

ARIN

BGP Tutorial

Avi Freedman

avi@freedman.net

<http://avi.freedman.net>

Index

- BGP Humor
- Internet Connectivity Overview
- Multihoming Concepts
- Multihoming Without BGP
- Multihoming - Address Space Complications
- Then the BGP Stuff

BGP Humor

BGP Movie Titles

I Still Know What You Announced Last Session
Crouching Announcements, Hidden Withdrawals
RIP Strikes Back
Fatal Announcements
O Prefix Where Art Thou
"Shall We Announce"
The Death of A Prefix
4 Announcements & A Withdrawal
Much Ado About Flapping
A Few Good Announcements
Grumpy Old BGP
The Dead Prefix Society
7007: A BGP Odyssey

BGP Movie Titles

Return of BGP

RIP Strikes Back

Revenge of the BGP

Scary BGP; Chasing BGP

The Wrath of BGP

Fatal Announcements

Fried Green BGP

Silent Route Strikes Back

Being BGP

"Shall We Announce"

Good Flap Hunting

A Route's Life

O Prefix Where Art Thou

The Death of a Prefix

The Unreachables

BGP Inc.; BGP Wars

Crouching Announcement,
Hidden Withdrawals

A Few Good Announcements

Grumpy old BGP

The Dead Prefix Society

4 Announcements & A
Withdrawal

7007: A BGP Oddysey

Much Ado About Flapping

Sense & Reachability

BGPless in Seattle

While You Were Announcing

I Know What You Announced
Last Session

The BGP Song

Yesterday

All the withdrawals seemed so far away
I thought my prefixes were here to stay
Oh, I believe in Yesterday.

Suddenly

It's not half the table it used to be
There's a black hole hanging over me
Oh, I believe in Yesterday.

Why they had to flap, announce and draw
away?

They sent something bad, now I long for
yesterday.

Yesterday

Routing was such an easy game to play
Now my packets all hide away
Oh, I believe in Yesterday

Index

- Basic BGP - The BGP Route
- Basic BGP - Inserting Routes into BGP
- Basic BGP - Advertising Routes
- Basic BGP - Other BGP Route Attributes
- Basic BGP - Selecting Routes
- Tuning Traffic Flow
- Research Problems

Internet Connectivity Overview

Having Internet Connectivity

- To have complete Internet connectivity you must be able to reach all destinations on the net.
- Your packets have to get delivered to every destination. This is easy (default routes).
- Packets from everywhere else have to “find you”. This is done by having your ISP(s) advertise routes for you.

Multihoming Without BGP

Multihoming Without BGP

- To get Internet connectivity, you can just default route your traffic to your upstream providers.
- To get traffic back **from** the Internet, you need to have your providers tell all of the rest of the Internet “where you are”.

BGP Route Advertisement (1)

- Think of a BGP route as a “promise”.
- If I advertise 207.8.128.0/17, I promise that if you deliver traffic to me for anywhere in 207.8.128.0/17, I know how to deliver it at least as well as anyone else.
- If my customer has 207.8.140.0/24, I generally will not announce that route separately since it is covered by my 207.8.128.0/17 aggregate route.

BGP Route Advertisement (2)

- By making sure these routes, or “promises”, are heard by ALL providers on the ‘net, your provider ensures a return path for all of your packets.
- Remember, sending packets OUT is easier than getting them back.
- Also, remember - sending routes OUT causes IP traffic to come IN.

BGP Route Advertisement (3)

- But the most specific route wins, so if one of my customers' ISPs is advertising 207.8.240.0/24, all incoming traffic from other networks will start flowing in that pipe.
- So I must “punch a hole” in my aggregate announcement and advertise 207.8.128.0/17 and 207.8.240.0/24.

BGP Route Advertisement (4)

- The complete set of routes advertised by all BGP speakers on the net is about 55,000 routes as of 10/98.
- If your route is missing in the “view” of any major provider, you will not have connectivity to them.

Multihoming Without BGP - How it Works

Customer Side - Outbound

- All you need to do is to put in static default route(s). To prefer two upstreams equally:
 - ip route 0.0.0.0 0.0.0.0 s4/0
 - ip route 0.0.0.0 0.0.0.0 s4/1
- To use one link as a backup only for outbound packets:
 - ip route 0.0.0.0 0.0.0.0 s4/0
 - ip route 0.0.0.0 0.0.0.0 s4/1 10
 - why? S4/1 could be a 56k or backup link

Cisco Load Balancing

- The way Ciscos (except for big new ones running “CEF” [aka the “Customer Enragement Feature”]) work if there are two “equal-cost” routes to the same place is -
 - Option 1 - Round-robin the packets without “route caching”. This goes through the slowest sections of the router’s OS. Bad. Also, if you are connected to different ISPs, packets can arrive out of order, etc...
 - Option 2 - Use route caching (default). Traffic to the same dest IP will always use the same interface, until the cache entry expires.

Customer Side - Inbound

- Just tell your ISP what address space you are bringing, if any.
- Your ISP may allocate you space out of their larger address blocks.
- If so, they need to announce your space “more specifically”.
- But you do no work other than tell your ISP what to do.

Provider Side (1)

- If both providers don't advertise your routes with the same specificity, you might have -
 - netaxs saying “4969 sez 207.8.128.0/17”
 - uunet saying “701 sez 207.8.195.0/24”
- Bad, because almost all traffic on the ‘net will come into you via UUNET.
- Why?
- {note} - talk about address filters

Provider Side (2)

- What you need is -
 - netaxs saying “4969 sez 207.8.128.0/17”
 - netaxs saying “4969 sez 207.8.195.0/24”
 - uunet saying “701 sez 207.8.195.0/24”
- Good, because -
 - 1) Because the two 207.8.195.0/24 routes are of the same specificity, providers CAN choose btwn netaxs and uunet to get to you; and
 - 2) For some people who don’t listen to /24s and such in new address space, they still have the 207.8.128.0/17 route to use to get to you.

Address Space Complications

- So, in the case of -
 - netaxs saying “4969 sez 207.8.128.0/17”
 - netaxs saying “4969 sez 207.8.195.0/24”
 - uunet saying “701 sez 207.8.195.0/24”
- “Some people won’t listen to the /24, so what happens if my netaxs connection goes down?”
- Not a problem!!! Because netaxs will hear the UUNET /24. Sprint send traffic to netaxs; netaxs to uunet; and uunet to you.

Disadvantages of not using BGP

- You gain a bit more control of your destiny when you speak BGP yourself. You can break up your routes in an emergency, or to tune traffic. You can “pad” your announcements to de-prefer one or more upstreams.
- Also, you lose the ability to fine-tune outbound traffic flow to the “best” upstream.

Why BGP?

- BGP is a multi-vendor “open” protocol with multiple implementations, all mostly interoperable. It is the only actively used EGP on the Internet.
- The main design feature of BGP was to allow ISPs to richly express their routing policy, both in selecting outbound paths and in announcing internal routes. Keep this in mind as we progress.

What is BGP?

BGP is ... (1)

- An Exterior Gateway Protocol (EGP), used to propagate tens or hundreds of thousands of routes between networks (ASs).
- The only protocol used to do this on the Internet today.

BGP is ... (2)

- The Border Gateway Protocol, currently Version 4 - defined in RFC 1771, and extended (with additional optional attributes) in other RFCs.
- A “distance-vector” routing protocol, running over TCP port 179.
- Supports modern “classless” routing. BGP3, RIPv1, and some others do NOT.

Purpose of BGP

Purpose of BGP

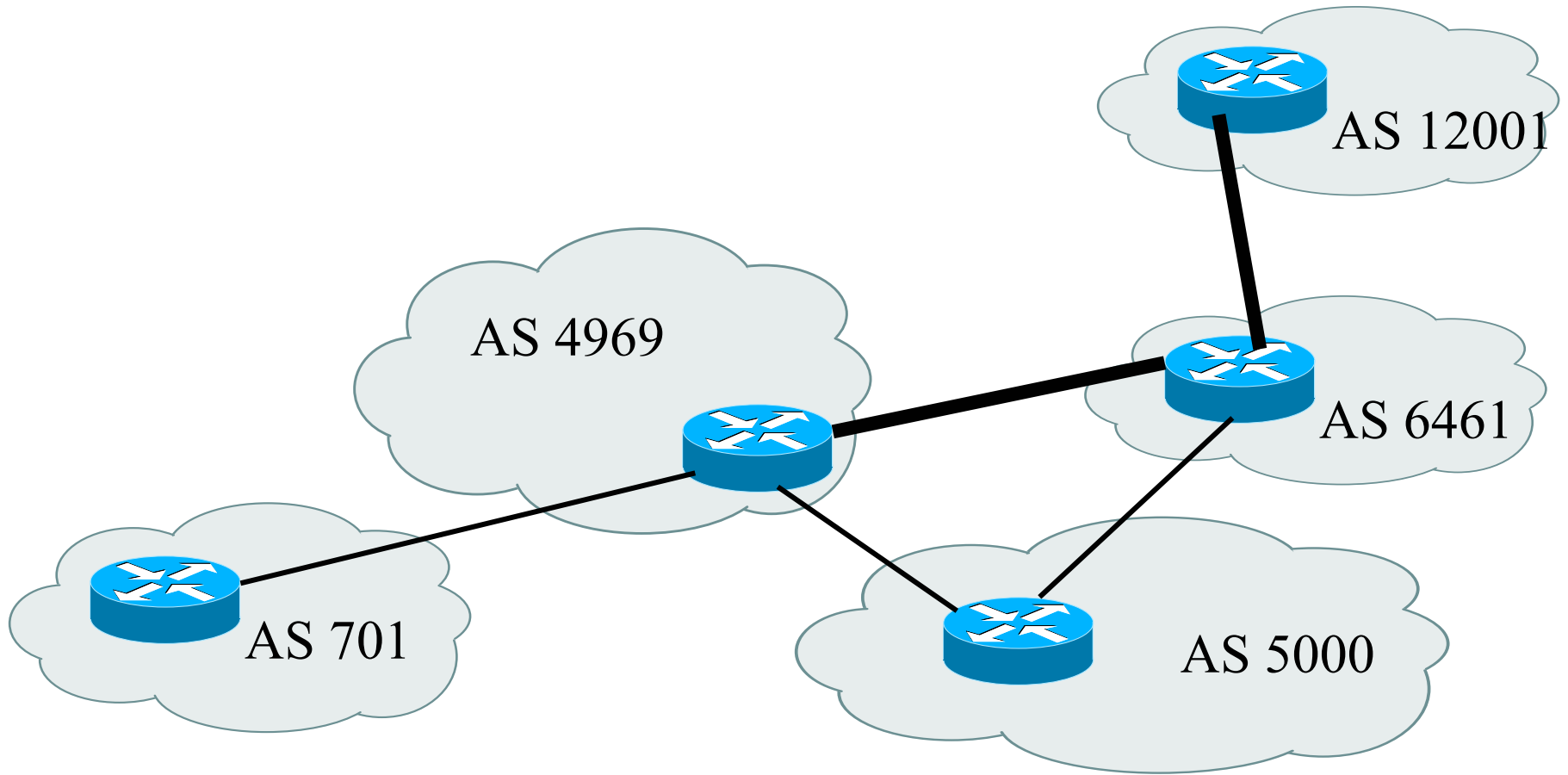
- To allow networks to tell other networks about routes (parts of the IP address space) that they are “responsible” for.
- Using “route advertisements”, or “promises” - also called “NLRI” or “network-layer reachability information”.
- Networks are “Autonomous Systems”.
- Identified in BGP by a number, called the ASN (“Autonomous System Number”)

