BGP 101

Avi Freedman
avi@servercentral.net

Index

- Internet Connectivity Overview
- Multihoming Concepts
- Multihoming Without BGP
- Multihoming - Address Space Complications

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

Index

- Multihoming with BGP - an Introduction
- Interlude - Hardware for BGP
- Multihoming with BGP - Taking Full Routes
- Default Routing in BGP

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 on the outbound side (with default routes).
- Packets from everywhere else have to “find you”. This is done by having your upstream provider(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 - sending routes OUT causes IP traffic to come IN.

BGP Route Advertisement (3)

- But the most specific route wins, so if one of our customers' providers is advertising 207.8.240.0/24 and we are advertising 207.8.128.0/17, all incoming traffic from other networks will start flowing in that pipe.
- So we must advertise both 207.8.128.0/17 and 207.8.240.0/24 so that we match the more specific being advertised elsewhere.

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 g4/0
  - – ip route 0.0.0.0 0.0.0.0 g5/0
- To use one link as a backup only for outbound packtes:
  - – ip route 0.0.0.0 0.0.0.0 g5/0
  - – ip route 0.0.0.0 0.0.0.0 g5/0 10
  - – why? g5/1 could be a more expensive link

Load Balancing

- The way most routers and switchrouters work 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 packet hashing or route caching (default). Traffic to the same dest IP will always use the same interface (potentially until the cache entry expires, depending on the device).

Customer Side - Inbound

- Just tell your provider what address space you are bringing, if any.
- Your provider 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 provider what to do.

Address Space and ASNs

- First, you’ll need some address space. Assuming your provider gives you permission, you can use their address space to multi-home
- Second, you’ll need an ASN (Autonomoous System Number). You can get this from an RIR (Regional Internet Registry) once you show them that you have at least 2 bandwidth contracts.
- RIRs include ARIN, RIPE NCC, APNIC, LACNIC, AfriNIC.

Routing Registries

- Your providers may also require that you enter your routes (or your intention to advertise those routes) into a routing registry.
- Some providers run their own; if not, we recommend altdb (which is free).
- This is done for easy (re)-build of filters by your providers.

Provider Side (1)

- If both providers don’t advertise your routes with the same specificity, you might have -
  - nLayer saying “4436 says 207.8.128.0/17”
  - Sprint saying “1239 says 207.8.195.0/24”
- Bad, because almost all traffic on the ‘net will come into you via Sprint.

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 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 “classless” routing.

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

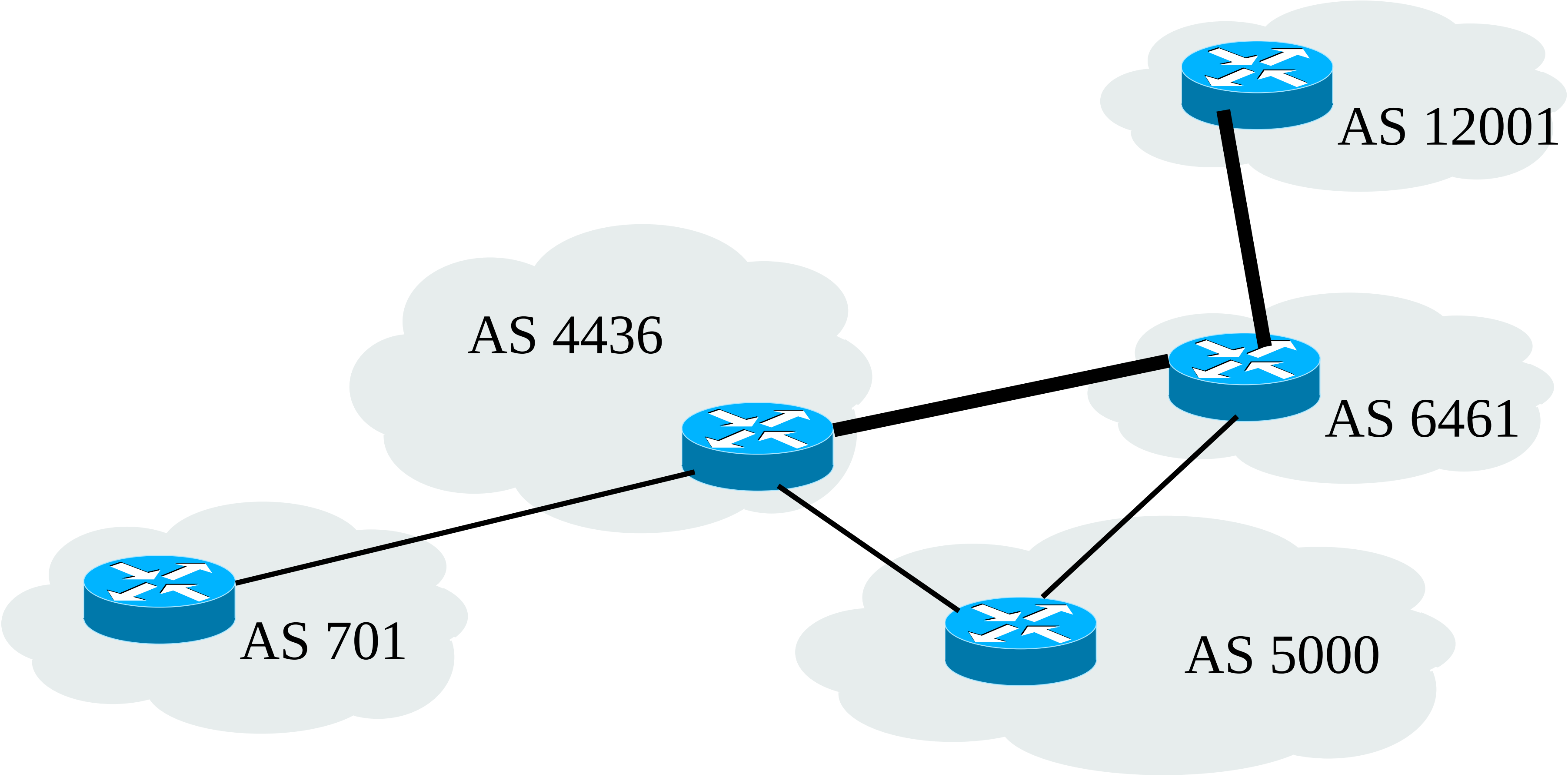
Basic BGP Concepts (1)

- BGP exchanges routes between networks (Autonomous Systems or ASs). Each AS has an Autonomous System Number, or ASN.
- When routes are exchanged, ASNs are stamped on the routes *on the way out* - adding one “AS hop” per network traversed. (1-64511)
- No concept of pipe size, internal router hop-count, congestion - in some sense BGP treats all ASs the same.

