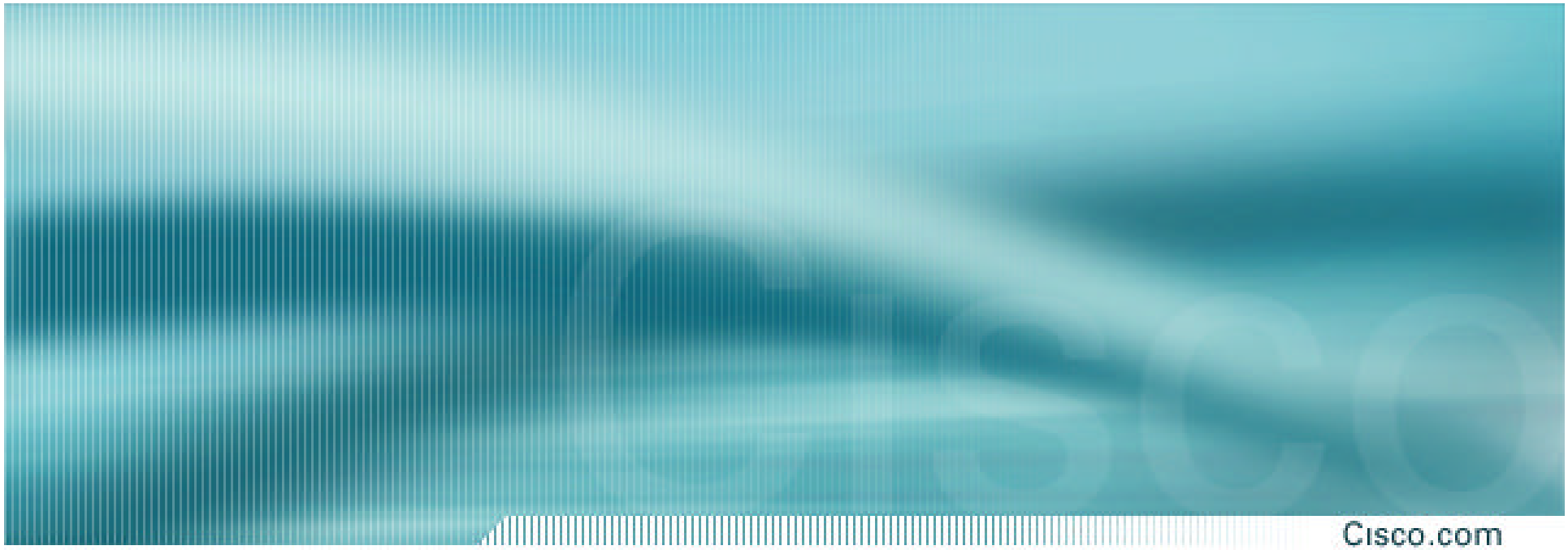


BGP and the Internet

Service Provider Multihoming

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**

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**

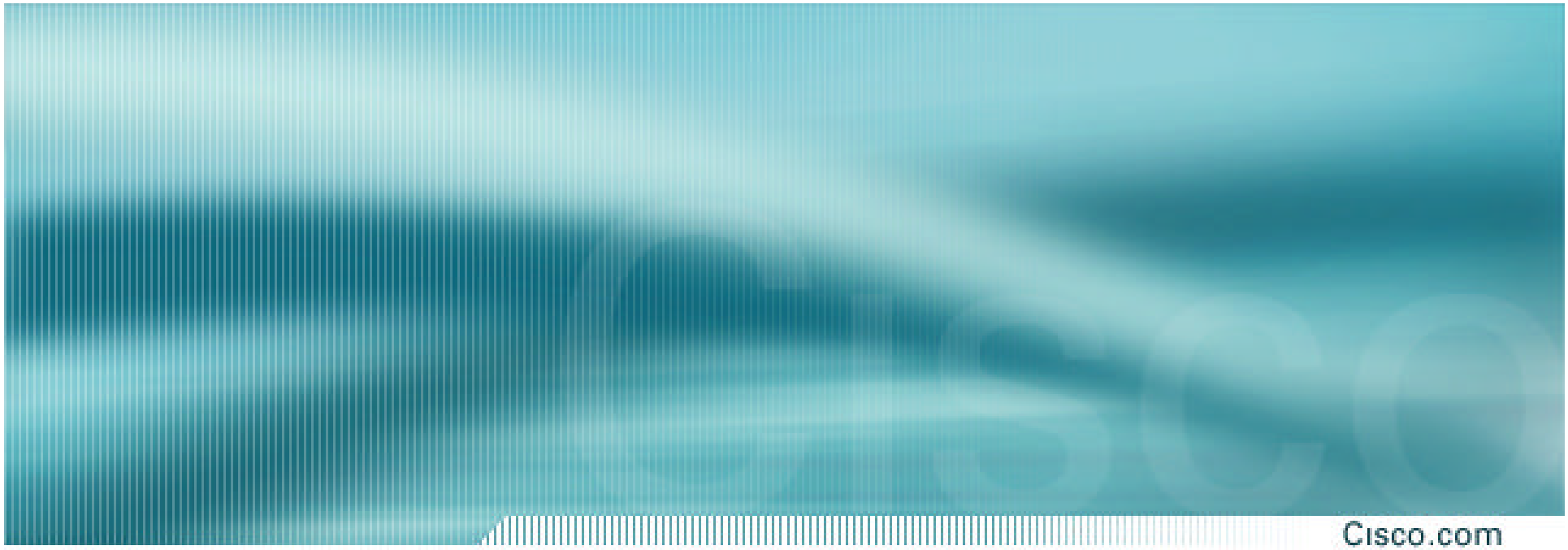
Service Provider Multihoming MYTHS!!

- 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

Cisco.com

- **Examples**
  - **One upstream, one local peer**
  - **One upstream, local exchange point**
  - **Two upstreams, one local peer**
  - **Tier-1 and regional upstreams, with local peers**
  - **Disconnected Backbone**
  - **IDC Multihoming**
- **All examples require BGP and a public ASN**



