BGP Multihoming

APRICOT 2001 Tutorial – Kuala Lumpur

1st March 2001

Cisco Systems

Introduction

- **Presenter:**

  Philip Smith, Consulting Engineer

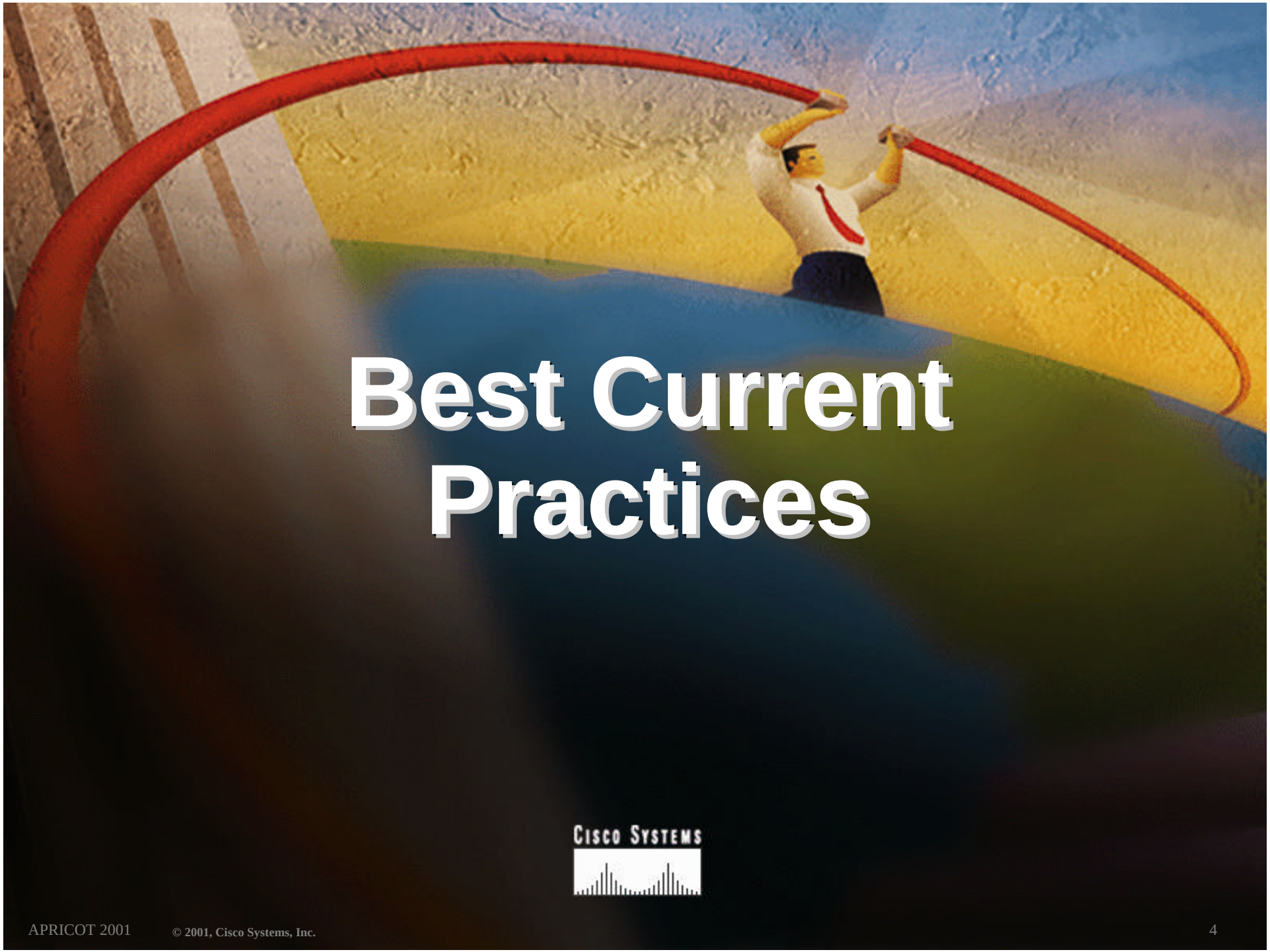
  Office of the CTO, Cisco Systems