Basic BGP Concepts (2)

- Why does BGP have this “AS Path” that gets built as routes flow between networks?
  - Loop detection – BGP when used **between providers** has no other way to avoid loops.
  - Setting policy – BGP was developed at the dawn of the commercial Internet, and was designed to allow providers to express ‘policy’ decisions (prefer nLayer over Sprint to get to Comcast).
  - And when your routing fu grows, you can inspect a routing table and scan for networks you know by watching for their ASN in paths.

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 internal tables (Routing/Forwarding).
- Once a session comes up, all best-routes are exchanged. Then over time, just “topology updates” are exchanged.
- You can ONLY exchange “best” routes.

Routing Tables

- Inside the router you can model routing tables as follows:
  - One for each dynamic routing protocol (BGP, OSPF, IS-IS, etc), plus one for static routes and one for connected;
  - A routing table which is where the best route from all of the other IP routing tables goes;
  - A forwarding table where just the prefix and destinations go (potentially sent to line cards)
- Each time a route flows downwards, information is stripped.

Basic BGP Configuration

Basic eBGP Configuration

```
interface g1/3
  ip address 198.7.0.6 255.255.255.252

router bgp 7007
  network 207.108.10.0 mask 255.255.254.0
  neighbor 198.7.0.5 remote-as 6450
  neighbor 198.7.0.5 prefix-list mine out
  neighbor 198.7.0.5 prefix-list no-bogons in
```

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?

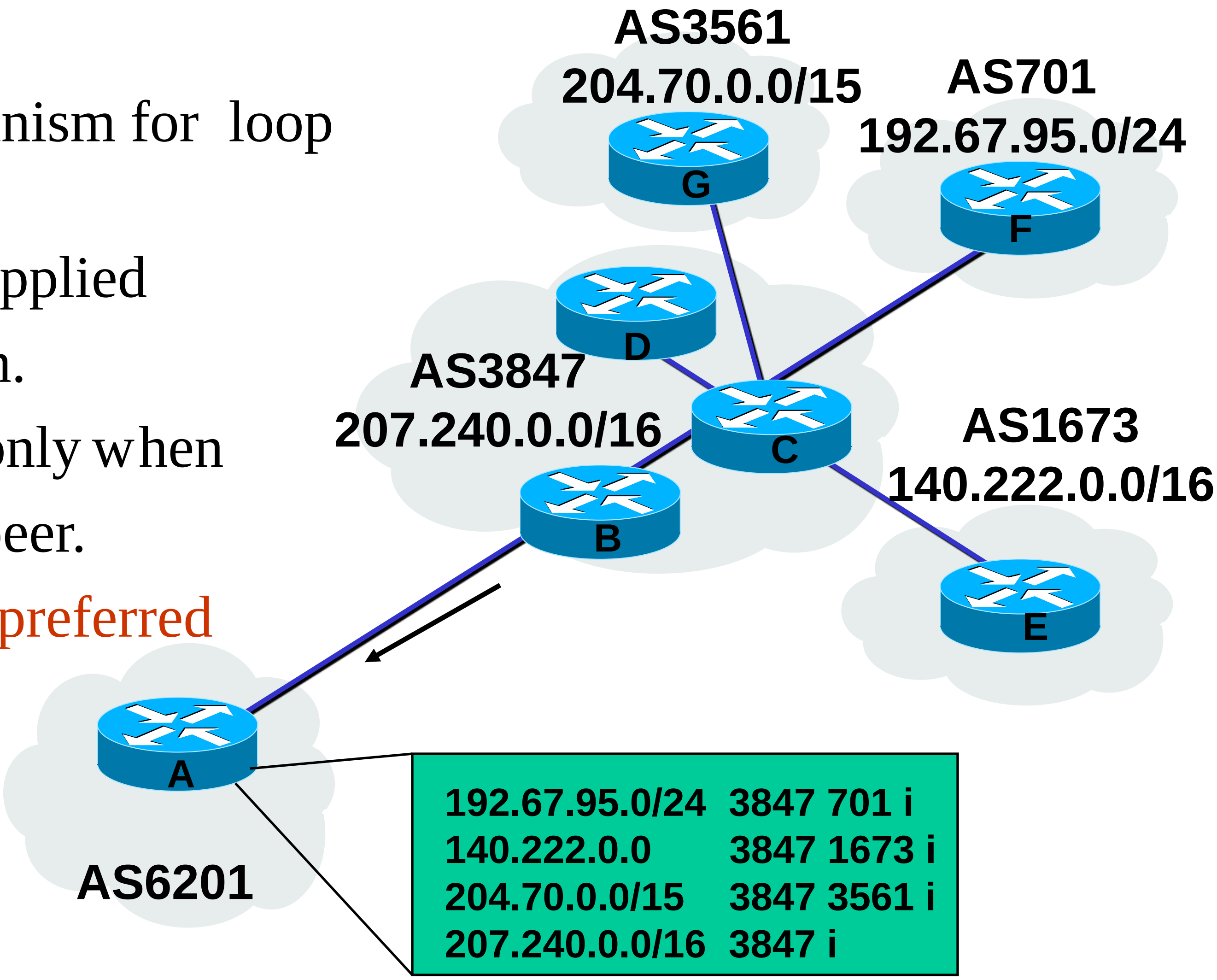
...	Next Hop	AS Path	MED	...	...

- 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.

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



192.67.95.0/24	3847 701 i
140.222.0.0	3847 1673 i
204.70.0.0/15	3847 3561 i
207.240.0.0/16	3847 i

AS Path Attribute (2)

