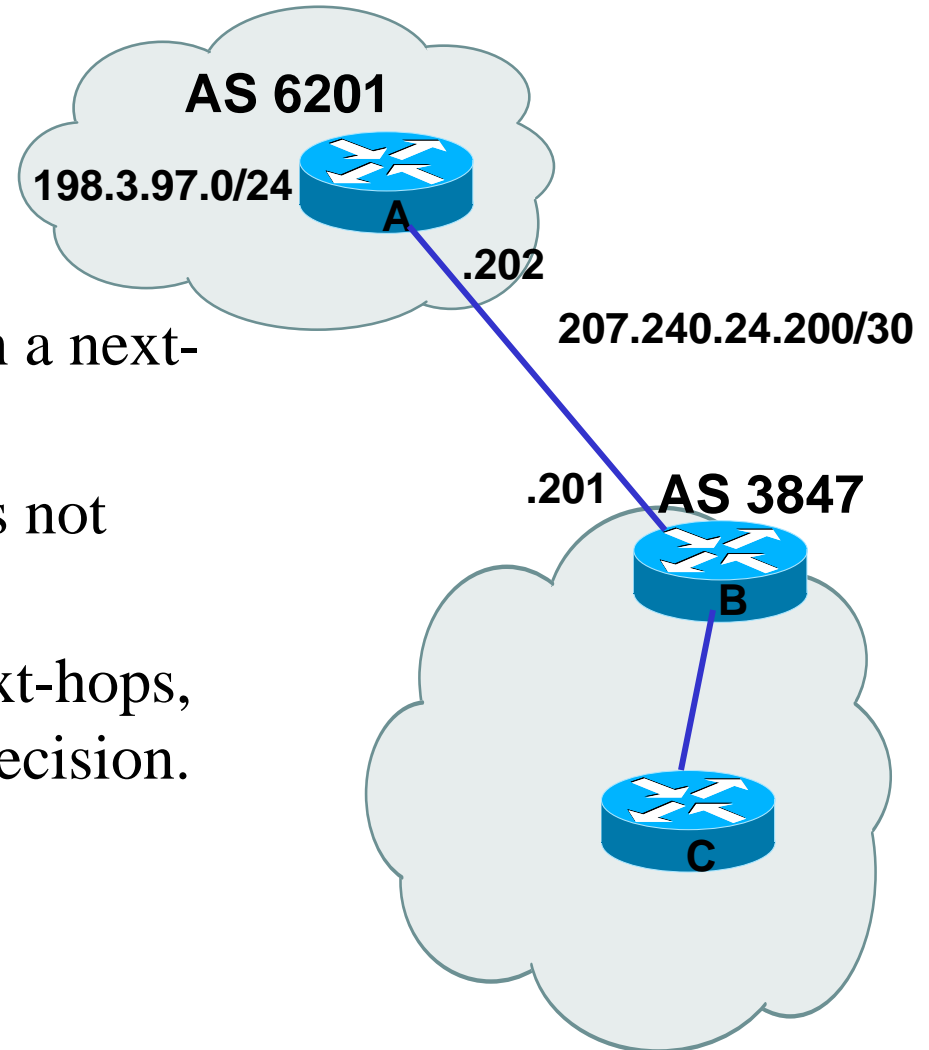
- Sequence of AS(s) a route has traversed.
- Provides a mechanism for **loop detection**.
- Policies may be applied based on AS path.
- Local AS added only when send to external peer.

*Shortest AS path preferred

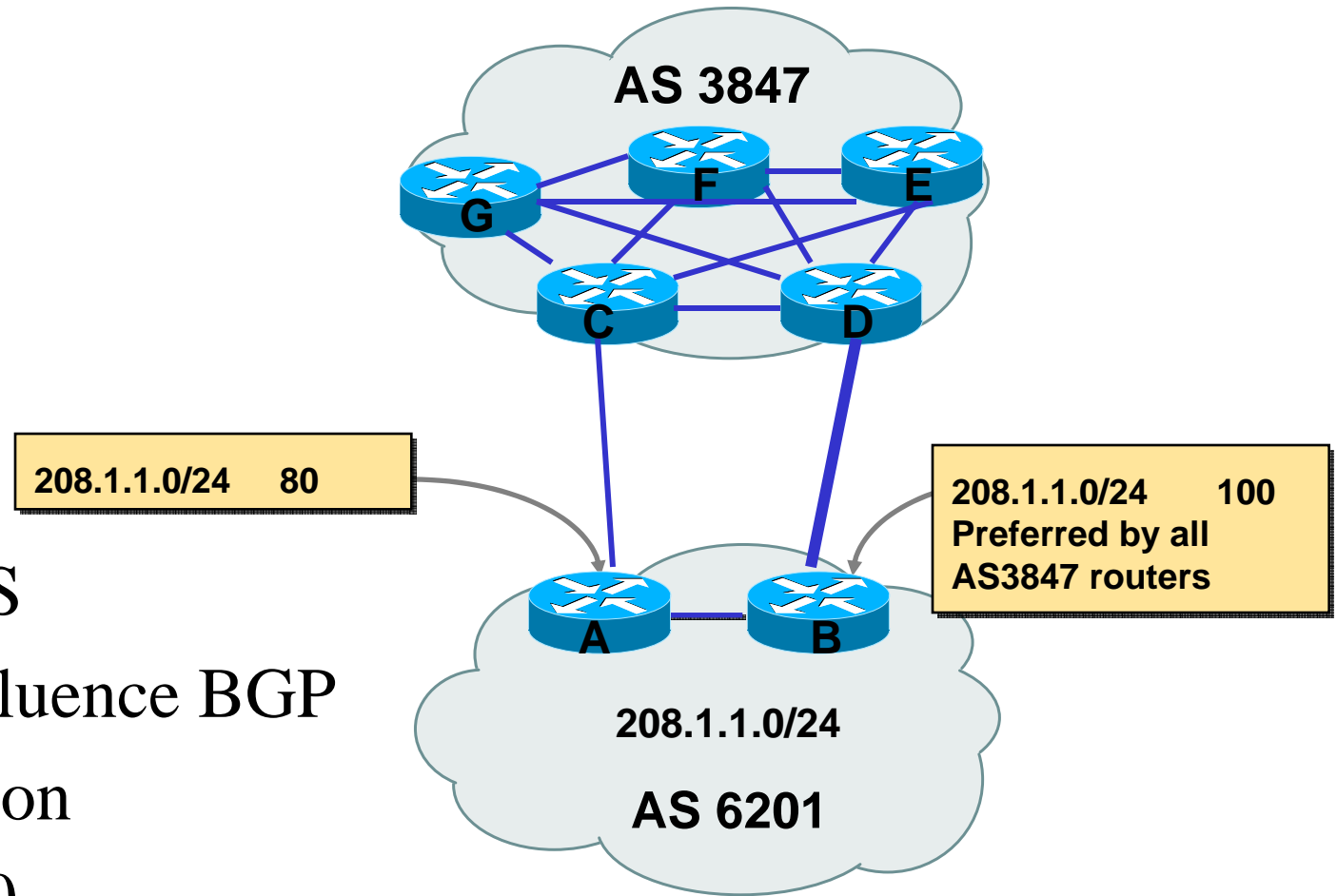


Next Hop

- Next-hop IP address to reach a network.
- Router A will advertise 198.3.97.0/24 to router B with a next-hop of 207.240.24.202.
- With IBGP, the next-hop does not change.
- IGP should carry route to next-hops, using intelligent forwarding decision.