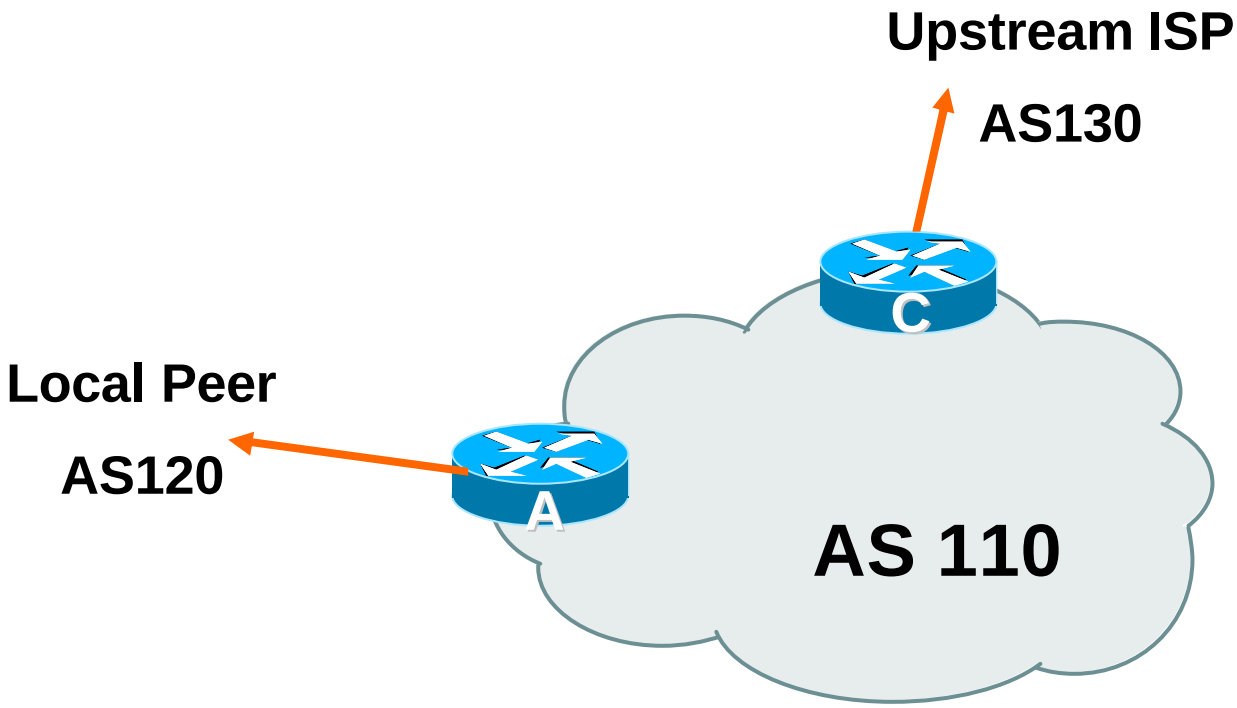
Service Provider Multihoming

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 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 120
 neighbor 222.222.10.2 prefix-list my-block out
 neighbor 222.222.10.2 prefix-list AS120-peer in
!
ip prefix-list AS120-peer permit 222.5.16.0/19
ip prefix-list AS120-peer permit 221.240.0.0/20
ip prefix-list my-block permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.224.0 null0
```

One Upstream, One Local Peer

- **Router A – Alternative Configuration**

```
router bgp 110
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 120
 neighbor 222.222.10.2 prefix-list my-block out
 neighbor 222.222.10.2 filter-list 10 in
!
ip as-path access-list 10 permit ^(120_)+$
!
ip prefix-list my-block permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.224.0 null0
```

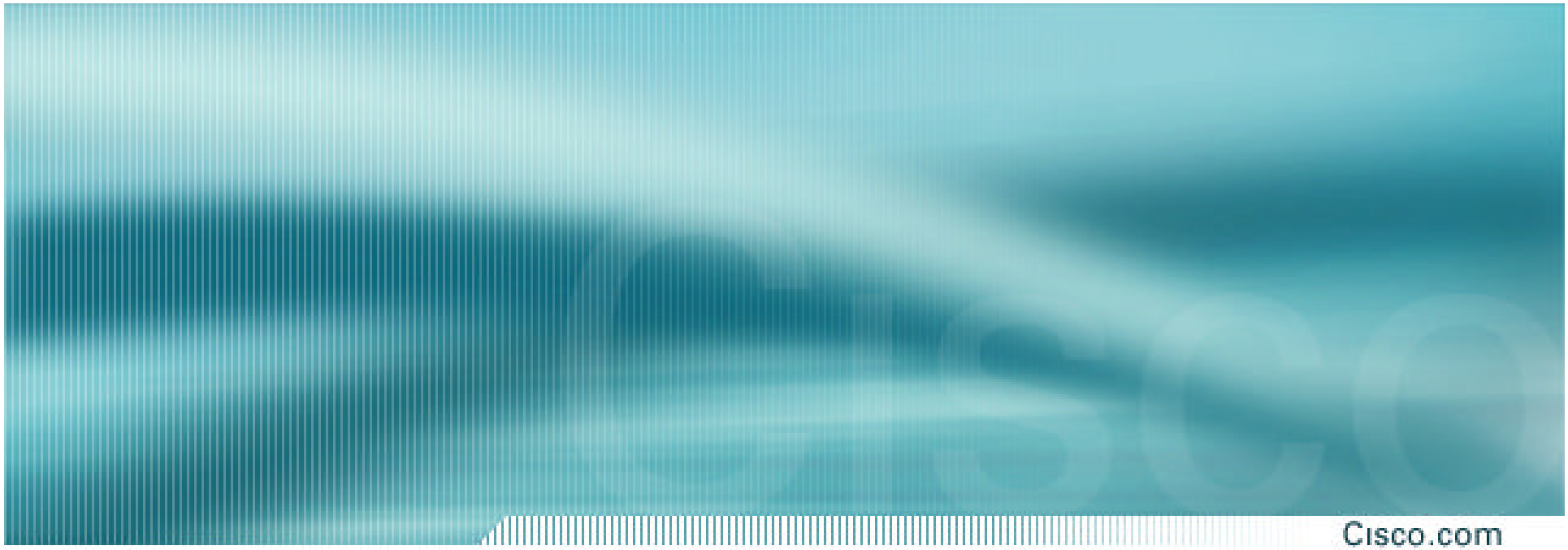
One Upstream, One Local Peer

- **Router C Configuration**

```
router bgp 110
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 130
 neighbor 222.222.10.1 prefix-list default in
 neighbor 222.222.10.1 prefix-list my-block out
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 221.10.0.0 255.255.224.0 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**