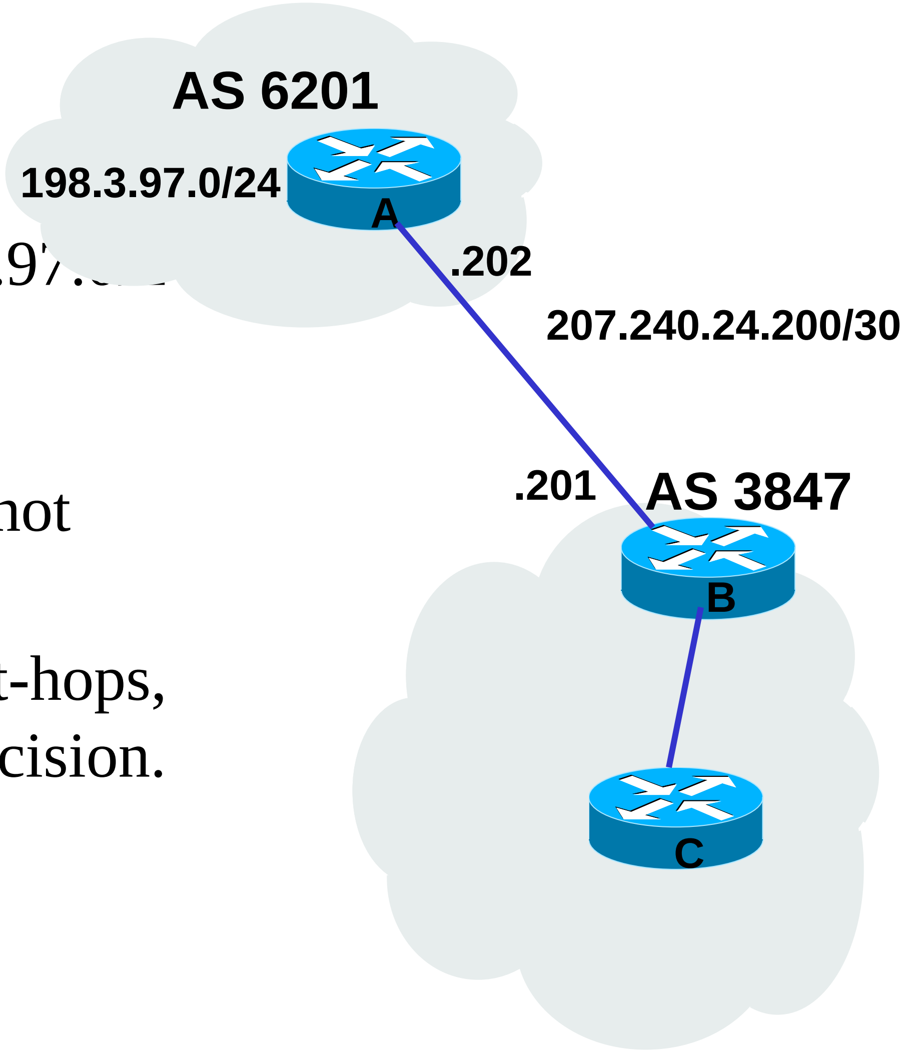
- **NOTE: There are now 4-byte ASNs, support for which will be mandatory within 1-2 years…**
- 0 and 65536 are reserved; 64512-65534 are for ‘private use’ only.
- nLayer is 4436; Sprint is 1239; Verizon/UUNet is 701.
- When pattern-matching, or regexping, AS_PATHs, ^ means “match beginning”, and $ means “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 4436$ is how a Sprint customer would see a nLayer route.
- ^1239 4436 11023$ is how a Sprint customer would see a nLayer BGP customer’s route.
- ^4436 11023$ is how Sprint itself sees that same route.

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.
- IGPs should carry route to next-hops, using intelligent forwarding decision.



Multi-Exit Discriminator (MED)

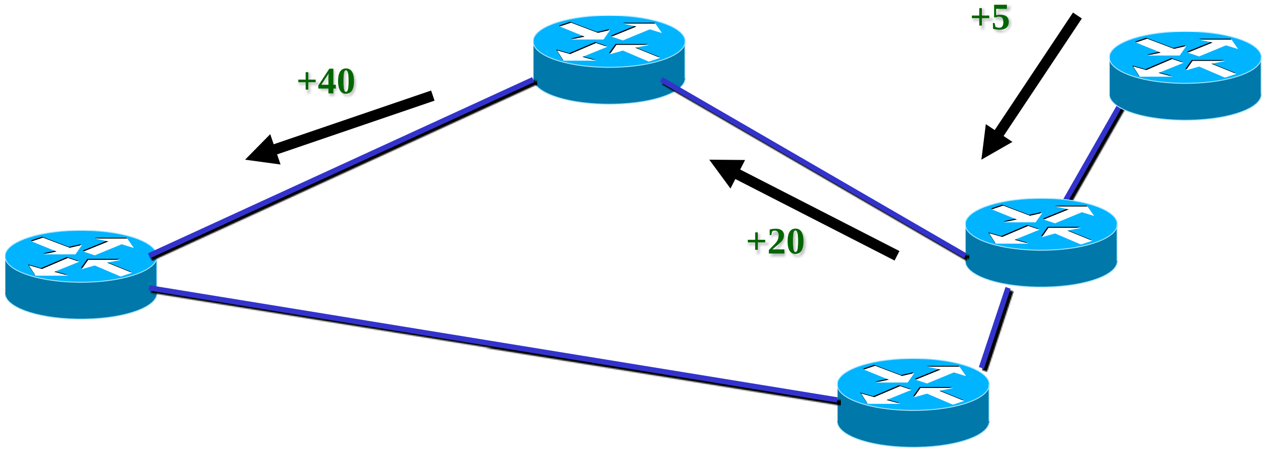
- Indication to external peers of the preferred path into an AS.
- Affects routes with same AS path.
- Advertised to external neighbors
- Usually based on IGP metric

* 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.
- Providers advertise MEDs to each other to let it be known which POP the route is “closest” to.

MED Attribute (3)



- Applies intelligently only to decisions between paths from the same provider
- But… There are some issues with using MED.
- Many prefer to send/receive communities and let your peers set their own metrics based on those.