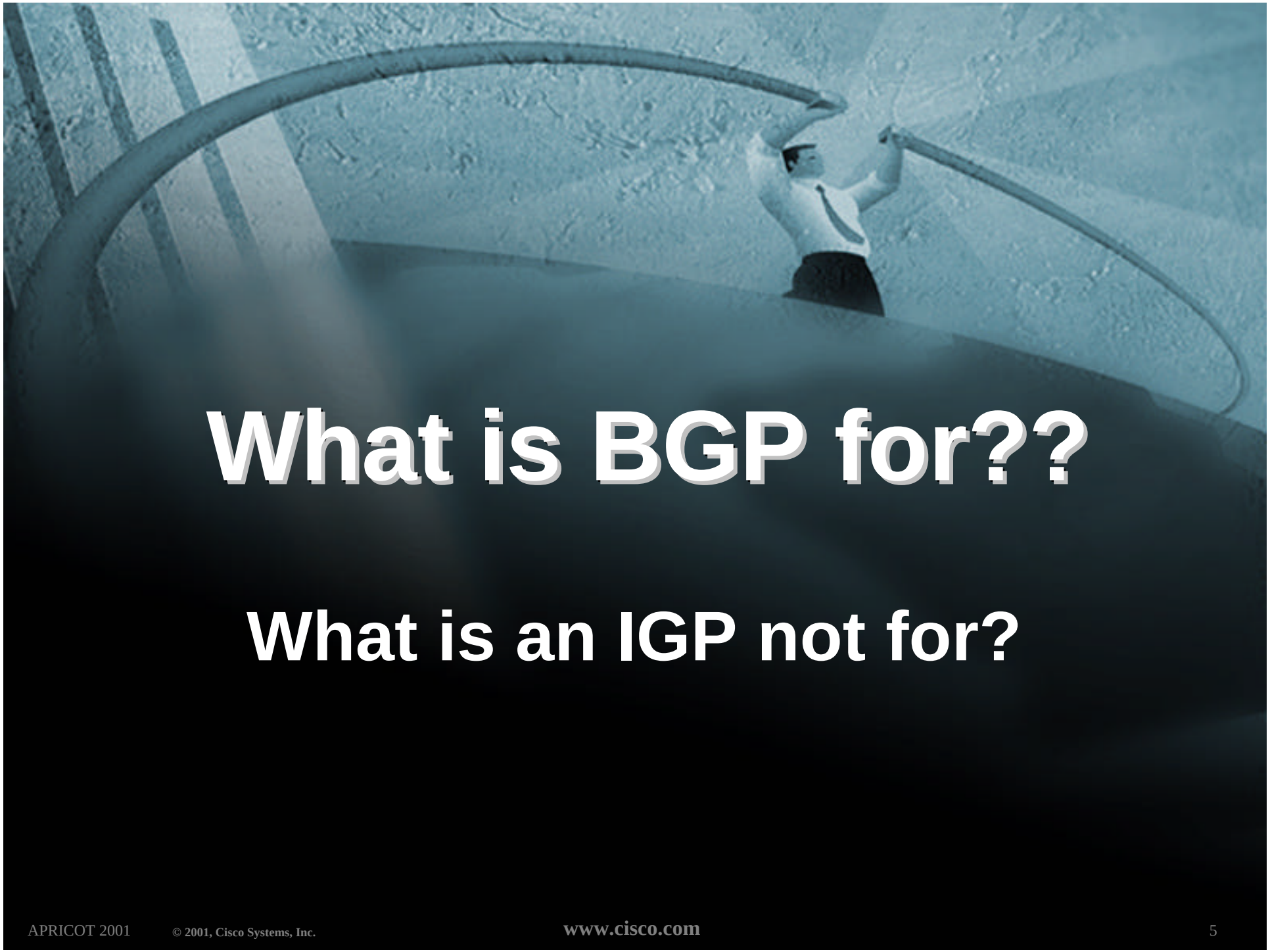
  e-mail: pfs@cisco.com

- **Please ask questions**

Agenda

- Best Current Practices
- Multihoming Examples
- Transit and IXPs
- Some more Examples





What is BGP for??