Local Preference

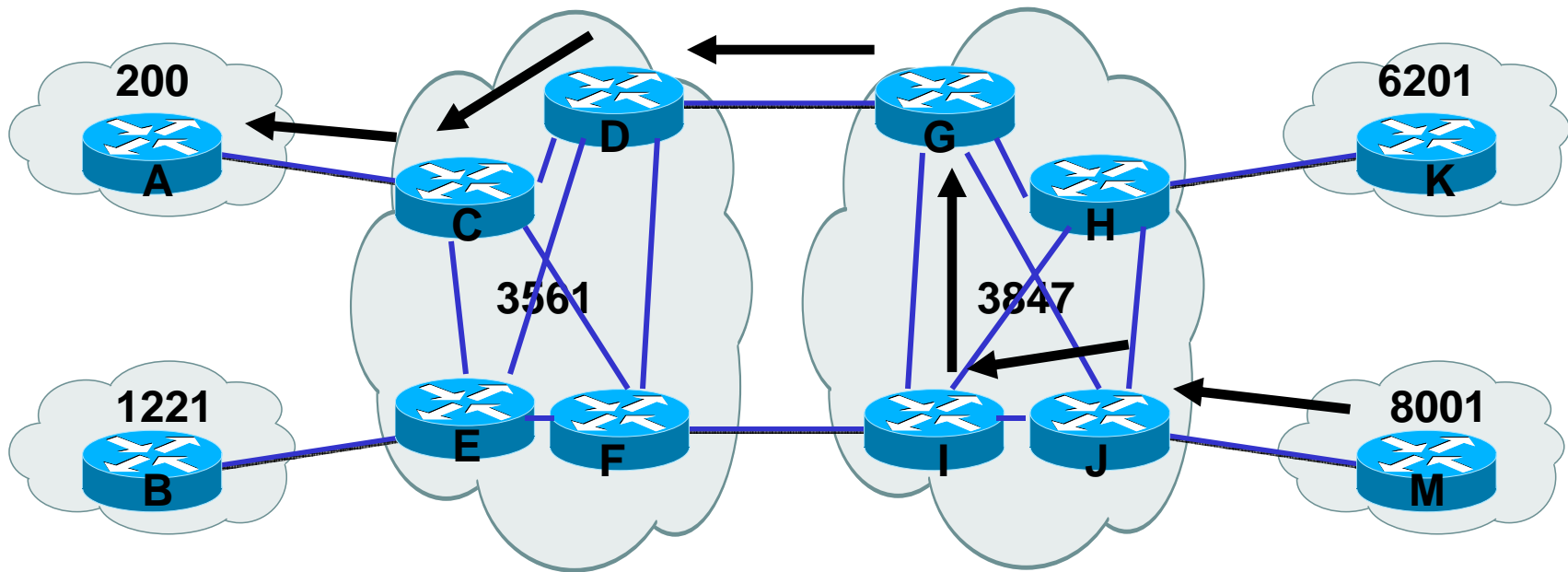


- Local to AS
- Used to influence BGP path selection
- Default 100
- * Highest local-pref preferred

Multi-Exit Discriminator (MED)

- Indication to external peers of the preferred path into an AS.
 - Affects routes with same AS path.
 - Advertised to external neighbors
 - Usually based on IGP metric
- * Lowest MED preferred

MEDs (cont.)

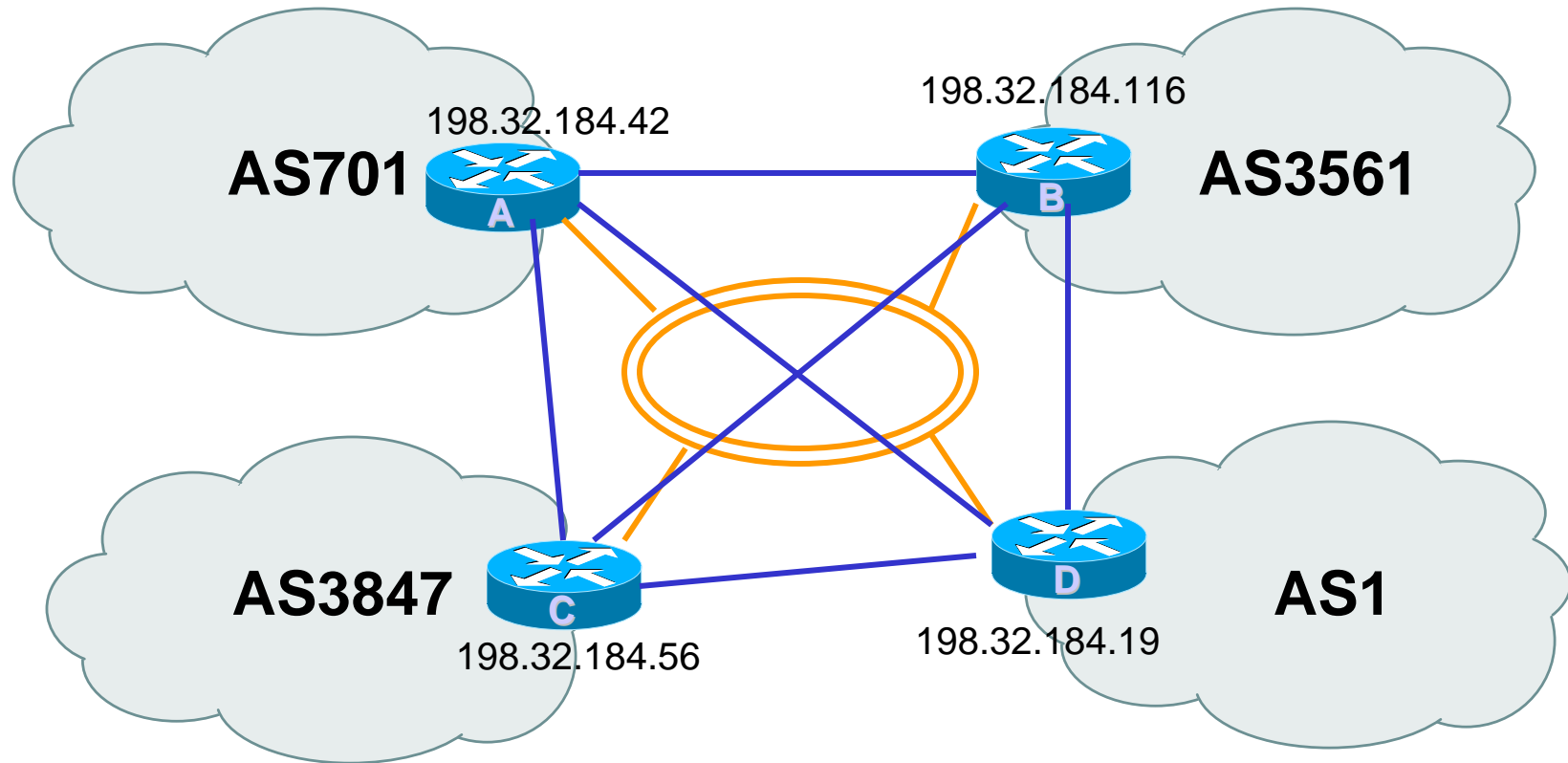


- Applies on a AS path basis
- Current aggregation schemes significantly lessen value.

Origin

- IGP (i)
 - Network statement under router BGP
- EGP (e)
 - Redistributed from EGP
- Incomplete (?)
 - Redistributed from IGP

Next Hop Self



BGP Policy

- BGP was designed to allow ASs to express a routing policy. This is done by filtering certain routes, based on prefix, as-path, or other attributes - or by adjusting some of the attributes to influence the best-route selection process.

BGP Best-Route Selection

- With all of the paths that a router may accumulate to a given prefix, how does the BGP router choose which is the “best” path?
- Through an RFC-specified (mostly) route selection algorithm.
- Watch out for weights (can cause routing loops) and be aware that local-prefs override what remote providers are trying to tell you with as-path padding.

