BGP Multihoming Techniques

Philip Smith <pfs@cisco.com>

SANOG 12

6th-14th August 2008

Kathmandu

Presentation Slides

- Available on

  ftp://ftp-eng.cisco.com

  /pfs/seminars/SANOG12-Multihoming.pdf

  And on the SANOG website

- Feel free to ask questions any time

Preliminaries

- Presentation has many configuration examples
  - Uses Cisco IOS CLI
- Aimed at Service Providers
  - Techniques can be used by many enterprises too

BGP Multihoming Techniques

- **Why Multihome?**
- Definition & Options
- Preparing the Network
- Basic Multihoming
- Service Provider Multihoming

Why Multihome?

It’s all about redundancy, diversity & reliability

Why Multihome?

- Redundancy

    One connection to internet means the network is dependent on:

        Local router (configuration, software, hardware)

        WAN media (physical failure, carrier failure)

        Upstream Service Provider (configuration, software, hardware)

Why Multihome?

- Reliability
  - Business critical applications demand continuous availability
  - Lack of redundancy implies lack of reliability implies loss of revenue

Why Multihome?

- Supplier Diversity

  Many businesses demand supplier diversity as a matter of course

  Internet connection from two or more suppliers

    With two or more diverse WAN paths

    With two or more exit points

    With two or more international connections

    Two of everything

Why Multihome?

- Not really a reason, but oft quoted…
- Leverage:

  Playing one ISP off against the other for:

  - Service Quality
  - Service Offerings
  - Availability

Why Multihome?

- Summary:

  Multihoming is easy to demand as requirement for any service provider or end-site network

  But what does it really mean:

    In real life?

    For the network?

    For the Internet?

  And how do we do it?

BGP Multihoming Techniques

- Why Multihome?
- **Definition & Options**
- Preparing the Network
- Basic Multihoming
- Service Provider Multihoming

Multihoming: Definitions & Options

What does it mean, what do we need, and how do we do it?

Multihoming Definition

- More than one link external to the local network

  two or more links to the same ISP

  two or more links to different ISPs

- Usually **two** external facing routers

  one router gives link and provider redundancy only

Autonomous System Number (ASN)

- An ASN is a 16 bit integer
  - 1-64511 are for use on the public Internet
  - 64512-65534 are for private use only
  - 0 and 65535 are reserved
- ASNs are now extended to 32 bit!
  - RFC4893 is standards document describing 32-bit ASNs
  - Representation still under discussion:
    - 32-bit notation or “16.16” notation
      - **draft-michaelson-4byte-as-representation-05.txt**
  - AS 23456 is used to represent 32-bit ASNs in 16-bit ASN world

Autonomous System Number (ASN)

- ASNs are distributed by the Regional Internet Registries
  - They are also available from upstream ISPs who are members of one of the RIRs
- Current 16-bit ASN allocations up to 48127 have been made to the RIRs
  - Around 29000 are visible on the Internet
- The RIRs also have received 1024 32-bit ASNs each
  - Around 10 are visible on the Internet (early adopters)
- See **www.iana.org/assignments/as-numbers**

Private-AS – Application

- Applications

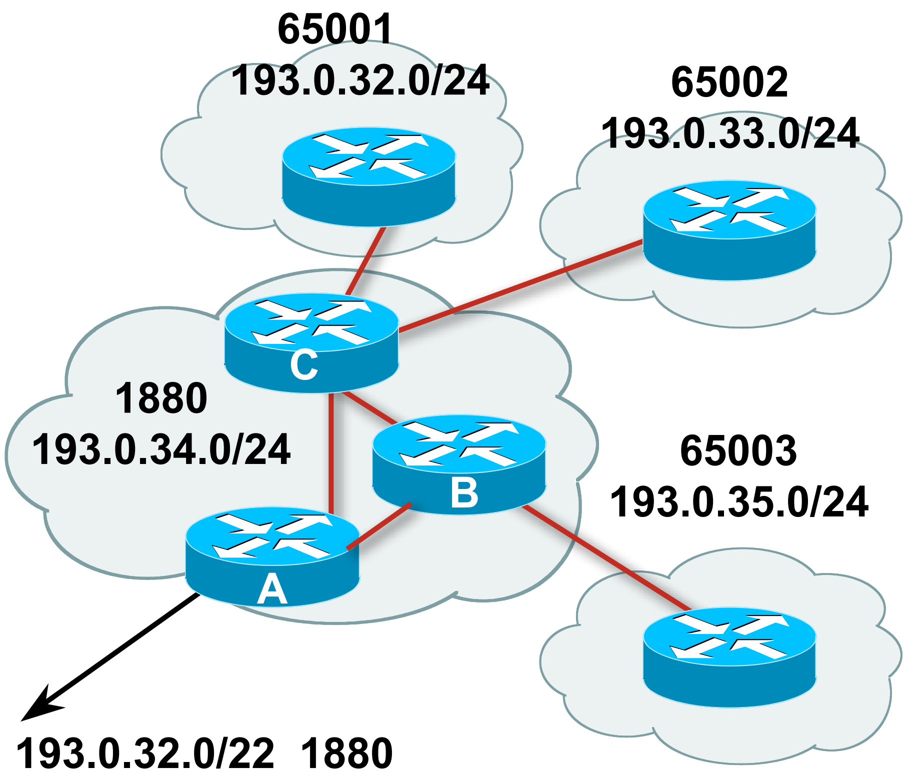
  An ISP with customers multihomed on their backbone (RFC2270)

  -or-

  A corporate network with several regions but connections to the Internet only in the core

  -or-

  Within a BGP Confederation



Private-AS – Removal

- Private ASNs MUST be removed from all prefixes announced to the public Internet

  Include configuration to remove private ASNs in the eBGP template

- As with RFC1918 address space, private ASNs are intended for internal use

  They should not be leaked to the public Internet

- Cisco IOS

```
neighbor x.x.x.x remove-private-AS
```


Configuring Policy

- Three BASIC Principles for IOS configuration examples throughout presentation:
  - prefix-lists to filter prefixes
  - filter-lists to filter ASNs
  - route-maps to apply policy
- Route-maps can be used for filtering, but this is more "advanced" configuration

Policy Tools

- Local preference
  - outbound traffic flows
- Metric (MED)
  - inbound traffic flows (local scope)
- AS-PATH prepend
  - inbound traffic flows (Internet scope)
- Communities
  - specific inter-provider peering

Originating Prefixes: Assumptions

- **MUST** announce assigned address block to Internet
- MAY also announce subprefixes – reachability is not guaranteed
- Current minimum allocation is from /20 to /22 depending on the RIR
  - Several ISPs filter RIR blocks on this boundary
  - Several ISPs filter the rest of address space according to the IANA assignments
  - This activity is called “Net Police” by some

Originating Prefixes

- The RIRs publish their minimum allocation sizes per /8 address block
  - AfriNIC: www.afrinic.net/docs/policies/afpol-v4200407-000.htm
  - APNIC: www.apnic.net/db/min-alloc.html
  - ARIN: www.arin.net/reference/ip_blocks.html
  - LACNIC: lacnic.net/en/registro/index.html
  - RIPE NCC: www.ripe.net/ripe/docs/smallest-alloc-sizes.html
  - Note that AfriNIC only publishes its current minimum allocation size, not the allocation size for its address blocks
- IANA publishes the address space it has assigned to end-sites and allocated to the RIRs:
  - www.iana.org/assignments/ipv4-address-space
- Several ISPs use this published information to filter prefixes on:
  - What should be routed (from IANA)
  - The minimum allocation size from the RIRs

"Net Police" prefix list issues

- Meant to "punish" ISPs who pollute the routing table with specifics rather than announcing aggregates
- Impacts legitimate multihoming especially at the Internet's edge
- Impacts regions where domestic backbone is unavailable or costs $$$ compared with international bandwidth
- Hard to maintain – requires updating when RIRs start allocating from new address blocks
- Don't do it unless consequences understood and you are prepared to keep the list current

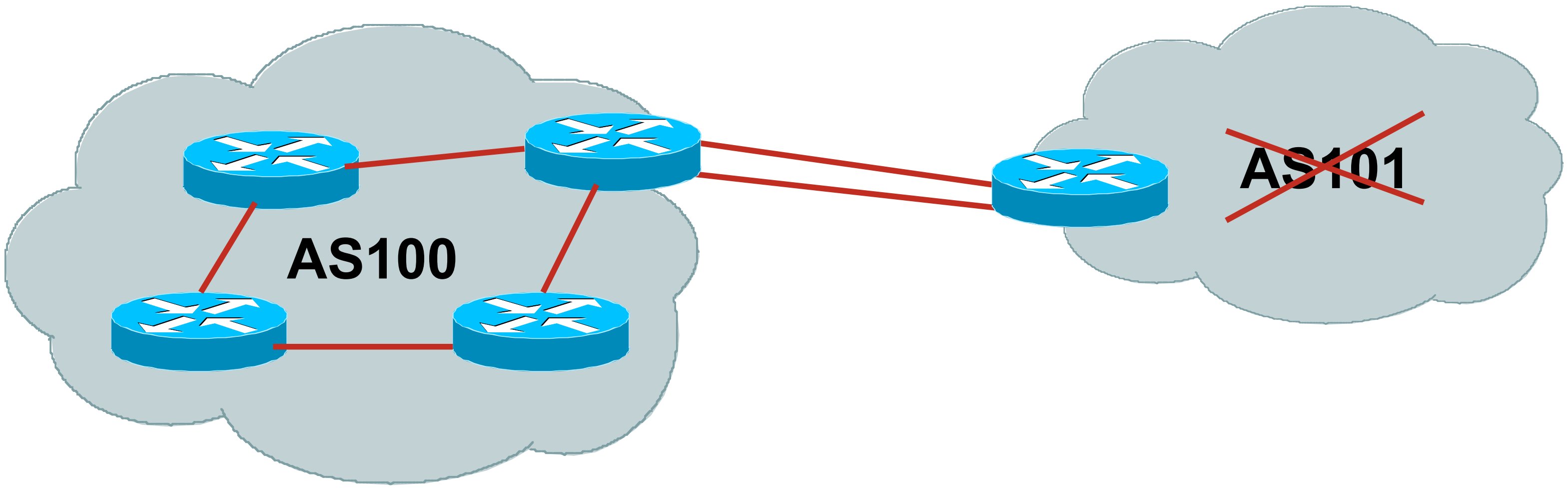
  Consider using the Team Cymru or other reputable bogon BGP feed:

  http://www.team-cymru.org/Services/Bogons/routeserver.html

Multihoming Scenarios

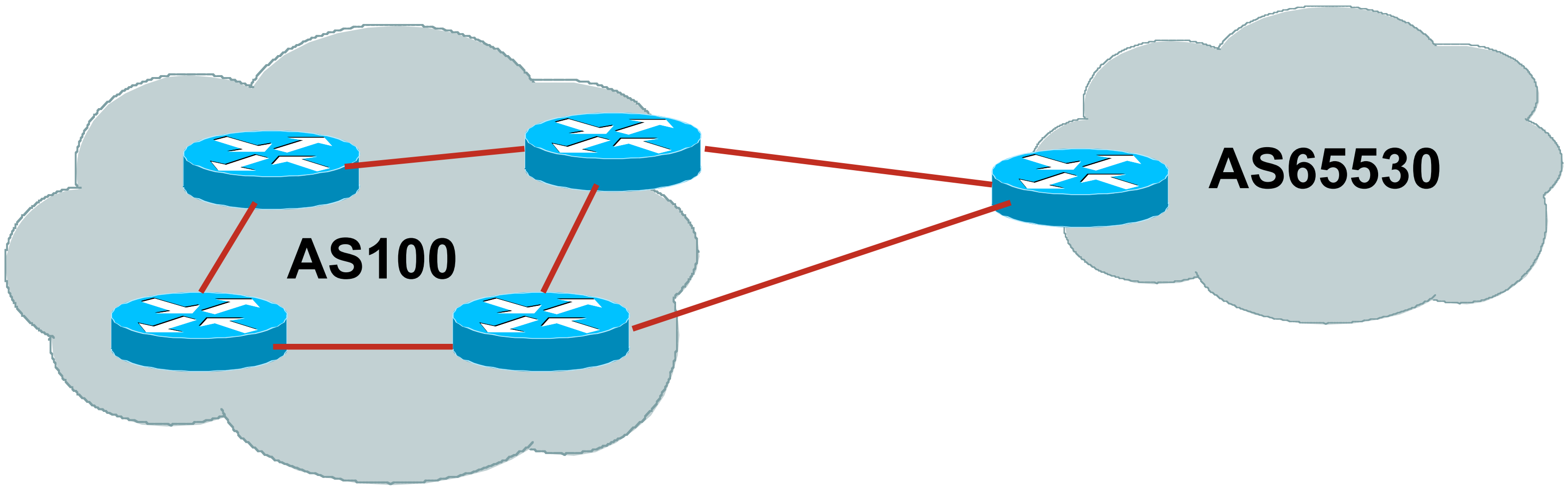
- Stub network
- Multi-homed stub network
- Multi-homed network
- Configuration Options