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

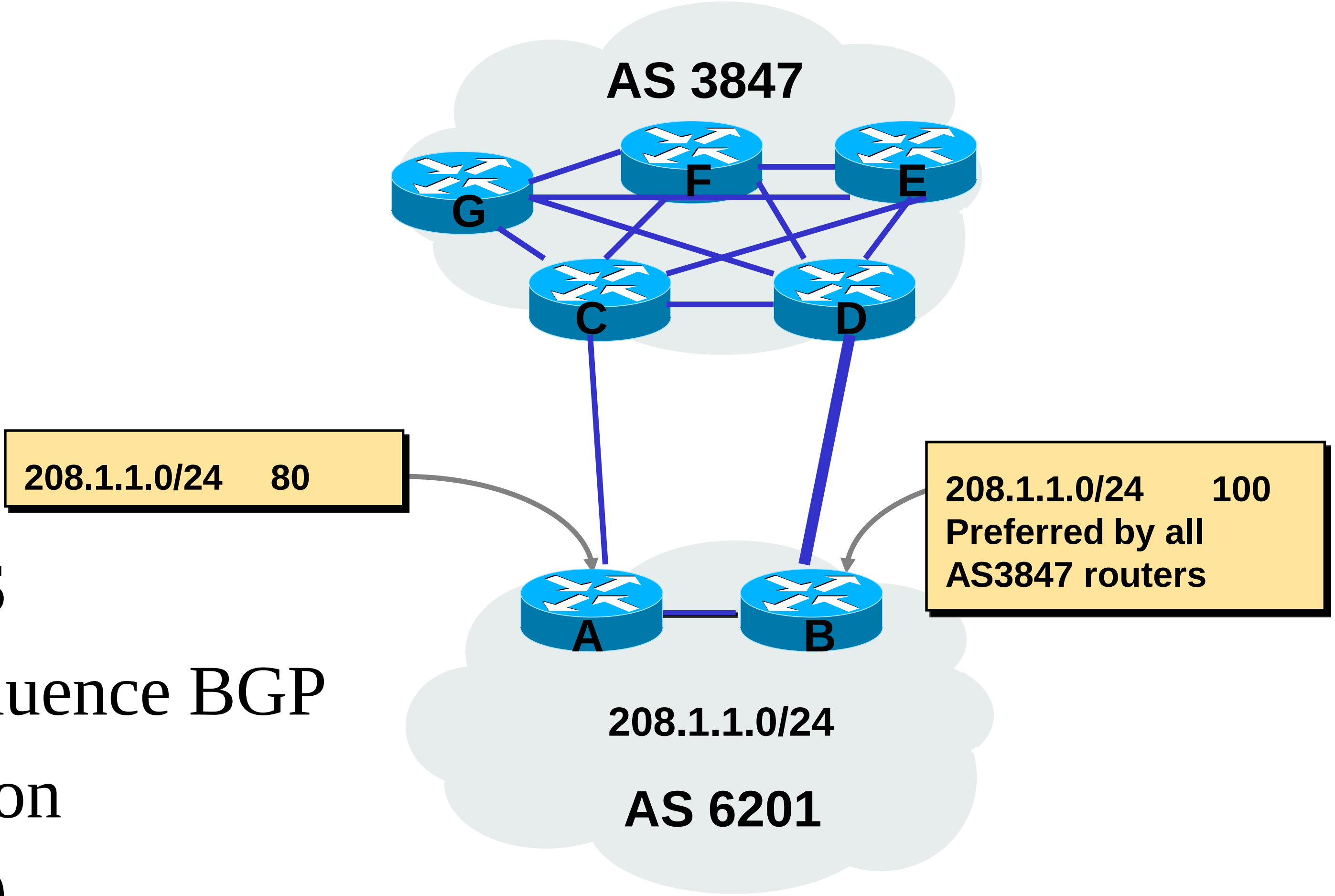
- Cisco introduced this.
- DO NOT USE IT until your fu is strong, if ever! Because…
- Weight is local to the router.
- So it's easy to get routing loops.
- Value 0-65535 (default if originated by router - 32768, other - 0)

* 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



208.1.1.0/24 80

208.1.1.0/24 100
Preferred by all
AS3847 routers

- 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

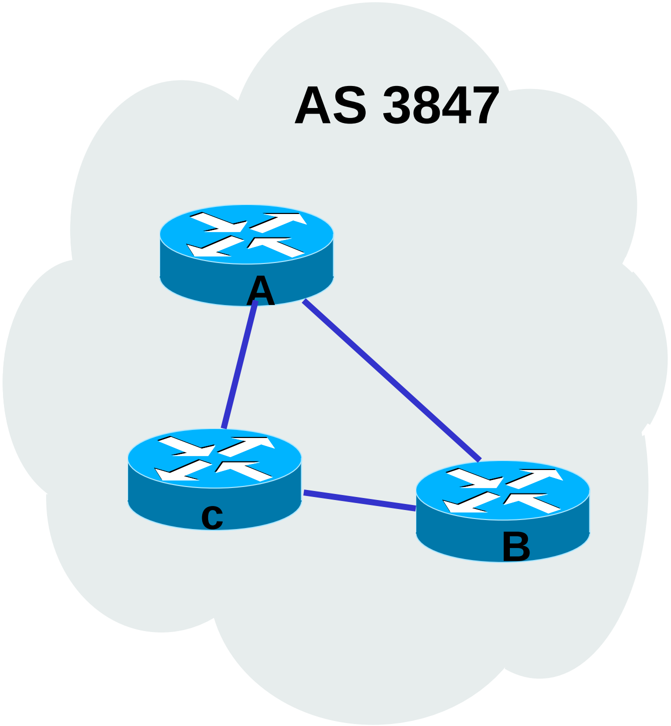
iBGP vs. eBGP

- BGP is very strange. It is promiscuous with external routes, making it very easy for you to (try) to blow up the Internet, 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!

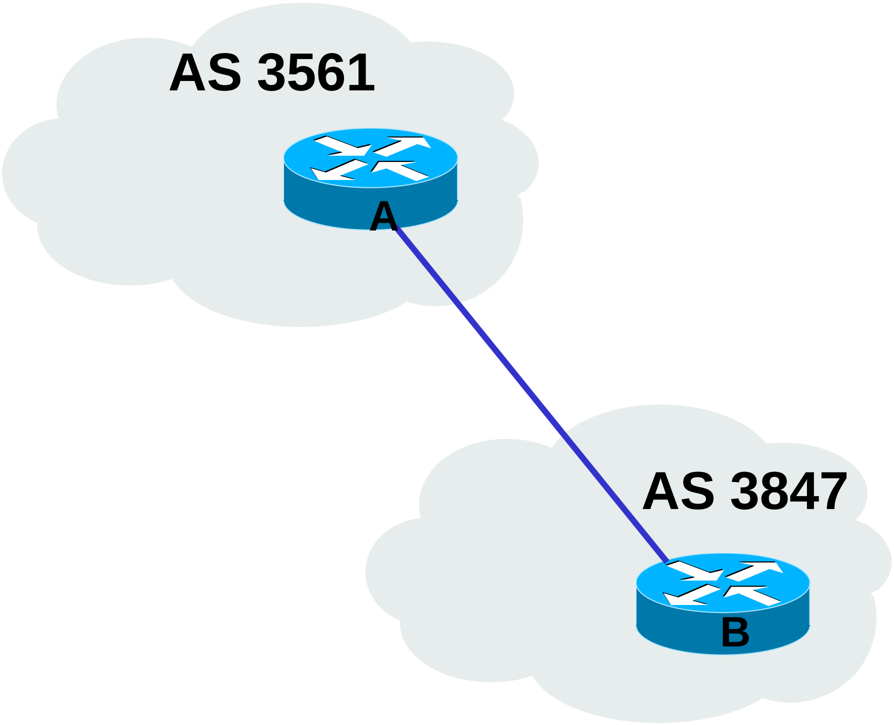
iBGP

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. This has scaling issues that are covered in BGP 102.