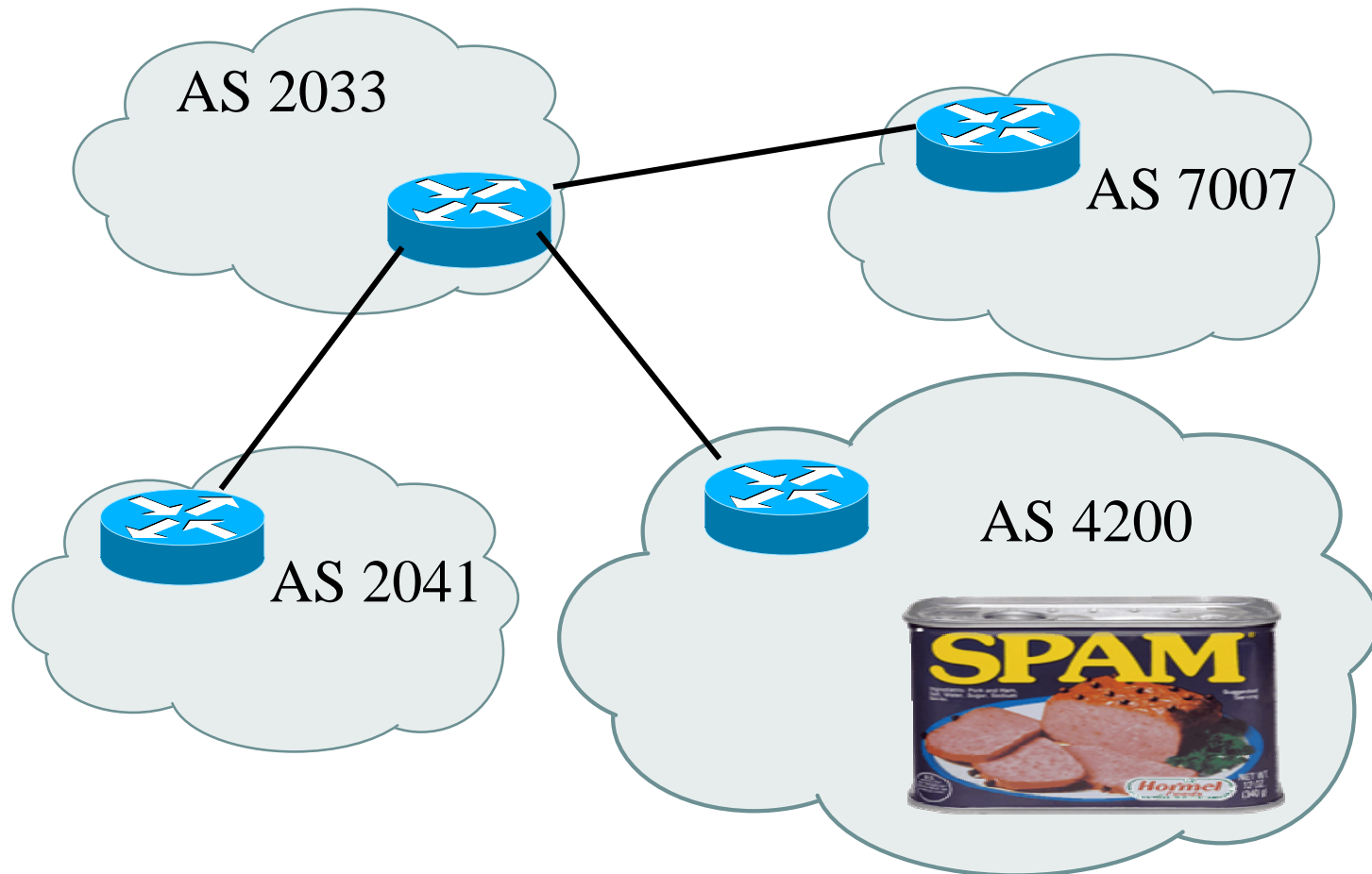
BGP Decision Algorithm

- Do not consider IBGP path if not synchronized
- Do not consider path if no route to next hop
- Do not consider IBGP path if not synchronized
- Highest weight (local to router)
- Do not consider path if no route to next hop
- Highest weight (prefer to other)
- Highest local preference (global within AS)
- Shortest AS path
- Shortest AS path
- Lowest origin code IGP < EGP < incomplete
- Lowest origin code IGP < EGP < incomplete
- Lowest MED
- Lowest MED
- Prefer EBGP path over IBGP path
- Prefer EBGP path over IBGP path
- Path with shortest next hop metric wins
- Lowest router id
- Lowest router id

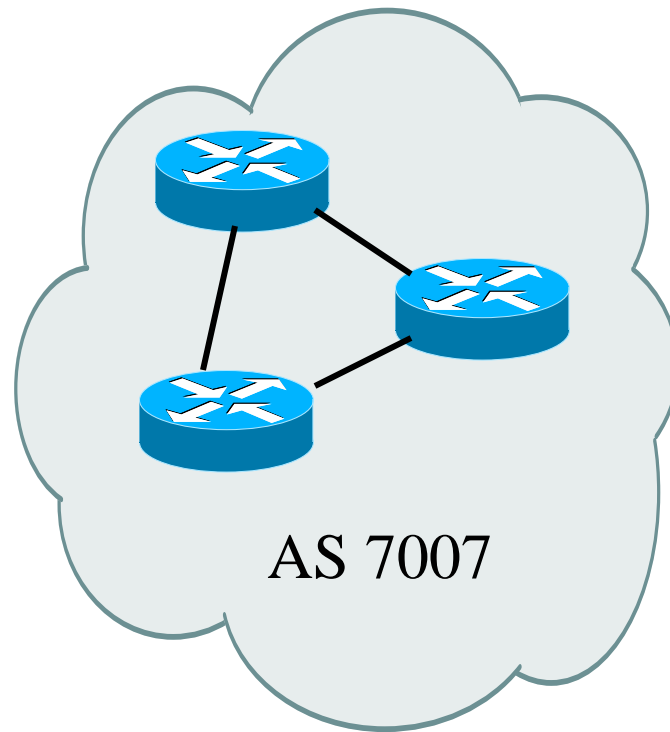
Communities

- Used to group destinations to which routing decisions can be applied.
- Each destination can belong to multiple communities.
- Usually applied with route-maps.

