# Introduction to BGP

William Waites
wwaites@tardis.ed.ac.uk

HUBS AS60241
&
School of Informatics
University of Edinburgh

September 16th 2015

# What is BGP-4?

- Obvious acronym expansion: *Border Gateway Protocol 4**
- It is for exchanging routing information between Autonomous Systems (AS)
- It is not about hop-by hop routing, it is
  - a Path-Vector protocol concerned with macroscopic AS-path
  - not a Distance-Vector protocol (like RIP, OSPF, etc) concerned with microscopic links
- Routes are announced using update messages from one AS to another
- Routes are similarly withdrawn by the same mechanism
- Updates about the route are forwarded onwards

*Rekhter, Y., Li, T., and Hares, S. (2005). RFC4271: A border gateway protocol 4 (BGP-4)

# Routing vs. Forwarding

Forwarding: when you receive a packet, looking for the best match for the destination in a table, and sending it on to the next hop.

Routing: deciding what to put in that table.

- Information used for forwarding is held in the Forwarding Information Base (FIB)
- Information used for routing is held in the Routing Information Base (RIB)
- BGP is a way to populate the RIB
- The FIB is made from the best entries in the RIB

# What goes into the FIB? (idealisation)

|   | Network | Netmask | Gateway |
|---|---|---|---|
| 1 | 192.0.2.0 | 255.255.255.240 | ether0 |
| 2 | 192.0.2.16 | 255.255.255.240 | 192.0.2.1 |
| 3 | 192.0.2.0 | 255.255.255.0 | 192.0.2.2 |

Lookup of a destination address to find the gateway is *always* done on a most specific basis (radix tree). For example,

lookup(192.0.2.18) → 192.0.2.1

lookup(192.0.2.1) → ether0

lookup(192.0.2.36) → 192.0.2.2

lookup(192.0.2.2) → ether0

# Populating the FIB

There are many ways that routing information can get from the RIB into the FIB:

- IP address configured on an interface
- Static route put in by hand
- A protocol like OSPF or RIP *within* an AS
- BGP from *outwith* an AS (eBGP)
- BGP from *within* an AS (iBGP)

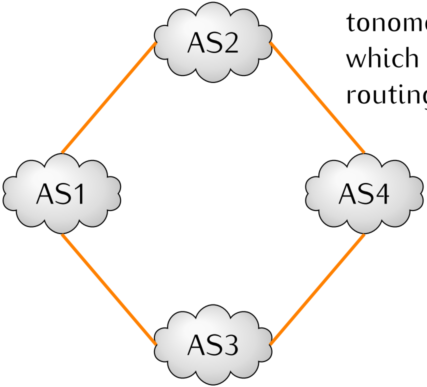
**Admin Distance**

| | |
|---|---|
| Connected | 0 |
| Static | 1 |
| eBGP | 20 |
| OSPF | 110 |
| RIP | 120 |
| iBGP | 200 |
| Unknown | 255 |

Each of these routing information sources has an administrative distance. For equal routes, lowest distance wins. Can be overridden if necessary.

But we haven't talked about how BGP works!

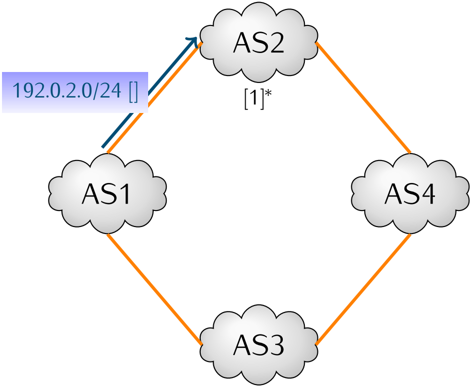
# BGP Update Messages



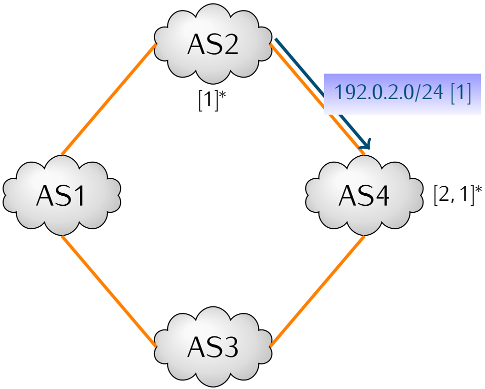
We start with four autonomous systems, none of which have any external routing information.