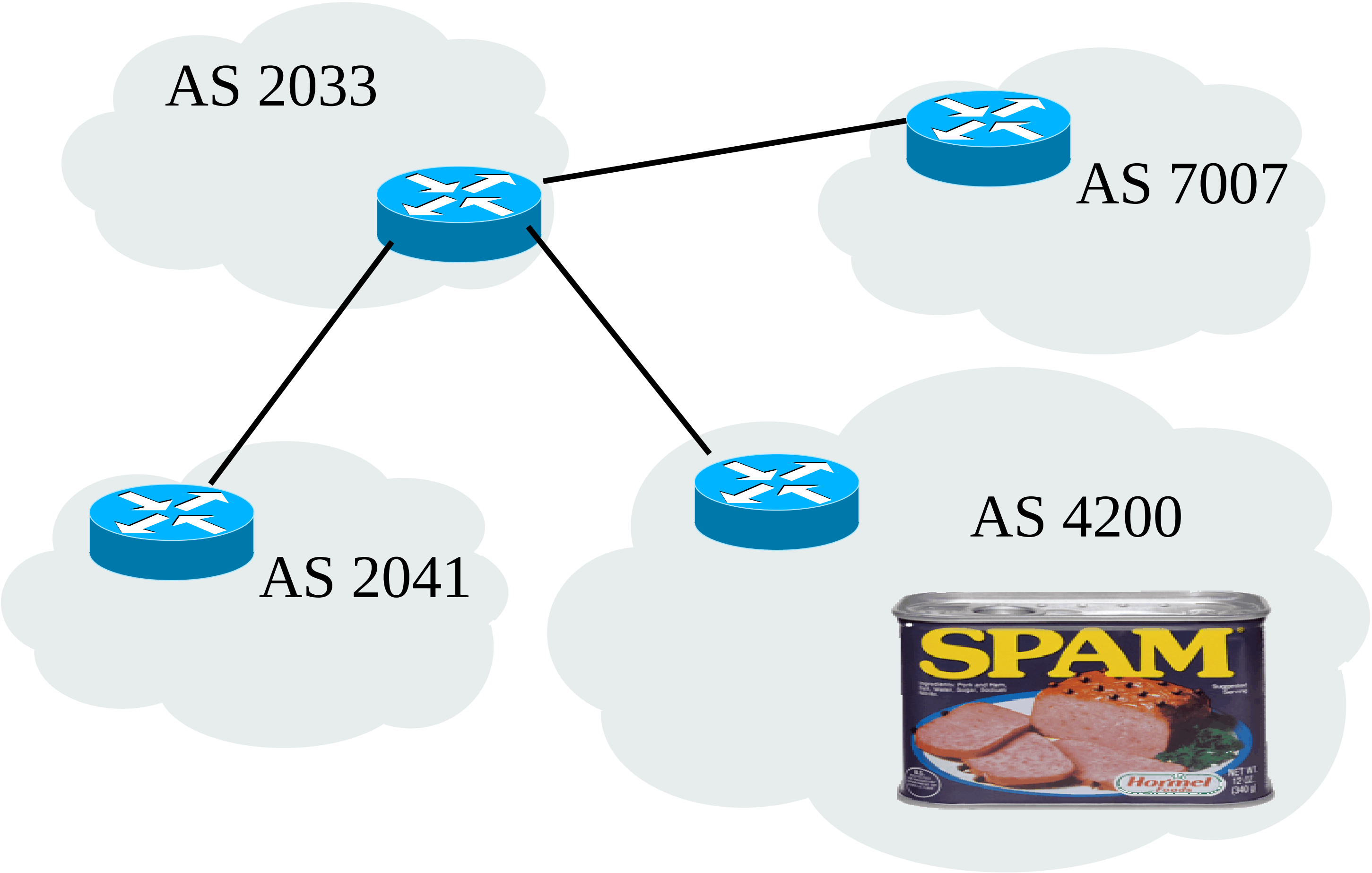
eBGP (1)



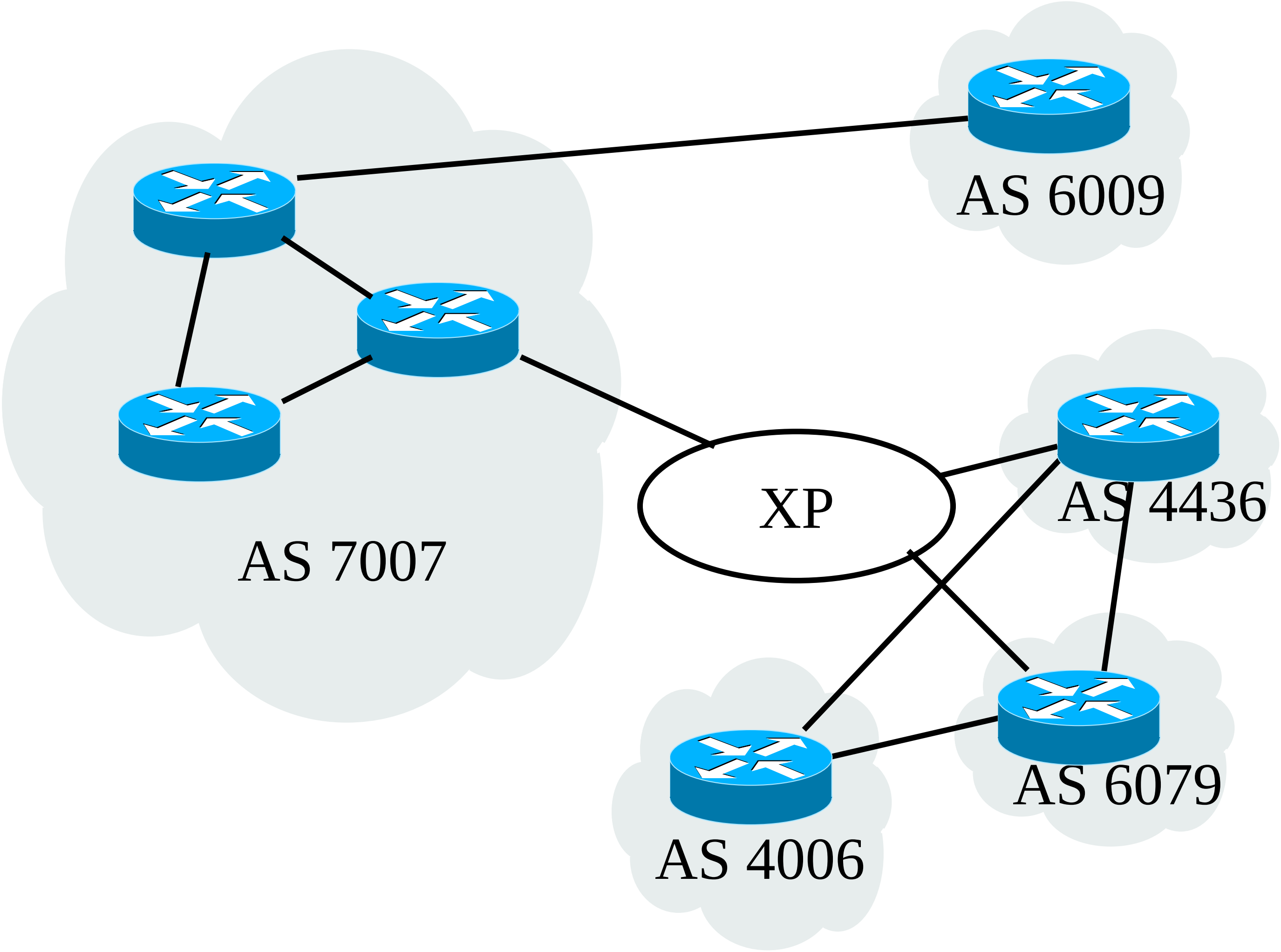
When BGP speakers in different ASs form a BGP connection for the purpose of exchanging routing information, they are said to be running EBGP or *external* BGP.