Stub Network



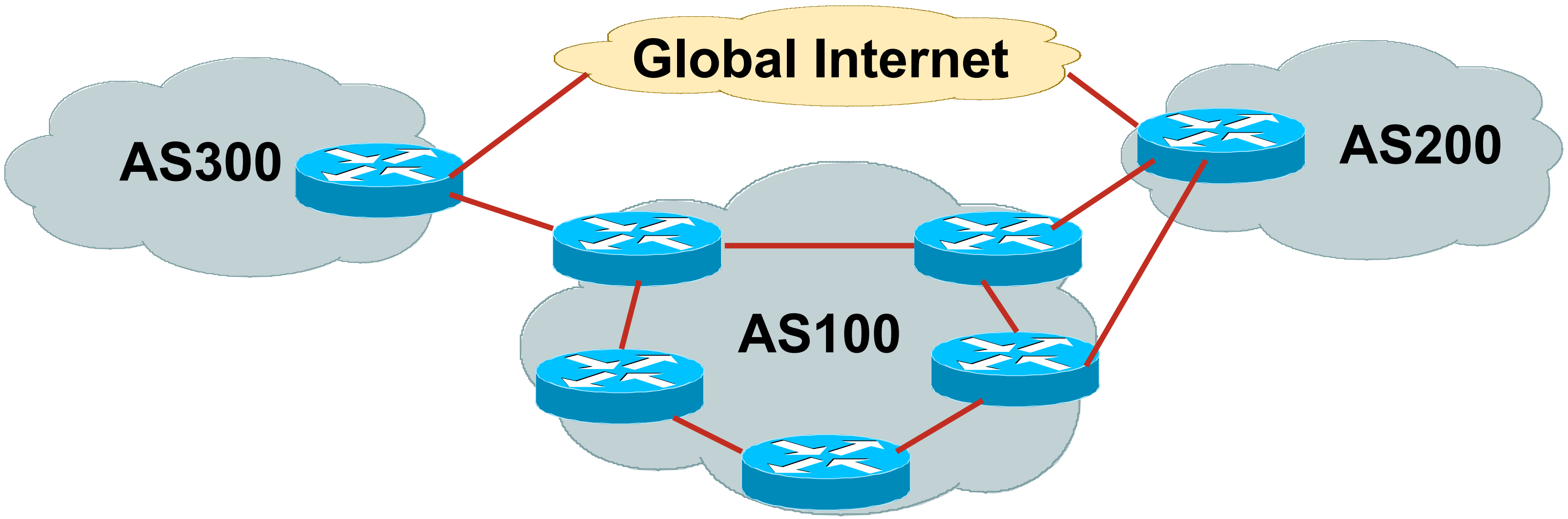
- No need for BGP
- Point static default to upstream ISP
- Upstream ISP advertises stub network
- Policy confined within upstream ISP's policy

Multi-homed Stub Network



- Use BGP (not IGP or static) to loadshare
- Use private AS (ASN > 64511)
- Upstream ISP advertises stub network
- Policy confined within upstream ISP's policy

Multi-homed Network

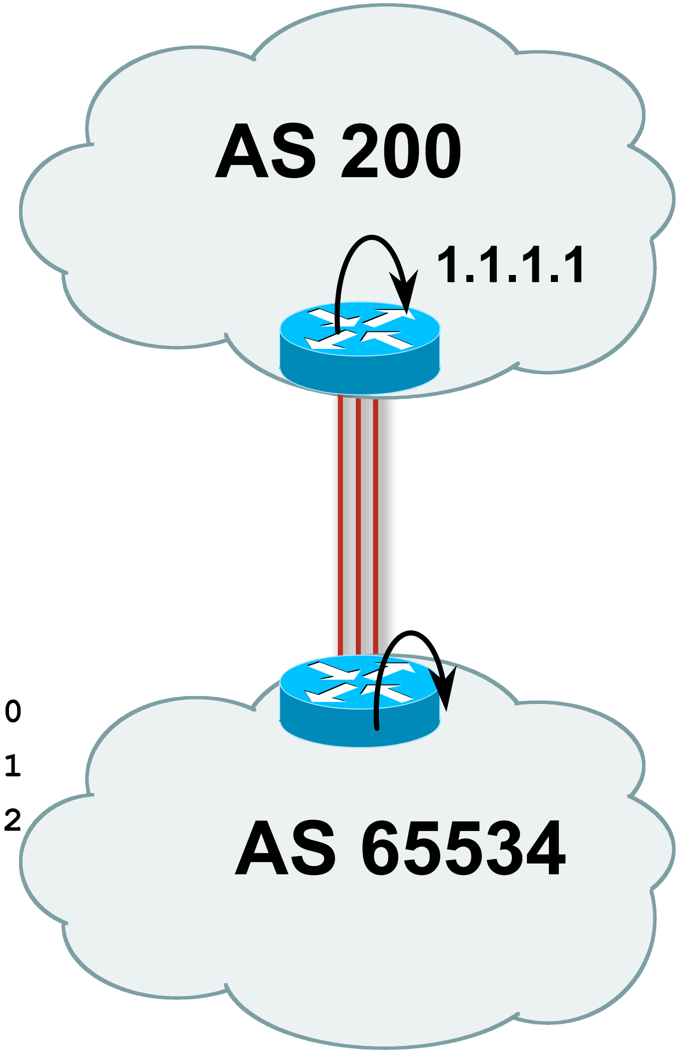


- Many situations possible
  - multiple sessions to same ISP
  - secondary for backup only
  - load-share between primary and secondary
  - selectively use different ISPs

Multiple Sessions to an ISP – Example One

- Use eBGP multihop
  - eBGP to loopback addresses
  - eBGP prefixes learned with loopback address as next hop
- Cisco IOS

```
router bgp 65534
 neighbor 1.1.1.1 remote-as 200
 neighbor 1.1.1.1 ebgp-multihop 2
 !
 ip route 1.1.1.1 255.255.255.255 serial 1/0
 ip route 1.1.1.1 255.255.255.255 serial 1/1
 ip route 1.1.1.1 255.255.255.255 serial 1/2
```



Multiple Sessions to an ISP – Example One

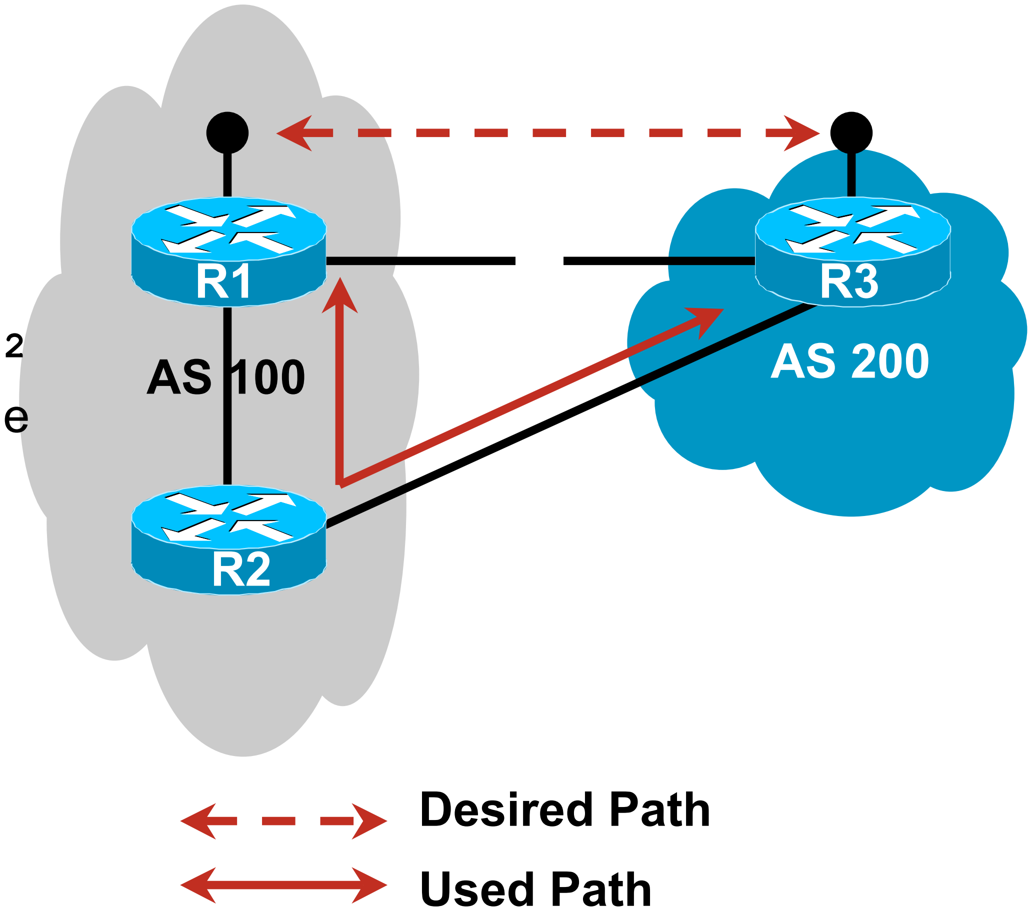
- One eBGP-multihop gotcha:

  R1 and R3 are eBGP peers that are loopback peering

  Configured with:

  `neighbor x.x.x.x ebgp-multihop 2`

  If the R1 to R3 link goes down the session could establish via R2



Multiple Sessions to an ISP – Example One

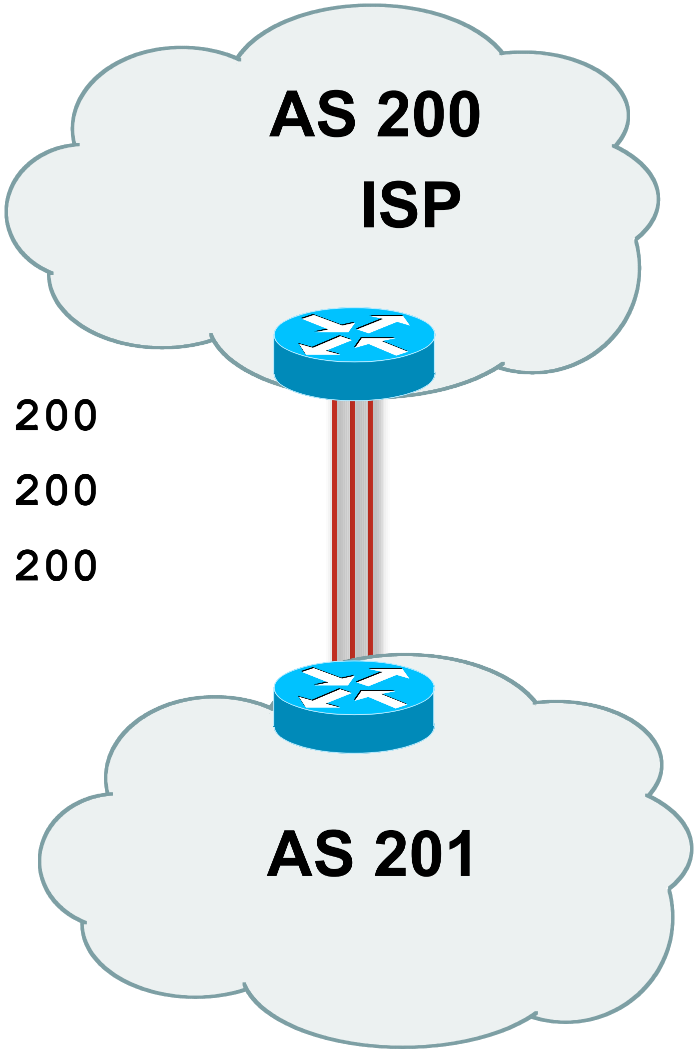
- Try and avoid use of ebgp-multihop unless:
  - It's absolutely necessary –or–
  - Loadsharing across multiple links
- Many ISPs discourage its use, for example:

We will run eBGP multihop, but do not support it as a standard offering because customers generally have a hard time managing it due to:
• routing loops
• failure to realise that BGP session stability problems are usually due connectivity problems between their CPE and their BGP speaker

Multiple Sessions to an ISP
bgp multi path

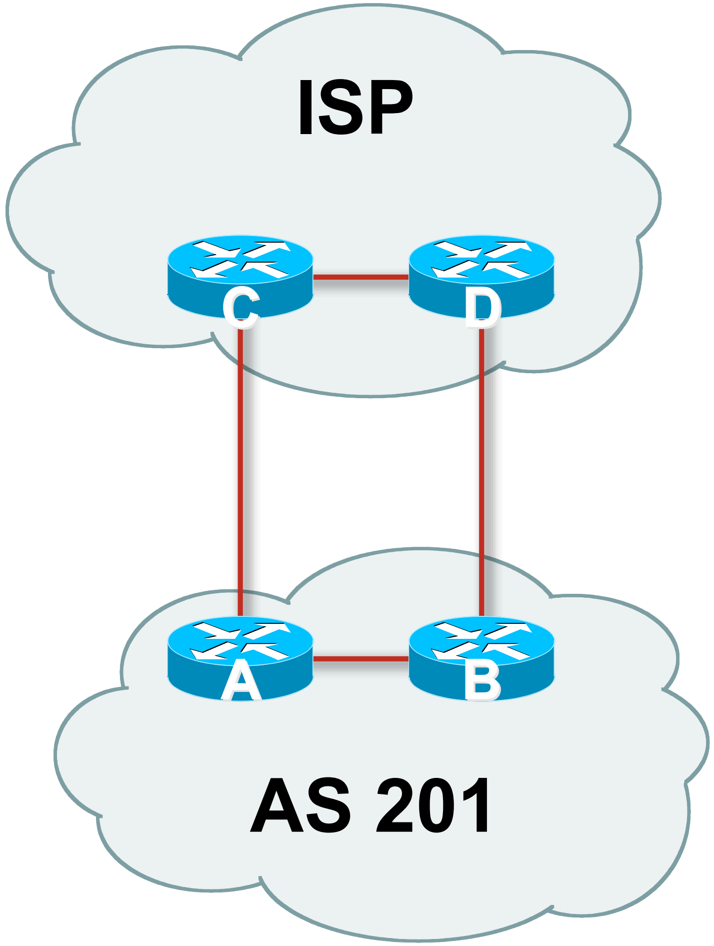
- Three BGP sessions required
- limit of 6 parallel paths

```
router bgp 201
 neighbor 1.1.2.1 remote-as 200
 neighbor 1.1.2.5 remote-as 200
 neighbor 1.1.2.9 remote-as 200
 maximum-paths 3
```