**Basic
BGP
Concepts**

Basic BGP Concepts (1)

- BGP exchanges routes between ASs.
- When routes are exchanged, ASNs are stamped on the routes *on the way out* - adding one “AS hop” per network traversed. (0-65535)
- No concept of pipe size, internal router hop-count, congestion - in some sense BGP treats all ASs the same.
- ASs allow administrative debugging, “policy” routing, and *loop detection*.

BGP AND ASNs



Basic BGP Concepts (2)

- Routes are exchanged over “peering sessions”, which run on top of TCP.
- Keepalives are used to avoid needed to re-send the whole table periodically.
- The routes are “objects”, or “bags” of “attributes” - really mini-databases.
- BGP is actually two protocols - iBGP, designed for internal routing, and eBGP, designed for external routing.

Basic BGP Concepts (3)

- There is only one “best” BGP route for any given IP block at one time.
- This “best” BGP route is not always the route that gets “installed” into the router’s RIB/FIB.
- Once a session comes up, all best-routes are exchanged. Then over time, just “topology updates” are exchanged.
- You can ONLY exchange “best” routes.

Basic BGP Concepts (4)

- Policy
 - The Internet was a strange place before the modern commercial Internet evolved in 1992-1993.
 - Some networks had policies about what kind of traffic they would carry.
 - BGP was designed to allow network operators to make routing decisions based on whatever “policy” they wanted (or HAD) to use.

Basic Router/Protocol Arch.

- (Somewhat Cisco-specific)
- BGP, OSPF, etc routing tables exist “above”
- The IP routing table (the RIB), which exists “above” the
- Forwarding table FIB (sometimes distributed)
- Each time you go “down” a level you lose info (i.e. first lose as-path/BGP attribs, then at the bottom it’s basically just a MAC address/interface and prefix, +/- load-balancing goo)

**Basic BGP Concepts -
The BGP Route
and
Route Attributes**

The BGP Route

- A BGP “route” is a “bag” of objects, or “attributes”.
- The “prefix” is the section of address space being advertised. A prefix consists of:
 - A starting point (i.e. 207.8.128.0)
 - A netmask (i.e. /24, aka 255.255.255.0)

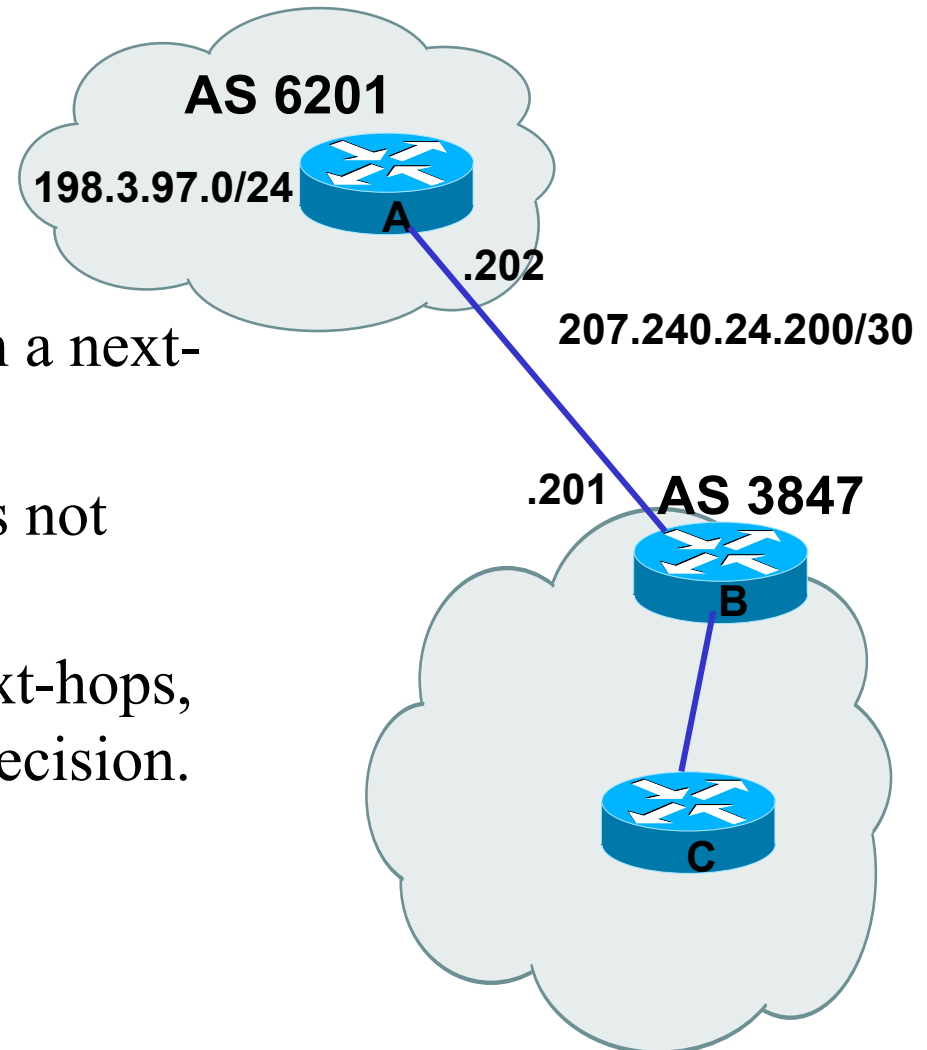
What Is an Attribute?



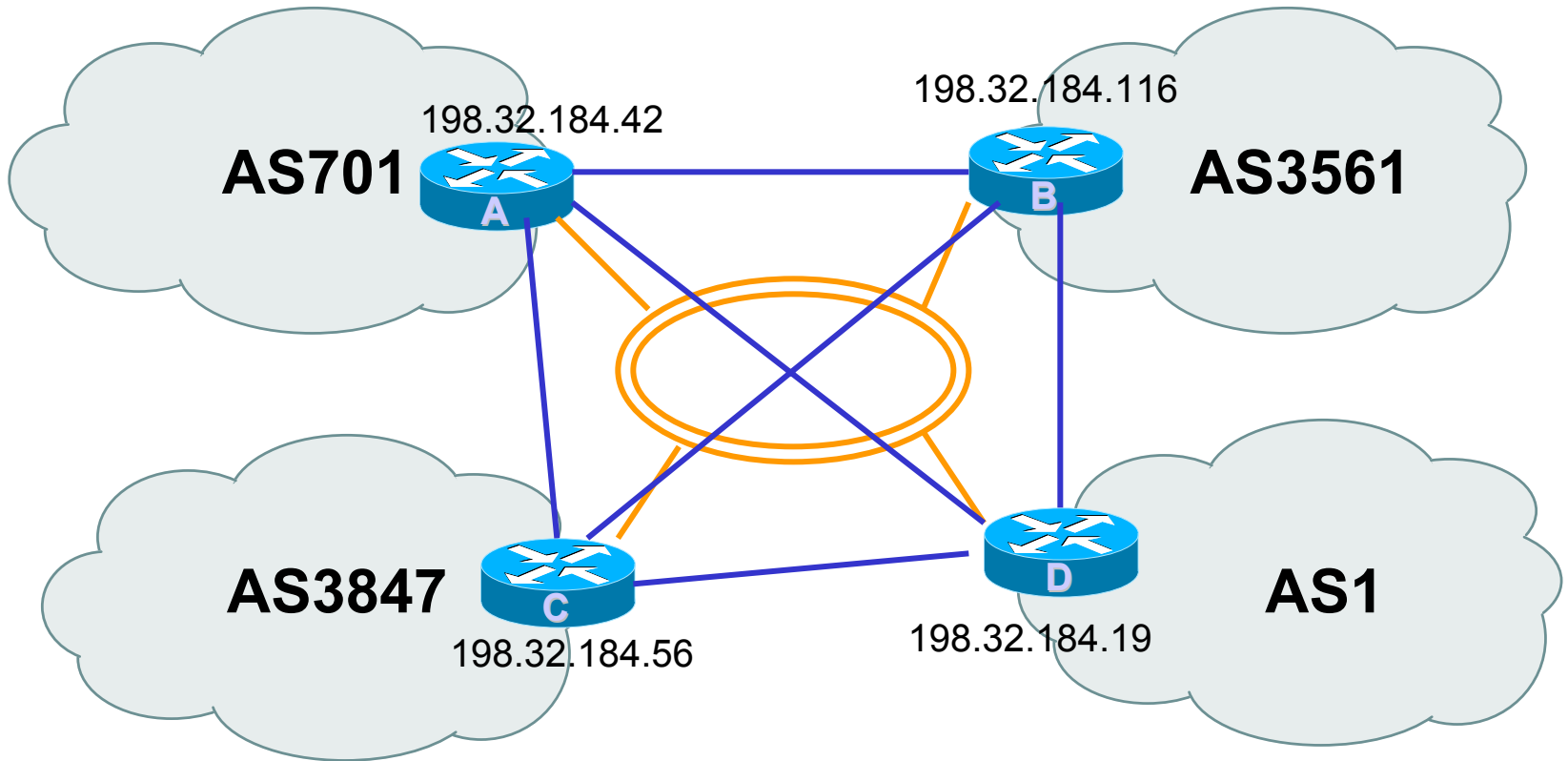
- A BGP message consists of a prefix and information about that prefix (i.e., local-pref, med, next-hop, originator, etc...). Each piece of information is encoded as an attribute in a TLV (type-length-value) format. The attribute length is 4 bytes long, and new attributes can be added by simply appending a new attribute.
- Attributes can be transitive or non-transitive, some are mandatory.