EBGP peers are usually directly connected.

eBGP (2)



iBGP and eBGP Diagram



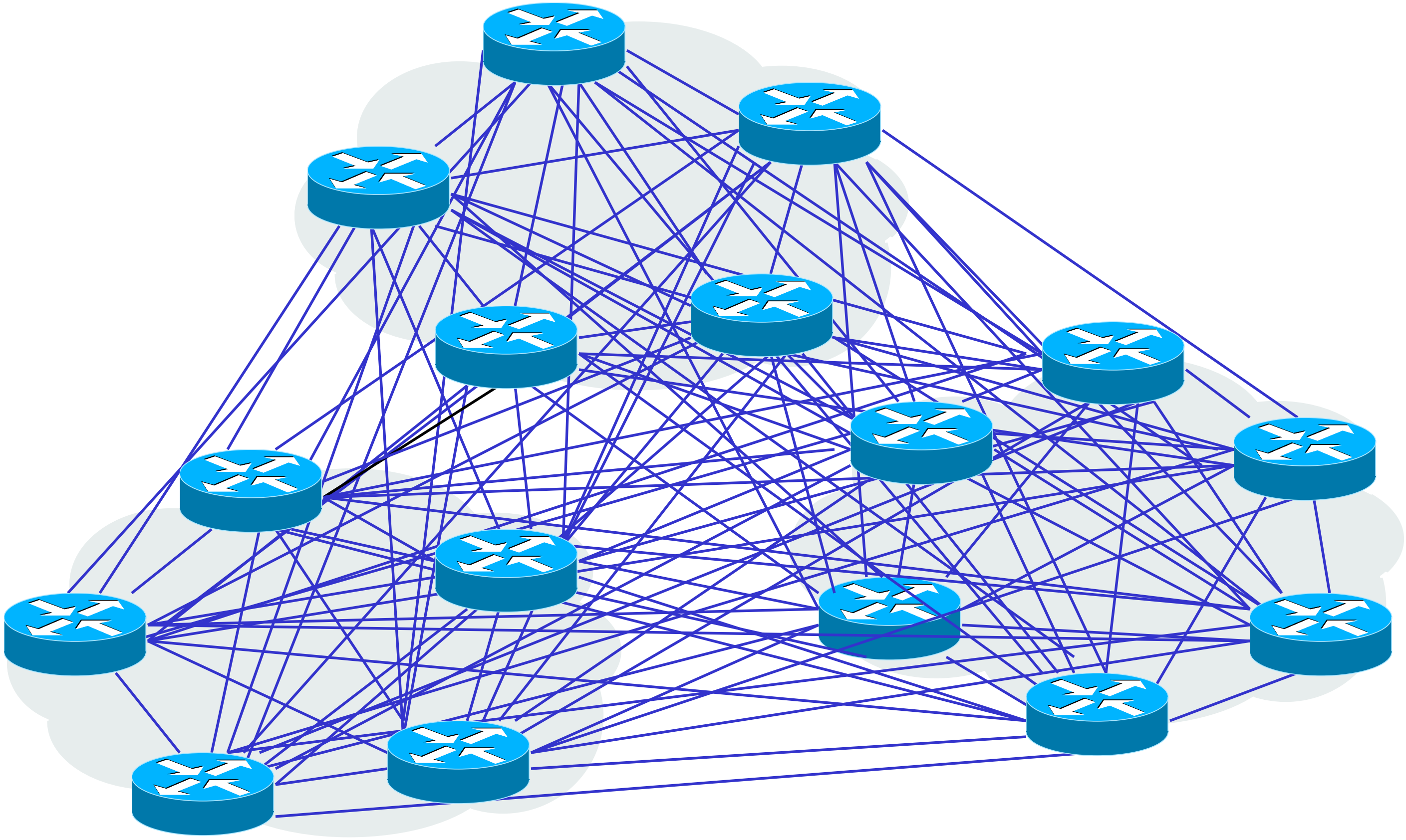
eBGP Rules

- 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

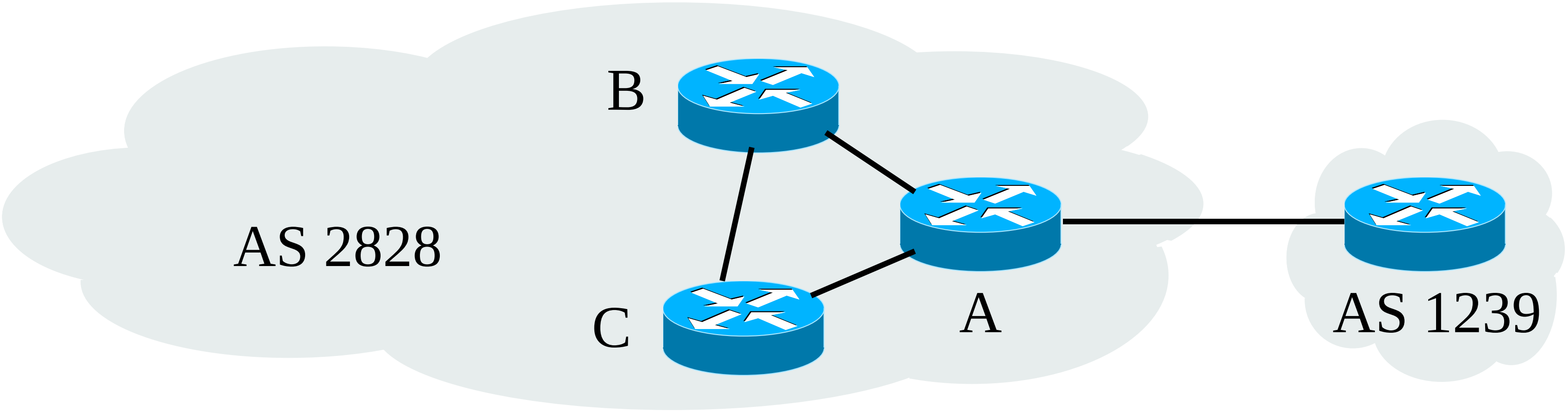
- Can talk to routers many hops away by default.
- Can only send routes it “injects”, or routes heard DIRECTLY from an external peer.
- Thus, requires a FULL mesh.
- Sends all attributes.
- Leaves the as-path attribute alone.
- Doesn’t touch the “next hop” attribute.

