Cisco Systems

Service Provider Multihoming

BGP Traffic Engineering

Service Provider Multihoming

- Previous examples dealt with loadsharing inbound traffic
  - Of primary concern at Internet edge
  - What about outbound traffic?
- Transit ISPs strive to balance traffic flows in both directions
  - Balance link utilisation
  - Try and keep most traffic flows symmetric
  - Some edge ISPs try and do this too
- The original "Traffic Engineering"

Service Provider Multihoming

- Balancing outbound traffic requires inbound routing information
  - Common solution is "full routing table"
  - Rarely necessary
    - Why use the "routing mallet" to try solve loadsharing problems?
  - "Keep It Simple" is often easier (and $$$ cheaper) than carrying N-copies of the full routing table

Service Provider Multihoming MYTHS!!

- Common MYTHS
- 1: You need the full routing table to multihome
  - People who sell router memory would like you to believe this
  - Only true if you are a transit provider
  - Full routing table can be a significant hindrance to multihoming
- 2: You need a BIG router to multihome
  - Router size is related to data rates, not running BGP
  - In reality, to multihome, your router needs to:
    - Have two interfaces,
    - Be able to talk BGP to at least two peers,
    - Be able to handle BGP attributes,
    - Handle at least one prefix
- 3: BGP is complex
  - In the wrong hands, yes it can be! Keep it Simple!

Service Provider Multihoming: Some Strategies

- Take the prefixes you need to aid traffic engineering
  - Look at NetFlow data for popular sites
- Prefixes originated by your immediate neighbours and their neighbours will do more to aid load balancing than prefixes from ASNs many hops away
  - Concentrate on local destinations
- Use default routing as much as possible
  - Or use the full routing table with care

Service Provider Multihoming

- Examples
  - One upstream, one local peer
  - One upstream, local exchange point
  - Two upstreams, one local peer
  - Tier-1 and regional upstreams, with local peers
- Require BGP and a public ASN
- Examples assume that the local network has their own /19 address block

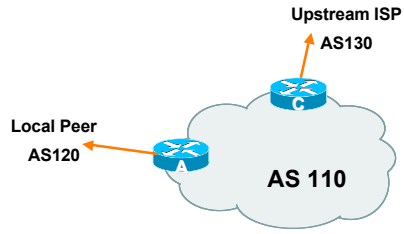
Service Provider Multihoming

One Upstream, One local peer

One Upstream, One Local Peer

- Very common situation in many regions of the Internet
- Connect to upstream transit provider to see the "Internet"
- Connect to the local competition so that local traffic stays local
  - Saves spending valuable $ on upstream transit costs for local traffic

One Upstream, One Local Peer



One Upstream, One Local Peer

- Announce /19 aggregate on each link
- Accept default route only from upstream
  - Either 0.0.0.0/0 or a network which can be used as default
- Accept all routes from local peer

One Upstream, One Local Peer

- Router A Configuration

Prefix filters inbound

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.2 remote-as 120
 neighbor 122.102.10.2 prefix-list my-block out
 neighbor 122.102.10.2 prefix-list AS120-peer in
!
ip prefix-list AS120-peer permit 122.5.16.0/19
ip prefix-list AS120-peer permit 121.240.0.0/20
ip prefix-list my-block permit 121.10.0.0/19
!
ip route 121.10.0.0 255.255.224.0 null0
```


One Upstream, One Local Peer

- Router A – Alternative Configuration

AS Path filters – more "trusting"

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.2 remote-as 120
 neighbor 122.102.10.2 prefix-list my-block out
 neighbor 122.102.10.2 filter-list 10 in
!
ip as-path access-list 10 permit ^(120_)+$
!
ip prefix-list my-block permit 121.10.0.0/19
!
ip route 121.10.0.0 255.255.224.0 null0
```