Next Hop Attribute

- 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.



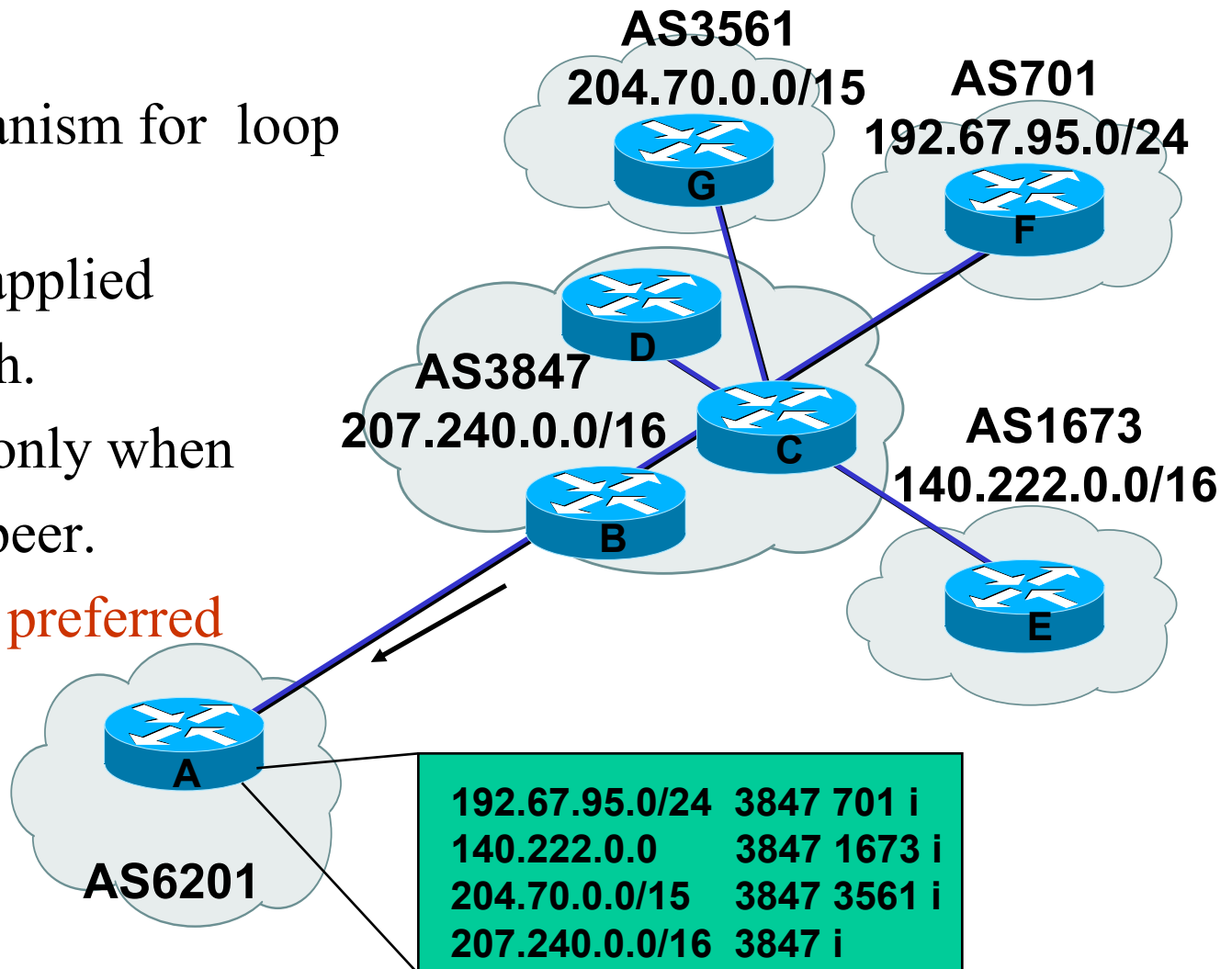
Next Hop Self



AS Path Attribute (1)

- 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



AS Path Attribute (2)

- Sprint is 1239; UUNET is 701; Net Access is 4969.
- When pattern-matching, or regexping, AS_PATHS, ^ means “match beginning”, and \$ means “match end”.
- The null AS-Path is ^\$ - if the AS-Path is null, the BGP route originated inside the same AS.

AS Path Attribute (3)

- ^1239 4969\$ is how a Sprint customer would see a Net Access route.
- ^1239 4969 11023\$ is how a Sprint customer would see a Net Access BGP customer's route.
- ^4969 11023\$ is how Sprint itself sees that same route.

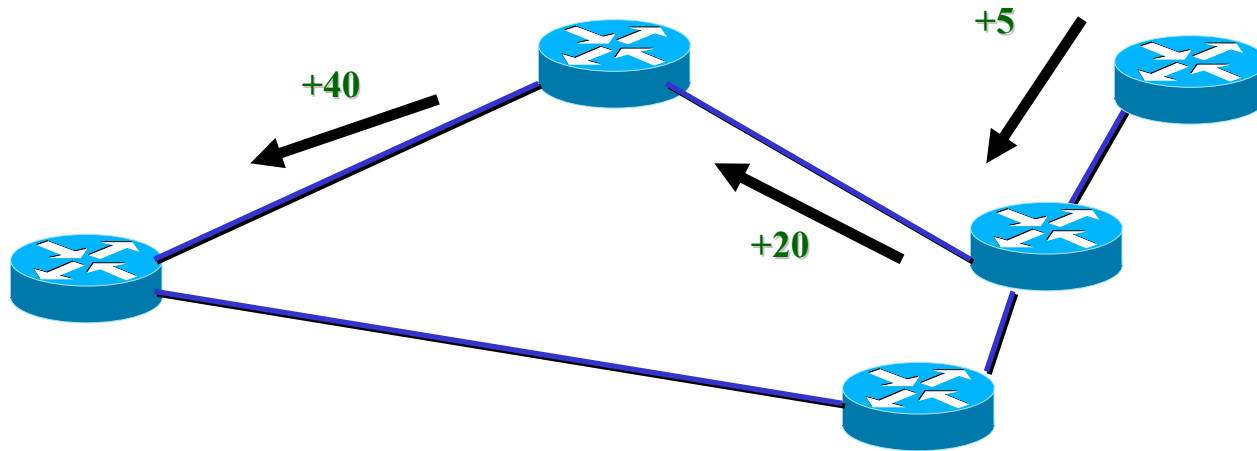
Multi-Exit Discriminator (MED)

- 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

MED Attribute (2)

- The MED (multi-exit discriminator) is a commonly used attribute. It comes after the AS_PATH in evaluation, and thus isn't quite as much of a “hammer” as local-pref.
- Commonly, MED is used to tack a distance on BGP routes as they move within your network.
- NSPs advertise MEDs to each other to let it be known which POP the route is “closest” to.

MED Attribute (3)



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

Origin Attribute

- One of the mandatory, but minor, attributes of a BGP route is the origin. It is one of (in order of preference):
 - IGP (i) (from a network statement)
 - EGP (e) (from an external peer)
 - Unknown (?) (from IGP redistribution)
- It can be re-set, but that is not often done.
- It is almost-last in the selection algorithm.