Logical view of 16 routers, fully meshed



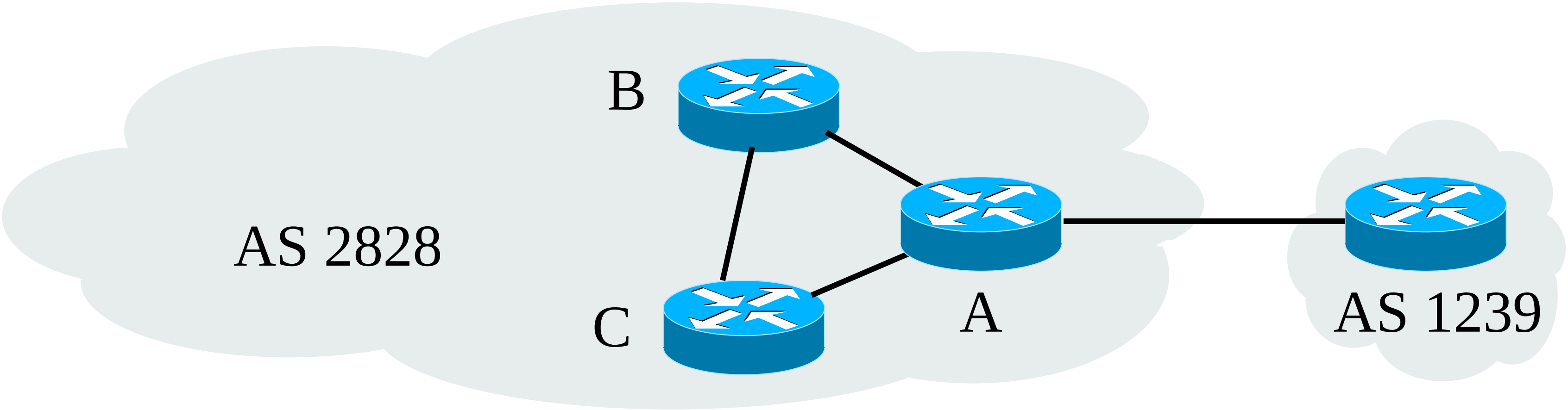
iBGP Restriction (1)

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



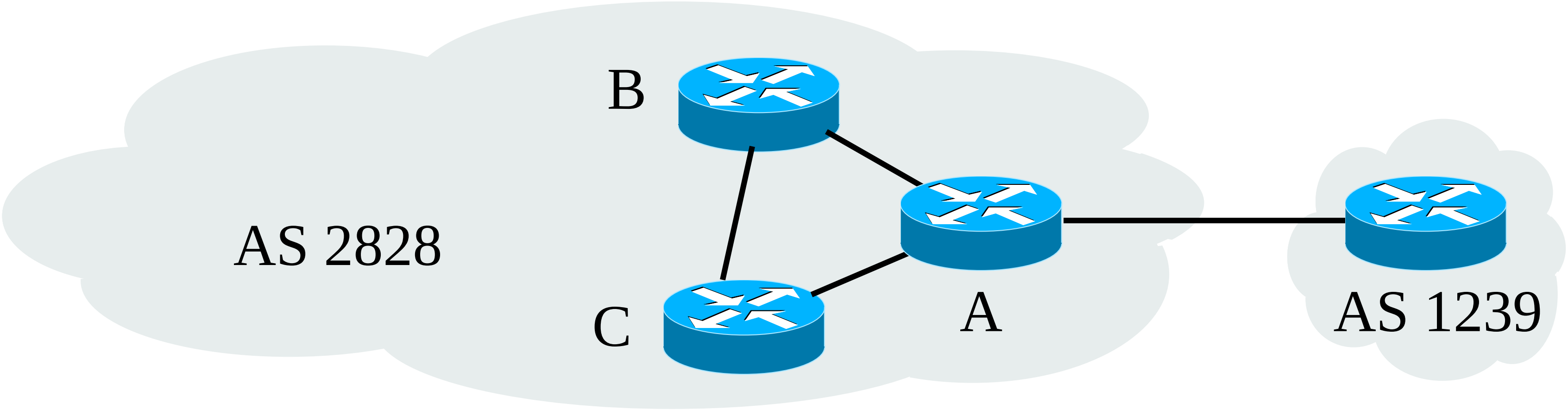
iBGP Restriction (2)

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 table.



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... (we are skipping this discussion for BGP 101)

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, as you can insert routes in a distributed fashion.
  - Makes scaling easier when you have to support multi-homed customers

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:
  - “prefix 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

Prefix List (1)

- Per neighbor access list applied to BGP routes
- Inbound or outbound
- Based upon IP prefixes

Prefx List (2)

```
 router bgp 3847
  neighbor 207.240.8.246 remote-as 8130
  neighbor  207.240.8.246 prefix-list mine out
  neighbor 207.240.8.246 prefix-list no-bogons in

ip prefix-list mine seq 100 permit 207.240.10.0/23 ge 24
! explicit deny if not specified

ip prefix-list no-bogons seq 100 deny 10.0.0.0 0/8 ge 8
ip prefix-list no-bogons seq 105 deny 127.0.0.0/8 ge 8
…
ip prefix-list no-bogons seq 105 deny 192.168.0.0/16 ge 16
ip prefix-list no-bogons seq 105 deny 224.0.0.0/8 ge 8
ip prefix-list no-bogons seq 105 deny 207.240.10.0/23 ge 23
prefix-list no-bogons  seq 200 permit 0.0.0.0/0 ge 24
```

AS-PATH Access 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 prefix-list mine-only out
 neighbor 207.240.10.100 prefix-list no-bogons in
 neighbor 207.240.10.100 filter-list 10 in

ip as-path access-list 10 permit ^$$
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.