Multiple Sessions to an ISP

- Simplest scheme is to use defaults
- Learn/advertise prefixes for better control
- Planning and some work required to achieve loadsharing
  - Point default towards one ISP
  - Learn selected prefixes from second ISP
  - Modify the number of prefixes learnt to achieve acceptable load sharing
- **No magic solution**



BGP Multihoming Techniques

- Why Multihome?
- Definition & Options
- **Preparing the Network**
- Basic Multihoming
- Service Provider Multihoming

Preparing the Network

Putting our own house in order first…

Preparing the Network

- We will deploy BGP across the network before we try and multihome
- BGP will be used therefore an ASN is required
- If multihoming to different ISPs, public ASN needed:
  - Either go to upstream ISP who is a registry member, or
  - Apply to the RIR yourself for a one off assignment, or
  - Ask an ISP who is a registry member, or
  - Join the RIR and get your own IP address allocation too (this option strongly recommended)!

Preparing the Network

- Will look at two examples of BGP deployment:
  - Example One: network uses only static routes
  - Example Two: network is currently running an IGP

Preparing the Network
Example One

- The network is not running any BGP at the moment

  single statically routed connection to upstream ISP

- The network is not running any IGP at all

  Static default and routes through the network to do “routing”

Preparing the Network
IGP

- Decide on IGP: OSPF or ISIS ☺
- Assign loopback interfaces and /32 addresses to each router which will run the IGP
  - Loopback is used for OSPF and BGP router id anchor
  - Used for iBGP and route origination
- Deploy IGP (e.g. OSPF)
  - IGP can be deployed with NO IMPACT on the existing static routing
    - OSPF distance is 110, static distance is 1
    - Smallest distance wins

Preparing the Network
IGP (cont)

- Be prudent deploying IGP – keep the Link State Database Lean!
  - Router loopbacks go in IGP
  - WAN point to point links go in IGP
  - (In fact, any link where IGP dynamic routing will be run should go into IGP)
  - Summarise on area/level boundaries (if possible) – i.e. think about your IGP address plan

Preparing the Network
IGP (cont)

- Routes which don't go into the IGP include:
  - Dynamic assignment pools (DSL/Cable/Dial/Wireless)
  - Customer point to point link addressing
    - (using next-hop-self in iBGP ensures that these do NOT need to be in IGP)
  - Static/Hosting LANs
  - Customer assigned address space
  - Anything else not listed in the previous slide

Preparing the Network
Introduce OSPF

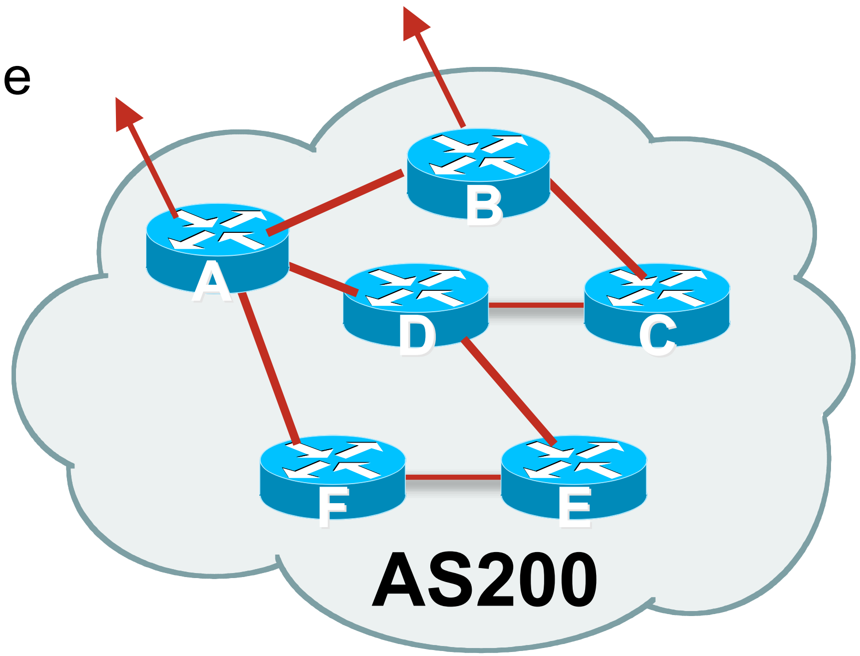
```
interface loopback 0
 ip address 121.10.255.1 255.255.255.255
!
interface Ethernet 0/0
 ip address 121.10.2.1 255.255.255.240
!
interface serial 0/0
 ip address 121.10.0.1 255.255.255.252
!
interface serial 0/1
 ip address 121.10.0.5 255.255.255.252
!
router ospf 100
 network 121.10.255.1 0.0.0.0 area 0
 network 121.10.2.0 0.0.0.15 area 0
 passive-interface default
 no passive-interface Ethernet 0/0
!
ip route 121.10.24.0 255.255.252.0 serial 0/0
ip route 121.10.28.0 255.255.254.0 serial 0/1
```

Add loopback configuration

Customer connections

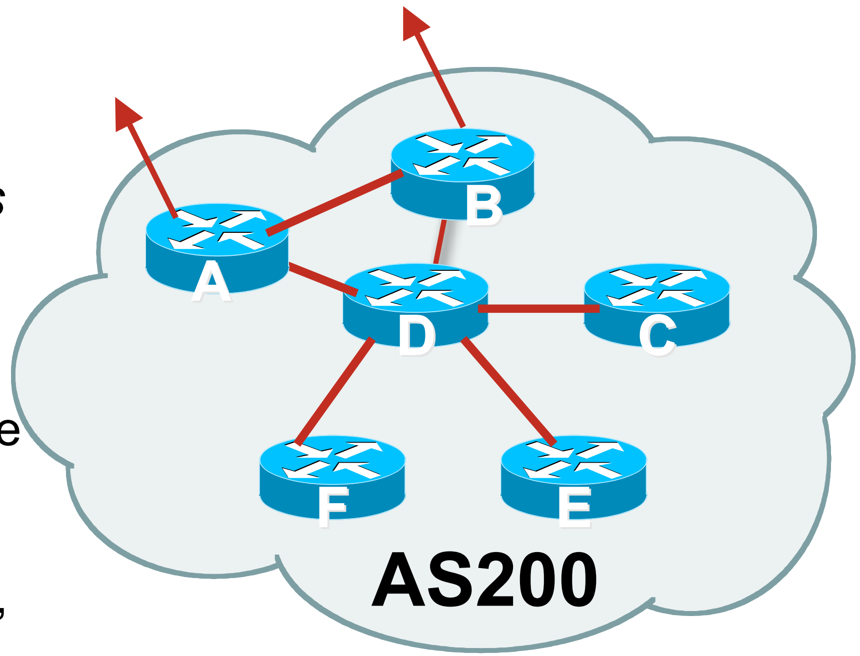
Preparing the Network
iBGP

- Second step is to configure the local network to use iBGP
- iBGP can run on
  - all routers, or
  - a subset of routers, or
  - just on the upstream edge
- *iBGP must run on all routers which are in the transit path between external connections*



Preparing the Network
iBGP (Transit Path)

- *iBGP must run on all routers which are in the transit path between external connections*
- Routers C, E and F are not in the transit path
  - Static routes or IGP will suffice
- Router D is in the transit path
  - Will need to be in iBGP mesh, otherwise routing loops will result



Preparing the Network
Layers

- Typical SP networks have three layers:
  - Core – the backbone, usually the transit path
  - Distribution – the middle, PoP aggregation layer
  - Aggregation – the edge, the devices connecting customers

Preparing the Network Aggregation Layer

- iBGP is optional
  - Many ISPs run iBGP here, either partial routing (more common) or full routing (less common)
  - Full routing is not needed unless customers want full table
  - Partial routing is cheaper/easier, might usually consist of internal prefixes and, optionally, external prefixes to aid external load balancing
    - Communities and peer-groups make this administratively easy
- Many aggregation devices can't run iBGP
  - Static routes from distribution devices for address pools
  - IGP for best exit

Preparing the Network Distribution Layer

- Usually runs iBGP
  - Partial or full routing (as with aggregation layer)
- But does not have to run iBGP
  - IGP is then used to carry customer prefixes (does not scale)
  - IGP is used to determine nearest exit
- Networks which plan to grow large should deploy iBGP from day one
  - Migration at a later date is extra work
  - No extra overhead in deploying iBGP, indeed IGP benefits

Preparing the Network
Core Layer

- Core of network is usually the transit path
- iBGP necessary between core devices
  - Full routes or partial routes:
    - Transit ISPs carry full routes in core
    - Edge ISPs carry partial routes only
- Core layer includes AS border routers

Preparing the Network
iBGP Implementation

- Decide on:
- Best iBGP policy
    - Will it be full routes everywhere, or partial, or some mix?
- iBGP scaling technique
    - Community policy?
    - Route-reflectors?
    - Techniques such as peer groups and templates?

Preparing the Network
iBGP Implementation

- Then deploy iBGP:

  Step 1: Introduce iBGP mesh on chosen routers

  make sure that iBGP distance is greater than IGP distance (it usually is)

  Step 2: Install "customer" prefixes into iBGP

  Check! Does the network still work?

  Step 3: Carefully remove the static routing for the prefixes now in IGP and iBGP

  Check! Does the network still work?

  Step 4: Deployment of eBGP follows

Preparing the Network
iBGP Implementation

Install "customer" prefixes into iBGP?

- Customer assigned address space
  - Network statement/static route combination
  - Use unique community to identify customer assignments
- Customer facing point-to-point links
  - Redistribute connected through filters which only permit point-to-point link addresses to enter iBGP
  - Use a unique community to identify point-to-point link addresses (these are only required for your monitoring system)
- Dynamic assignment pools & local LANs
  - Simple network statement will do this
  - Use unique community to identify these networks

Preparing the Network
iBGP Implementation

Carefully remove static routes?

- Work on one router at a time:

    Check that static route for a particular destination is also learned either by IGP or by iBGP

    If so, remove it

    If not, establish why and fix the problem

    (Remember to look in the RIB, not the FIB!)

- Then the next router, until the whole PoP is done
- Then the next PoP, and so on until the network is now dependent on the IGP and iBGP you have deployed

Preparing the Network Completion

- Previous steps are NOT flag day steps
  - Each can be carried out during different maintenance periods, for example:
  - Step One on Week One
  - Step Two on Week Two
  - Step Three on Week Three
  - And so on
  - And with proper planning will have NO customer visible impact at all

Preparing the Network
Example Two

- The network is not running any BGP at the moment
  - single statically routed connection to upstream ISP
- The network is running an IGP though
  - All internal routing information is in the IGP
  - By IGP, OSPF or ISIS is assumed

Preparing the Network
IGP

- If not already done, assign loopback interfaces (with /32 addresses) to each router which is running the IGP
  - Loopback is used for OSPF and BGP router id anchor
  - Used for iBGP and route origination
- Ensure that the loopback /32s are appearing in the IGP

Preparing the Network
iBGP

- Go through the iBGP decision process as in Example One
- Decide full or partial, and the extent of the iBGP reach in the network

Preparing the Network iBGP Implementation

- Then deploy iBGP:
  - Step 1: Introduce iBGP mesh on chosen routers
    - make sure that iBGP distance is greater than IGP distance (it usually is)
  - Step 2: Install "customer" prefixes into iBGP
    - Check! Does the network still work?
  - Step 3: Reduce BGP distance to be less than the IGP
    - (so that iBGP routes take priority)
  - Step 4: Carefully remove the "customer" prefixes from the IGP
    - Check! Does the network still work?
  - Step 5: Restore BGP distance to be greater than IGP
  - Step 6: Deployment of eBGP follows

Preparing the Network
iBGP Implementation

Install "customer" prefixes into iBGP?

- Customer assigned address space

  Network statement/static route combination

  Use unique community to identify customer assignments

- Customer facing point-to-point links

  Redistribute connected through filters which only permit point-to-point link addresses to enter iBGP

  Use a unique community to identify point-to-point link addresses (these are only required for your monitoring system)

- Dynamic assignment pools & local LANs

  Simple network statement will do this

  Use unique community to identify these networks

Preparing the Network
iBGP Implementation

Carefully remove "customer" routes from IGP?