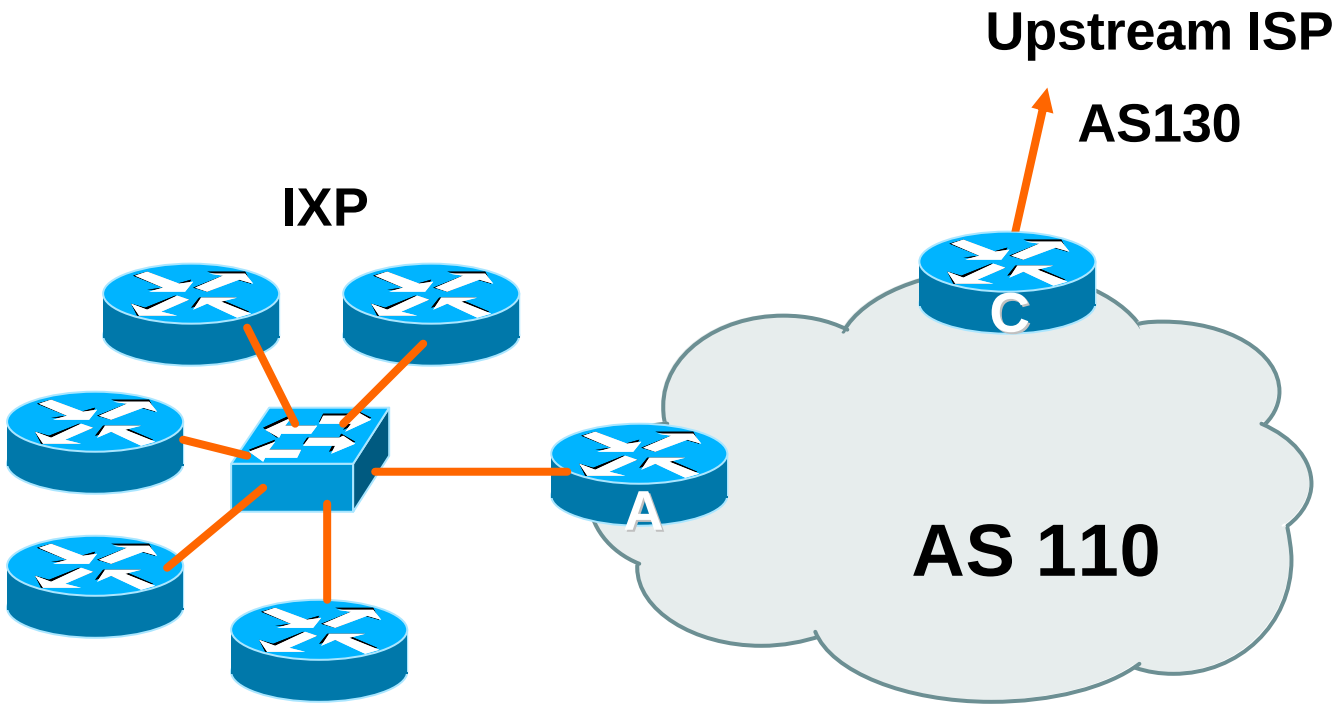
Service Provider Multihoming

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 from IXP peers

One Upstream, Local Exchange Point

- **Router A Configuration**

```
interface fastethernet 0/0
 description Exchange Point LAN
 ip address 220.5.10.1 mask 255.255.255.224
 ip verify unicast reverse-path
 no ip directed-broadcast
 no ip proxy-arp
 no ip redirects
!
router bgp 110
 network 221.10.0.0 mask 255.255.224.0
 neighbor ixp-peers peer-group
 neighbor ixp-peers soft-reconfiguration in
 neighbor ixp-peers prefix-list my-block out
..next slide
```

One Upstream, Local Exchange Point


```
neighbor 220.5.10.2 remote-as 100
neighbor 222.5.10.2 peer-group ixp-peers
neighbor 222.5.10.2 prefix-list peer100 in
neighbor 220.5.10.3 remote-as 101
neighbor 222.5.10.3 peer-group ixp-peers
neighbor 222.5.10.3 prefix-list peer101 in
neighbor 220.5.10.4 remote-as 102
neighbor 222.5.10.4 peer-group ixp-peers
neighbor 222.5.10.4 prefix-list peer102 in
neighbor 220.5.10.5 remote-as 103
neighbor 222.5.10.5 peer-group ixp-peers
neighbor 222.5.10.5 prefix-list peer103 in
..next slide
```

One Upstream, Local Exchange Point

Cisco.com

```
ip route 221.10.0.0 255.255.224.0 null0
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list peer100 permit 222.0.0.0/19
ip prefix-list peer101 permit 222.30.0.0/19
ip prefix-list peer102 permit 222.12.0.0/19
ip prefix-list peer103 permit 222.18.128.0/19
!
```

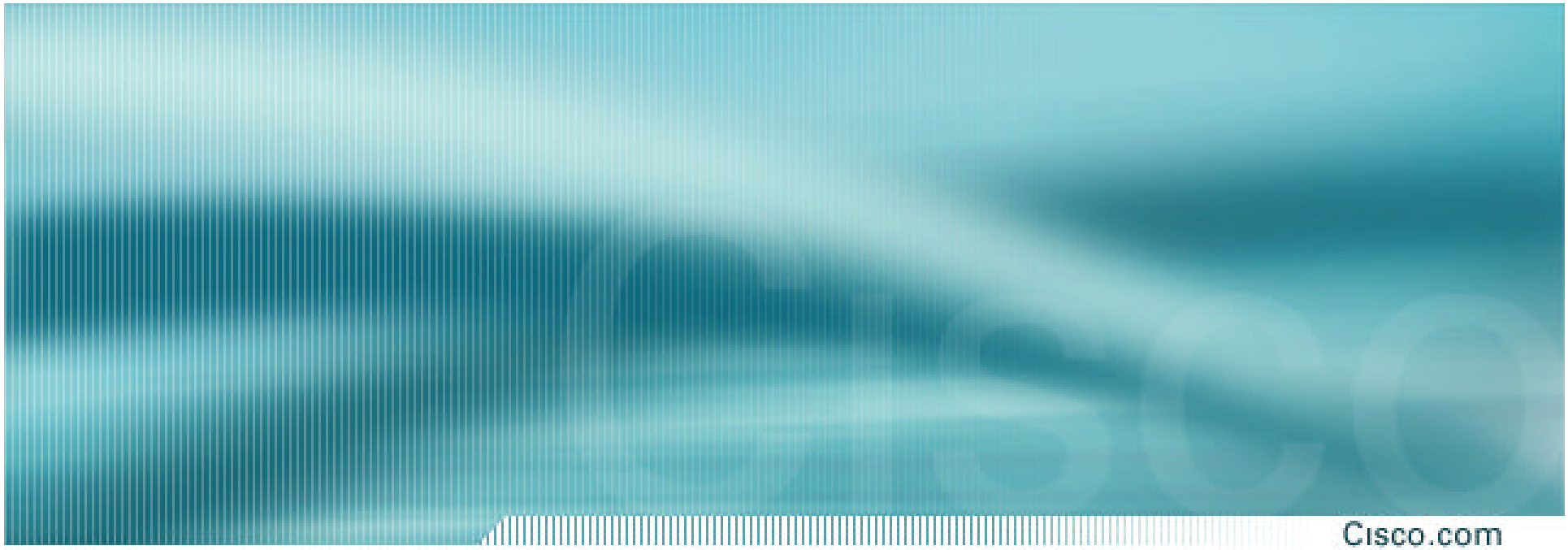
One Upstream, Local Exchange Point

- **Router C Configuration**

```
router bgp 110
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 130
 neighbor 222.222.10.1 prefix-list default in
 neighbor 222.222.10.1 prefix-list my-block out
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 221.10.0.0 255.255.224.0 null0
```