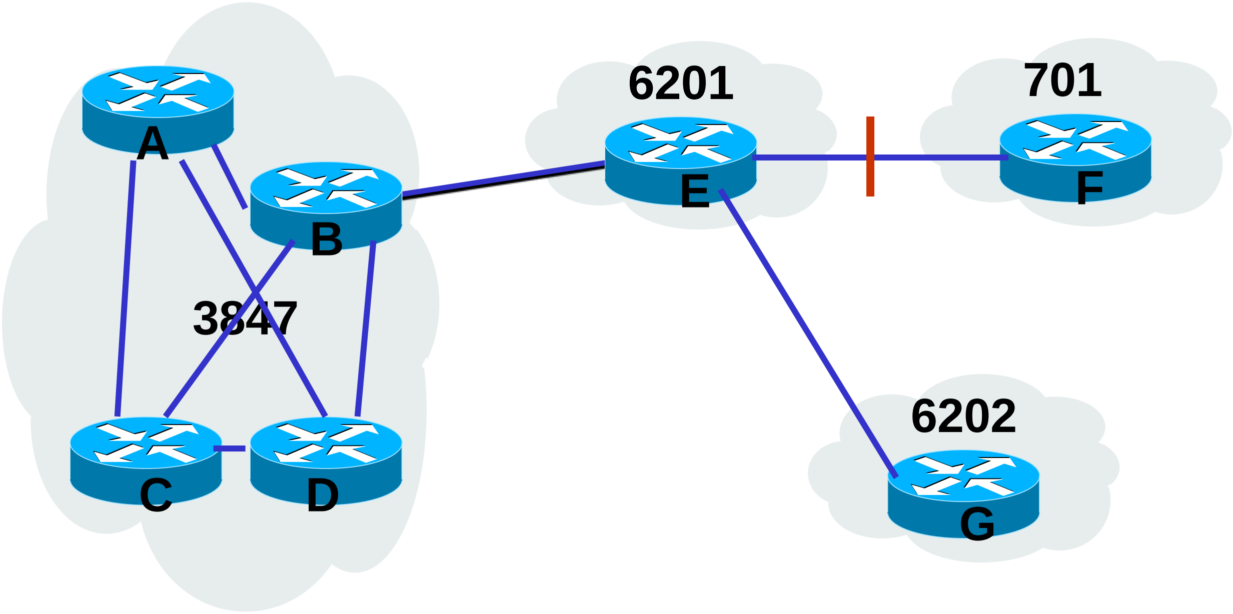
Cisco Regular Expressions (3)

[] 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 .*
```

Simple AS-Path ACLs

- When starting, make at least 3 access-lists you keep handy:
  - Permit all
  - Deny all
  - Permit only your own routes (and those of your downstream customers)

```
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 satements]

[repeat with unique sequence
 numbers as needed]
```

Route Maps (3)

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 (4)

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

Preferring Routes: Outbound

- The easiest way is to match on AS-PATH and prepend or use local-pref.
- To depref:

```
ip as access 50 permit _701_

! the first route-map only works w/ a 4436 peer
route-map depref-701
  match as 50
  set as pre 4436

route-map also-depref-701
  match as 50
  set local 90
```

Hardware
for
BGP

Router Vendors (1)

- Cisco
- Juniper
- PC router w/Unix and zebra/quagga/gated
- Anything else is… Adventurous for the beginner but Foundry and Force10 work for simple configs.

Router Vendors (2)

- Cisco and Juniper rule the BGP world. At least you’ll have the same bugs as everyone else.
- But for a small network, servers as routers can work to 3-5 gigabits. Get a pair of Nehalem 1us, do BGP, VRRP, OSPF, etc.
- Foundry and Force10 are annoying by being ‘mostly’ Cisco and have some gotchas but work reasonably.

Products

- The most common small provider router is probably the Cisco Catalyst 6500/7600. For < $20k you can get a router with 2 x GigE and add 48xGigE or 4x10GigE cards for < $10k per. Typical hosting and regional provider router.
- The Juniper MX is a bit more expensive and probably not a ‘starter’ router unless you’re a telco.
- The Foundry RX and MLX/XMR series are lower-end alternatives for full BGP routing.

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?
- Verizon’s lawyers show up at your doors and you look like an idiot.

Warning - I am Blackholio (2)

- What happens if you have a GigE to Sprint and a GigE 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 GigE.
- 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.

Logistics of becoming Multihomed

Multihoming Logistics

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

Multihoming Example

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 holdup routes

```
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 start by configuring the session to be shut down.
- 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 nLayer
 neigh 207.106.2.45 remote-as 4436
 neigh 207.106.2.45 next-hop-self
 neigh 207.106.2.45 version 4
 neigh 207.106.2.45 prefix-list mine-only out
 neigh 207.106.2.45 prefix-list no-bogons in
 neigh 207.106.2.45 filter 3 out
 neigh 207.106.2.45 filter 1 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 prefix-list no-bogons in
 neigh 10.40.4.81 prefix-list mine-only out
 neigh 10.40.4.81 filter 3 out
 neigh 10.40.4.81 filter 1 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.

Preferring Routes

Preferring Routes: Outbound

- The easiest way is to match on AS-PATH and prepend or use local-pref.
- To depref:

```
ip as access 50 permit _701_

! the first route-map only works w/ a 4436 peer
route-map depref-701
  match as 50
  set as pre 4436

route-map also-depref-701
  match as 50
  set local 90
```

Preferring Routes: Inbound

- Controlling inbound is less… subtle. You can't get down to the granularity of making one remote prefix get back to you via a different path.
- First method: Advertise more specific only to one provider. This is a bit rude to the Internet routing table as a whole and is still very brute force, but it is a widely-used practice.

Preferring Routes: More Specific

- To *also* advertise a /24 inside your /23 to nLayer:

```
ip prefix-list us-plus-specific seq 10 permit 207.106.8.0/23
ip- prefix-list us-plus-specific seq 20 permit 207.106.8.0/24
Ip prefix-list us-only seq 10 permit 207.106.8.0/23

router bgp 7007
  ! <snippets>
  neigh 198.186.9.3 remote-as 4436
  neigh 198.186.9.3 prefix-list us-plus-specific out
  neigh 209.209.209.3 remote-as 701
  neigh 209.209.209.3 prefix-list us-only out
  ! <plus other filters…>
```

Preferring Routes: Communities

- The second way to influence inbound traffic is BGP communities (covered in BGP 102).
- This will depend on your provider – each provider makes a system of communities that they parse.
- These communities are just numbers that go on the routes as you send them to your upstream.
- That say things like “Prepend once to 701” or “Prefer this route at another location if you hear it from me anywhere else”.

IOS Commands

IOS Commands

- sho ip bgp summ
- sho ip bgp <ip>
- sho ip bgp neigh <neighip> advertised
- clear ip bgp neigh <neighip> soft in/out
- Capturing the routing table with ‘ter len 0’ and ‘sho ip bgp’ (into a Unix ‘script’).

Best Practices
(covered in BGP 102)

Best Practices (1)

- Of course, filter – don’t advertise too much; block RFC1918 space and your routes inbound.
- BGP TTL ‘hack’
- Use next-hop-self and set BGP version to 4
- BGP Passwords
- Peering between loopbacks
- Enable soft-reconfig

Best Practices (2)

- Using peer-groups where possible
- Configuring for route-reflectors
- Make sure you pref internal routes over BGP
- Don’t use flap damping until you understand it
- If in doubt, use max-prefix
- Log BGP session state change (bgp log-neighbor-changes)
- If you use private ASNs internally, use ‘remote-private-as’ to external peers