The network 192.0.2.0/24 is in AS1, and it is about to send an update announcing this fact. . .

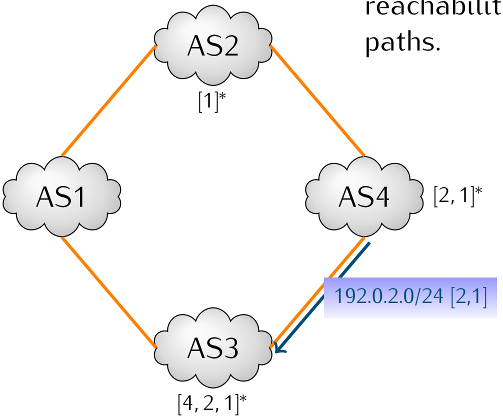
# BGP Update Messages



# BGP Update Messages

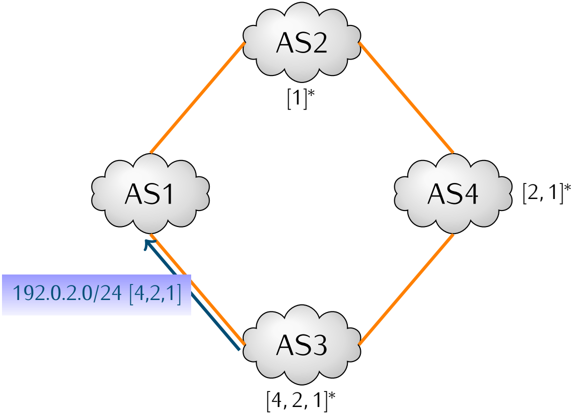


# BGP Update Messages



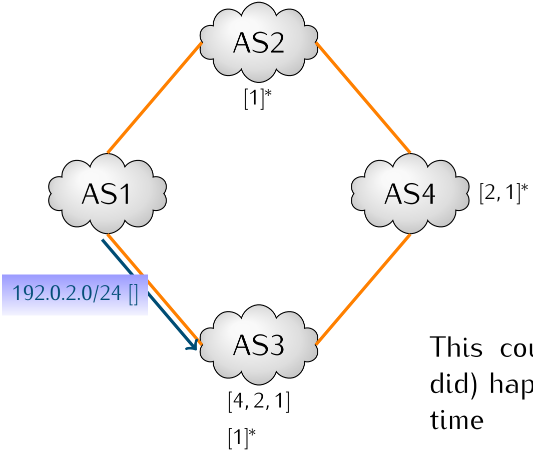
At this stage we have global reachability, but not optimal paths.

# BGP Update Messages



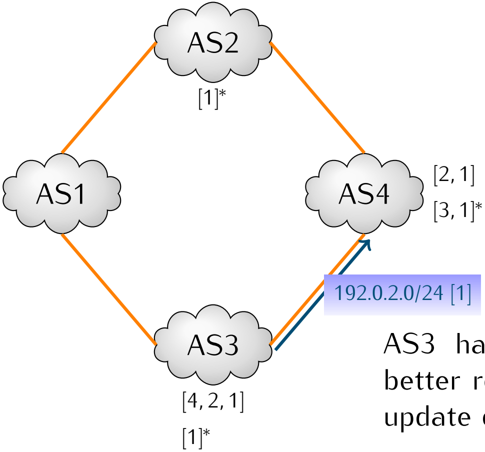
AS1 ignores the update because it contains its own ASN.