What is an IGP not for?

BGP versus OSPF/ISIS

- **Internal Routing Protocols (IGPs)**
  - **examples are ISIS and OSPF**
  - **used for carrying infrastructure addresses**
  - **NOT used for carrying Internet prefixes or customer prefixes**

BGP versus OSPF/ISIS

- BGP used internally (iBGP) and externally (eBGP)
- iBGP used to carry
  - some/all Internet prefixes across backbone
  - customer prefixes
- eBGP used to
  - exchange prefixes with other ASes
  - implement routing policy

BGP versus OSPF/ISIS

- **DO NOT:**

  distribute BGP prefixes into an IGP

  distribute IGP routes into BGP

  use an IGP to carry customer prefixes

- **YOUR NETWORK WILL NOT SCALE**

Announcing Prefixes

Route Aggregation

Aggregation

- **ISPs receive address block from Regional Registry or upstream provider**
- **Aggregation means announcing the address block only, not subprefixes**
- **Aggregate should be generated internally**

Configuring Aggregation - Cisco IOS

- ISP has 221.10.0.0/19 address block
- To put into BGP as an aggregate:

```
router bgp 100
 network 221.10.0.0 mask 255.255.224.0
ip route 221.10.0.0 255.255.224.0 null0
```

- The static route is a “pull up” route

  more specific prefixes within this address block ensure connectivity to ISP’s customers

  “longest match lookup”

Aggregation

- **Address block should be announced to the Internet as an aggregate**
- **Subprefixes of address blocks should NOT be announced to the Internet unless they provide additional and unique reachability information**