One Upstream, One Local Peer

- Router C Configuration

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.1 remote-as 130
 neighbor 122.102.10.1 prefix-list default in
 neighbor 122.102.10.1 prefix-list my-block out
!
ip prefix-list my-block permit 121.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 121.10.0.0 255.255.224.0 null0
```


One Upstream, One Local Peer

- Two configurations possible for Router A
  - Filter-lists assume peer knows what they are doing
  - Prefix-list higher maintenance, but safer
  - Some ISPs use both
- Local traffic goes to and from local peer, everything else goes to upstream

CISCO SYSTEMS

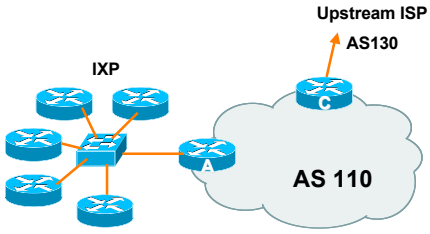
Service Provider Multihoming

One Upstream, Local Exchange Point

One Upstream, Local Exchange Point

- Very common situation in many regions of the Internet
- Connect to upstream transit provider to see the "Internet"
- Connect to the local Internet Exchange Point so that local traffic stays local
  - Saves spending valuable $ on upstream transit costs for local traffic

One Upstream, Local Exchange Point



One Upstream, Local Exchange Point

- Announce /19 aggregate to every neighbouring AS
- Accept default route only from upstream
  - Either 0.0.0.0/0 or a network which can be used as default
- Accept all routes originated by IXP peers

One Upstream, Local Exchange Point

- **Router A Configuration**

```
interface fastethernet 0/0
 description Exchange Point LAN
 ip address 120.5.10.1 mask 255.255.255.224
 ip verify unicast reverse-path
!
router bgp 110
 neighbor ixp-peers peer-group
 neighbor ixp-peers prefix-list my-block out
 neighbor ixp-peers remove-private-AS
..next slide
```


One Upstream, Local Exchange Point

```
neighbor 120.5.10.2 remote-as 100
neighbor 120.5.10.2 peer-group ixp-peers
neighbor 120.5.10.2 prefix-list peer100 in
neighbor 120.5.10.3 remote-as 101
neighbor 120.5.10.3 peer-group ixp-peers
neighbor 120.5.10.3 prefix-list peer101 in
neighbor 120.5.10.4 remote-as 102
neighbor 120.5.10.4 peer-group ixp-peers
neighbor 120.5.10.4 prefix-list peer102 in
neighbor 120.5.10.5 remote-as 103
neighbor 120.5.10.5 peer-group ixp-peers
neighbor 120.5.10.5 prefix-list peer103 in
..next slide
```


One Upstream, Local Exchange Point

```
!
ip prefix-list my-block permit 121.10.0.0/19
ip prefix-list peer100 permit 122.0.0.0/19
ip prefix-list peer101 permit 122.30.0.0/19
ip prefix-list peer102 permit 122.12.0.0/19
ip prefix-list peer103 permit 122.18.128.0/19
!
```

- **Note that Router A does not generate the aggregate for AS110**
  - **If Router A becomes disconnected from backbone, then the aggregate is no longer announced to the IX**
  - **BGP failover works as expected**

One Upstream, Local Exchange Point

- **Router C Configuration**

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.1 remote-as 130
 neighbor 122.102.10.1 prefix-list default in
 neighbor 122.102.10.1 prefix-list my-block out
!
ip prefix-list my-block permit 121.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 121.10.0.0 255.255.224.0 null0
```


One Upstream, Local Exchange Point

- **Note Router A configuration**
  - **Prefix-list higher maintenance, but safer**
  - **uRPF on the IX facing interface**
  - **No generation of AS110 aggregate**
- **IXP traffic goes to and from local IXP, everything else goes to upstream**

Aside: Configuration Recommendations (1)

- **Private Peers**
  - **The peering ISPs exchange prefixes they originate**
  - **Sometimes they exchange prefixes from neighbouring ASNs too**
- **Be aware that the private peer eBGP router should carry only the prefixes you want the private peer to receive**
  - **Otherwise they could point a default route to you and unintentionally transit your backbone**

Aside: Configuration Recommendations (2)

- IXP peers
  - The peering ISPs at the IXP exchange prefixes they originate
  - Sometimes they exchange prefixes from neighbouring ASNs too
- Be aware that the IXP border router should carry only the prefixes you want the IXP peers to receive and the destinations you want them to be able to reach
  - Otherwise they could point a default route to you and unintentionally transit your backbone
- If IXP router is at IX, and distant from your backbone
  - Don't originate your address block at your IXP router