# BGP Update Messages



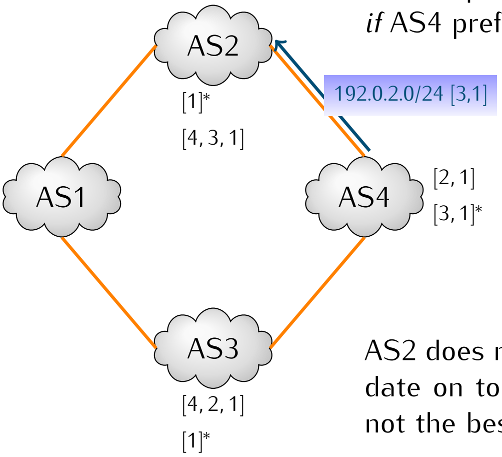
This could (and probably did) happen at any earlier time

# BGP Update Messages



AS3 has now learned a better route, so sends the update on

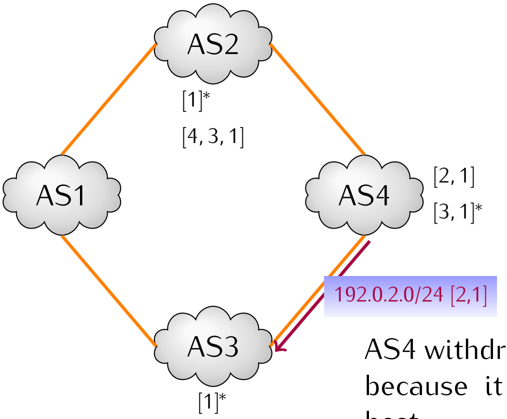
# BGP Update Messages



This step happens *if and only if* AS4 prefers the [3, 1] path.

AS2 does not forward the update on to AS1 because it is not the best route.

# BGP Update Messages



AS4 withdraws the [2, 1] route because it is no longer the best.

# BGP Route Selection

We saw that update messages carry the AS-path for a network, and AS4 decided that one path was better than another. How does BGP decide which updates to pass along? Routes actually have several attributes to enable this choice:

- Weight (highest)
- Local preference (highest)
- Prefer locally originated
- AS-path length (lowest)
- Multi-Exit Descriminator(MED) (lowest)
- Origin (igp, egp or incomplete)
- Originator router ID
- Neighbour address (lowest)

# BGP Route Selection (cont'd)

Not all route attributes have the same propagation characteristics:

| | |
|---|---|
| Weight | local to a router, not propagated |
| Local preference | local to an AS, not propagated beyond |
| AS-path | global, incrementally constructed |
| MED | global, sometimes overwritten |
| Origin | global |
| Originator ID | global |
| Neighbour address | local to an AS, sometimes overwritten |

Also BGP communities are (nearly) arbitrary tags with opaque semantics that are propagated across AS boundaries.

# Filters and Policy

BGP filters allow the setting of policy. They allow to:

- Accept or reject update according to attributes present
- Set attributes like weight or local preference to govern the local decision (outbound traffic)
- Set attributes like MED or communities to influence a neighbour AS's decision (inbound traffic)

These are *not* filters like you find in a firewall. Firewalls have packet filters that affect *forwarding* and say if a packet is allowed or not. BGP filters affect how the forwarding table is *created*. Read this again, it has been a cause of confusion.

# Policy Examples

- Set local preference on received routes to prefer a primary link over a backup link for outbound traffic
- Set the MED to cause the neighbour to prefer a primary link over a backup link for inbound traffic
- Add a community to tell the neighbour to forward updates only to particular peers e.g. at an exchange point
- Send updates to a peer (e.g. at an exchange point) if and only if a particular community is present (or absent)

It happens that these are exactly the policies that HUBS members should use. Adding the community 60241:8714 will suppress forwarding updates to IX Scotland (but why would you want to do that?).

# Other methods for influencing route selection

Prepending
: Add multiple copies of (your own, not someone else's!) AS number to the head of the path. Works when the choice is made on path length. Tends to be work reliably two or more AS hops out. Can also not work if a foreign AS applies local preference or weight where their policy conflicts with your intent.

Deaggregation
: Announce more specific networks. These are always preferred because of the *forwarding* (not the routing!) algorithm. Works reliably if you have enough address space to do it, exposes some information about internal network structure to the world. Increases resource (chiefly RAM) requirements.

# iBGP vs eBGP

What's the difference?