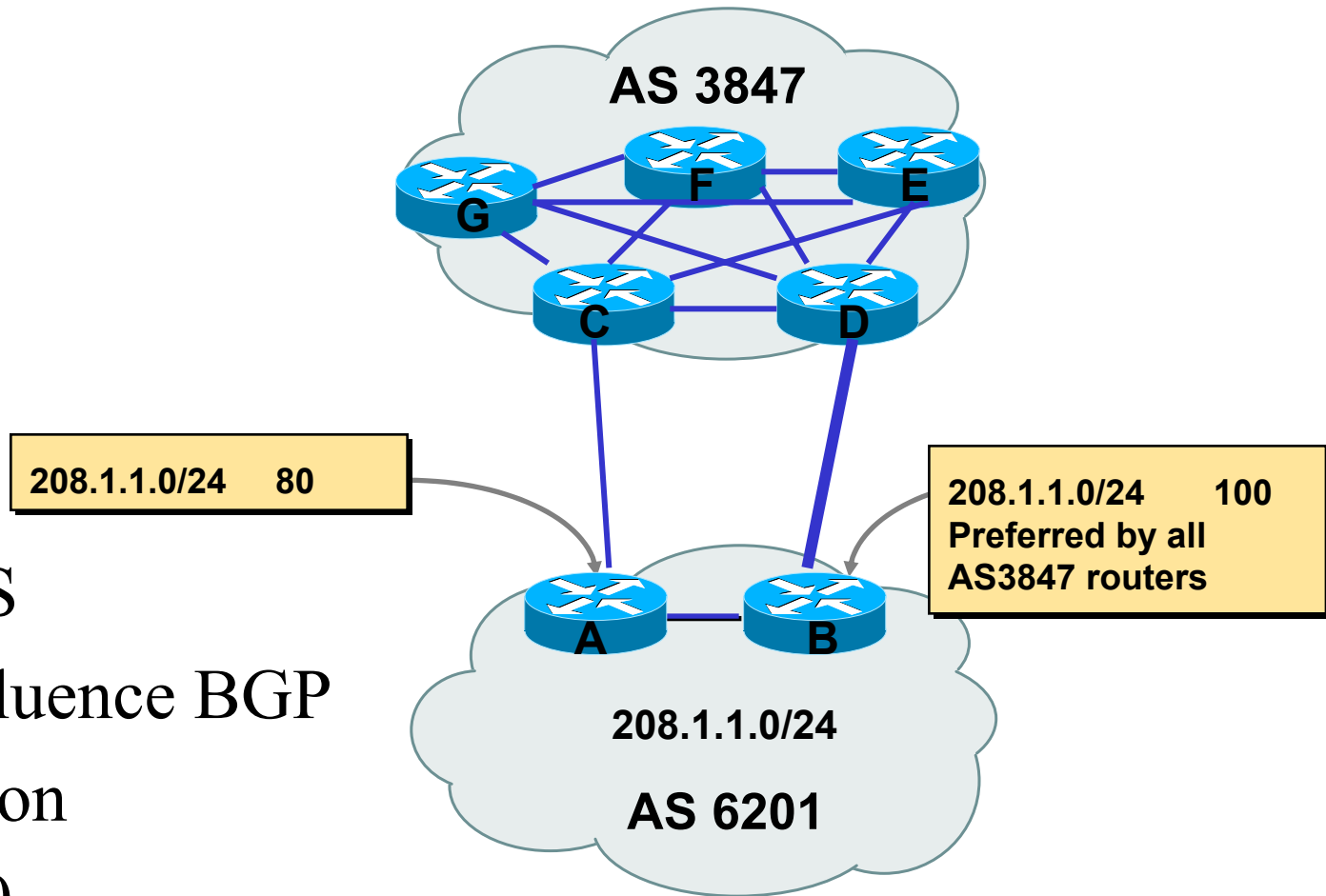
Weight Attribute

- Cisco proprietary, not part of any spec.
 - Local to router.
 - Value 0-65535 (default if originated by router - 32768, other - 0)
- * Highest weight preferred

Weight Attribute (ctd)

- Weight is rarely used. It overrides almost all other attributes in the decision path, and is local to a specific router - it is never sent to other routers, even ones inside your ASN.
- Usually used for temporary “I-don’t-have-time-to-think-about-it” fixes.

Local Preference Attribute



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

Local-Pref Attribute (2)

- An often-used attribute, local-pref (normally 100) overrides AS_PATH, and is transitive throughout your network. It is never advertised to an eBGP peer.
- For example, you can express the policy “prefer private interconnects” by making the local_pref be 150 and leaving all other peers at 100.
- Best used as an intermediate-level knob.

iBGP

vs.

eBGP

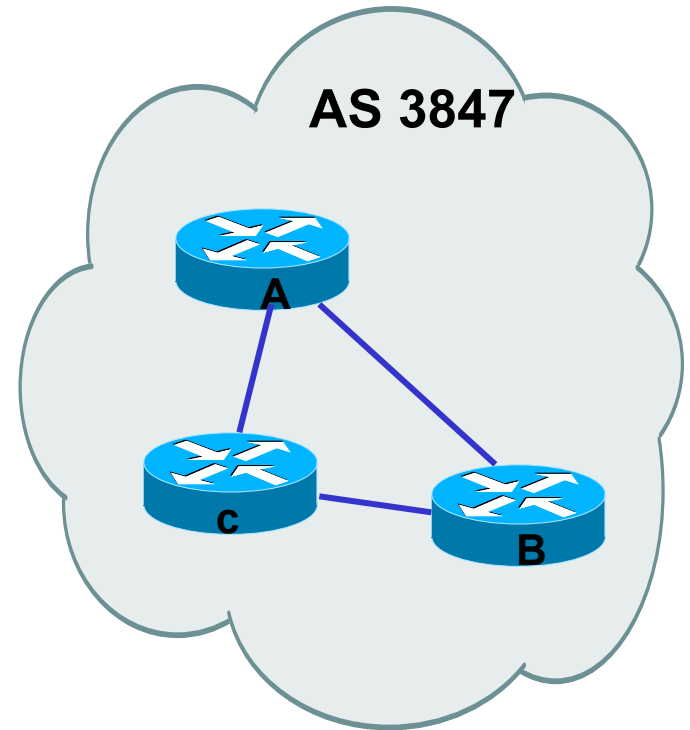
iBGP vs. eBGP

- BGP is very strange. It is promiscuous with external routes, making it very easy for you to become “MAE-Clueless”, yet it makes it very hard to advertise routes thoroughly inside your network.
- iBGP sessions are established when peering with the same AS; eBGP otherwise.
- Same protocols; different route install rules.
- **YOU MUST STRONGLY FILTER ALL eBGP SESSIONS!**

iBGP

When BGP speakers in the same AS form a BGP connection for the purpose of exchanging routing information, they are said to be running IBGP or *internal* BGP.

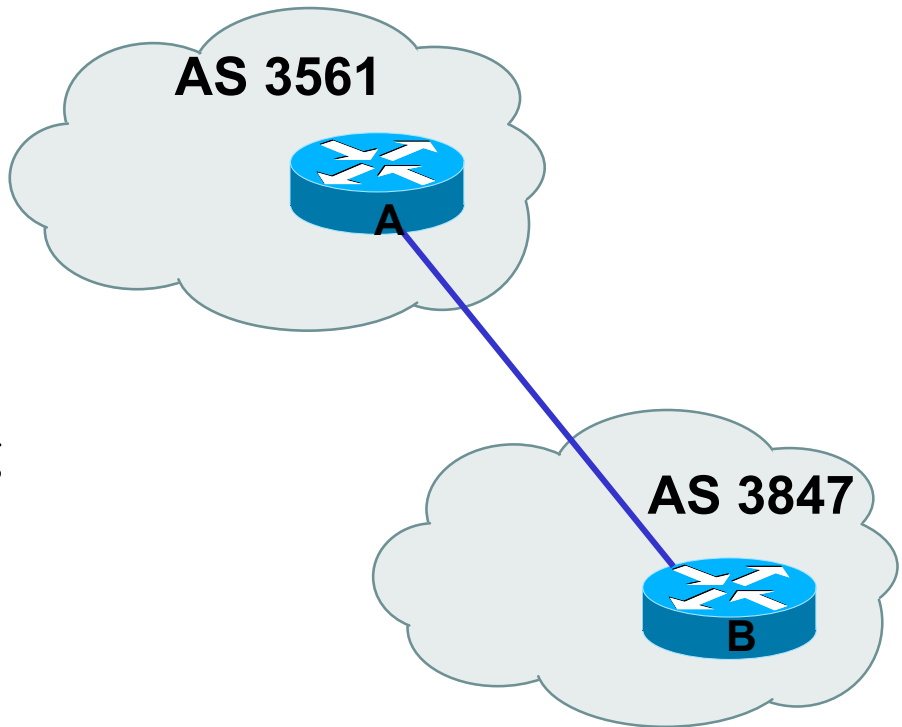
IBGP speakers are usually fully-meshed.