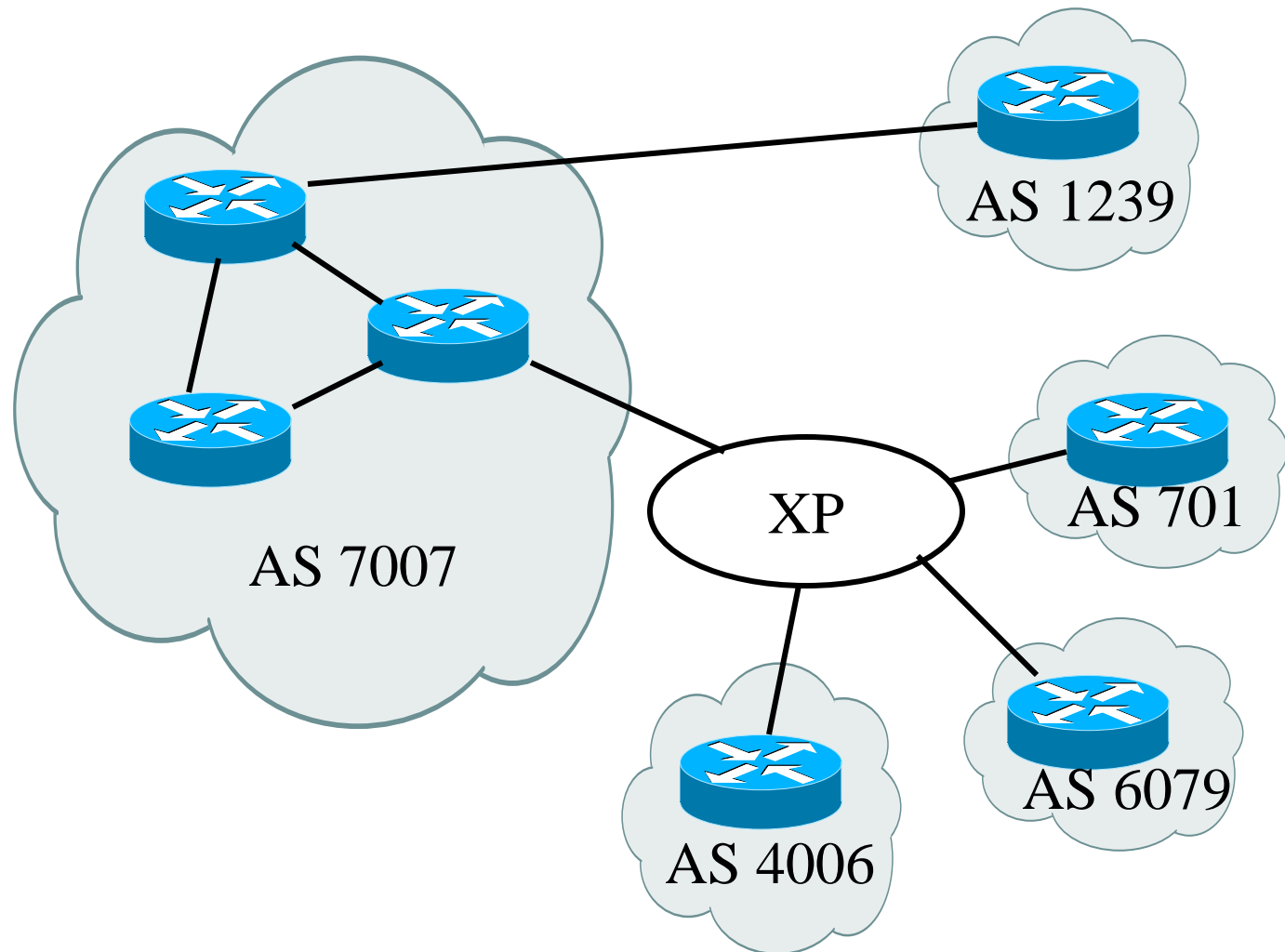
eBGP



iBGP



iBGP and eBGP



Determining Policy

- What do you want to do?
- The tricky part.
- Configuring is easy...

Typical Starting Point

- Use network statements to inject.
- Use AS-Path lists to control advertisement.
- Use AS-Path padding to prefer or de-prefer externally-heard paths.
- Have full iBGP mesh.

Inserting Routes into BGP

Route Insertion Methods

- network statement - most common
 - Sometimes thought of as “non-scalable”
- aggregate-address statement
 - difficult to punch holes
- redistributing through filters (usually with aggregate-address statements)
 - difficult to punch holes
 - dangerous as filters are altered

Using network statements

- Best to use network statements. Don't worry about not being fancy. Stick the network statement on the router the customer is on, or on multiple routers for LAN-attach customers.
- Easy to support customers who want to advertise more specifics with BGP.
- Also easy to apply per-route route-maps.

Stable Routing and Scaling with Loopbacks

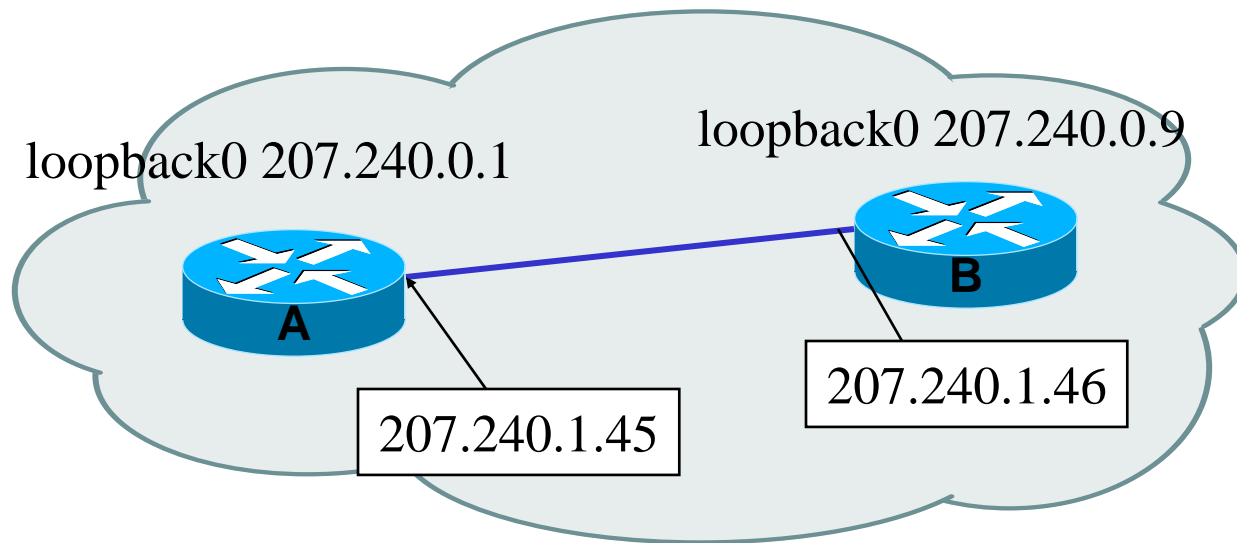
Stable BGP - Loobacks (1)

- Watch out for flapping routes.
- Sites think that if a site shows instability, it is worth blackholing for some time (30-90 minutes) until it stabilizes.
- Dampening hurts.
- So, nail non-multi-homed routes to loopback.

Stable BGP - Loopbacks (2)

- Also - peering between loopbacks enhances stability, since loopbacks don't go down.
- Also, good for load-balancing (balanced statics used underlying one peering session caused load-balancing for BGP-heard routes).
- Set up lo0, then
- “neigh x.y.z.q update-source looback0”

Update-Source Loopback0



Router A and router B peer with one another's loopback address. Normally, the source address of packets sent from router A to router B would be 207.240.1.45. If router B were to receive BGP packets from router A, the packets would be dropped because router B doesn't peer with 207.240.1.45. Because of this, "update-source loopback0" should be applied to the neighbor statements on both routers, thus telling the routers to set the source address to that of the specified interface for all BGP packets sent to that peer.

Scaling with Loopbacks

- Only have to remember loopback IP of each router.
- Easy to make sure you've "got" all routers for iBGP mesh.
- You know you have a configured loopback interface, with in-addr, to nail routes to.
- Good for logging and tac authentication - eliminates multiple serials showing up.

BGP Stability - soft-reconfig

- Instead of hammering a session to cause reevaluation (“clear ip bgp” drops the TCP session), “clear ip bgp soft” can be used.
- “clear ip bgp x.y.z.q soft out” is low cpu; it issues withdrawals for all currently-advertised routes and recomputes and re-sends routes.
- “clear ip bgp x.y.z.q soft in” can cause memory issues, as it needs to keep copy of all routes received.

Save CPU and Typing
with peer-groups

Peer Groups (1)

- Peer-groups were not designed to save typing, actually.
- By grouping neighbors with common policy together, routers can save lots of CPU by creating once a route object and then advertising that object to multiple peers.
- Also, saves typing :)

Peer Groups (2)

- Major restriction - next-hop is part of the object (one of the attributes), so a given peer-group can/should only be applied for peers on a common interface.
- So, useful for eBGP peers but sometimes not for iBGP peers.
- Still, can express different inbound policy per peer.

Sample peer-group

```
neighbor public-peer peer-group
neighbor public-peer next-hop-self
neighbor public-peer distribute 100 in
neighbor public-peer route-map public-in in
neighbor public-peer route-map public-out out
neighbor public-peer filter-list 30 in
```

Scalable Advertisements with Communities

AS-Path Filtering

- You can either announce routes by prefix or by as-path filtering. Updating a distributed prefix table is more difficult; as-path filtering (allowing routes from you or from customer ASs to be advertised), combined with aggressive inbound prefix-based filtering, is a good first approach.
- But...

Limitation of AS-filtering

- Either have to list all peers, or all customers. Gets really tricky when you peer with customers, or customers of peers, or peers of customers.
- These lists get difficult to read and distribute as you grow.
- So... Look at Communities to express policy.

BGP Communities - What

- Easier control of where routes go.
- Just a number (or numbers) that get stamped on BGP routes.
- ‘neigh x.y.z.q send-comm’ to send

```
ip comm 4 permit 4969:1200
route-map give-transit
    set comm 4969:1200 additive
route-map send-transit
    match community 4
```

BGP Communities - Why

- Give customers control of how you announce them
- Let customers see where you get routes
- Peering community; transit community; partial-transit community.
- Example – Philly-area ISP uses community 1601 to transit some PHL-area providers to each other; 1601 is the address of a PHL pop.