Cisco Systems

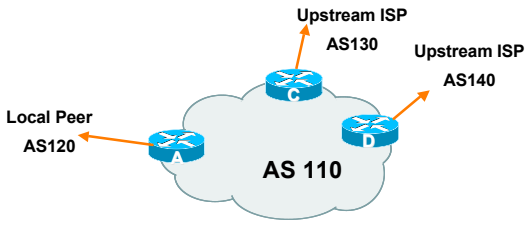
Service Provider Multihoming

Two Upstreams, One local peer

Two Upstreams, One Local Peer

- Connect to both upstream transit providers to see the "Internet"
  - Provides external redundancy and diversity – the reason to multihome
- Connect to the local peer so that local traffic stays local
  - Saves spending valuable $ on upstream transit costs for local traffic

Two Upstreams, One Local Peer



Two Upstreams, One Local Peer

- Announce /19 aggregate on each link
- Accept default route only from upstreams
  - Either 0.0.0.0/0 or a network which can be used as default
- Accept all routes from local peer

Two Upstreams, One Local Peer

- Router A
  - Same routing configuration as in example with one upstream and one local peer
  - Same hardware configuration

Two Upstreams, One Local Peer

- Router C Configuration

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.1 remote-as 130
 neighbor 122.102.10.1 prefix-list default in
 neighbor 122.102.10.1 prefix-list my-block out
!
ip prefix-list my-block permit 121.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 121.10.0.0 255.255.224.0 null0
```


Two Upstreams, One Local Peer

- Router D Configuration

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.5 remote-as 140
 neighbor 122.102.10.5 prefix-list default in
 neighbor 122.102.10.5 prefix-list my-block out
!
ip prefix-list my-block permit 121.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 121.10.0.0 255.255.224.0 null0
```


Two Upstreams, One Local Peer

- This is the simple configuration for Router C and D
- Traffic out to the two upstreams will take nearest exit
  - Inexpensive routers required
  - This is not useful in practice especially for international links
  - Loadsharing needs to be better

Two Upstreams, One Local Peer

- Better configuration options:
  - Accept full routing from both upstreams
    - Expensive & unnecessary!
  - Accept default from one upstream and some routes from the other upstream
    - The way to go!

Two Upstreams, One Local Peer Full Routes

- Router C Configuration

Allow all prefixes in apart from RFC1918 and friends

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.1 remote-as 130
 neighbor 122.102.10.1 prefix-list rfc1918-deny in
 neighbor 122.102.10.1 prefix-list my-block out
 neighbor 122.102.10.1 route-map AS130-loadshare in
!
ip prefix-list my-block permit 121.10.0.0/19
! See www.cymru.com/Documents/bogon-list.html
! ...for "RFC1918 and friends" list
..next slide
```


Two Upstreams, One Local Peer Full Routes

```
ip route 121.10.0.0 255.255.224.0 null0
!
ip as-path access-list 10 permit ^(130_)+$
ip as-path access-list 10 permit ^(130_)+_[0-9]+$
!
route-map AS130-loadshare permit 10
 match ip as-path 10
 set local-preference 120
route-map AS130-loadshare permit 20
 set local-preference 80
!
```

Two Upstreams, One Local Peer
Full Routes

- Router D Configuration

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.5 remote-as 140
 neighbor 122.102.10.5 prefix-list rfc1918-deny in
 neighbor 122.102.10.5 prefix-list my-block out
!
ip prefix-list my-block permit 121.10.0.0/19
! See www.cymru.com/Documents/bogon-list.html
! ...for "RFC1918 and friends" list
```

Allow all prefixes in apart from RFC1918 and friends

Two Upstreams, One Local Peer
Full Routes

- Router C configuration:
  - Accept full routes from AS130
  - Tag prefixes originated by AS130 and AS130's neighbouring ASes with local preference 120
    - Traffic to those ASes will go over AS130 link
  - Remaining prefixes tagged with local preference of 80
    - Traffic to other all other ASes will go over the link to AS140
- Router D configuration same as Router C without the route-map

Two Upstreams, One Local Peer
Full Routes

- Full routes from upstreams
  - Expensive – needs lots of memory and CPU
  - Need to play preference games
  - Previous example is only an example – real life will need improved fine-tuning!
  - Previous example doesn't consider inbound traffic – see earlier in presentation for examples