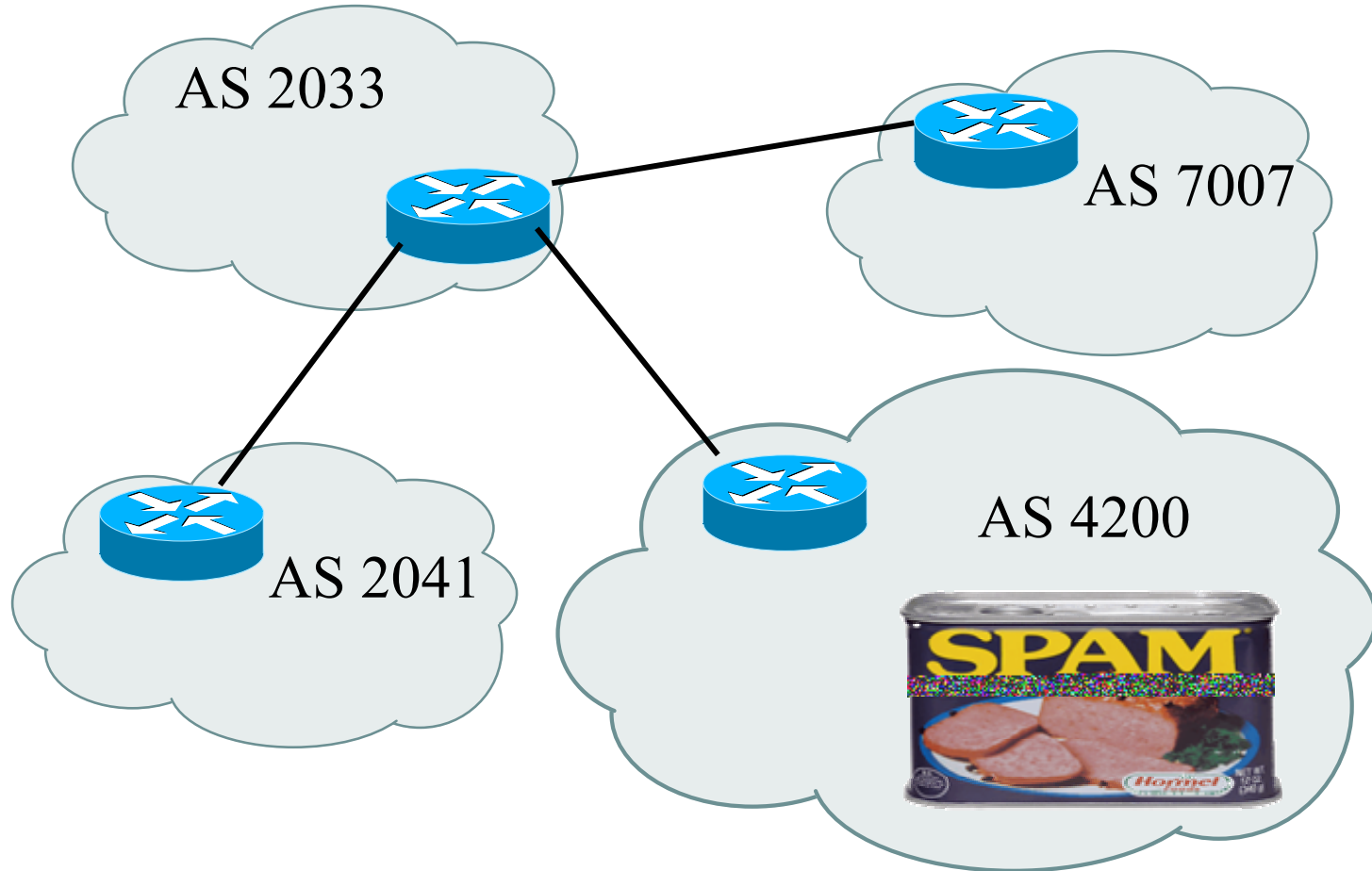
eBGP (1)

When BGP speakers in different ASs form a BGP connection for the purpose of exchanging routing information, they are said to be running EBGP or *external* BGP.

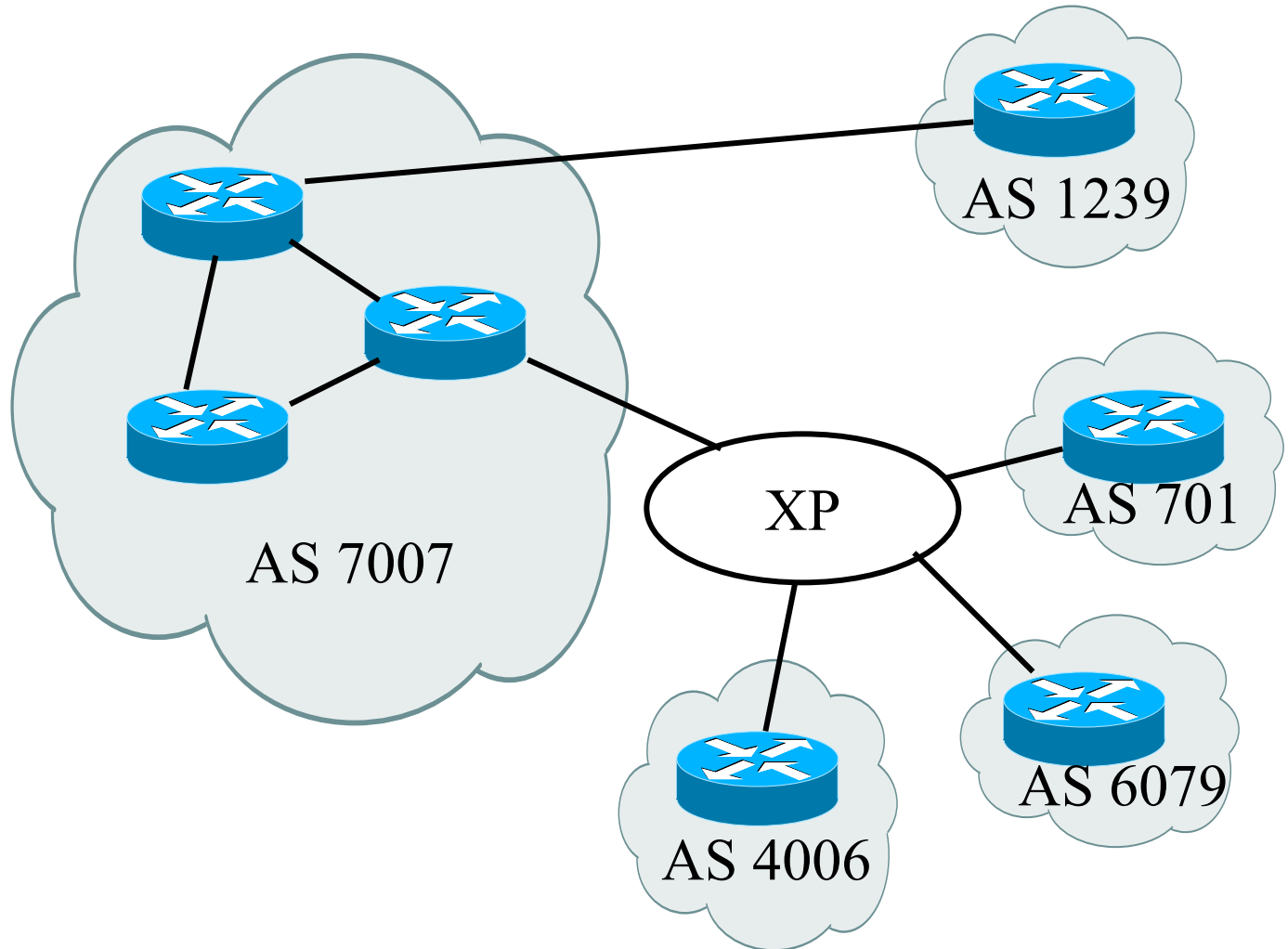
EBGP peers are usually directly connected.



eBGP (2)



iBGP and eBGP Diagram

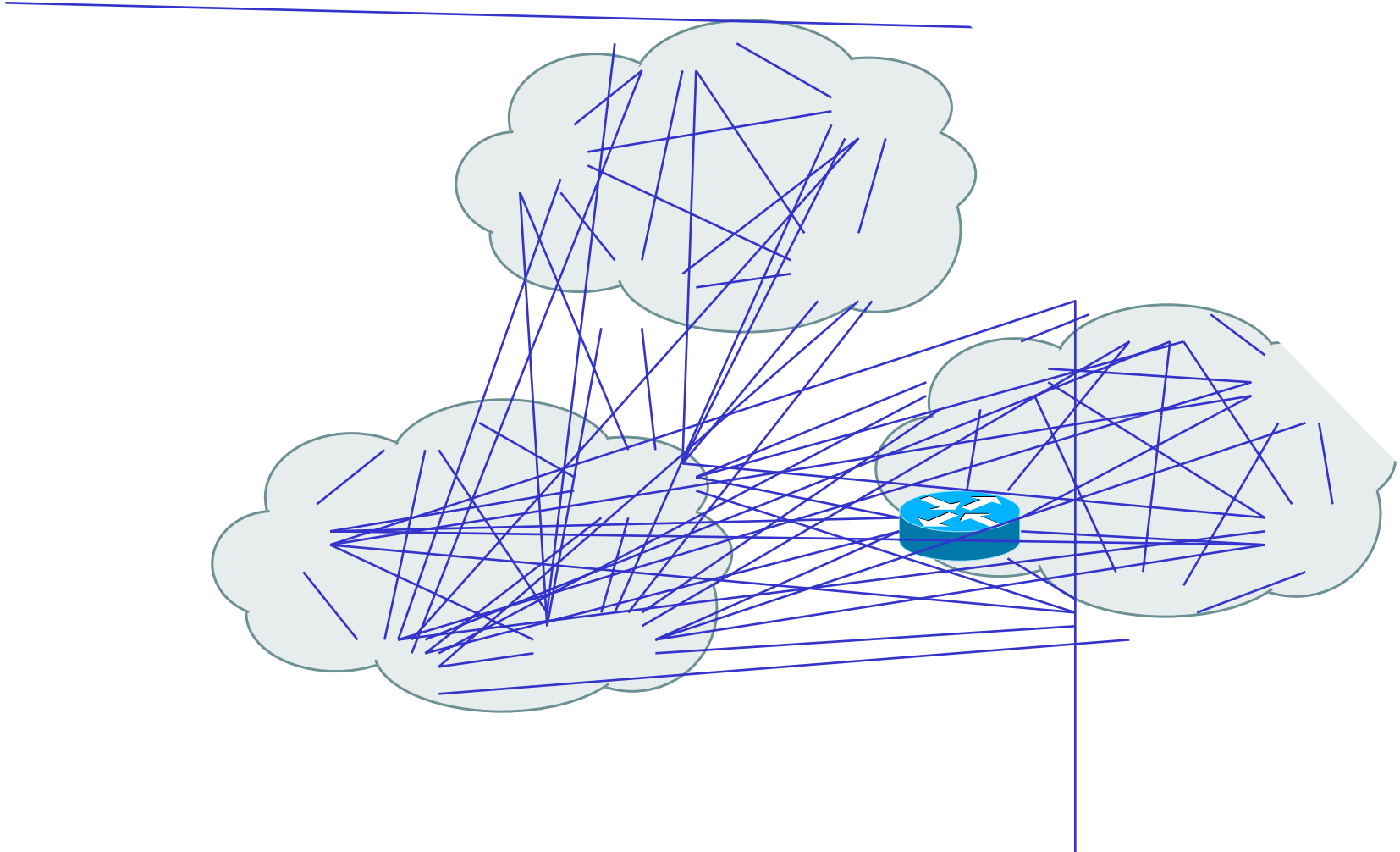


eBGP Rules

- By default, only talks to directly-connected router.
- Sends the one best BGP route for each destination.
- Sends all of the important “attributes”; omits the “local preference” attribute.
- Adds (prepends) the speaker’s ASN to the “as-path” attribute.
- Usually rewrites the “next-hop” attribute.