One Upstream, Local Exchange Point

- **Note Router A configuration**
  - **Prefix-list higher maintenance, but safer**
  - **uRPF on the FastEthernet interface**
- **IXP traffic goes to and from local IXP, everything else goes to upstream**



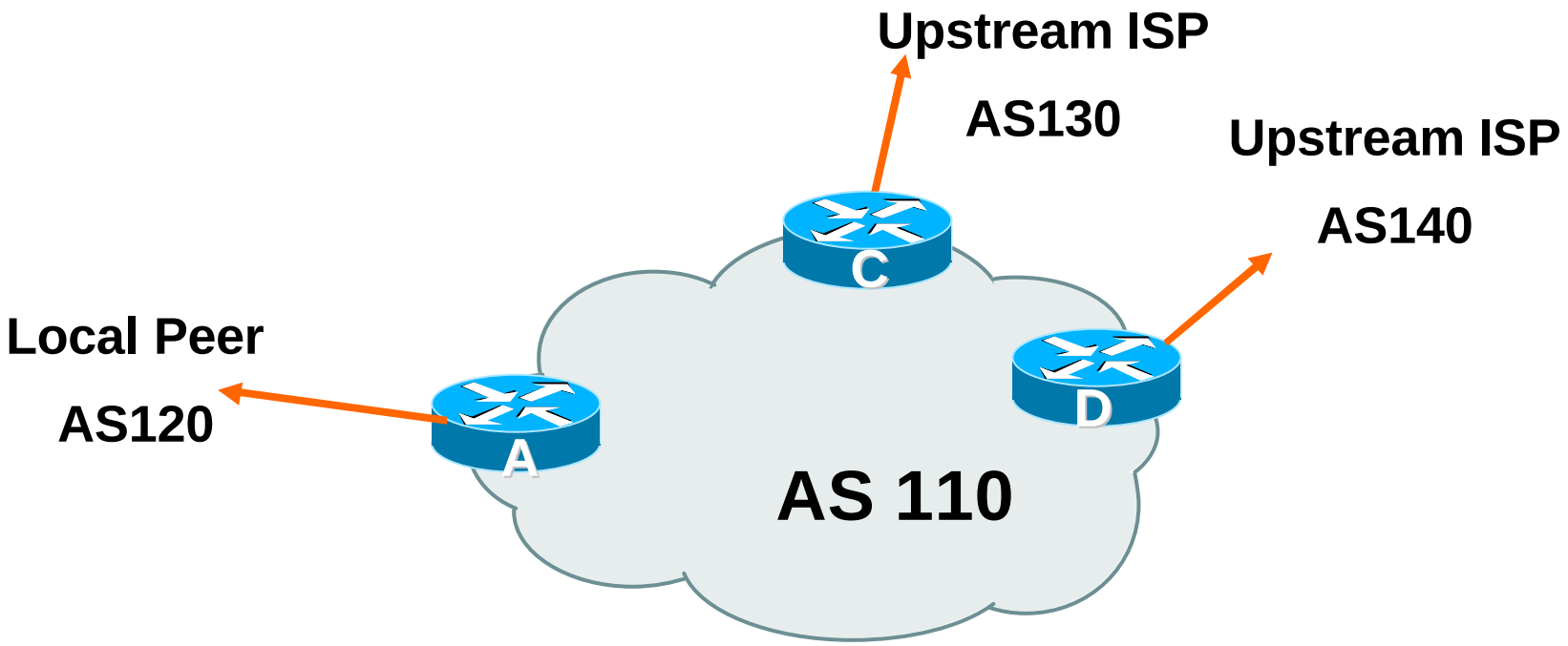
Service Provider Multihoming

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 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 130
 neighbor 222.222.10.1 prefix-list default in
 neighbor 222.222.10.1 prefix-list my-block out
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 221.10.0.0 255.255.224.0 null0
```

Two Upstreams, One Local Peer

- **Router D Configuration**

```
router bgp 110
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.5 remote-as 140
 neighbor 222.222.10.5 prefix-list default in
 neighbor 222.222.10.5 prefix-list my-block out
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 221.10.0.0 255.255.224.0 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 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 130
 neighbor 222.222.10.1 prefix-list rfc1918-deny in
 neighbor 222.222.10.1 prefix-list my-block out
 neighbor 222.222.10.1 route-map AS130-loadshare in
!
ip prefix-list my-block permit 221.10.0.0/19
! See earlier presentation for RFC1918 list
..next slide
```

Two Upstreams, One Local Peer Full Routes


```
ip route 221.10.0.0 255.255.224.0 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 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.5 remote-as 140
 neighbor 222.222.10.5 prefix-list rfc1918-deny in
 neighbor 222.222.10.5 prefix-list my-block out
!
ip prefix-list my-block permit 221.10.0.0/19
! See earlier in presentation for RFC1918 list
```

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

- **Router C Configuration**

```
router bgp 110
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 130
 neighbor 222.222.10.1 prefix-list rfc1918-nodef-deny in
 neighbor 222.222.10.1 prefix-list my-block out
 neighbor 222.222.10.1 filter-list 10 in
 neighbor 222.222.10.1 route-map tag-default-low in
!
..next slide
```

Two Upstreams, One Local Peer Partial Routes


```
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 221.10.0.0 255.255.224.0 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 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.5 remote-as 140
 neighbor 222.222.10.5 prefix-list default in
 neighbor 222.222.10.5 prefix-list my-block out
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
ip route 221.10.0.0 255.255.224.0 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 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 130
 neighbor 222.222.10.1 prefix-list rfc1918-deny in
 neighbor 222.222.10.1 prefix-list my-block out
 neighbor 222.222.10.1 filter-list 10 in
!
..next slide
```

Two Upstreams, One Local Peer Partial Routes


```
ip prefix-list my-block permit 221.10.0.0/19
! See earlier for RFC1918 list
!
ip route 221.10.0.0 255.255.224.0 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**