- Work on one router at a time:

  Check that IGP route for a particular destination is also learned by iBGP

  If so, remove it from the IGP

  If not, establish why and fix the problem

  (Remember to look in the RIB, not the FIB!)

- Then the next router, until the whole PoP is done

- Then the next PoP, and so on until the network is now dependent on the iBGP you have deployed

Preparing the Network
Example Two Configuration – Before BGP

```
interface loopback 0
 ip address 121.10.255.1 255.255.255.255
!
interface serial 0/0
 ip address 121.10.0.1 255.255.255.252
!
interface serial 0/1
 ip address 121.10.0.5 255.255.255.252
!
router ospf 100
 network 121.10.255.1 0.0.0.0 area 0
 passive-interface loopback 0
 redistribute connected subnets        ! Point-to-point links
 redistribute static subnets           ! Customer networks
!
ip route 121.10.24.0 255.255.252.0 serial 0/0
ip route 121.10.28.0 255.255.254.0 serial 0/1
```

Add loopback configuration if not already there

Preparing the Network
Example Two Configuration – Steps 1 & 2

```
! interface and OSPF configuration unchanged
!
router bgp 100
 redistribute connected subnets route-map point-to-point
 neighbor 121.10.1.2 remote-as 100
 neighbor 121.10.1.2 next-hop-self
 ...
 network 121.10.24.0 mask 255.255.252.0
 network 121.10.28.0 mask 255.255.254.0
 distance bgp 200 200 200
!
ip route 121.10.24.0 255.255.252.0 serial 0/0
ip route 121.10.28.0 255.255.254.0 serial 0/1
!
route-map point-to-point permit 5
 match ip address 1
 set community 100:1
!
access-list 1 permit 121.10.0.0 0.0.255.255
```

Add BGP and related configuration in red

Preparing the Network
Example Two Configuration – Steps 3 & 4

```
router ospf 100
 network 121.10.255.1 0.0.0.0 area 0
 network 121.10.2.0 0.0.0.15 area 0
 passive-interface default
 no passive-interface ethernet 0/0
!
router bgp 100
 redistribute connected route-map point-to-point
 neighbor 121.10.1.2 remote-as 100
 neighbor 121.10.1.2 next-hop-self
 ...
 network 121.10.24.0 mask 255.255.252.0
 network 121.10.28.0 mask 255.255.254.0
 distance bgp 20 20 20         ! reduced BGP distance
!
ip route 121.10.24.0 255.255.252.0 serial 0/0
ip route 121.10.28.0 255.255.254.0 serial 0/1
!
...etc...
```

OSPF redistribution has been removed, OSPF tidied up

Preparing the Network
Example Two Configuration – Step 5

```
router ospf 100
 network 121.10.255.1 0.0.0.0 area 0
 network 121.10.2.0 0.0.0.15 area 0
 passive-interface default
 no passive-interface ethernet 0/0
!
router bgp 100
 redistribute connected route-map point-to-point
 neighbor 121.10.1.2 remote-as 100
 neighbor 121.10.1.2 next-hop-self
 ...
 network 121.10.24.0 mask 255.255.252.0
 network 121.10.28.0 mask 255.255.254.0
 distance bgp 200 200 200      ! BGP distance restored
!
ip route 121.10.24.0 255.255.252.0 serial 0/0
ip route 121.10.28.0 255.255.254.0 serial 0/1
!
...etc...
```

Preparing the Network Completion

- Previous steps are NOT flag day steps
  - Each can be carried out during different maintenance periods, for example:
  - Step One on Week One
  - Step Two on Week Two
  - Step Three on Week Three
  - And so on
  - And with proper planning will have NO customer visible impact at all

Preparing the Network Configuration Summary

- IGP essential networks are in IGP
- Customer networks are now in iBGP
  - iBGP deployed over the backbone
  - Full or Partial or Upstream Edge only
- BGP distance is greater than any IGP
- Now ready to deploy eBGP

BGP Multihoming Techniques

- Why Multihome?
- Definition & Options
- Preparing the Network
- **Basic Multihoming**
- “BGP Traffic Engineering”

Basic Multihoming

Learning to walk before we try running

Basic Multihoming

- No frills multihoming
- Will look at two cases:
  - Multihoming with the same ISP
  - Multihoming to different ISPs
- Will keep the examples easy
  - Understanding easy concepts will make the more complex scenarios easier to comprehend
  - All assume that the site multihoming has a /19 address block

Basic Multihoming

- This type is most commonplace at the edge of the Internet
  - Networks here are usually concerned with inbound traffic flows
  - Outbound traffic flows being “nearest exit” is usually sufficient
- Can apply to the leaf ISP as well as Enterprise networks

Basic Multihoming

Multihoming to the Same ISP

Basic Multihoming: Multihoming to the same ISP

- Use BGP for this type of multihoming
  - use a private AS (ASN > 64511)
  - There is no need or justification for a public ASN
    - Making the nets of the end-site visible gives no useful information to the Internet
- Upstream ISP proxy aggregates
  - in other words, announces only your address block to the Internet from their AS (as would be done if you had one statically routed connection)

Two links to the same ISP

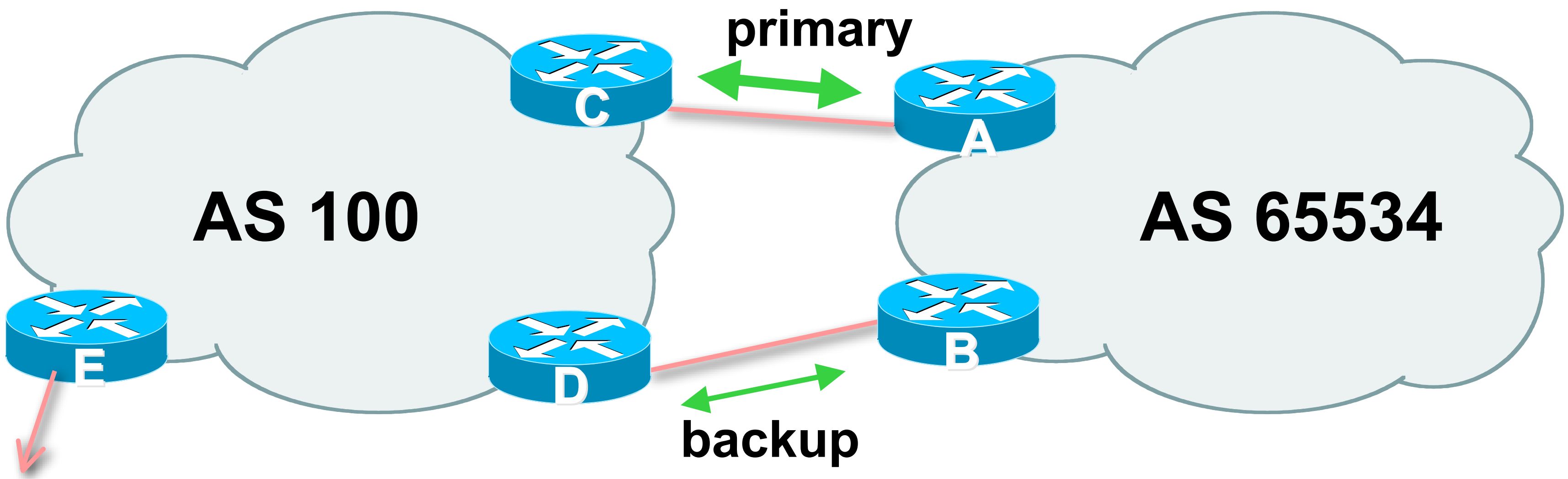
One link primary, the other link backup only

Two links to the same ISP (one as backup only)

- Applies when end-site has bought a large primary WAN link to their upstream a small secondary WAN link as the backup

  For example, primary path might be an E1, backup might be 64kbps

Two links to the same ISP (one as backup only)



- AS100 removes private AS and any customer subprefixes from Internet announcement

Two links to the same ISP (one as backup only)

- Announce /19 aggregate on each link

  primary link:

    Outbound – announce /19 unaltered

    Inbound – receive default route

  backup link:

    Outbound – announce /19 with increased metric

    Inbound – received default, and reduce local preference

- When one link fails, the announcement of the /19 aggregate via the other link ensures continued connectivity

Two links to the same ISP (one as backup only)

- Router A Configuration

```
router bgp 65534
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.2 remote-as 100
 neighbor 122.102.10.2 description RouterC
 neighbor 122.102.10.2 prefix-list aggregate out
 neighbor 122.102.10.2 prefix-list default in
!
ip prefix-list aggregate 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 links to the same ISP (one as backup only)

- Router B Configuration

```
router bgp 65534
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.6 remote-as 100
 neighbor 122.102.10.6 description RouterD
 neighbor 122.102.10.6 prefix-list aggregate out
 neighbor 122.102.10.6 route-map routerD-out out
 neighbor 122.102.10.6 prefix-list default in
 neighbor 122.102.10.6 route-map routerD-in in
!
```

..next slide

Two links to the same ISP (one as backup only)

```
ip prefix-list aggregate 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
!
route-map routerD-out permit 10
 match ip address prefix-list aggregate
 set metric 10
route-map routerD-out permit 20
!
route-map routerD-in permit 10
 set local-preference 90
!
```


Two links to the same ISP (one as backup only)

- Router C Configuration (main link)

```
router bgp 100
 neighbor 122.102.10.1 remote-as 65534
 neighbor 122.102.10.1 default-originate
 neighbor 122.102.10.1 prefix-list Customer in
 neighbor 122.102.10.1 prefix-list default out
!
ip prefix-list Customer permit 121.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
```

Two links to the same ISP (one as backup only)

- Router D Configuration (backup link)

```
router bgp 100
 neighbor 122.102.10.5 remote-as 65534
 neighbor 122.102.10.5 default-originate
 neighbor 122.102.10.5 prefix-list Customer in
 neighbor 122.102.10.5 prefix-list default out
!
ip prefix-list Customer permit 121.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
```

Two links to the same ISP (one as backup only)

- Router E Configuration

```
router bgp 100
 neighbor 122.102.10.17 remote-as 110
 neighbor 122.102.10.17 remove-private-AS
 neighbor 122.102.10.17 prefix-list Customer out
!
ip prefix-list Customer permit 121.10.0.0/19
```

- Router E removes the private AS and customer's subprefixes from external announcements
- Private AS still visible inside AS100

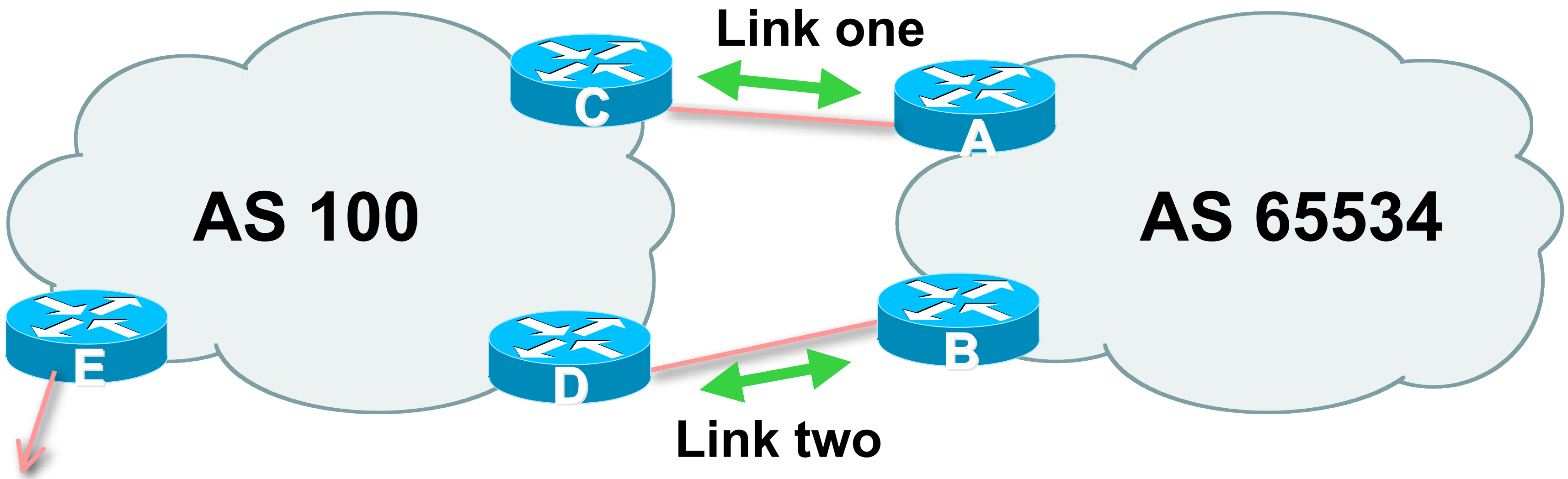
Two links to the same ISP

With Loadsharing

Loadsharing to the same ISP

- More common case
- End sites tend not to buy circuits and leave them idle, only used for backup as in previous example
- This example assumes equal capacity circuits
    - Unequal capacity circuits requires more refinement – see later

Loadsharing to the same ISP



- Border router E in AS100 removes private AS and any customer subprefixes from Internet announcement

Loadsharing to the same ISP

- Announce /19 aggregate on each link
- Split /19 and announce as two /20s, one on each link

  basic inbound loadsharing

  assumes equal circuit capacity and even spread of traffic across address block
- Vary the split until “perfect” loadsharing achieved
- Accept the default from upstream

  basic outbound loadsharing by nearest exit

  okay in first approx as most ISP and end-site traffic is inbound

Loadsharing to the same ISP

- Router A Configuration