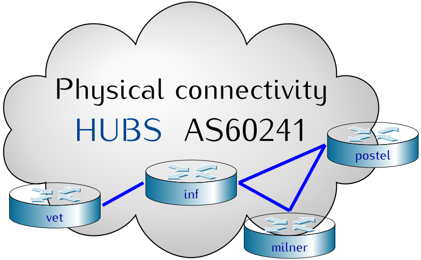
eBGP happens when peers have different AS numbers

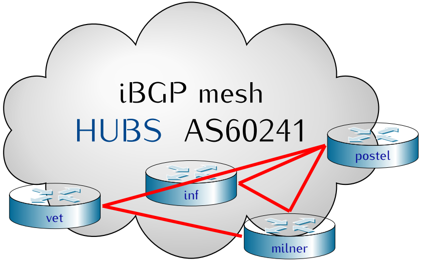
iBGP happens when the AS numbers are the same

Other than administrative distance, there are important differences between internal and external BGP.

- Next hop address is *preserved*. This means the next hop for a route may not be directly connected and has to be found by other means. (remember the FIB idealisation?)
- *Only* updates from eBGP peers are forwarded to iBGP peers. This means that either all BGP routers within an AS must peer with each other – $\mathcal{O}(n^2)$ configuration needed – or use route-reflectors – just $\mathcal{O}(n)$
- Local preference and AS path are unchanged with iBGP

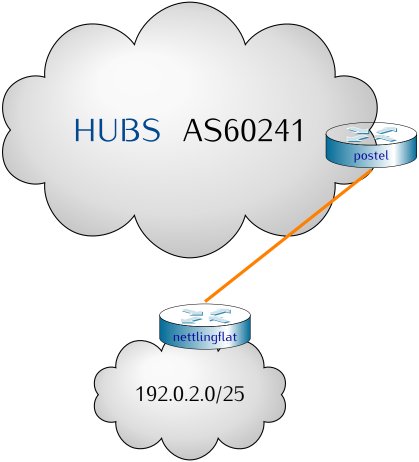
# The AS60241 Transit Path





- External connections at Easter Bush (`vet`) and Summerhall (`postel` and `milner`)
- `postel` and `milner` are route reflectors
- Connections into Summerhall have redundant sessions
- HUBS honours MED so members can signal inbound path preference
- Usually announces a default route and member routes to downstream peers

# Heriot – Nettlingflat

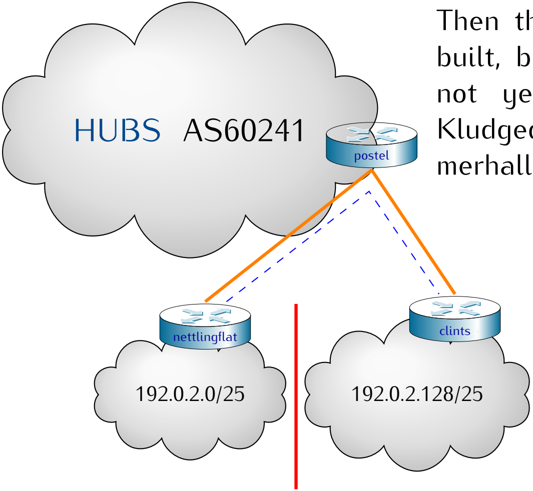


The Heriot network is interesting because it grew up along side HUBS . It started with a single link to Summerhall (via) Appleton Tower, and a single, very simple BGP session. No sophisticated policy was required because there was only a single link.

# Heriot – Nettlingflat: Configuration

```
/routing filter
  add chain=nef-addrs prefix=192.0.2.0/25 action=accept
  add chain=nef-addrs action=reject
  add chain=hubs-main-in action=accept
  add chain=hubs-main-out match-chain=nef-addrs \
      action=accept
  add chain=hubs-main-out action=reject
/routing bgp instance
  set default router-id=192.0.2.0 as=65529
/routing bgp network
  add network=192.0.2.0/25 synchronize=no
/routing bgp peer
  add name=hubs-main \
      remote-address=198.51.100.1 remote-as=60241 \
      in-filter=hubs-main-in out-filter=hubs-main-out
```

# Heriot – Clints



Then the Clint's Hill mast was built, but Heriot's network was not yet internally connected. Kludged with a VLAN via Summerhall.