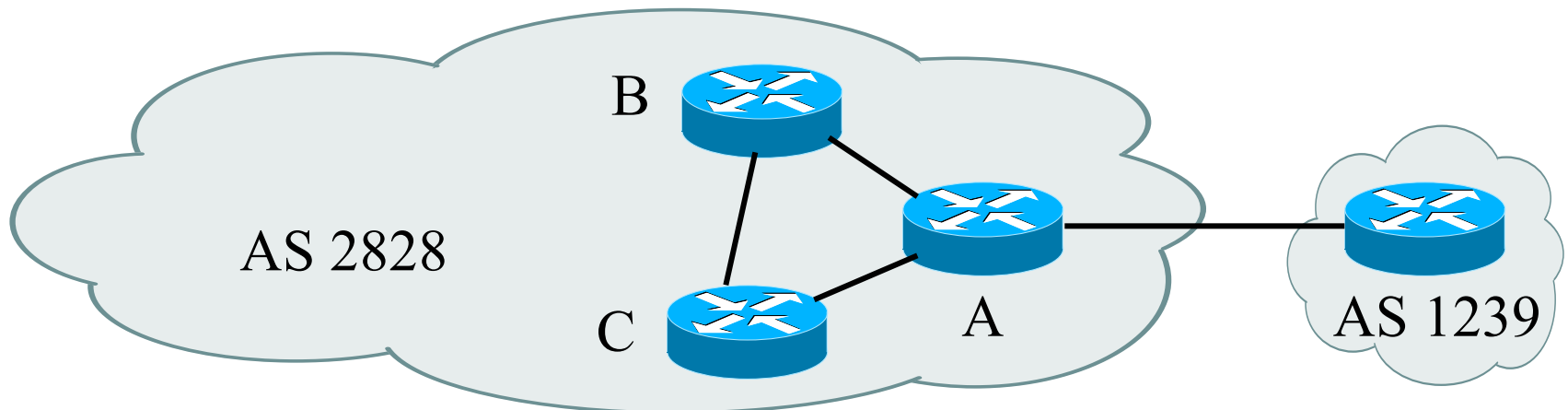
iBGP Rules

- Can talk to routers many hops away by default.
- Can only send routes it “injects”, or routes heard DIRECTLY from an external peer.
- Thus, requires a FULL mesh.
- Sends all attributes.
- Leaves the as-path attribute alone.
- Doesn't touch the “next hop” attribute.



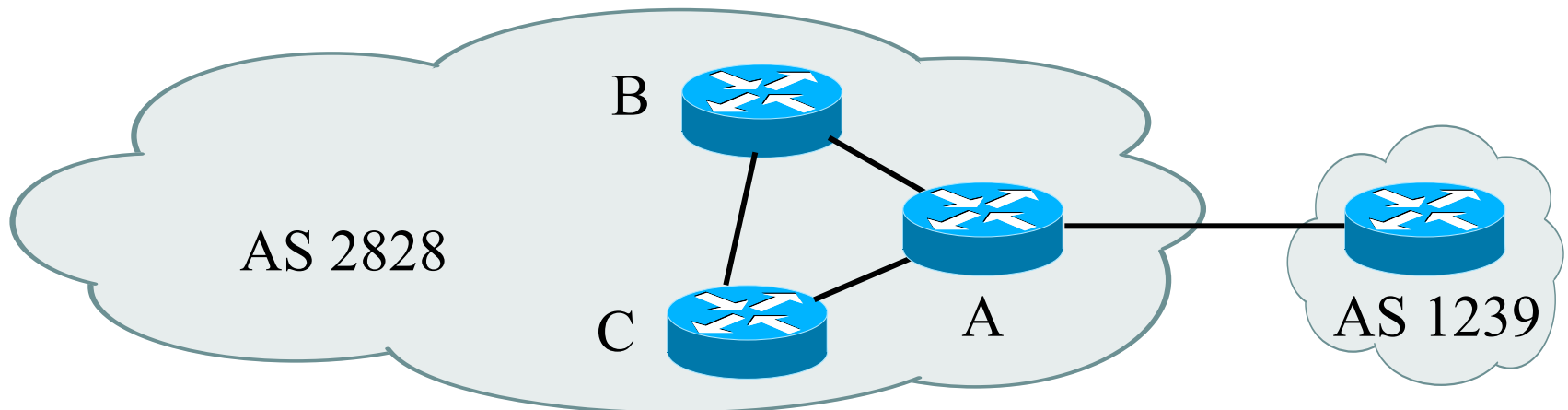
iBGP Restriction (1)

Assume AS1239 sends route 10.0.0.0/8 to AS2828. Router A will send that route to Routers B and C.



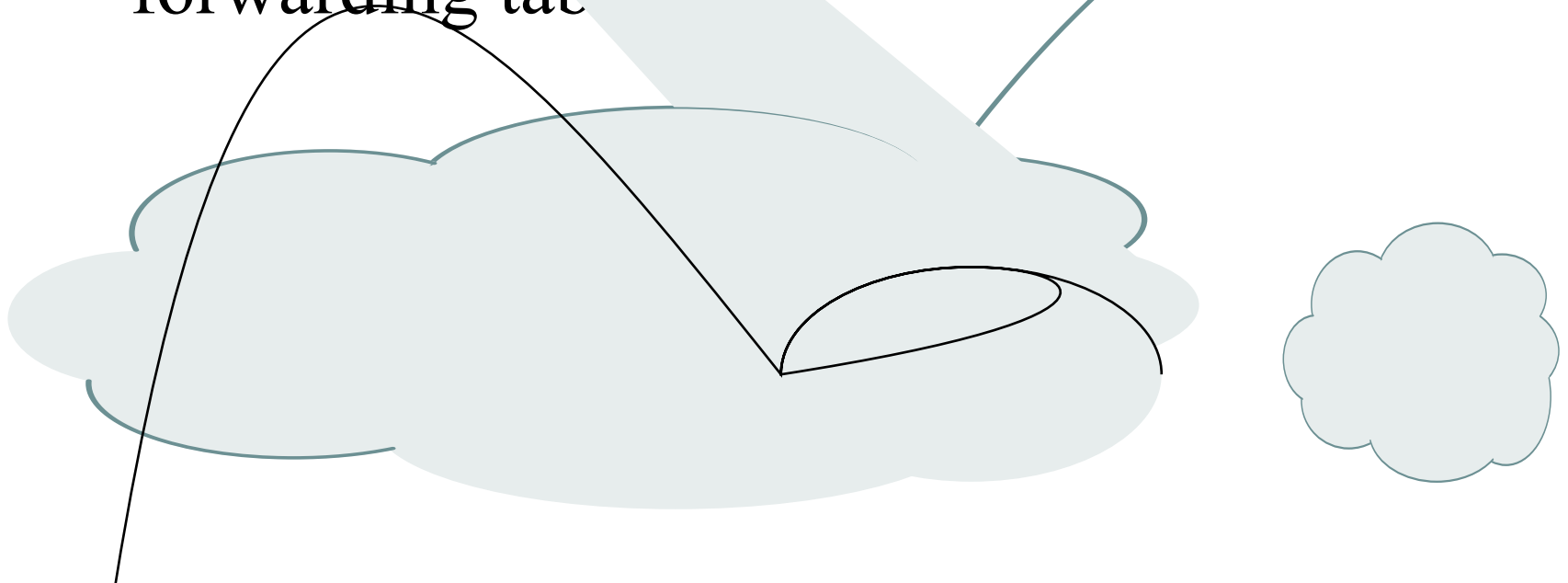
iBGP Restriction (2)

When Router B receives 10.0.0.0/8, it will not propagate that route to Router C because it was learned from an iBGP neighbor. Router C will behave similarly.



iBGP and next-hop (1)

Furthermore, the Next Hop for 10.0.0.0/8 will be the serial interface on the AS1239 router, even in Router B's and Router C's forwarding tables.



iBGP and next-hop (2)

- With iBGP, next-hop is not a router directly connected.
- So a “recursive lookup” is needed.
- After the next-hop is found, a second lookup is made to figure out how to send the packet “in the direction” of the next-hop.

Basic BGP Concepts

Inserting Routes

into BGP

Inserting Routes into BGP (1)

- How do routes get into BGP? They have to come from somewhere. You have to insert routes into BGP, and someone had to insert external routes that you get into BGP somewhere else in the first place.
- Two main ways:
 - network statements (like static BGP routes)
 - redistributing from OSPF, static, etc...

Inserting Routes into BGP (2)

- network statements
 - “network x.y.z.q [mask a.b.c.d]”
 - MUST have an EXACTLY-matching IGP route
 - specificity must be an exact match
 - Doesn’t scale beyond 200 or so network statements per routers; not a problem, though.
 - Makes scaling easier when you have to support multi-homed customers

Inserting Routes into BGP (3)

- aggregate-address statements
 - “aggregate-address x.y.z.q a.b.c.d [aggregate-only] [suppress-map XXX]”
 - (Really a relative of the network statement)
 - Brings up the given network if there are any more specific BGP routes for the prefix specified.
 - Usually used with aggregate-only to suppress more specifics.
 - Usually used in conjunction with redistribution.

Inserting Routes into BGP (4)

- Redistribution
 - ALWAYS redistribute through an address filter! Otherwise you will have crud in your BGP!
 - Examples later on...
- Default route is a special case. More soon.