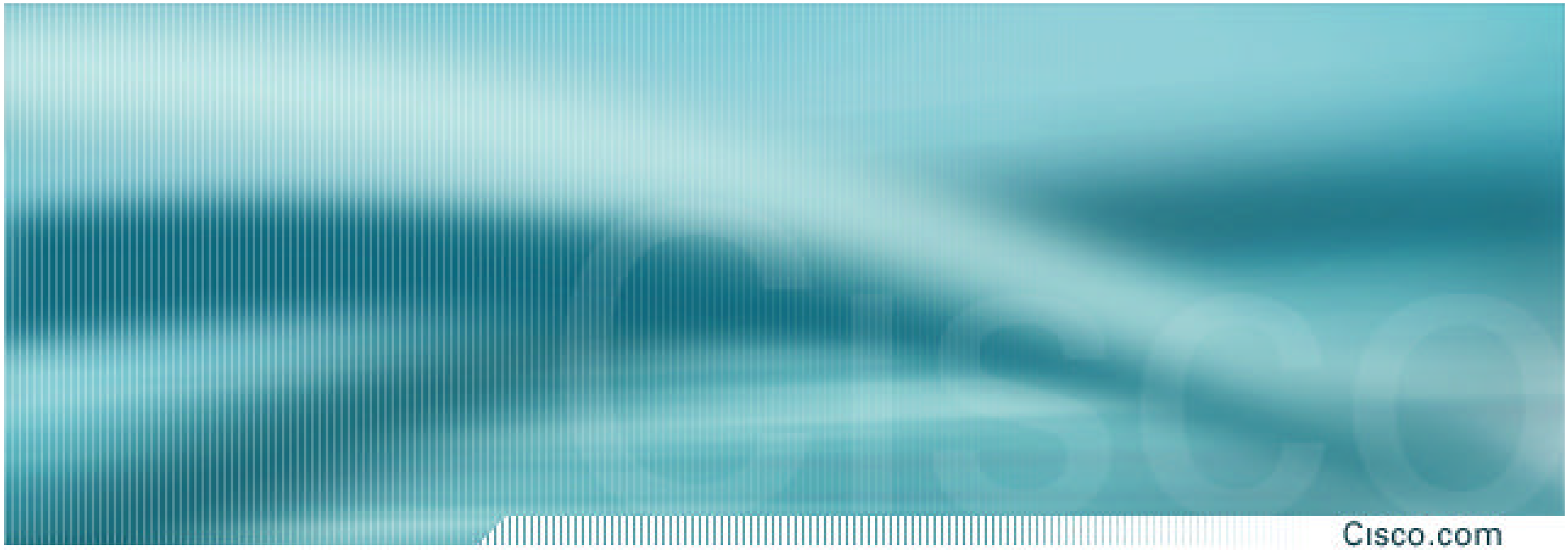
```
router ospf 110
 default-information originate metric 10
 passive-interface Serial 0/0
!
router bgp 110
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.5 remote-as 140
 neighbor 222.222.10.5 prefix-list deny-all in
 neighbor 222.222.10.5 prefix-list my-block out
!
..next slide
```

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 221.10.0.0/19
! See earlier in presentation for RFC1918 list
!
ip route 221.10.0.0 255.255.224.0 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**



Service Provider Multihoming

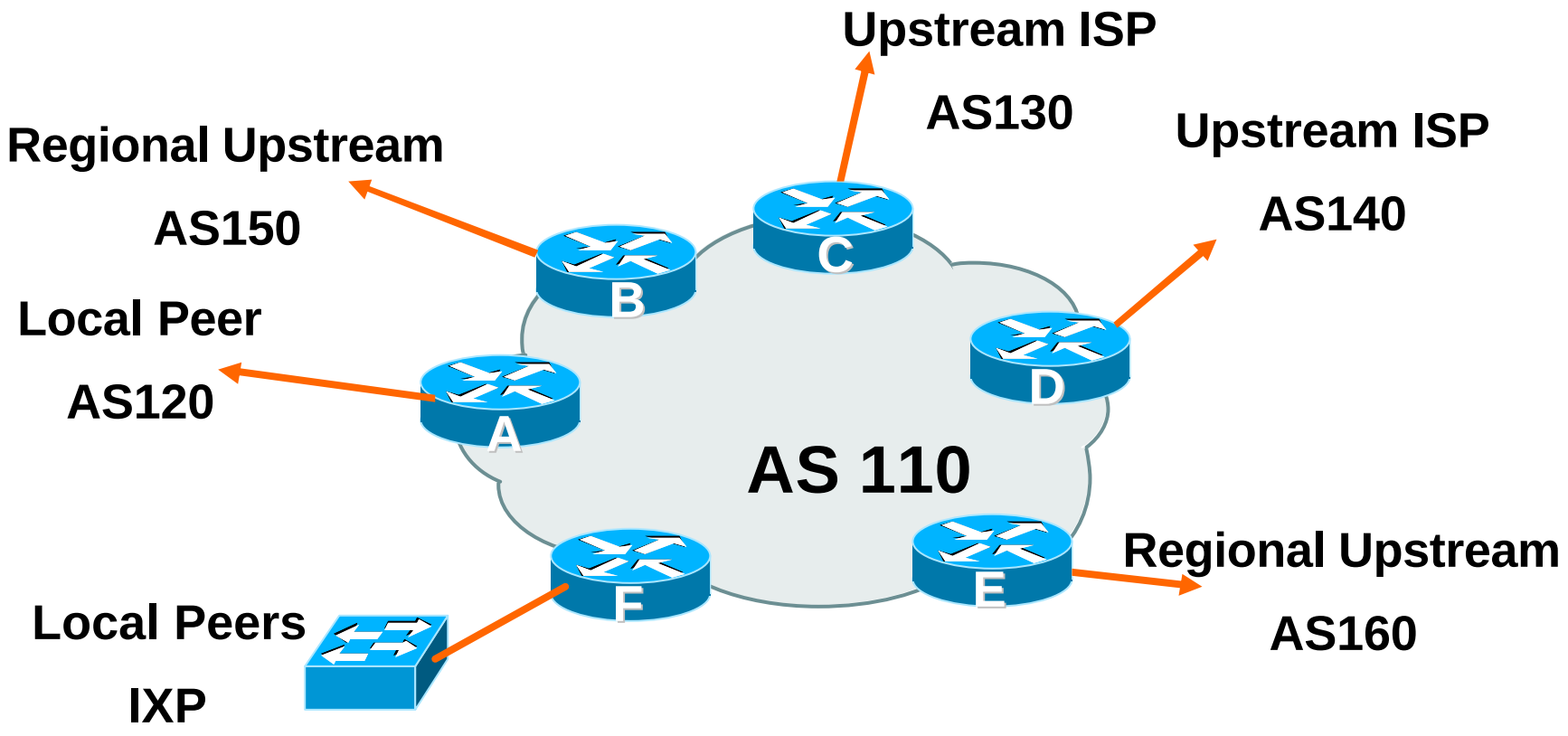
Two Tier-1 upstreams, two regional upstreams, and local peers