BGP Communities

- Well-known communities -
 - no-export - don't advertise to eBGP peers
 - no-advertise - don't advertise to any peer

Sample Communities

- 4969:12392 means “pad towards sprint 2 times”
- 49690:7010 means “don’t announce me to uunet”
- 4969:2 means “pad me twice”
- Some providers will make communities flexibly; some won’t. You can always ask for a community list/document as part of the (pre-)sales process.

Scaling with Local-Prefs

AS-Path Padding

- A 1st-cut approach to load-balancing or quality-balancing might be to de-prefer any routes heard via a provider you're seeing problems with. How?
- First approach is to add an extra copy of the next-hop AS to the AS-Path, so $^7007\$$ becomes $^7007\ 7007\$$. Longer AS-Paths are less preferred, all else being equal.
- You can implement complex policy with this, in fact.

Limitations of AS-padding

- A typical first way to select between multiple outbound paths is by padding the less-preferred paths as they come into your network.
- This works reasonably well, unless you have to redistribute these paths to others.
- Local-prefs make implementing this easier, though there is a caveat.

Local-Prefs

- The local-pref is a “powerful” BGP attribute - it comes before as-path length in the selection algorithm.
- Setting can override as-path length - consider the provider with a Gige and a 10Gig who WANTS you to pay attention to the 7-times-padded path...
- Come up with a unified scheme.
- CUSTOMER ROUTES ARE SACRED.

Typical local-pref Scheme

- 80 de-preferred routes
- 100 questionable xp routes
- <101-115> better xp routes
- <116-119> transit pipes
- 120 private xcon routes
- <121-139> better private routes
- 140+ customer routes

Implementing Local-pref

```
route-map public-in  
  set local 100  
  set comm 15000:8100 15000:666
```

```
route-map pni-in  
  set local 125  
  set comm 15000:609 15000:666
```

```
route-map set-transit  
  set local 140  
  set comm 15000:1200 add
```

Scaling iBGP with Confederations

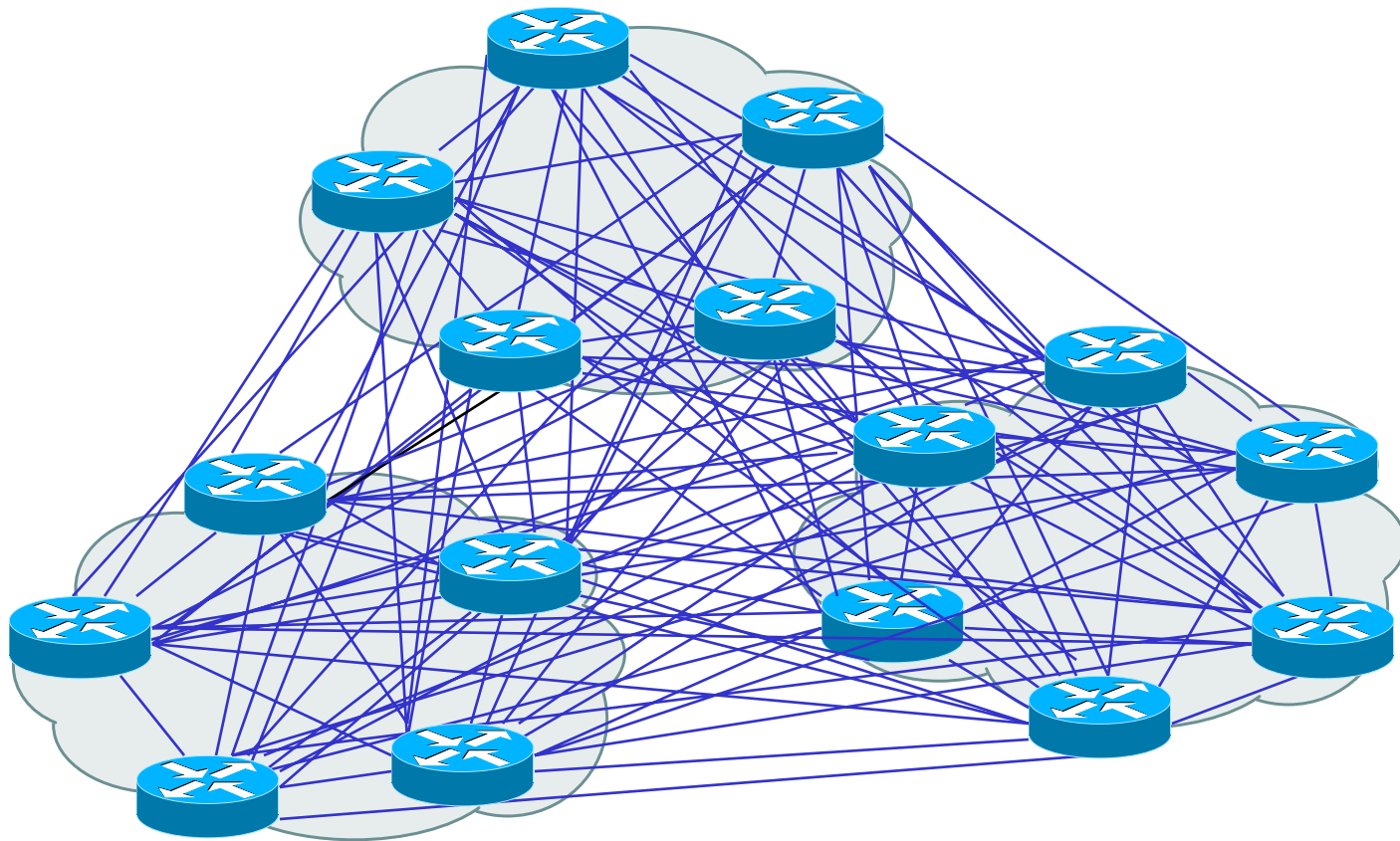
iBGP vs. eBGP Review

- iBGP and eBGP are the same protocol; just different rules.
- Rules are counter-intuitive -
 - eBGP advertises everything to everyone by default. OOPS - don't be MAE-Clueless.
 - iBGP does NOT advertise “3rd-party routes” to other iBGP peers. Why?
 - No way to do loop detection with iBGP, so this solves it.

iBGP Scaling Issues

- So you have to have ALL BGP-speaking routers in your AS peer with each other. Really.
- With 10 routers, an iBGP mesh is OK
- With 30 routes it is stretched
- With 100 it is taxed
- Eventually, CPU to deal with multiple sessions is nasty.