Basic BGP

Advertising Routes

BGP Peering Sessions (1)

- BGP Routes are exchanged inside of BGP peering sessions.
- BGP uses TCP to ensure reliable delivery of routing updates.
- If a TCP session dies, all associated routes must be withdrawn.
- BGP peers, or neighbors, must be specified explicitly. This is a good thing.

BGP Peering Sessions (2)

- Once a peering session is set up:
 - Both sides flood the other end with all of their best BGP routes. VERY IMPORTANT - there is one best route per prefix, and that is the route that is advertised. BGP can only advertise routes that are eligible for use or routing loops can occur.
 - Then, periodic updates send new routes and/or withdraw old ones, and keepalives are sent every N seconds.
 - On a very stable network, very little or no traffic should flow besides keepalives.

Peering - BGP State Machine

- There is a state machine that describes the setting up, use, and tearing down of BGP sessions. It's useful to know the states because Cisco uses them to describe session state.
- Idle -> Connect -> Active {send “startup” packet} -> OpenSent -> OpenConfirm {wait for ack} -> Established [... -> Idle]
- In “sho ip bgp summ”, “Active” does NOT mean Active, it means “waiting” - FYI.

Peering - Processing Routes

- For each route received:
 - If it's a valid route AND passes any filters, it must be put into the BGP routing table.
 - Then, unless it is replacing a duplicate, a best-path computation must be run on all candidate BGP routes of the same prefix.
 - Then, if the best route changed, the RIB and/or FIB must be updated.
 - This process is done for ALL incoming BGP routes.

Filtering BGP Routes - BGP Policy Control

BGP Policy Control

- To decide what routes can and can't go to various other routers, you can “filter” using:
 - “distribute lists” (“prefix filters”) - lists of routes
 - “filter lists” (“as-path filters”) - lists of regular expressions matching or denying ASs
 - “route maps” (“BGP Basic programs”) that allow you to match and change most BGP attributes

Distribute List (1)

- Per neighbor access list applied to BGP routes
- Inbound or outbound
- Based upon network numbers

Distribute List (2)

```
router bgp 3847
neighbor 207.240.8.246 remote-as 8130
neighbor 207.240.8.246 distribute-list 127 in
neighbor 207.240.8.246 distribute-list 101 out
```

```
access-list 127 permit ip host 207.19.74.0 host 255.255.255.0
access-list 127 permit ip host 208.198.100.0 host 255.255.252.0
access-list 127 permit ip host 208.204.80.0 host 255.255.252.0
access-list 127 permit ip host 208.212.249.0 host 255.255.255.0
access-list 127 permit ip host 207.240.120.0 host 255.255.255.0
access-list 127 permit ip host 208.220.144.0 host 255.255.248.0
access-list 127 permit ip host 208.225.192.0 host 255.255.240.0
access-list 127 deny ip any any
```

! explicit deny if not specified

Distribute List (3)

```
access-list 10 deny ip 10.0.0.0 0.255.255.255
access-list 10 deny ip 127.0.0.0 0.255.255.255
access-list 10 deny ip 128.0.0.0 0.0.255.255
access-list 10 deny ip 172.16.0.0 0.15.255.255
access-list 10 deny ip 191.255.0.0 0.0.255.255
access-list 10 deny ip 192.0.2.0 0.0.0.255
access-list 10 deny ip 192.168.0.0 0.0.255.255
access-list 10 deny ip 223.255.255.0 0.0.0.255
access-list 10 deny ip 224.0.0.0 31.255.255.255
access-list 10 deny ip 207.240.0.0 0.0.3.255
access-list 10 permit ip any
```

A sanity filter like this keeps your table neat and prevents you from advertising crud to your peers.

Filter List (1)

- Filter routes both inbound and outbound based on value of AS path attribute.
- Called “as-path” access, or filter, lists.
- Configuration

```
router bgp 3847
```

```
neighbor 207.240.10.100 remote-as 2900
```

```
neighbor 207.240.10.100 distribute-list 100 in
```

```
neighbor 207.240.10.100 distribute-list 101 out
```

```
neighbor 207.240.10.100 filter-list 10 in
```

```
ip as-path access-list 10 permit ^2900$
```

```
ip as-path access-list 10 deny .*
```

Cisco Regular Expressions (1)

- 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

Cisco Regular Expressions (2)

^ 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.

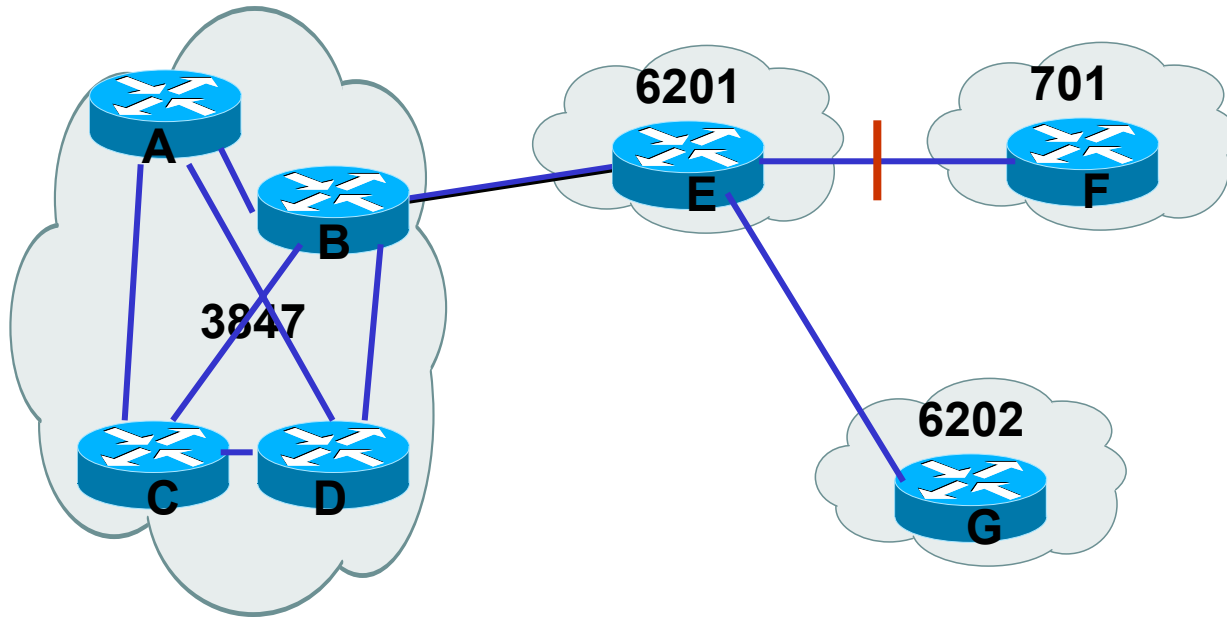
Cisco Regular Expressions (3)

[] Square brackets designate a range of single character patterns.

- Hyphen separates the endpoints of a range.

As you may have noticed, these are much like standard vi regular expressions.

Applying AS Path Filtering



The following configuration could be used on router B to accept routes from AS6201 & 6202 and deny all others.

```
ip as-path access-list 10 permit ^6201$  
ip as-path access-list 10 permit ^6201_6202$  
ip as-path access-list 10 deny .*
```

netaxs AS-Path ACLs

- 3 default lists
- (Permit all; Deny all; Permit only our routes)

```
ip as-path access-list 1 permit .*
```

```
ip as-path access-list 2 deny .*
```

```
ip as-path access-list 3 permit ^$
```

Route Maps (1)

Route-maps are cisco's mechanism to select and modify routes with if/then style algorithms.

Route-maps are used for more than just BGP in a cisco router, such as traffic shaping and policy routing.

Route Maps (2)

Route-maps follow this format:

```
route-map <name> <per|deny> <#>  
  [match statements]  
  [set statements]  
  
[repeat with unique sequence  
  numbers as needed]
```

Route Maps (3)

Route-maps follow this format:

```
route-map <name> <per|deny> <#>  
  [match statements]  
  [set statements]  
  
[repeat with unique sequence  
  numbers as needed]
```

Route Maps (4)

For route-maps with the keyword “permit”, if the prefix being examined passes the match statement, the set commands are executed and the route-map is exited.

If the match statement is not passed, the next sequence number is executed.

If there are no more sequence numbers, the prefix is filtered/dropped.

Route Maps (5)

For route-maps with the keyword “deny”, if the prefix being examined passes the match statement, the prefix in question is filtered and no more sequence numbers are executed.

If the prefix does not pass the match statements, the next sequence number is executed.

Basic BGP

Selecting Routes

Selecting BGP Routes

- Usually there will be 2, 3, 4, etc... ways to get to a given destination, all of which are represented by BGP routes.
- There is a way of picking the “best” one.
- Most important note -
 - Selection is NOT random between “similar” routes.
 - You can ALWAYS figure out why something is happening if you understand the rules.

Selecting BGP Routes - Basic

- ALWAYS find the most specific route.
- ONLY consider paths w/ reachable NEXT_HOPs.
- Prefer a route originated on the local rtr.
- Then, unless tuning has been done, pick the route with the shortest AS-PATH; then origin code; select on MED; then router ID.
- Or, if weight, LOCAL_PREF is set, or padding done to AS_PATH, look at those.

BGP Decision Algorithm

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

Multihoming with BGP

An Introduction

Step 1 - Determine Policy

- “You go find out what they want; we’ll start programming the routers” doesn’t work well.
- Before you step up to the router, determine what routing policy you want to express with your configuration.
- Plan your configuration, and ask how it could put you (in an unwelcome light) on the nanog mailing list.

Policy for Basic Multi-Homing

- We want to advertise our routes - all of them, but only OUR routes. So, assemble a list of our routes and masks.
- We want to accept all routes and let the router sort them out, initially based on AS-PATH length. If we don't have enough memory to take full routes, we'll start off taking none and then play later.