Announcing Aggregate - Cisco IOS

- **Configuration Example**

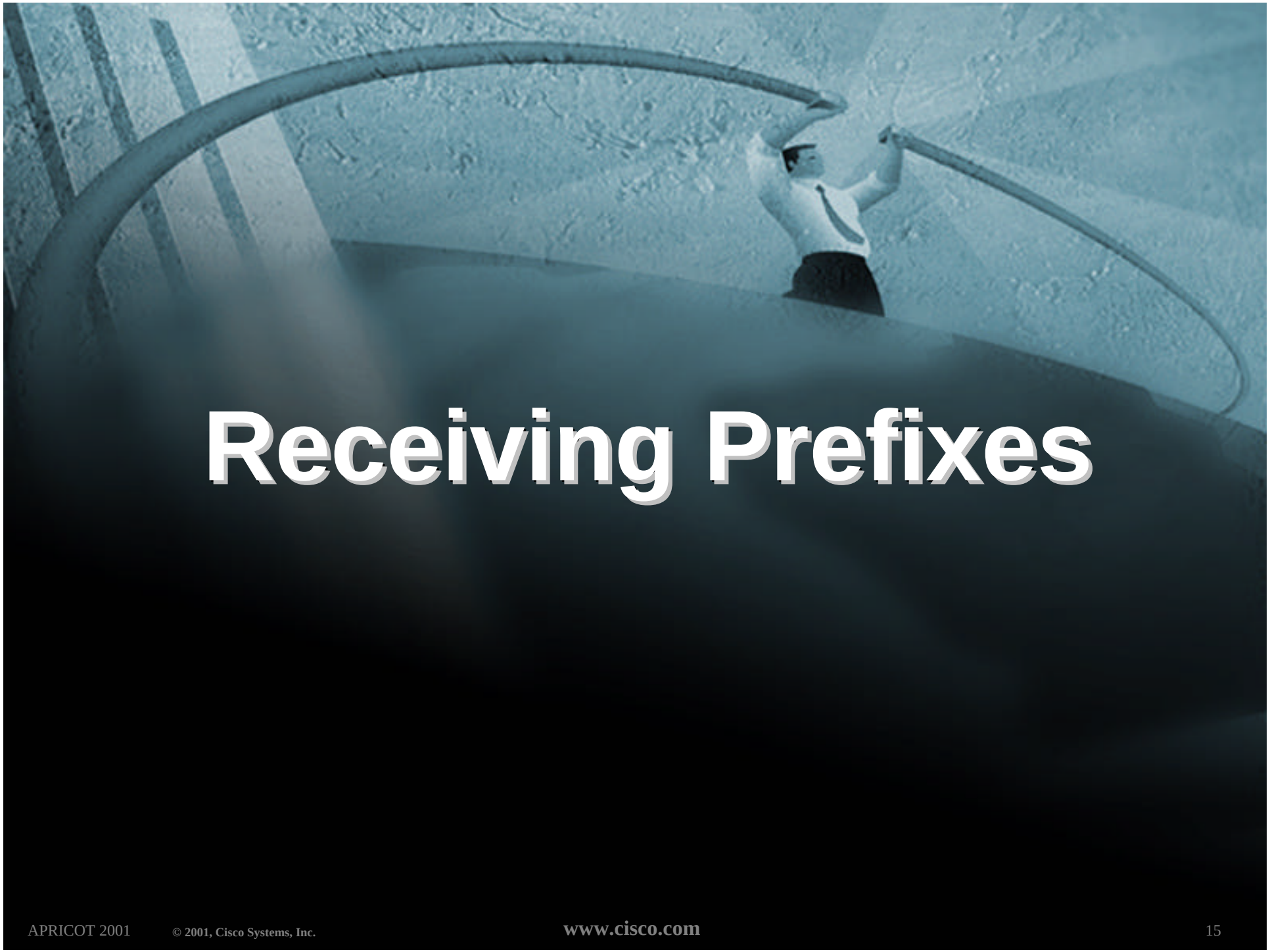
```
router bgp 100
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 101
 neighbor 222.222.10.1 prefix-list out-filter out
!
ip route 221.10.0.0 255.255.224.0 null0
!
ip prefix-list out-filter permit 221.10.0.0/19
ip prefix-list out-filter deny 0.0.0.0/0 le 32
```

Announcing an Aggregate

- ISPs who don't and won't aggregate are held in poor regard by community
- Registries' minimum allocation size is now a /20

  no real reason to see anything longer than a /21 prefix in the Internet

  BUT there are currently almost 60000 /24s!



Receiving Prefixes

Receiving Prefixes from downstream peers

- ISPs should only accept prefixes which have been assigned or allocated to their downstream peer
- For example

  downstream has 220.50.0.0/20 block

  should only announce this to peers

  peers should only accept this from them

Receiving Prefixes - Cisco IOS

- **Configuration Example on upstream**

```
router bgp 100
 neighbor 222.222.10.1 remote-as 101
 neighbor 222.222.10.1 prefix-list customer in
!
ip prefix-list customer permit 220.50.0.0/20
ip prefix-list customer deny 0.0.0.0/0 le 32
```

Receiving Prefixes from upstream peers

- **Not desirable unless really necessary**

  special circumstances

- **Ask upstream to either:**

  originate a default-route

  announce one prefix you can use as default

Receiving Prefixes from upstream peers

- **Downstream Router Configuration**

```
router bgp 100
 network 221.10.0.0 mask 255.255.224.0
 neighbor 221.5.7.1 remote-as 101
 neighbor 221.5.7.1 prefix-list infilt in
 neighbor 221.5.7.1 prefix-list outfilt out
!
ip prefix-list infilt permit 0.0.0.0/0
ip prefix-list infilt deny 0.0.0.0/0 le 32
!
ip prefix-list outfilt permit 221.10.0.0/19
ip prefix-list outfilt deny 0.0.0.0/0 le 32
```