Logical View of full 16-router Mesh

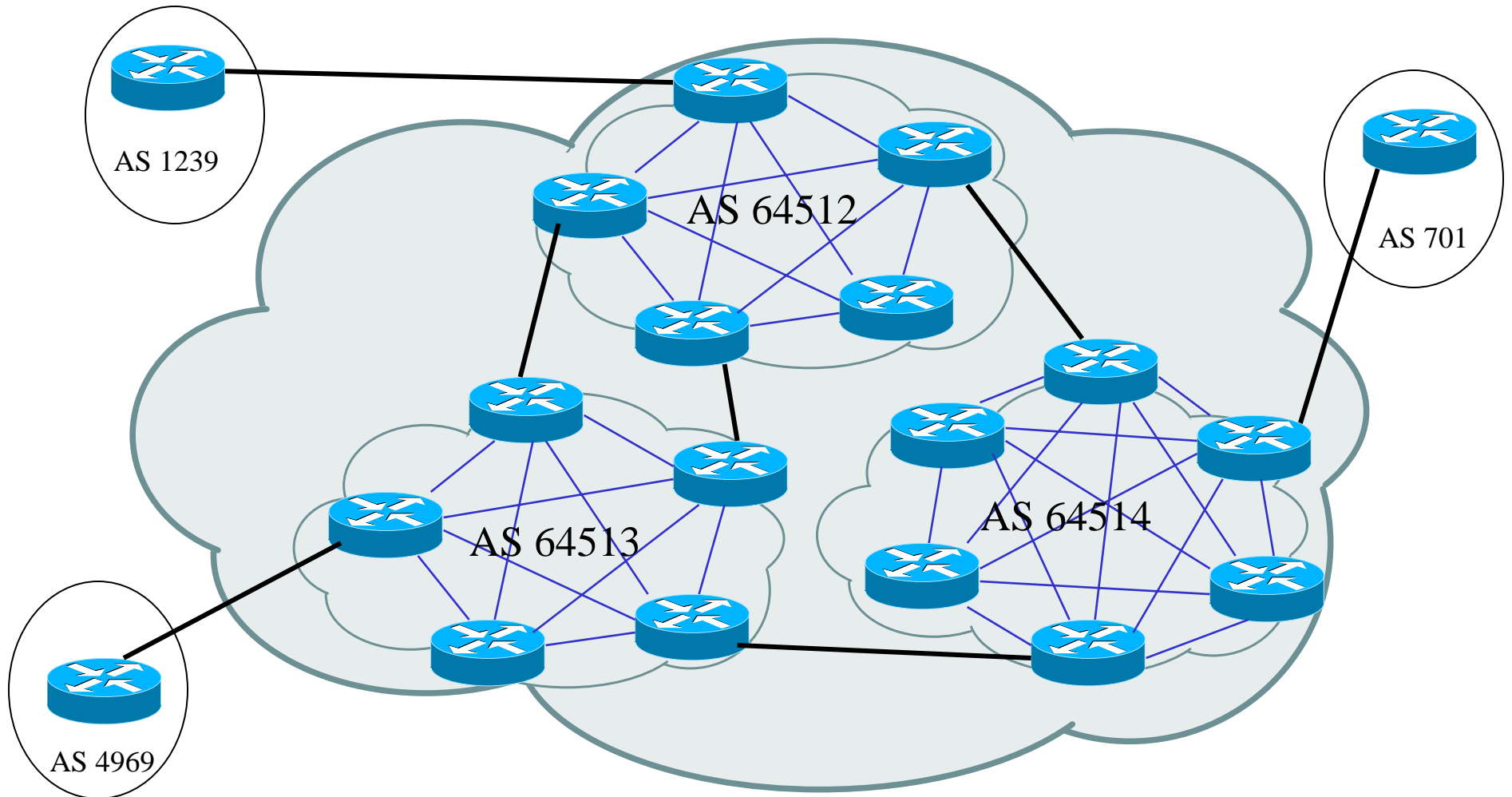


(kudos to danny ex-of-genuity-now-arbor)

Confederations (1)

- Makes iBGP more promiscuous
- How?
 - Fully-mesh all BGP speakers at a POP
 - Use fake ASNs at each POP
 - Between POPs, use eBGP rules (send everything)
 - Within POPs, use iBGP rules
 - Preserve local_prefs between POPs

Confederations, Illustrated



Confederations (2)

- Reduces CPU due to internal churn, but can increase CPU due to external churn in some cases.
- Trickier as-paths; use communities.
- Identified source of routes handily (just have to remember fake AS per POP, not one loopback for each router in a POP).
- Easier to apply MEDs.
- Makes iBGP more “hop-by-hop”.

Implementing Confederations

```
router bgp 64512
  bgp confederation identifier 15000
  bgp confederation peers 64512 64513 64514 64515
```

- note - put in extra confederation peers up-front
- as-path becomes (64512 64513) 7018 instead of 7018

AS-Path filters for confederations

- ^\$ Doesn't work any more...
- ^\$ matches internal routes in a given POP, but with confederations your routes will look like:
 - ^(64512 64513)\$ as well as ^\$
 - ip as acc 55 deny ^(\([0-9]*\))*\$

Supporting
Multi-Homed
Customers

Supporting Multi-Homed Custs

- What they need from you is routes to the ‘net, and some ability to be flexible in how you announce their routes.
- Routes to the ‘net - give them your communities (“neighbor x.y.z.q send-communities”). Publish your communities so they know what they mean. WARN if you change community semantics.

Supporting Multi-Homed Custs

- Be prepared to punch holes in your aggregates.
 - Using network statements, no problem.
 - Otherwise, be prepared to use suppress-maps with aggregate-address statements.
- Set up communities they can use to control which pipes you advertise them to, and what their routes look like.

Backup Transit

Mutual Backup Transit/Peering

- Make your network better AND help your competitor. Strange world we live in.
- Find a local competitor who has diverse connectivity and share the cost of a FastE/GigE. (Easy if you're both in a metro Ethernet cloud or at a local XP).
- Announce each other either:
 - Always, but padded (best, requires lots of coordination)
 - By request
 - Only if you can't hear them from the outside (communities-based and tricky)
- Local peering just for news often makes bandwidth-saving sense

Router Configs

Review - Basic Router Configuration

“How do I log config changes?”

- Run tacacs+ on most gear and it'll log all commands (including 'conf term' commands).
- You might want to look into rancid and other router-config tools.
- Once you start MacGuyver-ing things it's hard to go back

Cisco Regular Expressions

- . Period matches any single character, including white space.
- * Asterisk matches 0 or more sequences of the pattern.
- + Plus sign matches 1 or more sequences of the pattern.
- ? Question mark matches 0 or 1 occurrences of the pattern
- ^ Caret matches the beginning of the input string.
- \$ Dollar sign matches the end of the input string.
- _ Underscore matches a comma (,), left brace ({), right brace (}), left parenthesis, right parenthesis, the beginning or end of the input string, or a space.
- [] Brackets designate a range of single character patterns.
- Hyphen separates the endpoints of a range.

Basic Parameters (1)

hostname <some-hostname>

ip name <nameserver>

ip default-domain <yourdomain>

no service finger

<etc>

Basic Parameters (3)

ip bgp-community new-format

logging buffered

logging console informational

logging monitor informational

logging trap warnings

logging facility kern

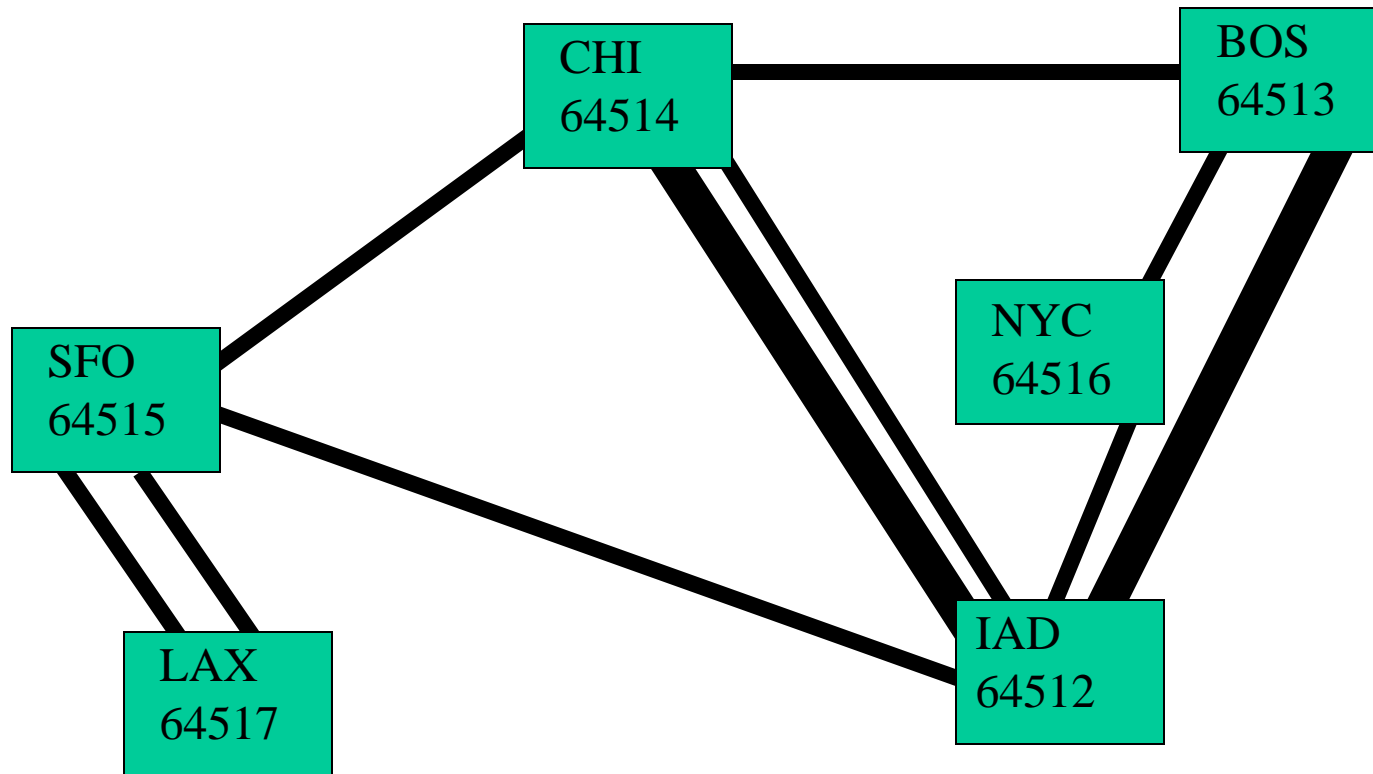
logging <logging ip server>

Basic Parameters (4)