But this raised a policy question: how to ensure traffic went on the direct path and not over the VLAN, e.g., Summerhall → Nettlingflat → Summerhall → Clint's Hill?

# Heriot – Nettlingflat & Clints: Policy

Let's be more specific about the desired policy.

- Traffic to Nettlingflat should go direct
- Traffic to Clints should go direct
- Traffic between Clints and Nettlingflat traverses the VLAN
- If the Nettlingflat session with HUBS is down, traffic should go to Clints and then via the VLAN, and vice-versa

An iBGP session between Clints and Nettlingflat (via the VLAN) is required, otherwise there would be no path between Clints and Nettlingflat. (Why?)

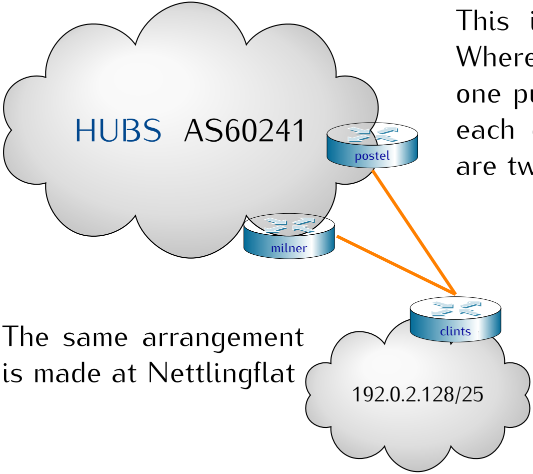
The solution is to announce the local /25 *and* 192.0.2.0/24 to HUBS because the /25 is preferred, if it exists, and HUBS will see the /24 supernet via both paths. (cf Deaggregation)

# Heriot – Nettlingflat & Clints: Configuration

```
/routing filter
  add chain=nef-addrs prefix=192.0.2.0/24 action=accept
  ...
/routing bgp network
  add network=192.0.2.0/24 synchronize=no
/routing bgp peer
  add name=clints \
      remote-address=192.0.2.128 remote-as=65529
```

Mutatis mutandis for Clints.

# HUBS Gains Redundant Routers



This is something different. Where before there was only one putatively “best” path to each destination, now there are two.

The same arrangement is made at Nettlingflat

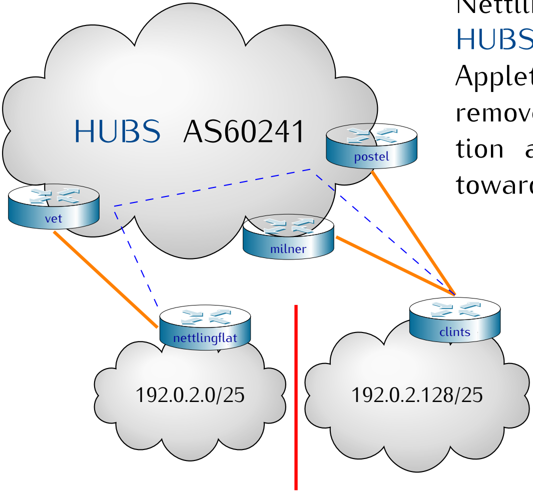
So we use local preference to select outbound routes, and MED to influence inbound routes.

# Redundant Links: Configuration

```
/routing filter
  add chain=hubs-main-in set-bgp-local-pref=200 \
      action=accept
  add chain=hubs-backup-in set-bgp-local-pref=100 \
      action=accept
  add chain=hubs-main-out match-chain=clints-addrs \
      set-bgp-med=10 action=accept
  add chain=hubs-main-out action=reject
  add chain=hubs-backup-out match-chain=clints-addrs \
      set-bgp-med=20 action=accept
  add chain=hubs-backup-out action=reject
```

Replacing any previous filters of the same name. Also note that there is no reason why the main and backup sessions *have* to be on the same router!