Receiving Prefixes from upstream peers

- **Upstream Router Configuration**

```
router bgp 101
 neighbor 221.5.7.2 remote-as 100
 neighbor 221.5.7.2 default-originate
 neighbor 221.5.7.2 prefix-list cust-in in
 neighbor 221.5.7.2 prefix-list cust-out out
!
ip prefix-list cust-in permit 221.10.0.0/19
ip prefix-list cust-in deny 0.0.0.0/0 le 32
!
ip prefix-list cust-out permit 0.0.0.0/0
ip prefix-list cust-out deny 0.0.0.0/0 le 32
```

Receiving Prefixes from upstream peers

- **If necessary to receive prefixes from upstream provider, care is required**
  - **don’t accept RFC1918 etc prefixes**
  - **don’t accept your own prefix**
  - **don’t accept default (unless you need it)**
  - **don’t accept prefixes longer than /24**

Receiving Prefixes

```
router bgp 100
 network 221.10.0.0 mask 255.255.224.0
 neighbor 221.5.7.1 remote-as 101
 neighbor 221.5.7.1 prefix-list in-filter in
!
ip prefix-list in-filter deny 0.0.0.0/0                ! Block default
ip prefix-list in-filter deny 0.0.0.0/8 le 32
ip prefix-list in-filter deny 10.0.0.0/8 le 32
ip prefix-list in-filter deny 127.0.0.0/8 le 32
ip prefix-list in-filter deny 169.254.0.0/16 le 32
ip prefix-list in-filter deny 172.16.0.0/12 le 32
ip prefix-list in-filter deny 192.0.2.0/24 le 32
ip prefix-list in-filter deny 192.168.0.0/16 le 32
ip prefix-list in-filter deny 221.10.0.0/19 le 32     ! Block local prefix
ip prefix-list in-filter deny 224.0.0.0/3 le 32       ! Block multicast
ip prefix-list in-filter deny 0.0.0.0/0 ge 25         ! Block prefixes >/24
ip prefix-list in-filter permit 0.0.0.0/0 le 32
```