Warning - I am Blackholio (1)

- Never blackhole someone.
- Say `www.uu.net` is `137.239.5.24`, and the best match for that IP is the prefix `137.239.0.0/16`.
- What happens if you announce `137.239.5.0/24`, by accident or on purpose?
- Worldcom's lawyers show up at your doors and you look like an idiot.

Warning - I am Blackholio (2)

- What happens if you have a T1 to Sprint and a T1 to UUNET, and you announce Sprint routes to UUNET? (Assume no sanity filters at the upstream, which is always a good assumption).
- Answer - you have become MAE-Clueless, and all of UUNET tries to get to Sprint through your T1.
- Why?

Warning - I am Blackholio (3)

- As your provider, I have to believe that your route is the best way to get to a given prefix.
- Why? Because otherwise I can't transit you
- I can only send routes to the other providers on the Internet if I believe they are the best ones.

Multihoming - Minimal BGP

(for cheap routers)

Insert Static Default Routes

- Insert static default routes, either load-balanced or with primary/backup, as per non-BGP multihoming.
- Either
 - `ip route 0.0.0.0 0.0.0.0 s4/0`
 - `ip route 0.0.0.0 0.0.0.0 s4/1`
- Or
 - `ip route 0.0.0.0 0.0.0.0 s4/0`
 - `ip route 0.0.0.0 0.0.0.0 s4/1 250`

Gather Networks

- Routes
 - 207.8.200.0/22
 - 198.69.44.0/24
- Holdup routes keep the routes in BGP so they don't "flap". "Flapping" can blackhole you.
- Then, build access-list and holdup routes

```
access 55 permit 207.8.200.0 0.0.3.255
access 55 permit 198.69.44.0 0.0.0.255
ip route 207.8.200.0 255.255.252.0 null0 250
ip route 198.69.44.0 255.255.255.0 null0 250
```

Set up BGP Base Config

```
ip as access 1 permit .*
```

```
ip as access 2 deny .*
```

```
ip as access 3 permit ^$
```

```
router bgp 22222
```

```
no sync
```

```
net 207.8.200.0 mask 255.255.252.0
```

```
net 198.69.44.0 mask 255.255.255.0
```

Configuring Neighbors - Note

- The best way to configure a neighbor is to use cut-and-paste, or to tftpboot a snippet or whole config.
- You have 30-60 seconds to type in the whole neighbor clause before the session could come up and start receiving and sending routes - WITHOUT FILTERS if you didn't type fast enough...

Neighbor Configuration (1)

```
router bgp 22222
```

```
  neigh 207.106.2.45 descr transit to netaxs
```

```
  neigh 207.106.2.45 remote-as 4969
```

```
  neigh 207.106.2.45 next-hop-self
```

```
  neigh 207.106.2.45 version 4
```

```
  neigh 207.106.2.45 dist 55 out
```

```
  neigh 207.106.2.45 filter 3 out
```

```
  neigh 207.106.2.45 filter 2 in
```

Neighbor Configuration (2)

```
router bgp 22222
```

```
neigh 10.40.4.81 descr transit to UUNET
```

```
neigh 10.40.4.81 remote-as 701
```

```
neigh 10.40.4.81 next-hop-self
```

```
neigh 10.40.4.81 version 4
```

```
neigh 10.40.4.81 dist 55 out
```

```
neigh 10.40.4.81 filter 3 out
```

```
neigh 10.40.4.81 filter 2 in
```

Test it

- Do a “`sho ip bgp`”. Only your 2 routes should show.
- Do a “`show ip bgp neigh <neighip> adv`”. You should show that you are advertising those 2 routes to your 2 neighbors.
- Go to nitrous.digex.net or another BGP looking glass, to see that the routes are being advertised under your AS, not the provider’s, and that both paths are there.

**Multihoming with BGP -
Taking Customer Routes
(an intermediate solution)**

Taking Just Customer Routes

- One option in-between default routing and taking full BGP is to at least take customer routes from each provider.
- This way, you'll be able to make some intelligent decisions, which can be especially important for news feeding and dns and mail exchange optimization.
- If your provider isn't Sprint or CW, you can probably fit "customer" routes in 16mb.

Taking Just Customer Routes (2)

- The best plan is to get your provider to advertise their customer routes **ONLY** to you. Still, use the KGB motto - “Trust, but verify”.
- Doesn't work on small routers if your upstream is MCI or UU.
- Or, community-based filtering (more later).

Taking Just Customer Routes (3)

- So, a sanity filter:

```
ip as acc 10 deny _701_
```

```
ip as acc 10 deny _1239_
```

```
ip as acc 10 deny _3561_
```

```
ip as acc 10 deny _1673_
```

```
ip as acc 10 deny _1_
```

```
ip as acc 10 permit .*
```

- (Prevent hearing routes from the big boys - eve)

Taking Just Customer Routes (4)

```
router bgp 22222
```

```
  neigh 207.106.2.45 descr transit to netaxs
```

```
  neigh 207.106.2.45 remote-as 4969
```

```
  neigh 207.106.2.45 next-hop-self
```

```
  neigh 207.106.2.45 version 4
```

```
  neigh 207.106.2.45 distribute 55 out
```

```
  neigh 207.106.2.45 filter 3 out
```

```
  neigh 207.106.2.45 filter 10 in
```

Multihoming with BGP - Taking Full Routes

Policy

- Actually, very easy.
- Continue to advertise your routes, as before.
- Take full routing info.
- Later on, you can tune if you find that as-path is not a good indicator to some sites.

So, what Policy?

- We'll do the same thing on advertisement, but we'll take all routes from both upstreams.

Configuring Full BGP

- Router bgp 22222
 - neigh 207.106.2.45 remote-as 4969
 - neigh 207.106.2.45 next-hop-self
 - neigh 207.106.2.45 version 4
 - neigh 207.106.2.45 distribute 55 out
 - neigh 207.106.2.45 filter 3 out
 - neigh 207.106.2.45 filter 1 in

**Logistics of
becoming
Multihomed**

Multihoming Logistics

- Address space.
- Redundant connectivity during switch.
- Test configs.
- Bring up outbound BGP first.

TUNING INBOUND BGP ANNOUNCEMENTS

Inbound BGP Routes

- Inbound BGP routes make traffic go out. Having a route means that an outbound packet can use it as the basis for a forwarding decision (well, the router can).
- It is far easier to adjust outbound routing than inbound.
- Goal is generally to provide fastest, lowest-loss, path for all destinations.

Tuning Inbound BGP Routes

- Policy
 - Generally, to optimize throughput and latency.
 - Could be to squash traffic to certain providers, though, depending on the time of night and state of mind of the network engineer in question.
 - Or, to reduce transit cost.
 - Generally, though, it is to optimize connectivity “quality”, whatever that is.

Tuning Inbound BGP Routes

- Many destinations that you tune make themselves known in the form of customer complaints.
- Otherwise, start focusing on the biggest providers (Sprint, UU, CW, ATT, GENU, ...).

Tuning Inbound BGP Routes

- Use traceroutes to determine connectivity.
- However, do the traceroute from the source IP of the provider you are testing.
- No problem - do it from the border router and the source IP will be that of the serial interface.
- So, just set a temporary static route to a given destination and trace away...

Tuning Inbound BGP Routes

- Once you identify better paths, use AS_PATH padding.
- Identify the providers in question.
- Pick out the relevant AS_PATH regexp.
- Build a route-map to apply inbound.

Tuning Inbound BGP Routes

- Simple route-map

```
ip as acc 20 permit ^701 1673_
```

```
route-map inbound-uu permit 10
```

```
match as 20
```

```
set as pre 701 701
```

```
route-map inbound-uu permit 20
```

```
match as 1
```

- Always best to leave a specific match all at the end.

Tuning Inbound BGP Routes

- Other methods:
- We'll talk about `local_prefs` later on...

TUNING OUTBOUND BGP ANNOUNCEMENTS

Tuning Outbound BGP

- This is harder, because all of the other networks implementing their own policies complicate your life.
- Your two main tools are:
 - Padding your outbound AS_PATHs
 - Deaggregating announcements
- And:
 - With a cooperative provider, using communities

Tuning Outbound - Padding

- When your router announces iBGP routes, it normally creates a 1-entry AS_PATH with your ASN. So, by adding one or more copy of your own ASN, you cause the providers who listen to that route to de-prefer it a bit (since the AS_PATH is now 1 longer, thus making it win less often).

Tuning Outbound - Padding

```
route-map pad-me-once  
  match as 1  
  set as prepend 22222
```

```
router bgp 22222  
  neigh 207.106.2.45 route-map pad-me-once out
```

Tuning Outbound - Communities

- If your providers are good (netaxs, above.net, some others), they'll give you the ability to control your destiny with communities.
- For example, netaxs honors the communities:

Tuning Outbound - deagg.

- I have 207.106.128.0/17.

I want to advertise 207.106.128.0/17 to spr and uu, and 207.106.128.0/18 to spr alone.

```
access 56 deny 207.106.128.0 0.0.63.255
```

```
access 56 <insert lines from access 55>
```

```
neigh <uunetip> dist 56 out
```

Research Issues

Research Problems

- Convergence
 - Loopy
 - Satisfiable policy?
 - Cpu/ram issues?
- RAM issues
 - Stupid storage structures
- MEDs/bilateral semantics
 - Poor performance
 - Causes looping

Research Problems (2)

- Hidden design problems
 - Lack of sender-side loop detection
 - Min route advertisement interval
- Lack of authentication
- Bad code
- NOTIFY behavior
- No intelligence/performance data

"Internet Dies,
Slides at 11"

Death of the Internet

- Blackholio: AS7007 deaggregates
- BGP -> OSPF redistribution
- 'test crash internet'
- Reserved ASs
- Malformed BGP updates
- Router hardening