Two Upstreams, One Local Peer
Partial Routes

- Strategy:
  - Ask one upstream for a default route
    - Easy to originate default towards a BGP neighbour
  - Ask other upstream for a full routing table
    - Then filter this routing table based on neighbouring ASN
    - E.g. want traffic to their neighbours to go over the link to that ASN
    - Most of what upstream sends is thrown away
    - Easier than asking the upstream to set up custom BGP filters for you

Two Upstreams, One Local Peer
Partial Routes

- Router C Configuration

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.1 remote-as 130
 neighbor 122.102.10.1 prefix-list rfc1918-nodef-deny in
 neighbor 122.102.10.1 prefix-list my-block out
 neighbor 122.102.10.1 filter-list 10 in
 neighbor 122.102.10.1 route-map tag-default-low in
!
..next slide
```

Allow all prefixes and default in; deny RFC1918 and friends

AS filter list filters prefixes based on origin ASN

Two Upstreams, One Local Peer
Partial Routes

```
ip prefix-list my-block permit 121.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 121.10.0.0 255.255.224.0 null0
!
ip as-path access-list 10 permit ^(130_)+$
ip as-path access-list 10 permit ^(130_)+_[0-9]+$
!
route-map tag-default-low permit 10
 match ip address prefix-list default
 set local-preference 80
route-map tag-default-low permit 20
!
```

Two Upstreams, One Local Peer Partial Routes

- Router D Configuration

```
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.5 remote-as 140
 neighbor 122.102.10.5 prefix-list default in
 neighbor 122.102.10.5 prefix-list my-block out
!
ip prefix-list my-block permit 121.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 121.10.0.0 255.255.224.0 null0
```


Two Upstreams, One Local Peer Partial Routes

- Router C configuration:
  - Accept full routes from AS130
    - (or get them to send less)
  - Filter ASNs so only AS130 and AS130's neighbouring ASes are accepted
  - Allow default, and set it to local preference 80
  - Traffic to those ASes will go over AS130 link
  - Traffic to other all other ASes will go over the link to AS140
  - If AS140 link fails, backup via AS130 – and vice-versa

Two Upstreams, One Local Peer Partial Routes

- Partial routes from upstreams
  - Not expensive – only carry the routes necessary for loadsharing
  - Need to filter on AS paths
  - Previous example is only an example – real life will need improved fine-tuning!
  - Previous example doesn't consider inbound traffic – see earlier in presentation for examples

Two Upstreams, One Local Peer

- When upstreams cannot or will not announce default route
  - Because of operational policy against using "default-originate" on BGP peering
  - Solution is to use IGP to propagate default from the edge/peering routers

Two Upstreams, One Local Peer Partial Routes

- Router C Configuration

```
router ospf 110
 default-information originate metric 30
 passive-interface Serial 0/0
!
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.1 remote-as 130
 neighbor 122.102.10.1 prefix-list rfc1918-deny in
 neighbor 122.102.10.1 prefix-list my-block out
 neighbor 122.102.10.1 filter-list 10 in
!
..next slide
```


Two Upstreams, One Local Peer Partial Routes

```
ip prefix-list my-block permit 121.10.0.0/19
! See www.cymru.com/Documents/bogon-list.html
! ...for "RFC1918 and friends" list
!
ip route 121.10.0.0 255.255.224.0 null0
ip route 0.0.0.0 0.0.0.0 serial 0/0 254
!
ip as-path access-list 10 permit ^(130_)+$
ip as-path access-list 10 permit ^(130_)+_[0-9]+$
!
```

Two Upstreams, One Local Peer Partial Routes

- **Router D Configuration**

```
router ospf 110
 default-information originate metric 10
 passive-interface Serial 0/0
!
router bgp 110
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.5 remote-as 140
 neighbor 122.102.10.5 prefix-list deny-all in
 neighbor 122.102.10.5 prefix-list my-block out
!
..next slide
```


Two Upstreams, One Local Peer Partial Routes