Receiving Prefixes

- **The prefix-list in-filter**

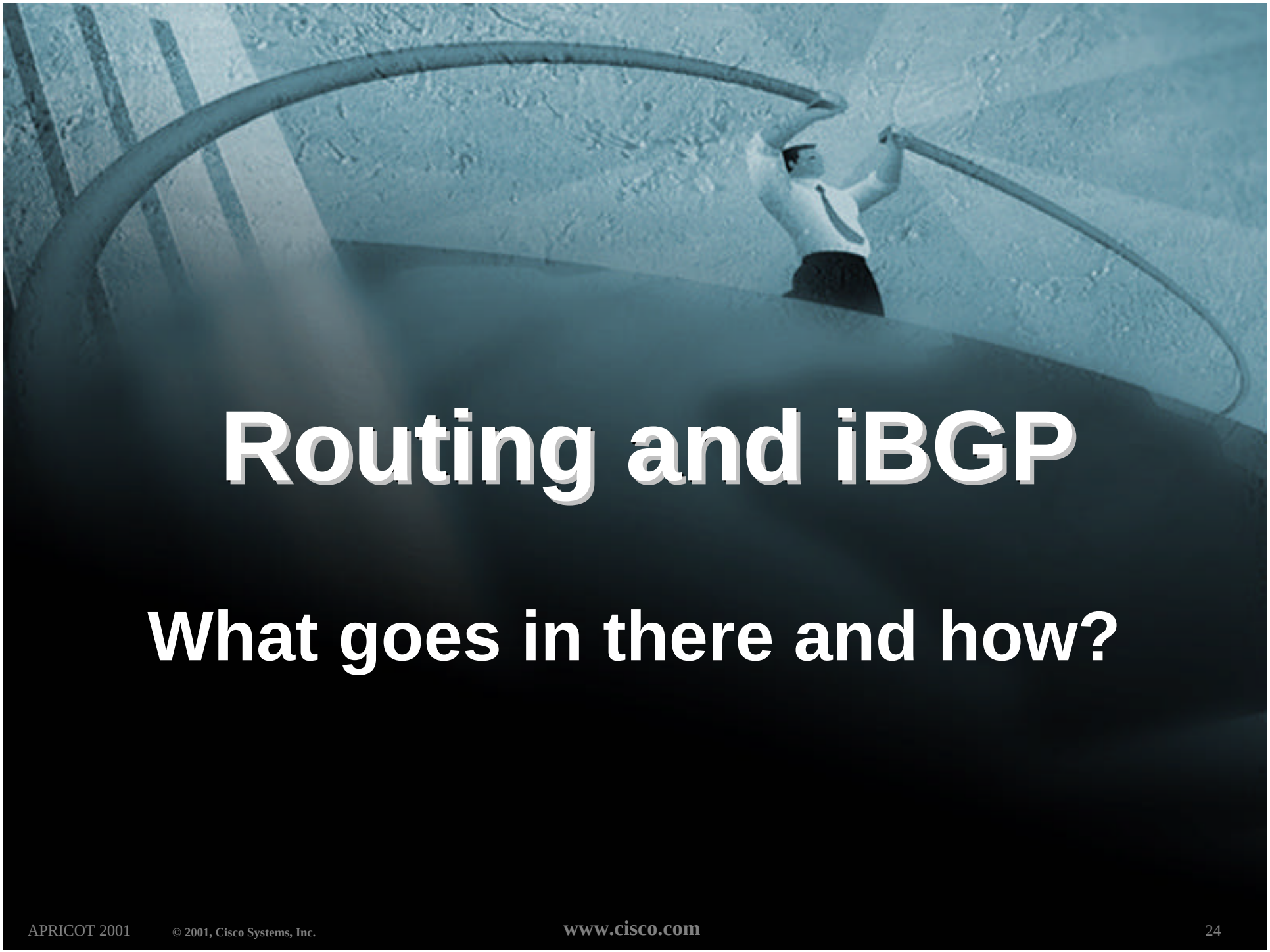
  MUST be applied to all BGP peerings where you are not specifically filtering prefixes

  CAN be applied inbound and outbound

  (but watch the logic)

- **Motivation behind filtering these prefixes is documented in:**

  http://www.ietf.org/internet-drafts/draft-manning-dsua-06.txt



Routing and iBGP

What goes in there and how?

Injecting prefixes into iBGP

- Use iBGP to carry customer prefixes

  Do not use IGP

- Point static route to customer interface
- Use BGP network statement
- As long as static route exists (interface active), prefix will be in BGP

Router Configuration network statement

- **Example:**

```
interface loopback 0
 ip address 215.17.3.1 255.255.255.255
!
interface Serial 5/0
 ip unnumbered loopback 0
 ip verify unicast reverse-path
!
ip route 215.34.10.0 255.255.252.0 Serial 5/0
!
router bgp 100
 network 215.34.10.0 mask 255.255.252.0
```

Injecting prefixes into iBGP

- **interface flap will result in prefix withdraw and reannounce**

  use “ip route…permanent”

- **many ISPs use redistribute static rather than network statement**

  only use this if you understand why

Router Configuration redistribute static

- **Example:**

```
ip route 215.34.10.0 255.255.252.0 Serial 5/0
!
router bgp 100
 redistribute static route-map static-to-bgp
<snip>
!
route-map static-to-bgp permit 10
 match ip address prefix-list ISP-block
 set origin igp
<snip>
!
ip prefix-list ISP-block permit 215.34.10.0/22 le 30
!
```