```
router bgp 65534
 network 121.10.0.0 mask 255.255.224.0
 network 121.10.0.0 mask 255.255.240.0
 neighbor 122.102.10.2 remote-as 100
 neighbor 122.102.10.2 prefix-list routerC out
 neighbor 122.102.10.2 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerC permit 121.10.0.0/20
ip prefix-list routerC permit 121.10.0.0/19
!
ip route 121.10.0.0 255.255.240.0 null0
ip route 121.10.0.0 255.255.224.0 null0
```

Loadsharing to the same ISP

- Router C Configuration

```
router bgp 100
 neighbor 122.102.10.1 remote-as 65534
 neighbor 122.102.10.1 default-originate
 neighbor 122.102.10.1 prefix-list Customer in
 neighbor 122.102.10.1 prefix-list default out
!
ip prefix-list Customer permit 121.10.0.0/19 le 20
ip prefix-list default permit 0.0.0.0/0
```

- Router C only allows in /19 and /20 prefixes from customer block
- Router D configuration is identical

Loadsharing to the same ISP

- Router E Configuration

```
router bgp 100
 neighbor 122.102.10.17 remote-as 110
 neighbor 122.102.10.17 remove-private-AS
 neighbor 122.102.10.17 prefix-list Customer out
!
ip prefix-list Customer permit 121.10.0.0/19
```

- Private AS still visible inside AS100

Loadsharing to the same ISP

- Default route for outbound traffic?

  Use default-information originate for the IGP and rely on IGP metrics for nearest exit

  e.g. on router A:

```
router ospf 65534
 default-information originate metric 2 metric-type 1
```

Loadsharing to the same ISP

- Loadsharing configuration is only on customer router
- Upstream ISP has to
  - remove customer subprefixes from external announcements
  - remove private AS from external announcements
- Could also use BGP communities

Two links to the same ISP

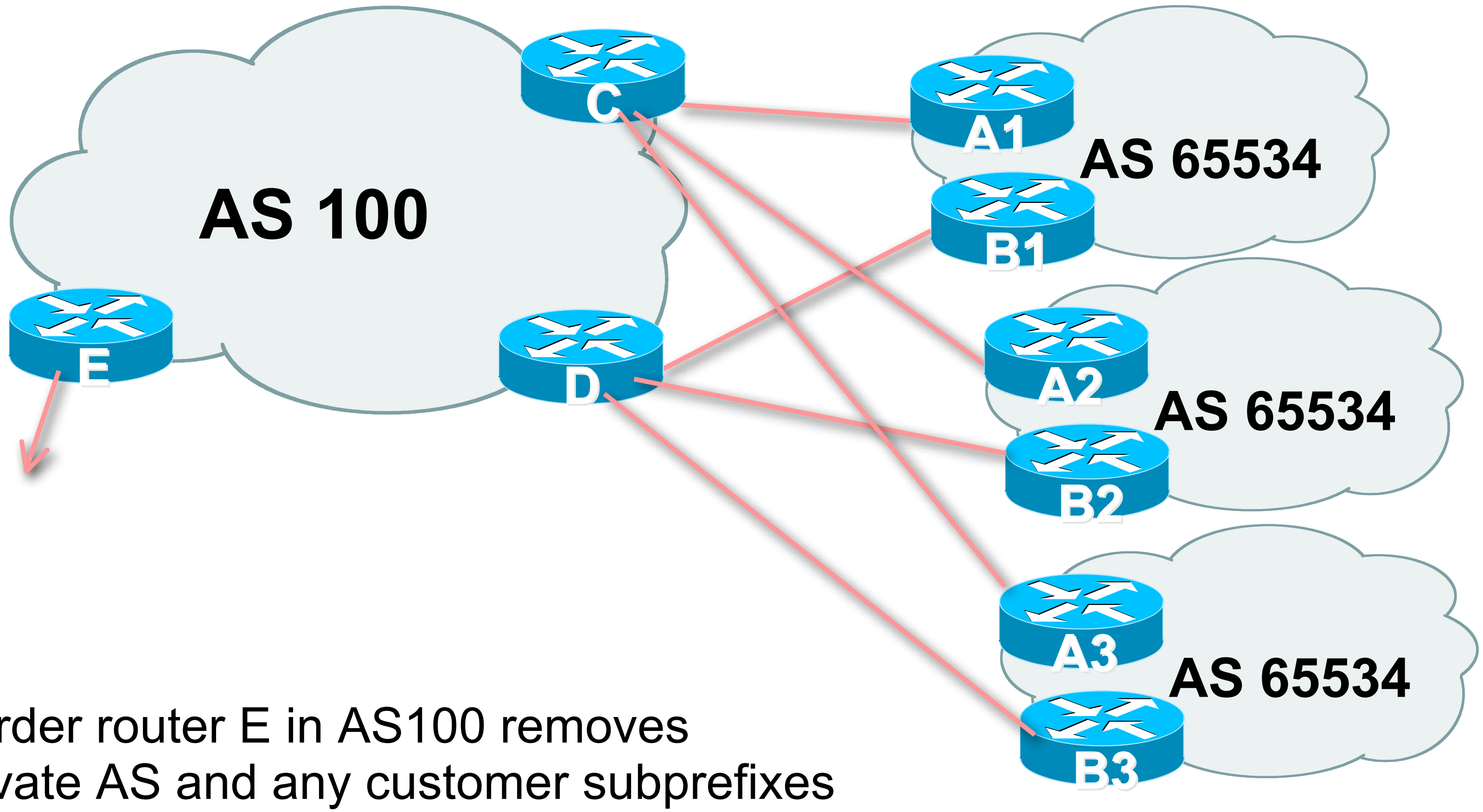
Multiple Dualhomed Customers

(RFC2270)

Multiple Dualhomed Customers (RFC2270)

- Unusual for an ISP just to have one dualhomed customer
  - Valid/valuable service offering for an ISP with multiple PoPs
  - Better for ISP than having customer multihome with another provider!
- Look at scaling the configuration
  - ⇒ Simplifying the configuration
  - Using templates, peer-groups, etc
  - Every customer has the same configuration (basically)

Multiple Dualhomed Customers (RFC2270)



- Border router E in AS100 removes private AS and any customer subprefixes from Internet announcement

Multiple Dualhomed Customers (RFC2270)

- Customer announcements as per previous example
- Use the same private AS for each customer
  - documented in RFC2270
  - address space is not overlapping
  - each customer hears default only
- Router An and Bn configuration same as Router A and B previously

Multiple Dualhomed Customers (RFC2270)

- Router A1 Configuration

```
router bgp 65534
 network 121.10.0.0 mask 255.255.224.0
 network 121.10.0.0 mask 255.255.240.0
 neighbor 122.102.10.2 remote-as 100
 neighbor 122.102.10.2 prefix-list routerC out
 neighbor 122.102.10.2 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerC permit 121.10.0.0/20
ip prefix-list routerC permit 121.10.0.0/19
!
ip route 121.10.0.0 255.255.240.0 null0
ip route 121.10.0.0 255.255.224.0 null0
```

Multiple Dualhomed Customers (RFC2270)

- Router C Configuration

```
router bgp 100
 neighbor bgp-customers peer-group
 neighbor bgp-customers remote-as 65534
 neighbor bgp-customers default-originate
 neighbor bgp-customers prefix-list default out
 neighbor 122.102.10.1 peer-group bgp-customers
 neighbor 122.102.10.1 description Customer One
 neighbor 122.102.10.1 prefix-list Customer1 in
 neighbor 122.102.10.9 peer-group bgp-customers
 neighbor 122.102.10.9 description Customer Two
 neighbor 122.102.10.9 prefix-list Customer2 in
```

Multiple Dualhomed Customers (RFC2270)

```
 neighbor 122.102.10.17 peer-group bgp-customers
 neighbor 122.102.10.17 description Customer Three
 neighbor 122.102.10.17 prefix-list Customer3 in
!
ip prefix-list Customer1 permit 121.10.0.0/19 le 20
ip prefix-list Customer2 permit 121.16.64.0/19 le 20
ip prefix-list Customer3 permit 121.14.192.0/19 le 20
ip prefix-list default permit 0.0.0.0/0
```

- Router C only allows in /19 and /20 prefixes from customer block
- Router D configuration is almost identical

Multiple Dualhomed Customers (RFC2270)

- Router E Configuration

  assumes customer address space is not part of upstream's address block

```
router bgp 100
 neighbor 122.102.10.17 remote-as 110
 neighbor 122.102.10.17 remove-private-AS
 neighbor 122.102.10.17 prefix-list Customers out
!
ip prefix-list Customers permit 121.10.0.0/19
ip prefix-list Customers permit 121.16.64.0/19
ip prefix-list Customers permit 121.14.192.0/19
```

- Private AS still visible inside AS100

Multiple Dualhomed Customers (RFC2270)

- If customers' prefixes come from ISP's address block

  do **NOT** announce them to the Internet

  announce ISP aggregate only

- Router E configuration:

```
router bgp 100
 neighbor 122.102.10.17 remote-as 110
 neighbor 122.102.10.17 prefix-list my-aggregate out
!
ip prefix-list my-aggregate permit 121.8.0.0/13
```

Basic Multihoming

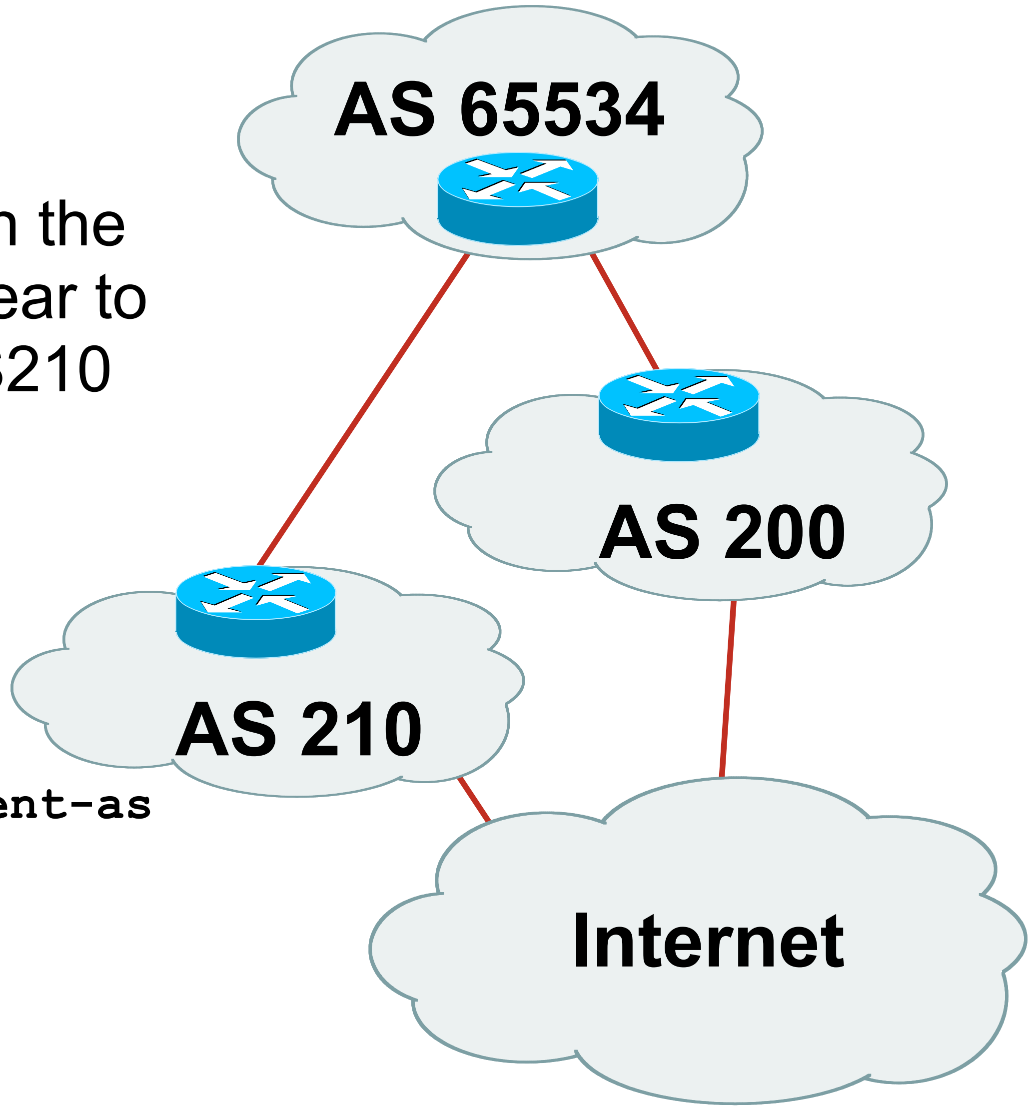
Multihoming to different ISPs

Two links to different ISPs

- Use a Public AS
  - Or use private AS if agreed with the other ISP
  - But some people don’t like the “inconsistent-AS” which results from use of a private-AS
- Address space comes from
  - both upstreams or
  - Regional Internet Registry
- Configuration concepts very similar

Inconsistent-AS?

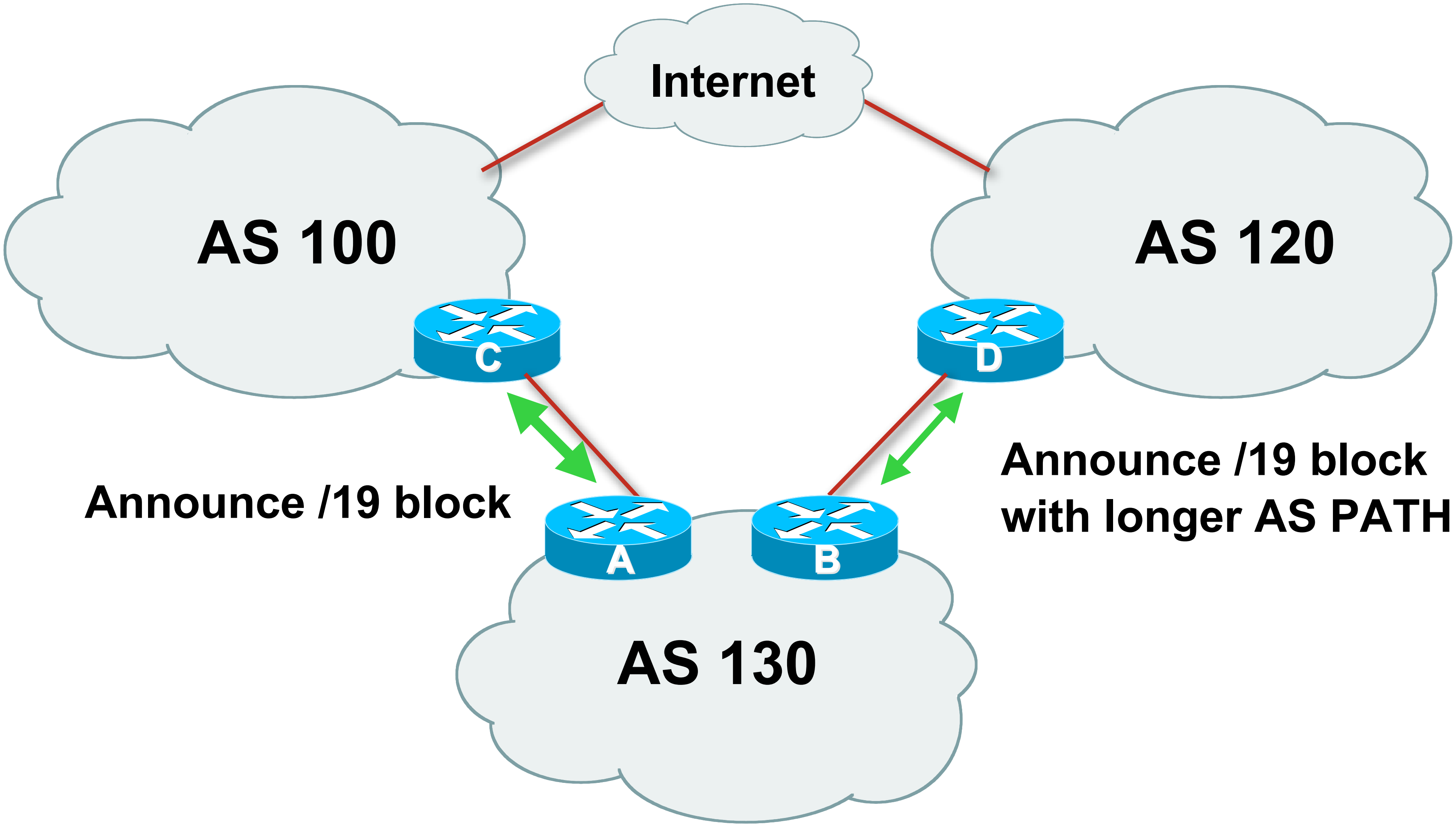
- Viewing the prefixes originated by AS65534 in the Internet shows they appear to be originated by both AS210 and AS200
  - This is NOT bad
  - Nor is it illegal
- IOS command is
  - **`show ip bgp inconsistent-as`**



Two links to different ISPs

One link primary, the other link backup only

Two links to different ISPs (one as backup only)



Two links to different ISPs (one as backup only)

- Announce /19 aggregate on each link
    - primary link makes standard announcement
    - backup link lengthens the AS PATH by using AS PATH prepend
- When one link fails, the announcement of the /19 aggregate via the other link ensures continued connectivity

Two links to different ISPs (one as backup only)

- Router A Configuration

```
router bgp 130
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.1 remote-as 100
 neighbor 122.102.10.1 prefix-list aggregate out
 neighbor 122.102.10.1 prefix-list default in
!
ip prefix-list aggregate 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 links to different ISPs (one as backup only)

- Router B Configuration

```
router bgp 130
 network 121.10.0.0 mask 255.255.224.0
 neighbor 120.1.5.1 remote-as 120
 neighbor 120.1.5.1 prefix-list aggregate out
 neighbor 120.1.5.1 route-map routerD-out out
 neighbor 120.1.5.1 prefix-list default in
 neighbor 120.1.5.1 route-map routerD-in in
!
ip prefix-list aggregate permit 121.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
route-map routerD-out permit 10
 set as-path prepend 130 130 130
!
route-map routerD-in permit 10
 set local-preference 80
```

Two links to different ISPs (one as backup only)

- Not a common situation as most sites tend to prefer using whatever capacity they have

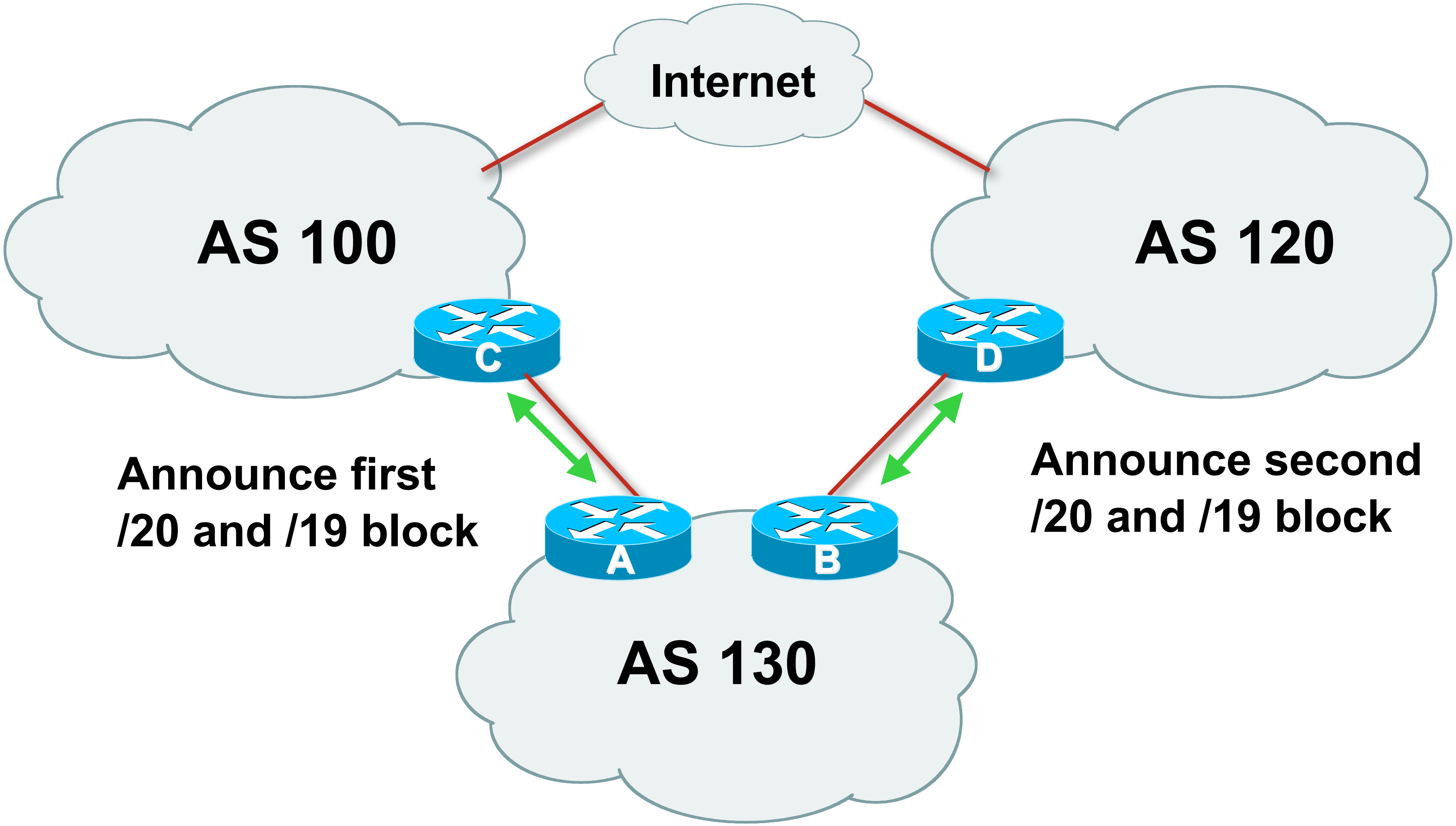
  (Useful when two competing ISPs agree to provide mutual backup to each other)

- But it shows the basic concepts of using local-prefs and AS-path prepends for engineering traffic in the chosen direction

Two links to different ISPs

With Loadsharing

Two links to different ISPs (with loadsharing)



Two links to different ISPs (with loadsharing)

- Announce /19 aggregate on each link
- Split /19 and announce as two /20s, one on each link

  basic inbound loadsharing

- When one link fails, the announcement of the /19 aggregate via the other ISP ensures continued connectivity

Two links to different ISPs (with loadsharing)

- Router A Configuration

```
router bgp 130
 network 121.10.0.0 mask 255.255.224.0
 network 121.10.0.0 mask 255.255.240.0
 neighbor 122.102.10.1 remote-as 100
 neighbor 122.102.10.1 prefix-list firstblock out
 neighbor 122.102.10.1 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
!
ip prefix-list firstblock permit 121.10.0.0/20
ip prefix-list firstblock permit 121.10.0.0/19
```

Two links to different ISPs (with loadsharing)

- Router B Configuration

```
router bgp 130
 network 121.10.0.0 mask 255.255.224.0
 network 121.10.16.0 mask 255.255.240.0
 neighbor 120.1.5.1 remote-as 120
 neighbor 120.1.5.1 prefix-list secondblock out
 neighbor 120.1.5.1 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
!
ip prefix-list secondblock permit 121.10.16.0/20
ip prefix-list secondblock permit 121.10.0.0/19
```

Two links to different ISPs (with loadsharing)

- Loadsharing in this case is very basic
- But shows the first steps in designing a load sharing solution

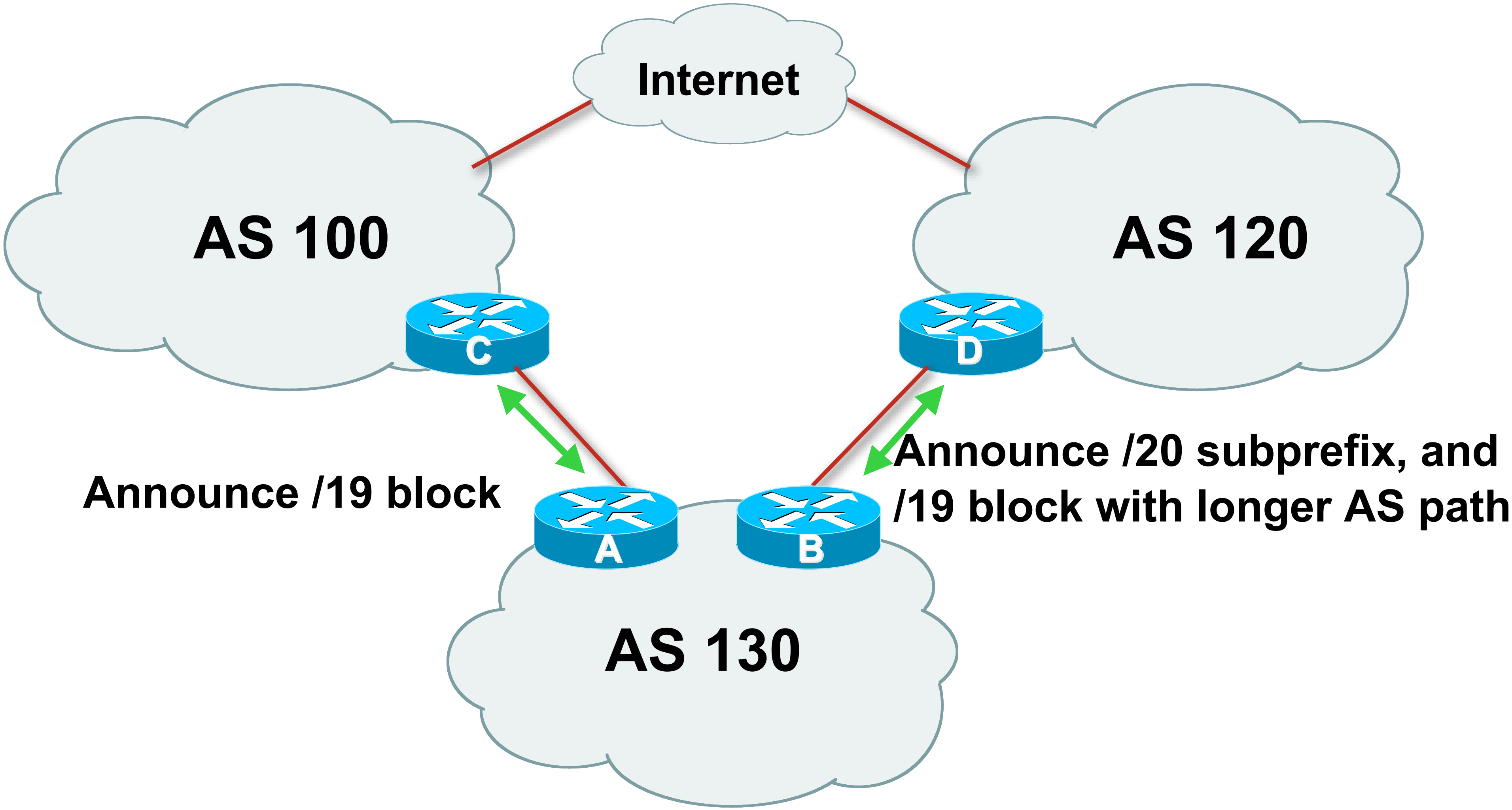
  Start with a simple concept

  And build on it…!