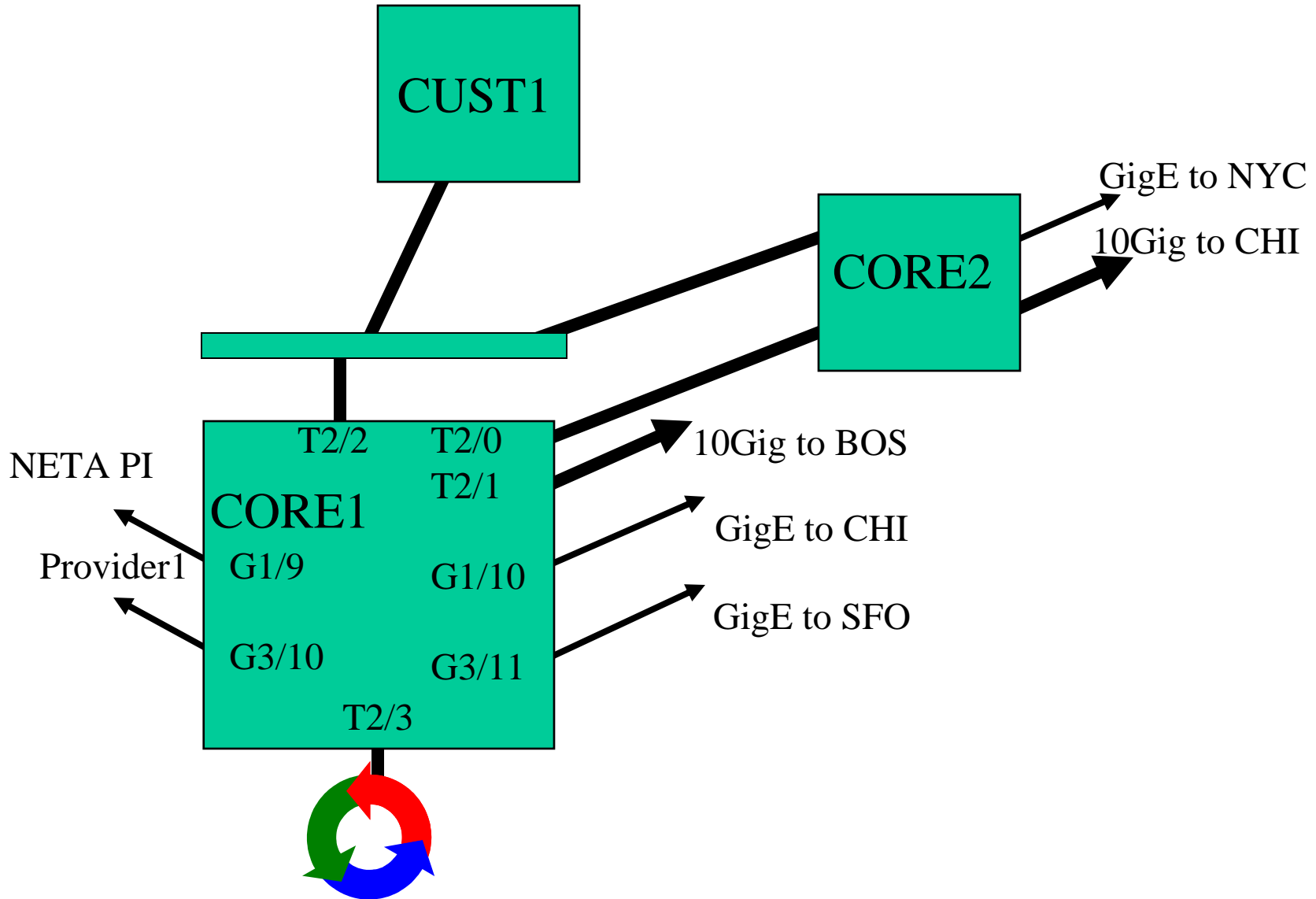
```
aaa new-model
aaa authentication login default tacacs+ local
aaa accounting commands 15 stop-only tacacs+
aaa accounting network start-stop tacacs+
aaa accounting connection start-stop tacacs+
aaa accounting system start-stop tacacs+
ip tacacs source-interface Loopback0
tacacs-server host 10.5.0.1
tacacs-server host 10.6.0.2
tacacs-server host 10.7.0.3
tacacs-server key smurfbded
```

Config for Sample Network

Sample Network



NoNameNet
Equinix Ashburn



Design Goals (1)

- Filter customer routes vigorously on inbound; assign (or let them assign) a transit community.
- Filter garbage (XP) routes inbound from everyone.
- No dampening.
- Allow customers to control how you advertise them.

Design Goals (2)

- Prefer customers, then private, then good public, then worse public, routes.
- Use confederations not because needed, but for scaling concerns.
- Use loopbacks for iBGP peering.

Interface Configs

```
interface Loopback0
  ip address 207.106.0.2 255.255.255.255
  ip route-cache flow
!
interface Fastethernet1/0/0
  description core1-core2 private
  ip add 207.106.2.89 255.255.255.252
  no ip directed-broadcast
  ip route-cache flow
!
interface Fastethernet2/0/0
  description POP Backbone
  ip address 207.106.4.1 255.255.255.224
  no ip directed-broadcast
  ip route-cache flow
!
interface Fddi3/0/0
  description MAE-East FDDI
  ip address 192.41.177.4 255.255.255.0
  no ip directed-broadcast
  ip route-cache flow
```

```
interface Posip9/0/0
  description OC3 to NYC
  ip address 207.106.2.5 255.255.255.252
  ip route-cache flow
!
interface Seral4/0/0
  description T3 to CHI
  ip address 207.106.2.9 255.255.255.252
  ip route-cache flow
!
Interface Serial4/0/1
  description T3 to SFO
  ip address 207.106.2.13 255.255.255.252
  ip route-cache flow
!
interface Serial5/0/0
  description PI to NetworkA
  ip address 10.50.1.2 255.255.255.252
  ip route-cache flow
!
interface Serial5/0/1
  description T3 to netaxs
  ip address 207.106.127.6 255.255.255.252
  ip route-cache flow
```

OSPF Configuration

```
router ospf 22
 redistribute connected subnets
 redistribute static subnets
 passive-interface Fastether2/0
 passive-interface Serial5/0/0
 passive-interface Serial5/0/1
 network 207.106.4.0 0.0.0.31 area 207.106.4.0
 network 207.106.2.0 0.0.0.255 area 0
 area 0 authentication
 area 207.106.4.0 authentication
! Plus appropriate costs on different-size links
```


BGP Config

```
ip as acc 1 permit .*
ip as acc 2 deny .*

router bgp 64512
  no synchronization
  bgp always-compare-med
  no bgp dampening
  confederation identifier 15000
  confederation peers 64512 64513 64514 64515
    64516 64517 64518 64519
  network 207.106.60.0 mask 255.255.255.0 route-
    map set-local-community

route-map set-local-community
  set comm 15000:123
```