Injecting prefixes into iBGP

- **Route-map ISP-block can be used for many things:**
  - **setting communities and other attributes**
  - **setting origin code to IGP, etc**
- **Be careful with prefix-lists and route-maps**
  - **absence of either/both means all statically routed prefixes go into iBGP**

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

Multihoming

- The scenarios described here apply equally well to end sites being customers of ISPs and ISPs being customers of other ISPs
- Implementation detail may be different

end site ® ISP	ISP controls config
ISP1 ® ISP2	ISPs share config

AS Numbers

- An Autonomous System Number is required by BGP
- Obtained from upstream ISP or Regional Registry
- Necessary when you have links to more than one ISP or exchange point

Configuring Policy

- **Assumptions:**

  prefix-lists are used throughout

  easier/better/faster than access-lists

- **Three BASIC Principles**

  prefix-lists to filter prefixes

  filter-lists to filter ASNs

  route-maps to apply policy

Originating Prefixes

- **Basic Assumptions**

  MUST announce assigned address block to Internet

  MAY also announce subprefixes - reachability is not guaranteed

  RIR minimum allocation is /20 - several ISPs filter RIR blocks on this boundary - “Net Police”

Part of the "Net Police" prefix list

```
!! RIPE
ip prefix-list FILTER permit 62.0.0.0/8 ge 12 le 20
ip prefix-list FILTER permit 193.0.0.0/8 ge 12 le 20
ip prefix-list FILTER permit 194.0.0.0/7 ge 12 le 20
ip prefix-list FILTER permit 212.0.0.0/7 ge 12 le 20
!! APNIC
ip prefix-list FILTER permit 61.0.0.0/8 ge 12 le 20
ip prefix-list FILTER permit 202.0.0.0/7 ge 12 le 20
ip prefix-list FILTER permit 210.0.0.0/7 ge 12 le 20
!! ARIN
ip prefix-list FILTER permit 63.0.0.0/8 le 20
ip prefix-list FILTER permit 64.0.0.0/8 le 20
ip prefix-list FILTER permit 199.0.0.0/8 le 20
ip prefix-list FILTER permit 200.0.0.0/8 le 20
ip prefix-list FILTER permit 204.0.0.0/6 le 20
ip prefix-list FILTER permit 208.0.0.0/7 le 20
ip prefix-list FILTER permit 216.0.0.0/8 le 20
```

“Net Police” prefix list issues

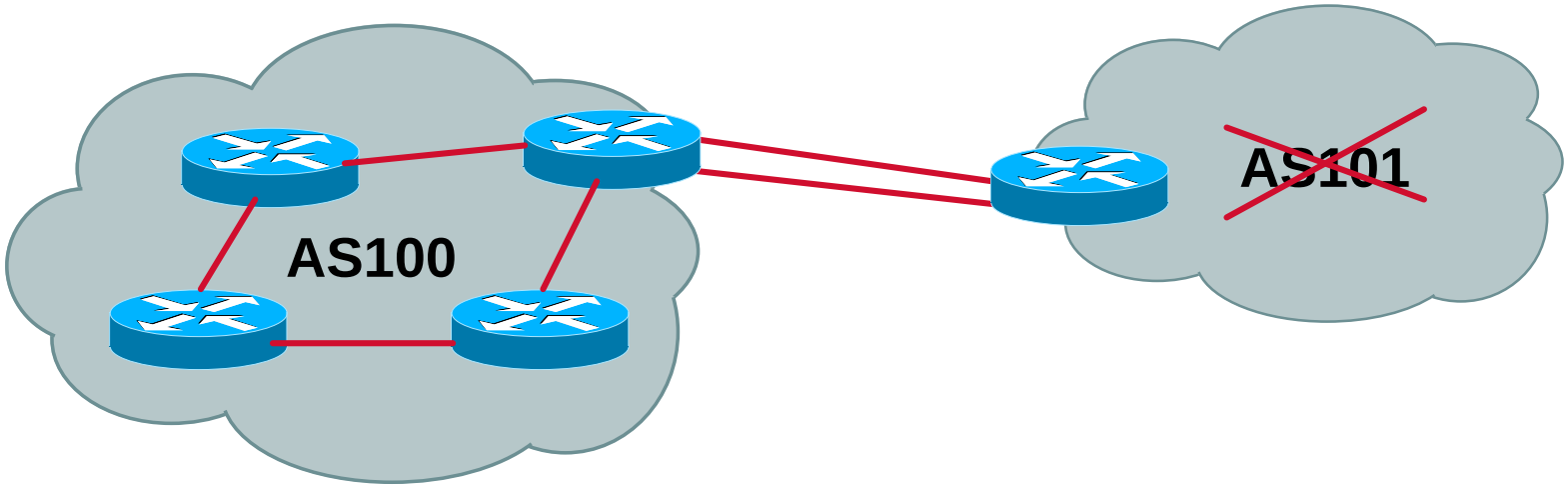
- meant to “punish” ISPs who won’t and don’t aggregate
- impacts legitimate multihoming
- impacts regions where domestic backbone is unavailable or costs $$$ compared with international bandwidth
- hard to maintain - requires updating when RIRs start allocating from new address blocks
- Do NOT do it unless consequences understood