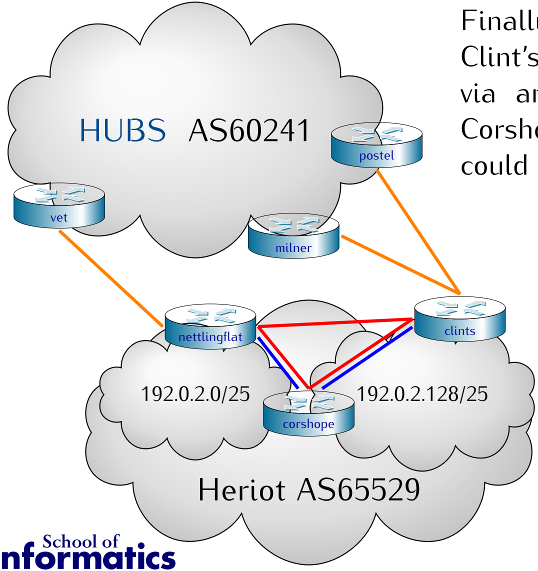
# Flight from Appleton Tower



Nettlingflat had reached HUBS through a radio link to Appleton Tower. We had to remove this due to construction and re-orient the link towards Easter Bush.

Nothing substantial changes, save that Nettlingflat no longer has redundant sessions, and the pesky VLAN to paper over a network partition takes a longer path.

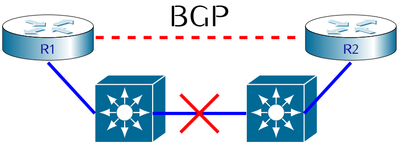
# Unification of Heriot



Finally Nettlingflat and Clint’s Hill were connected via an intermediate site at Corshope and the VLAN could be retired.

This implies completing the iBGP mesh since every router between iBGP speakers must also be part of iBGP (Why?). Any more and we’d want a route reflector.

# One way BGP fails



- BGP is a TCP protocol → TCP timeout?
- Timers: keepalive, hold time ≈ 90 seconds
- No direct way of knowing this link has failed!
- Pretty long wait until we notice and re-routing happens
- Solution: Bidirectional Forwarding Detection (BFD)*
- Session tied to BFD “Hello” protocol between R1 and R2
- Failure detection in $<$ 1 second
- Requires explicit configuration on both ends:

```
    /routing bgp peer ... use-bfd=yes
```

- Works for OSPF too!

*Katz, D. and Ward, D. (2010). RFC5880 bidirectional forwarding detection

# Summary: HUBS Member Interconnect Requirements

- One or more BGP sessions in one or more places
- HUBS advertises a default route and member routes
- Member advertises only routes it is allowed to
- HUBS only[*] accepts networks larger than /29[†]
- If $> 1$ sessions, member internal network must be coherent
- Use local preference to pick best outbound path
- Use MED to tell HUBS which inbound path is best
- If that's not enough, deaggregate carefully
- BFD for fast failure detection supported on request

[*] Except in special, pre-aggreed circumstances

[†] For IPv6 the smallest accepted network is /48

# Questions to Ponder

- Why is it a good idea to always use loopback interfaces and not physical interfaces for iBGP?
- What is the role of OSPF or similar intra-AS protocol and how does it interact with iBGP? (Hint, think about the lookups in the FIB idealisation slide).
- What could happen if a router accepted an announcement from an eBGP peer that contained its own AS in the path?

# Epilogue

- Everything in this presentation applies equally to IPv6 networks. Mind that the BGP router ID remains a 32-bit number and is not really an IPv4 address even though it is conventially represented like one.
- The IP networks used in these slides, 192.0.2.0/24 and 198.51.100.0/24, are special use networks reserved for documentation[*].
- The autonomous system number used by Heriot, 65529, is from the range reserved for private (i.e. not to be seen on the global Internet) use in RFC6996[†].

[*] Cotton, M. (2010). RFC5737: Special use IPv4 addresses

[†] Mitchell, J. (2013). RFC6996: Autonomous system (AS) reservation for private use