Public Peers (1)

```
router bgp 64512
  neighbor public-peer peer-group
  neighbor public-peer next-hop-self
  neighbor public-peer soft-reconfig in
  neighbor public-peer version 4
  neighbor public-peer send-community
  neighbor public-peer distribute-list 110 in
  neighbor public-peer route-map public-in in
  neighbor public-peer route-map send-transit out
  neighbor public-peer filter-list 4 in
```

Public Peers (2)

```
access-list 110 deny ip host 0.0.0.0 any
access-list 110 deny ip 192.41.177.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 192.157.69.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 198.32.128.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 198.32.130.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 198.32.136.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 198.32.146.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 198.32.146.0 0.0.1.255 255.255.254.0 0.0.1.255
access-list 110 deny ip 198.32.176.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 198.32.180.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 198.32.184.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 198.32.186.0 0.0.0.255 255.255.255.0 0.0.0.255
access-list 110 deny ip 127.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
access-list 110 deny ip 10.0.0.0 0.255.255.255 255.0.0.0 0.255.255.255
access-list 110 deny ip 172.16.0.0 0.15.255.255 255.240.0.0 0.15.255.255
access-list 110 deny ip 192.168.0.0 0.0.255.255 255.255.0.0 0.0.255.255
access-list 110 permit ip any any
```

Public Peers (3)

```
route-map public-in permit 10
  set community 15000:666 15000:8100
  set local 100

ip community-list 1 permit 15000:123
ip community-list 1 permit 15000:1200

route-map send-transit
  match community 1
```

Public Peers (4)

! Obviously, don't apply this to UU, Sprint,
! CW, ATT, BBN, etc...

```
ip as-path access-list 4 deny _701_  
ip as-path access-list 4 deny _1239_  
ip as-path access-list 4 deny _3561_  
ip as-path access-list 4 deny _7018_  
ip as-path access-list 4 deny _1_  
<etc>  
ip as-path access-list 4 permit .*
```

Private Peers (1)

```
router bgp 64512
  neighbor <peerip> next-hop-self
  neighbor <peerip> soft-reconfig in
  neighbor <peerip> version 4
  neighbor <peerip> send-community
  neighbor <peerip> distribute-list 110 in
  neighbor <peerip> route-map private-in in
  neighbor <peerip> route-map send-transit out
  neighbor <peerip> filter-list 4 in
! Sometimes insert route-map to do fixer-meds
```

Private Peers (2)

```
route-map public-in permit 10
  set community 15000:666 15000:8100
  set local 120
```

Customer Peer (1)

```
router bgp 64512
  neighbor <custip> next-hop-self
  neighbor <custip> soft-reconfig in
  neighbor <custip> version 4
  neighbor <custip> send-community
  neighbor <custip> distribute-list NNN in
  neighbor <custip> route-map set-transit in
  neighbor <custip> route-map send-transit out
```

! Distribute list is PER-CUSTOMER!!!

Customer Peer (2)

```
route-map set-transit
  set local-pref 140
  set community 15000:8100 15000:1200 additive
```

! Or, for customers who want flexibility
! Let them set themselves for transit

```
route-map allow-transit
  set local-pref 140
  set community 15000:8100 additive
```

!also, have communities for changing local-pref

Internal - Same or Diff Confed

```
router bgp 64512
  neighbor <custip> next-hop-self
  neighbor <custip> update-source Loopback0
  neighbor <custip> send-community
```

! Main thing is to set med on per-neighbor basis.

! No need for soft-reconfig in; can always clear
! it outbound from the other end.

To Sprintlink

```
ip community 25 permit 15000:12390
ip community 26 permit 15000:12392
ip community 27 permit 15000:12391
ip community 28 permit 15000:1239
ip community 28 permit 15000:1200
ip community 28 permit 15000:123
route-map 2sprint deny 10
  match comm 25
route-map 2sprint permit 20
  match comm 26
  set as pre 15000 15000
route-map 2sprint permit 30
  match comm 27
  set as pre 15000
route-map 2sprint permit 40
  match comm 28
```

{Backup} Transit

```
route-map backup-out permit 10  
  match community 1  
  set as pre 15000 15000 15000 15000 15000 15000
```

```
route-map send-transit permit 10  
  match community 1
```

```
route-map allow-transit  
  set local-pref 140  
  set community 15000:8100 additive
```

BGP Clause

```
router bgp 64512
no synchronization
  bgp always-compare-med
  no bgp dampening
  confederation identifier 15000
  confederation peers 64512 64513 64514 64515
    64516 64517 64518 64519
  network 207.106.60.0 mask 255.255.255.0
    route-map set-local-community
!
  neigh public-peer peer-group
  neigh public-peer next-hop-self
  neigh public-peer soft-reconfig in
  neigh public-peer version 4
  neigh public-peer send-community
  neigh public-peer distribute-list 110 in
  neigh public-peer route-map public-in in
  neigh public-peer route-map send-transit out
  neigh public-peer filter-list 4 in
!
  neigh 207.106. remote-as 64512
  neigh 207.106.0.3 descr IAD-aggregator1
  neigh 207.106.0.3 update-source lo0
  neigh 207.106.0.3 send-community

! 207.106.0.4 is preferred via f1/0/0
  neigh 207.106.0.4 remote-as 64512
  neigh 207.106.0.4 descr IAD-core2
  neigh 207.106.0.4 update-source lo0
  neigh 207.106.0.4 send-community
!
  neigh 207.106.0.8 remote-as 64513
  neigh 207.106.0.8 descr OC3 to BOS
  neigh 207.106.0.8 update-source lo0
  neigh 207.106.0.8 send-community
  neigh 207.106.0.8 route-map medplus1000 out
!
  neigh 207.106.0.11 remote-as 64514
  neigh 207.106.0.11 descr DS3 to CHI
  neigh 207.106.0.11 update-source lo0
  neigh 207.106.0.11 send-community
  neigh 207.106.0.11 route-map medplus2000 out
!
  neigh 207.106.0.14 remote-as 64515
  neigh 207.106.0.14 descr DS3 to SFO
  neigh 207.106.0.14 update-source lo0
  neigh 207.106.0.14 send-community
  neigh 207.106.0.14 route-map medplus6500 out
```

BGP Clause

```
neigh 10.5.1.1 remote-as 16040
neigh 10.5.1.1 descr private to NetA
neigh 10.5.1.1 next-hop-self
neigh 10.5.1.1 soft-reconfig in
neigh 10.5.1.1 version 4
neigh 10.5.1.1 send-community
neigh 10.5.1.1 distribute-list 110 in
neigh 10.5.1.1 route-map allow-transit in
neigh 10.5.1.1 route-map backup-out out
neigh 10.5.1.1 filter-list 4 in
!
neigh 207.106.2.5 remote-as 4969
neigh 207.106.2.5 descr t3 transit to netaxs
neigh 207.106.2.5 next-hop-self
neigh 207.106.2.5 soft-reconfig in
neigh 207.106.2.5 version 4
neigh 207.106.2.5 send-community
neigh 207.106.2.5 distribute-list 110 in
neigh 207.106.2.5 route-map send-transit out
```

```
neigh 192.41.177.241 remote-as 1239
neigh 192.41.177.241 next-hop-self
neigh 192.41.177.241 soft-reconfig in
neigh 192.41.177.241 distribute-list 110 in
neigh 192.41.177.241 route-map public-in in
neigh 192.41.177.241 route-map 2sprint out
!
neigh 192.41.177.A remote-as BBBB
neigh 192.41.177.A descr NetB
neigh 192.41.177.A peer-group public peer
!
neigh 192.41.177.C remote-as DDDD
neigh 192.41.177.C descr NetD
neigh 192.41.177.C peer-group public peer
!
neigh 192.41.177.E remote-as FFFF
neigh 192.41.177.E descr NetF
neigh 192.41.177.E peer-group public peer
!
neigh 192.41.177.G remote-as HHHH
neigh 192.41.177.G descr NetH
neigh 192.41.177.G peer-group public peer
! and so on...
! and so on...
! and so on...
```