Tier-1 & Regional Upstreams, Local Peers

Cisco.com

- 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

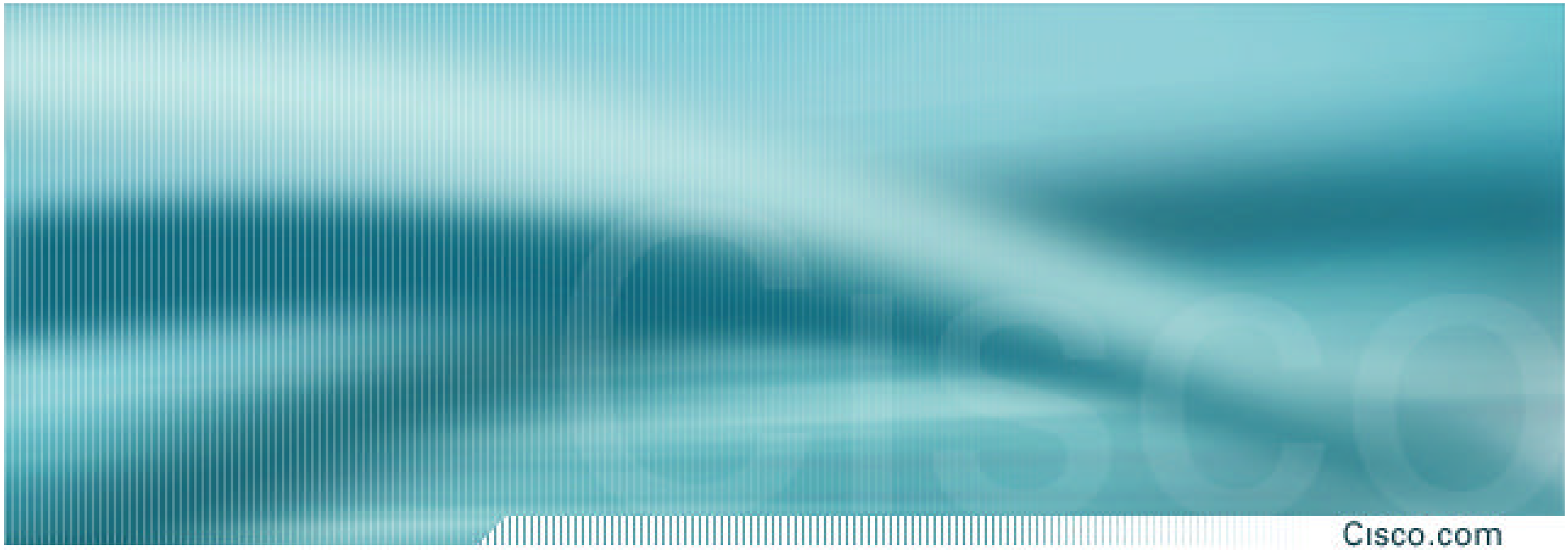
- Local traffic goes to local peer and IXP
- Regional traffic goes to two regional upstreams
- Everything else is shared between the two Tier-1s
- To modify loadsharing tweak what is heard from the two regionals and the first Tier-1
  - Best way is through modifying the AS-path filter

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**



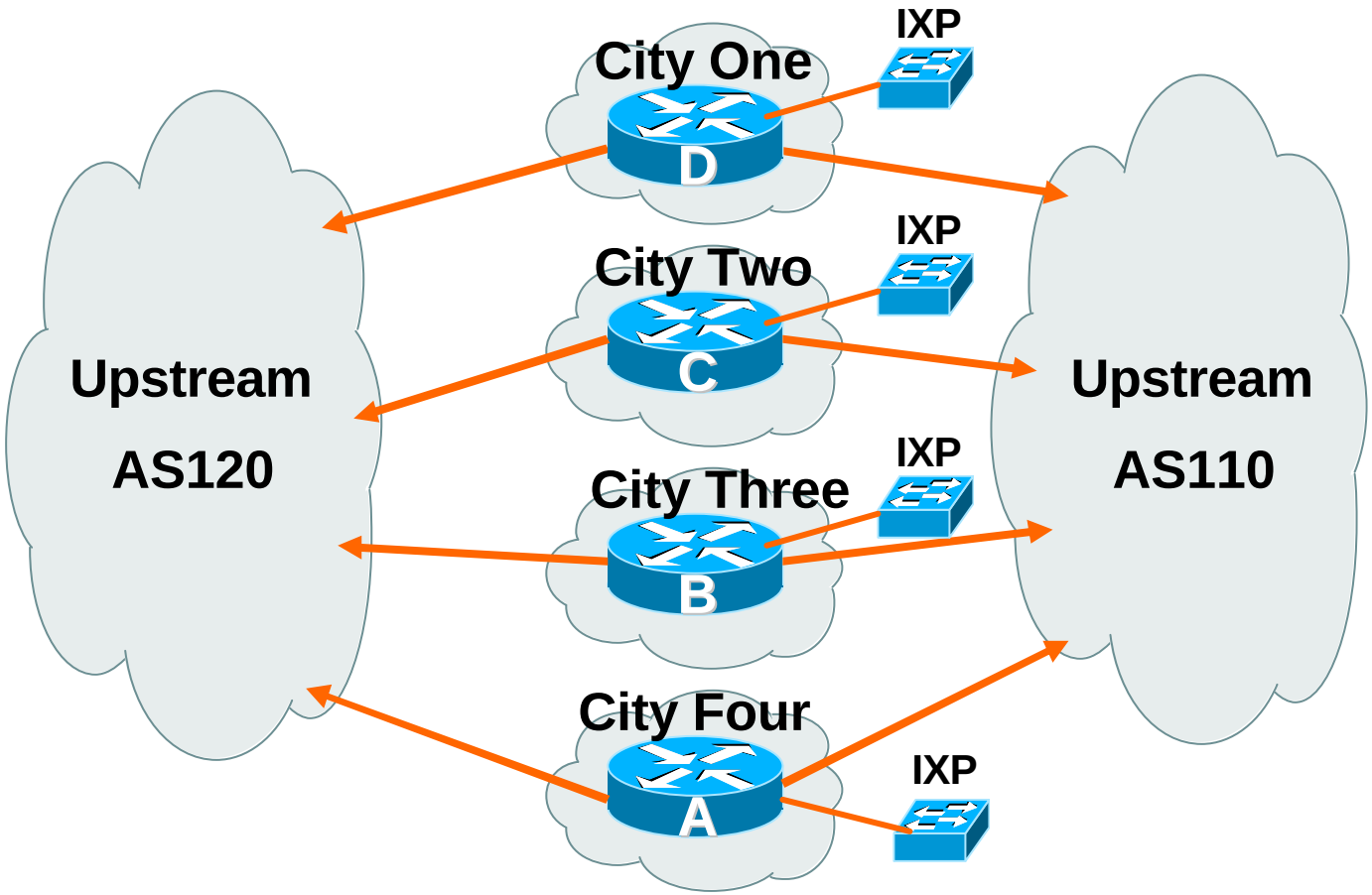
Service Provider Multihoming

Disconnected Backbone

Disconnected Backbone

- **ISP runs large network**
  - **Network has no backbone, only large PoPs in each location**
  - **Each PoP multihomes to upstreams**
  - **Common in some countries where backbone circuits are hard to obtain**
- **This is to show how it could be done**
  - **Not impossible, nothing “illegal”**