```
ip prefix-list deny-all deny 0.0.0.0/0 le 32
ip prefix-list my-block permit 121.10.0.0/19
!
ip route 121.10.0.0 255.255.224.0 null0
ip route 0.0.0.0 0.0.0.0 serial 0/0 254
!
```


Two Upstreams, One Local Peer Partial Routes

- Partial routes from upstreams
  - Use OSPF to determine outbound path
  - Router D default has metric 10 – primary outbound path
  - Router C default has metric 30 – backup outbound path
  - Serial interface goes down, static default is removed from routing table, OSPF default withdrawn

Cisco Systems

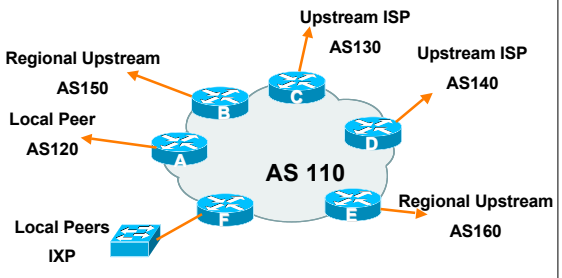
Service Provider Multihoming

Two Tier-1 upstreams, two regional upstreams, and local peers

Tier-1 & Regional Upstreams, Local Peers

- This is a complex example, bringing together all the concepts learned so far
- Connect to both upstream transit providers to see the "Internet"
  - Provides external redundancy and diversity – the reason to multihome
- Connect to regional upstreams
  - Hopefully a less expensive and lower latency view of the regional internet than is available through upstream transit provider
- Connect to private peers for local peering purposes
- Connect to the local Internet Exchange Point so that local traffic stays local
  - Saves spending valuable $ on upstream transit costs for local traffic

Tier-1 & Regional Upstreams, Local Peers



Tier-1 & Regional Upstreams, Local Peers

- Announce /19 aggregate on each link
- Accept partial/default routes from upstreams
  - For default, use 0.0.0.0/0 or a network which can be used as default
- Accept all routes from local peer
- Accept all partial routes from regional upstreams
- This is more complex, but a very typical scenario

Tier-1 & Regional Upstreams, Local Peers Detail

- Router A – local private peer
  - Accept all (local) routes
  - Local traffic stays local
  - Use prefix and/or AS-path filters
  - Use local preference (if needed)
- Router F – local IXP peering
  - Accept all (local) routes
  - Local traffic stays local
  - Use prefix and/or AS-path filters

Tier-1 & Regional Upstreams, Local Peers Detail

- Router B – regional upstream
  - They provide transit to Internet, but longer AS path than Tier-1s
  - Accept all regional routes from them
    - e.g. ^150_[0-9]+$
  - Ask them to send default, or send a network you can use as default
    - Set local pref on “default” to 60
  - Will provide backup to Internet only when direct Tier-1 links go down

Tier-1 & Regional Upstreams, Local Peers Detail

- Router E – regional upstream
  - They provide transit to Internet, but longer AS path than Tier-1s
  - Accept all regional routes from them
    - e.g. ^160_[0-9]+$
  - Ask them to send default, or send a network you can use as default
    - Set local pref on “default” to 70
  - Will provide backup to Internet only when direct Tier-1 links go down

Tier-1 & Regional Upstreams, Local Peers Detail

- Router C – first Tier-1
  - Accept all their customer and AS neighbour routes from them
    - e.g. ^130_[0-9]+$
  - Ask them to send default, or send a network you can use as default
    - Set local pref on “default” to 80
  - Will provide backup to Internet only when link to second Tier-1 goes down

Tier-1 & Regional Upstreams, Local Peers Detail

- Router D – second Tier-1
  - Ask them to send default, or send a network you can use as default
    - This has local preference 100 by default
  - All traffic without any more specific path will go out this way

Tier-1 & Regional Upstreams, Local Peers Summary

- Local traffic goes to local peer and IXP
- Regional traffic goes to two regional upstreams
- Everything else is shared between the two Tier-1s
- To modify loadsharing tweak what is heard from the two regionals and the first Tier-1
  - Best way is through modifying the AS-path filter

Tier-1 & Regional Upstreams, Local Peers

- What about outbound announcement strategy?
  - This is to determine incoming traffic flows
  - /19 aggregate must be announced to everyone!
  - /20 or /21 more specifics can be used to improve or modify loadsharing
  - See earlier for hints and ideas

Tier-1 & Regional Upstreams, Local Peers

- What about unequal circuit capacity?
  - AS-path filters are very useful
- What if upstream will only give me full routing table or nothing
  - AS-path and prefix filters are very useful

Service Provider Multihoming

BGP Traffic Engineering