Two links to different ISPs

More Controlled Loadsharing

Loadsharing with different ISPs



Loadsharing with different ISPs

- Announce /19 aggregate on each link
  - On first link, announce /19 as normal
  - On second link, announce /19 with longer AS PATH, and announce one /20 subprefix
    - controls loadsharing between upstreams and the Internet
- Vary the subprefix size and AS PATH length until “perfect” loadsharing achieved
- Still require redundancy!

Loadsharing with different ISPs

- Router A Configuration

```
router bgp 130
 network 121.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.1 remote-as 100
 neighbor 122.102.10.1 prefix-list default in
 neighbor 122.102.10.1 prefix-list aggregate out
!
ip prefix-list aggregate 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
```

Loadsharing with different ISPs

- Router B Configuration

```
router bgp 130
 network 121.10.0.0 mask 255.255.224.0
 network 121.10.16.0 mask 255.255.240.0
 neighbor 120.1.5.1 remote-as 120
 neighbor 120.1.5.1 prefix-list default in
 neighbor 120.1.5.1 prefix-list subblocks out
 neighbor 120.1.5.1 route-map routerD out
!
route-map routerD permit 10
 match ip address prefix-list aggregate
 set as-path prepend 130 130
route-map routerD permit 20
!
ip prefix-list subblocks permit 121.10.0.0/19 le 20
ip prefix-list aggregate permit 121.10.0.0/19
```

Loadsharing with different ISPs

- This example is more commonplace
- Shows how ISPs and end-sites subdivide address space frugally, as well as use the AS-PATH prepend concept to optimise the load sharing between different ISPs
- Notice that the /19 aggregate block is ALWAYS announced

BGP Multihoming Techniques

- Why Multihome?
- Definition & Options
- Preparing the Network
- Basic Multihoming
- **“BGP Traffic Engineering”**

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
  - Three upstreams, unequal link bandwidths
- Require BGP and a public ASN
- Examples assume that the local network has their own /19 address block

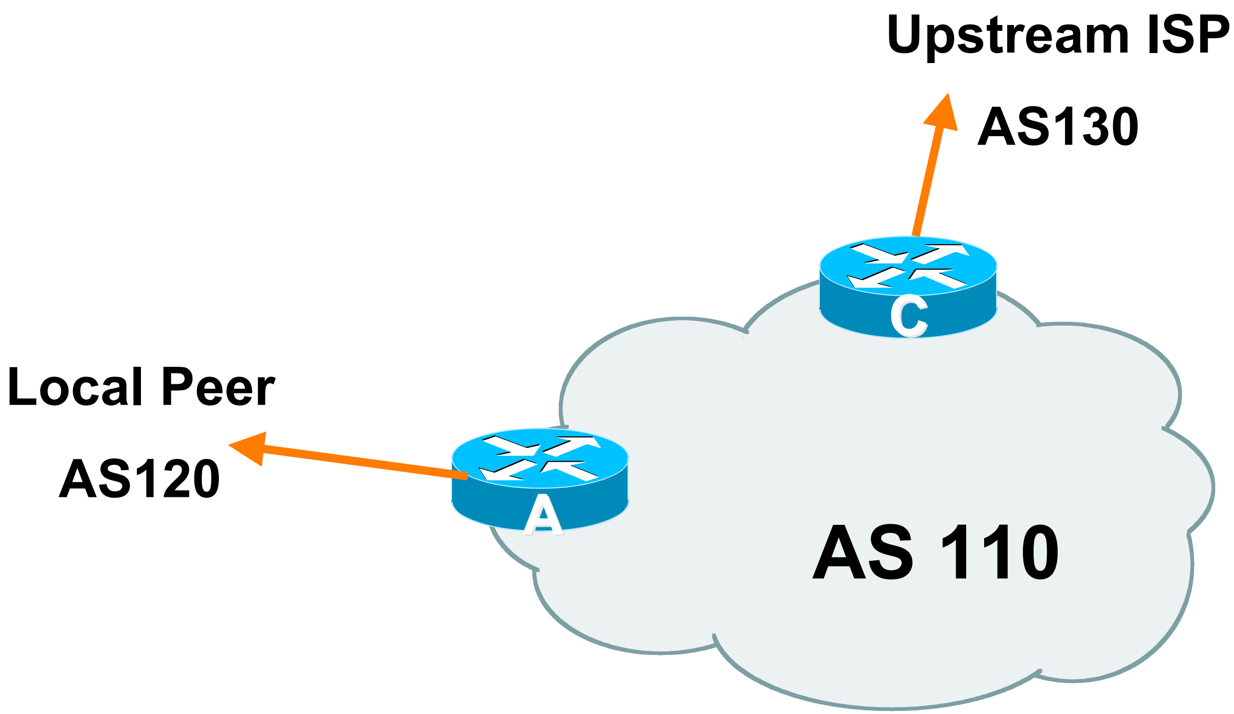
Service Provider Multihoming

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

```
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 250
```

Prefix filters inbound

One Upstream, One Local Peer

- Router A – Alternative Configuration

```
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
```

AS Path filters – more "trusting"

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

Aside: Configuration Recommendations

- 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

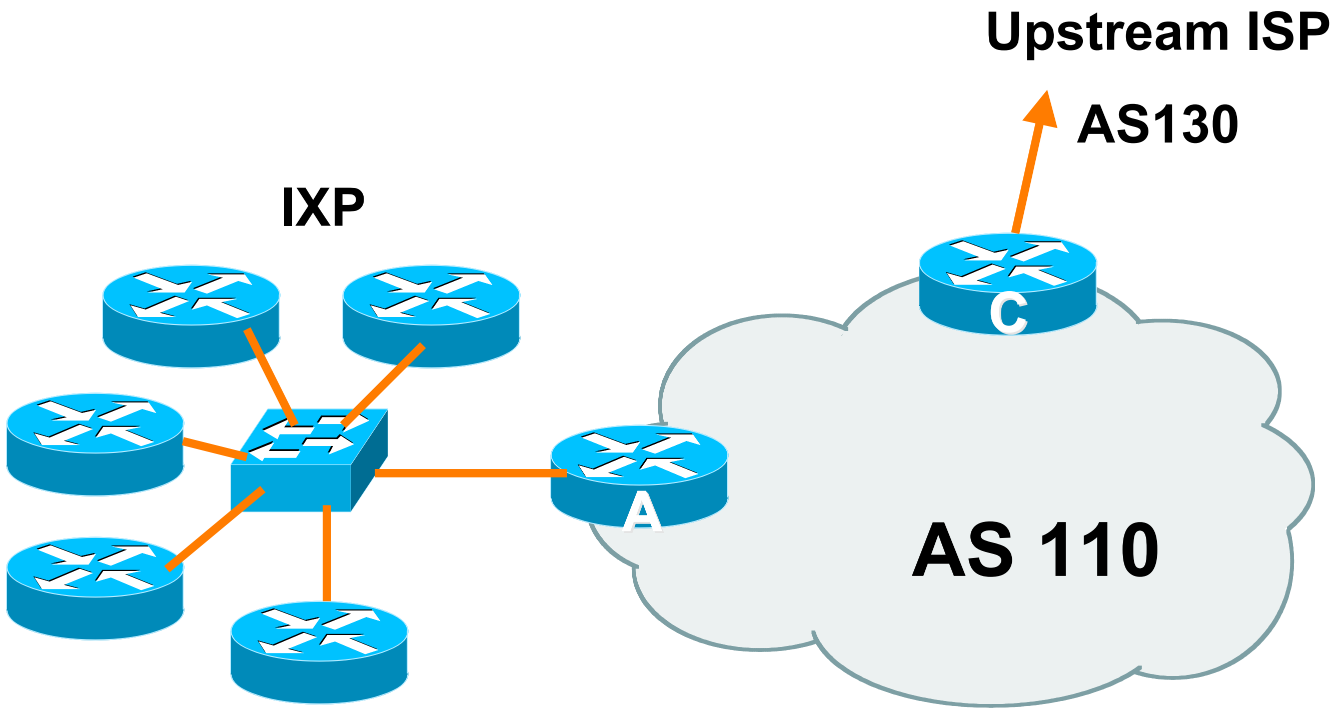
Service Provider Multihoming

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
 neighbor ixp-peers route-map set-local-pref in
```

…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
!
route-map set-local-pref permit 10
 set local-preference 150
!
```

One Upstream, Local Exchange

- 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

- Note the inbound route-map which sets the local preference higher than the default

    This ensures that local traffic crosses the IXP

    (And avoids potential problems with uRPF check)

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:
IXP Configuration Recommendations

- 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

Service Provider Multihoming

Two Upstreams, One local peer

Two Upstreams, One Local Peer

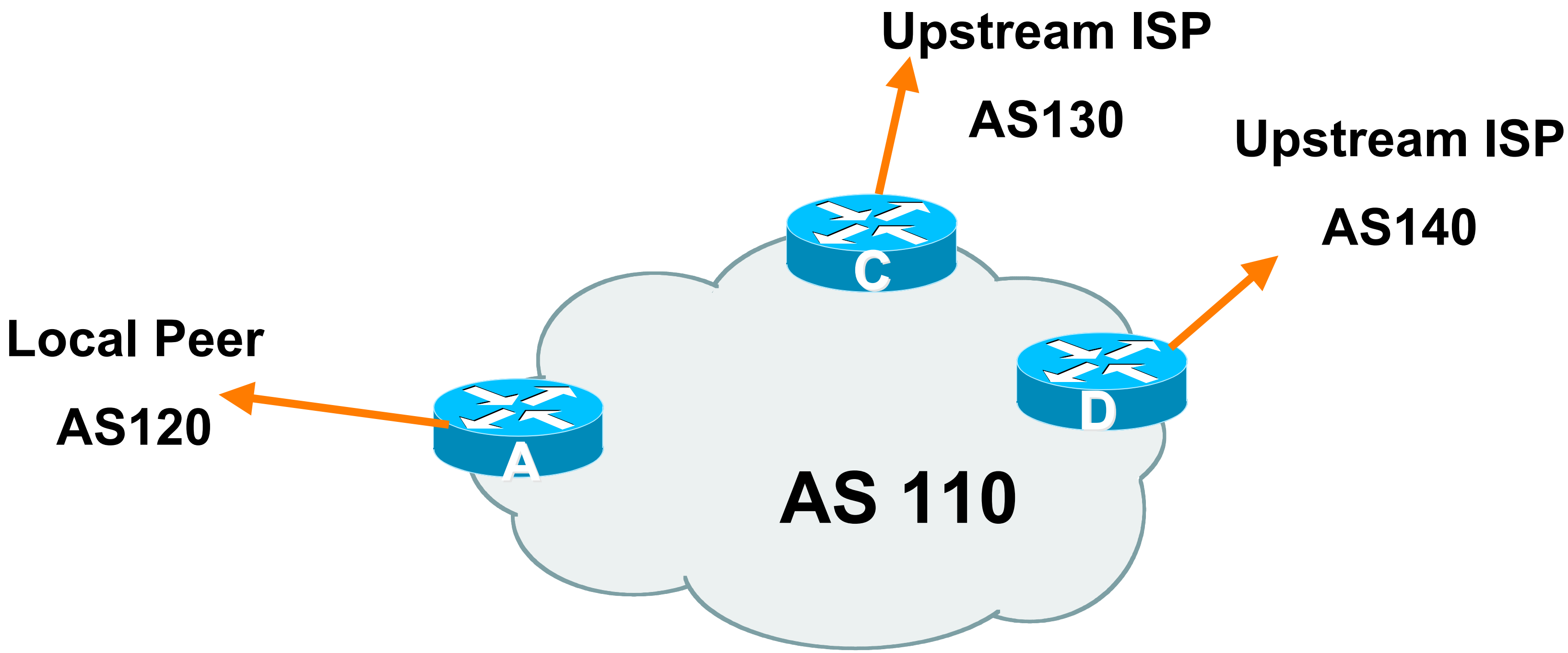
- Connect to both upstream transit providers to see the "Internet"