Disconnected Backbone



Disconnected Backbone

- **Works with one AS number**
  - **Not four – no BGP loop detection problem**
- **Each city operates as separate network**
  - **Uses defaults and selected leaked prefixes for loadsharing**
  - **Peers at local exchange point**

Disconnected Backbone

Cisco.com

- **Router A Configuration**

```
router bgp 100
 network 221.10.0.0 mask 255.255.248.0
 neighbor 222.200.0.1 remote-as 120
 neighbor 222.200.0.1 description AS120 – Serial 0/0
 neighbor 222.200.0.1 prefix-list default in
 neighbor 222.222.0.1 prefix-list my-block out
 neighbor 222.222.10.1 remote-as 110
 neighbor 222.222.10.1 description AS110 – Serial 1/0
 neighbor 222.222.10.1 prefix-list rfc1918-sua in
 neighbor 222.222.10.1 prefix-list my-block out
 neighbor 222.222.10.1 filter-list 10 in
!
```

…continued on next page…

Disconnected Backbone

Cisco.com

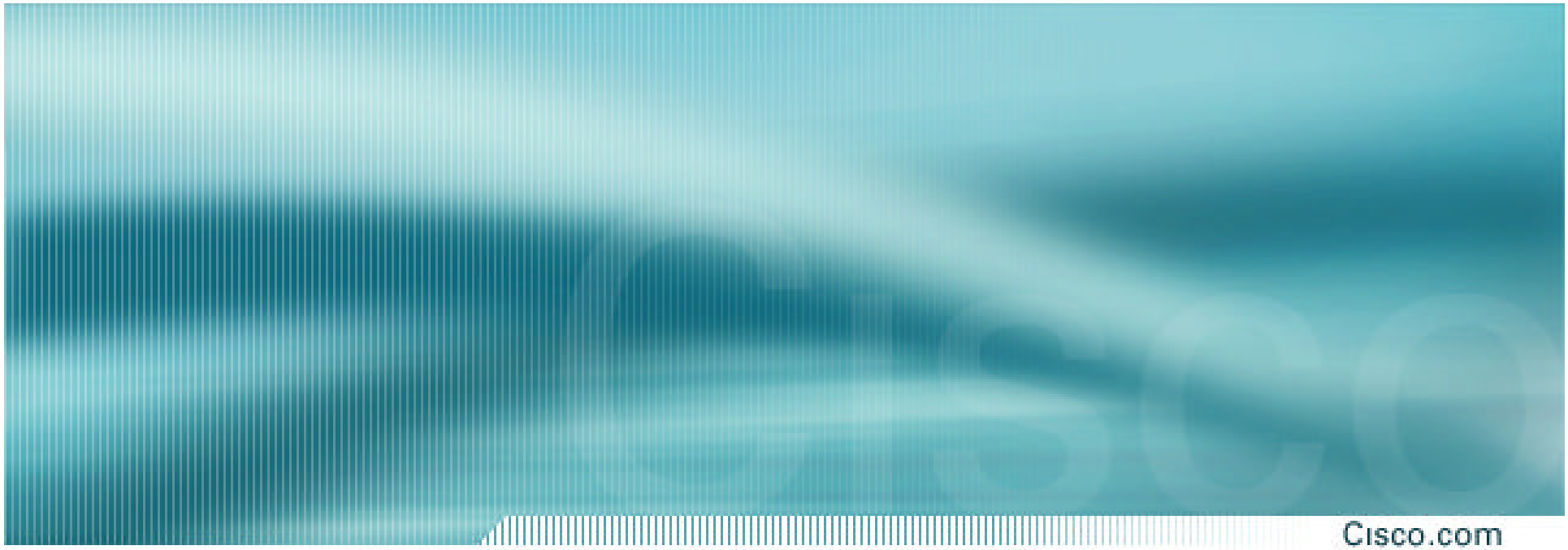
```
ip prefix-list my-block permit 221.10.0.0/21
ip prefix-list default permit 0.0.0.0/0
!
ip as-path access-list 10 permit ^(110_)+$
ip as-path access-list 10 permit ^(110_)+_[0-9]+$
!…etc to achieve outbound loadsharing
!
ip route 0.0.0.0 0.0.0.0 Serial 1/0 250
ip route 221.10.0.0 255.255.248.0 null0
!
```

Disconnected Backbone

- **Peer with AS120**
  - **Receive just default route**
  - **Announce /22 address**
- **Peer with AS110**
  - **Receive full routing table – filter with AS-path filter**
  - **Announce /22 address**
  - **Point backup static default – distance 252 – in case AS120 goes down**

Disconnected Backbone

- **Default ensures that disconnected parts of AS100 are reachable**
  - **Static route backs up AS120 default**
  - **No BGP loop detection – relying on default route**
- **Do not announce /19 aggregate**
  - **No advantage in announcing /19 and could lead to problems**