  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
- Note separation of Router C and D

  Single edge router means no redundancy
- Router A

  Same routing configuration as in example with one upstream and one local peer

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

```
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

Allow all prefixes in apart from RFC1918 and friends

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 its neighbouring ASes are accepted
    - Traffic to those ASes will go over AS130 link
    - Traffic to other all other ASes will go over the link to AS140
    - What about backup?

Two Upstreams, One Local Peer
Partial Routes

- Router C IGP Configuration

```
router ospf 110
default-information originate metric 30
passive-interface Serial 0/0
!
ip route 0.0.0.0 0.0.0.0 serial 0/0 254
```

- Router D IGP Configuration

```
router ospf 110
default-information originate metric 10
passive-interface Serial 0/0
!
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

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

Aside: Configuration Recommendation

- When distributing internal default by iBGP or OSPF
  - Make sure that routers connecting to private peers or to IXPs do NOT carry the default route
  - Otherwise they could point a default route to you and unintentionally transit your backbone
  - Simple fix for Private Peer/IXP routers:

```
ip route 0.0.0.0 0.0.0.0 null0
```

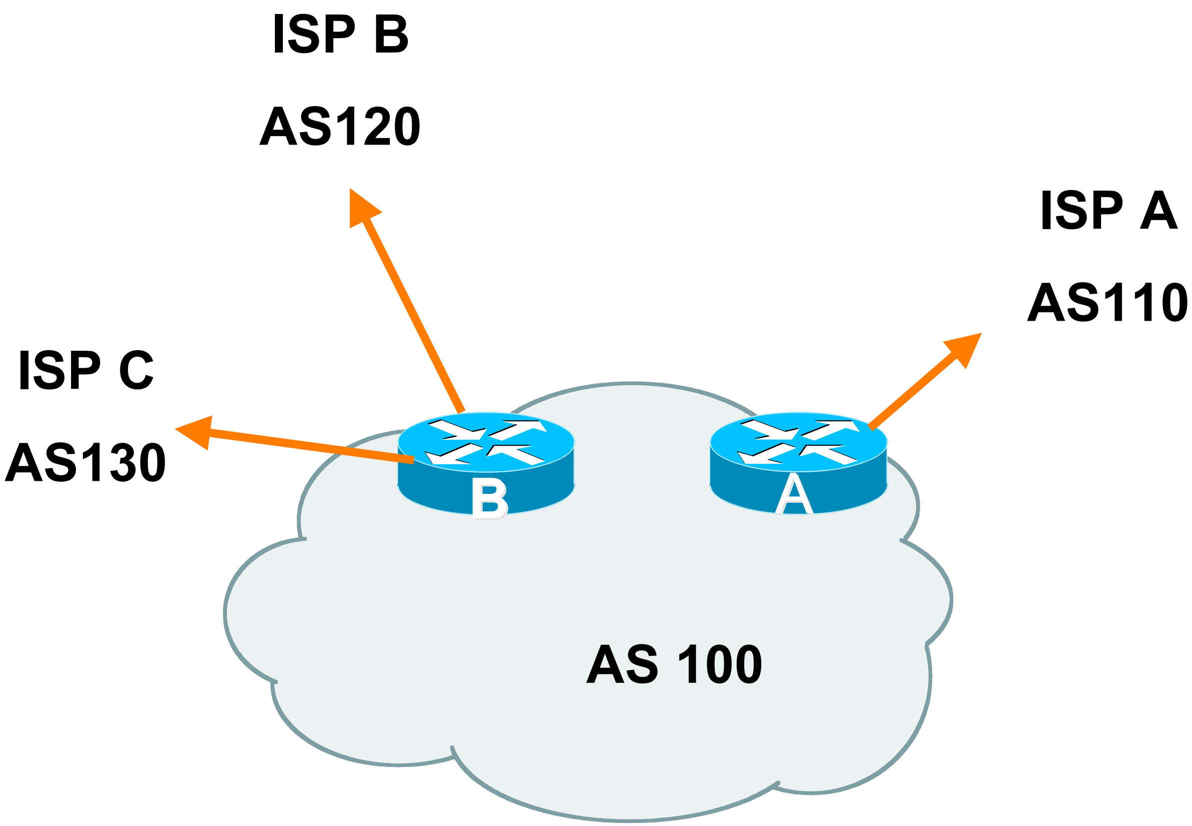
Service Provider Multihoming

Three upstreams, unequal bandwidths

Three upstreams, unequal bandwidths

- Autonomous System has three upstreams
  - 8Mbps to ISP A
  - 4Mbps to ISP B
  - 2Mbps to ISP C
- What is the strategy here?
  - One option is full table from each
    - 3x 270k prefixes ⇒ 810k paths
  - Other option is partial table and defaults from each
    - How??

Diagram



- Router A has 8Mbps circuit to ISP A
- Router B has 4Mbps and 2Mbps circuits to ISPs B&C

Outbound load-balancing strategy

- Available BGP feeds from Transit providers:
  - Full table
  - Customer prefixes and default
  - Default Route
- These are the common options
  - Very rare for any provider to offer anything different

Outbound load-balancing strategy

- Accept only a default route from the provider with the **largest** connectivity, ISP A
  - Because most of the traffic is going to use this link
- If ISP A won't provide a default:
  - Still run BGP with them, but discard all prefixes
  - Point static default route to the upstream link
  - Distribute the default in the IGP
- Request the full table from ISP B & C
  - Most of this will be thrown away
  - ("Default plus customers" is not enough)

Outbound load-balancing strategy

- How to decide what to keep and what to discard from ISPs B & C?

  Most traffic will use ISP A link — so we need to find a good/useful subset

- Discard prefixes transiting the global transit ISPs

  Global transit ISPs generally appear in most non-local or regional AS-PATHs

- Discard prefixes with ISP A's ASN in the path

  Makes more sense for traffic to those destinations to go via the link to ISP A

Outbound load-balancing strategy

- Global Transit ISPs include:

209	Qwest	3549	Global Crossing
701	VerizonBusiness	3356	Level 3
1239	Sprint	3561	Savvis
1668	AOL TDN	7018	AT&T
2914	NTT America		

ISP B peering Inbound AS-PATH filter

```
ip as-path access-list 1 deny _209_
ip as-path access-list 1 deny _701_
ip as-path access-list 1 deny _1239_
ip as-path access-list 1 deny _3356_
ip as-path access-list 1 deny _3549_
ip as-path access-list 1 deny _3561_
ip as-path access-list 1 deny _2914_
ip as-path access-list 1 deny _7018_
!
ip as-path access-list 1 deny _ISPA_
ip as-path access-list 1 deny _ISPC_
!
ip as-path access-list 1 permit _ISPB$
ip as-path access-list 1 permit _ISPB_[0-9]+$
ip as-path access-list 1 permit _ISPB_[0-9]+_[0-9]+$
ip as-path access-list 1 permit _ISPB_[0-9]+_[0-9]+_[0-9]+$
ip as-path access-list 1 deny   .*
```

Outbound load-balancing strategy: ISP B peering configuration

- Part 1: Dropping Global Transit ISP prefixes

  This can be fine-tuned if traffic volume is not sufficient

  (More prefixes in = more traffic out)

- Part 2: Dropping prefixes transiting ISP A & C network

- Part 3: Permitting prefixes from ISP B, their BGP neighbours, and their neighbours, and their neighbours

  More AS_PATH permit clauses, the more prefixes allowed in, the more egress traffic

  Too many prefixes in will mean more outbound traffic than the link to ISP B can handle

Outbound load-balancing strategy

- Similar AS-PATH filter can be built for the ISP C BGP peering
- If the same prefixes are heard from both ISP B and C, then establish proximity of their origin ASN to ISP B or C
  - e.g. ISP B might be in Japan, with the neighbouring ASN in Europe, yet ISP C might be in Europe
  - Transit to the ASN via ISP C makes more sense in this case

Inbound load-balancing strategy

- The largest outbound link should announce just the aggregate
- The other links should announce:

  a) The aggregate with AS-PATH prepend

  b) Subprefixes of the aggregate, chosen according to traffic volumes to those subprefixes, and according to the services on those subprefixes

- Example:

  Link to ISP B could be used just for Broadband/Dial customers — so number all such customers out of one contiguous subprefix

  Link to ISP C could be used just for commercial leased line customers — so number all such customers out of one contiguous subprefix

Router A: eBGP Configuration Example

```
router bgp 100
 network 100.10.0.0 mask 255.255.224.0
 neighbor 122.102.10.1 remote 110
 neighbor 122.102.10.1 prefix-list default in
 neighbor 122.102.10.1 prefix-list aggregate out
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list aggregate permit 100.10.0.0/19
!
```

Router B: eBGP Configuration Example

```
router bgp 100
 network 100.10.0.0 mask 255.255.224.0
 neighbor 120.103.1.1 remote 120
 neighbor 120.103.1.1 filter-list 1 in
 neighbor 120.103.1.1 prefix-list ISP-B out
 neighbor 120.103.1.1 route-map to-ISP-B out
 neighbor 121.105.2.1 remote 130
 neighbor 121.105.2.1 filter-list 2 in
 neighbor 121.105.2.1 prefix-list ISP-C out
 neighbor 121.105.2.1 route-map to-ISP-C out
!
ip prefix-list aggregate permit 100.10.0.0/19
!
..next slide
```

Router B: eBGP Configuration Example

```
ip prefix-list ISP-B permit 100.10.0.0/19
ip prefix-list ISP-B permit 100.10.0.0/21
!
ip prefix-list ISP-C permit 100.10.0.0/19
ip prefix-list ISP-C permit 100.10.28.0/22
!
route-map to-ISP-B permit 10
 match ip address prefix-list aggregate
 set as-path prepend 100
!
route-map to-ISP-B permit 20
!
route-map to-ISP-C permit 10
 match ip address prefix-list aggregate
 set as-path prepend 100 100
!
route-map to-ISP-C permit 20
```

/21 to ISP B "dial customers"

/22 to ISP C "biz customers"

e.g. Single prepend on ISP B link

e.g. Dual prepend on ISP C link

What about outbound backup?

- We have:

  Default route from ISP A by eBGP

  Mostly discarded full table from ISPs B&C

- Strategy:

  Originate default route by OSPF on Router A (with metric 10) — link to ISP A

  Originate default route by OSPF on Router B (with metric 30) — links to ISPs B & C

  Plus on Router B:

    Static default route to ISP B with distance 240

    Static default route to ISP C with distance 245

  When link goes down, static route is withdrawn

Outbound backup: steady state

- Steady state (all links up and active):

  Default route is to Router A — OSPF metric 10

  (Because default learned by eBGP ⇒ default is in RIB ⇒ OSPF will originate default)

  Backup default is to Router B — OSPF metric 20

  eBGP prefixes learned from upstreams distributed by iBGP throughout backbone

  (Default can be filtered in iBGP to avoid “RIB failure error”)

Outbound backup: failure examples

- Link to ISP A down, to ISPs B&C up:

  Default route is to Router B — OSPF metric 20

  (eBGP default gone from RIB, so OSPF on Router A withdraws the default)

- Above is true if link to B or C is down as well

- Link to ISPs B & C down, link to ISP A is up:

  Default route is to Router A — OSPF metric 10

  (static defaults on Router B removed from RIB, so OSPF on Router B withdraws the default)

Other considerations

- Default route should not be propagated to devices terminating non-transit peers and customers
- No need to carry default in iBGP
  - Filter out default in iBGP mesh peerings
- Still carry other eBGP prefixes across iBGP mesh
  - Otherwise routers will follow default route rules resulting in suboptimal traffic flow
  - Not a big issue because not carrying full table

Router A: iBGP Configuration Example

```
router bgp 100
 network 100.10.0.0 mask 255.255.224.0
 neighbor ibgp-peers peer-group
 neighbor ibgp-peers remote-as 100
 neighbor ibgp-peers prefix-list ibgp-filter out
 neighbor 100.10.0.2 peer-group ibgp-peers
 neighbor 100.10.0.2 prefix-list ibgp-filter out
 neighbor 100.10.0.3 peer-group ibgp-peers
 neighbor 100.10.0.3 prefix-list ibgp-filter out
!
ip prefix-list ibgp-filter deny 0.0.0.0/0
ip prefix-list ibgp-filter permit 0.0.0.0/0 le 32
!
```

Three upstreams, unequal bandwidths: Summary

- Example based on many deployed working multihoming/loadbalancing topologies
- Many variations possible — this one is:
  - Easy to tune
  - Light on border router resources
  - Light on backbone router infrastructure
  - Sparse BGP table ⇒ faster convergence

Summary

Summary

- Multihoming is not hard, really…

  Keep It Simple & Stupid!

- Full routing table is *rarely* required

  A default is often just as good

  If customers want 270k prefixes, charge them money for it

BGP Multihoming Techniques

End of Tutorial