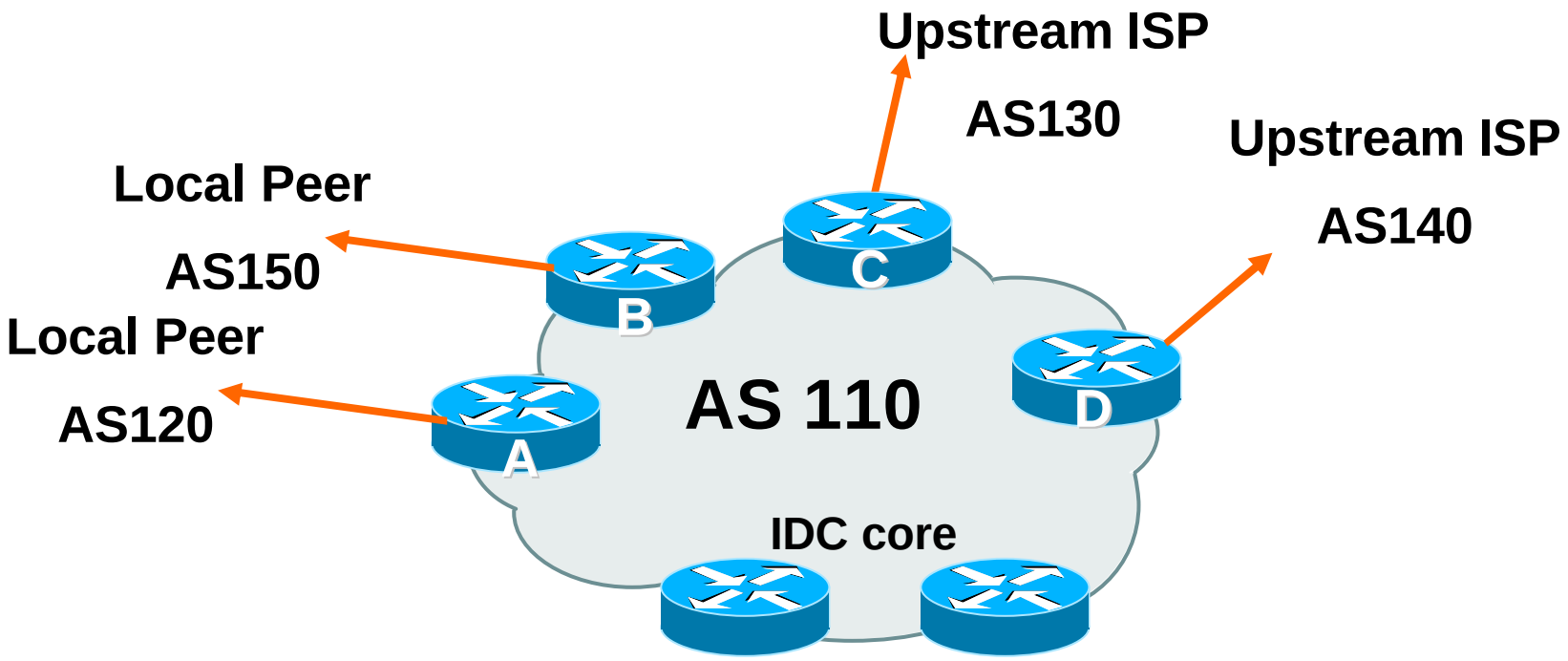


IDC Multihoming

IDC Multihoming

- **IDCs typically are not registry members so don't get their own address block**
  - **Situation also true for small ISPs and "Enterprise Networks"**
- **Smaller address blocks being announced**
  - **Address space comes from both upstreams**
  - **Should be apportioned according to size of circuit to upstream**
- **Outbound traffic paths matter**

Two Upstreams, Two Local Peers IDC



Assigned /24 from AS130 and /23 from AS140.

Circuit to AS130 is 2Mbps, circuit to AS140 is 4Mbps

IDC Multihoming

- **Router A and B configuration**

  In: Should accept all routes from AS120 and AS150

  Out: Should announce all address space to AS120 and AS150

  Straightforward

IDC Multihoming

- **Router C configuration**
  - **In: Accept partial routes from AS130**
    - **e.g. ^130_[0-9]+$**
  - **In: Ask for a route to use as default**
    - **set local preference on default to 80**
  - **Out: Send /24, and send /23 with AS-PATH prepend of one AS**

IDC Multihoming

- **Router D configuration**
  - **In: Ask for a route to use as default**
    - **Leave local preference of default at 100**
  - **Out: Send /23, and send /24 with AS-PATH prepend of one AS**

IDC Multihoming
Fine Tuning

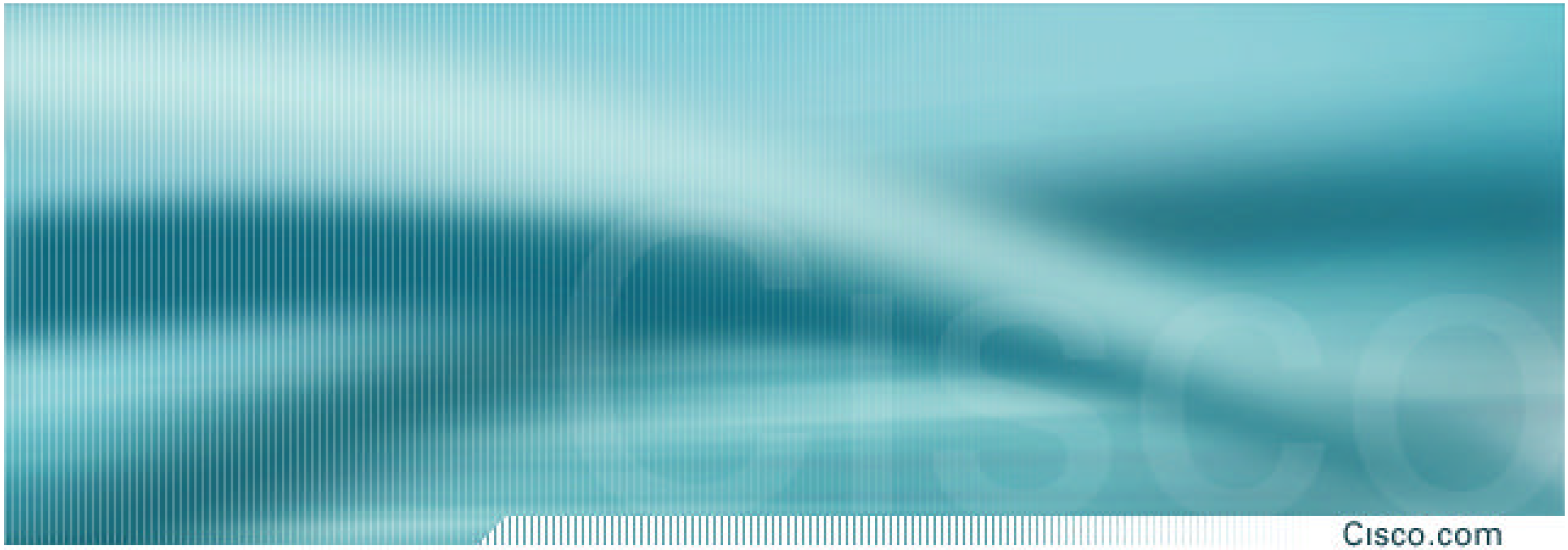
- **For local fine tuning, increase circuit capacity**
  - **Local circuits usually are cheap**
  - **Otherwise…**
- **For longer distance fine tuning**
  - **In: Modify as-path filter on Router C**
  - **Out: Modify as-path prepend on Routers C and D**
  - **Outbound traffic flow is usual critical for an IDC so inbound policies need to be carefully thought out**

IDC Multihoming
Other Details

- **Redundancy**
  - **Circuits are terminated on separate routers**
- **Apply thought to address space use**
  - **Request from both upstreams**
  - **Utilise address space evenly across IDC**
    - **Don’t start with /23 then move to /24 – use both blocks at the same time in the same proportion**
    - **Helps with loadsharing – yes, really!**

IDC Multihoming
Other Details

- **What about failover?**
  - **/24 and /23 from upstreams' blocks announced to the Internet routing table all the time**
  - **No obvious alternative at the moment**
    - **Conditional advertisement can help in steady state, but subprefixes still need to be announced in failover condition**



BGP and the Internet

Service Provider Multihoming