Multihoming Options

Multihoming Scenarios

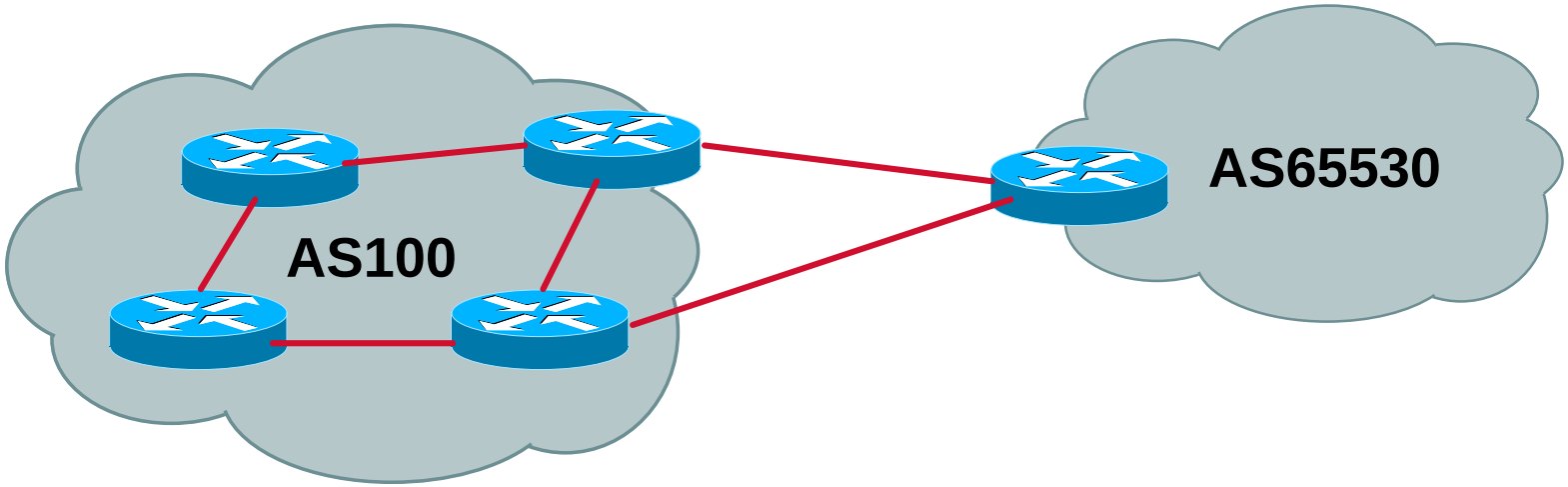
- Stub network
- Multi-homed stub network
- Multi-homed network
- Configuration Options

Stub Network



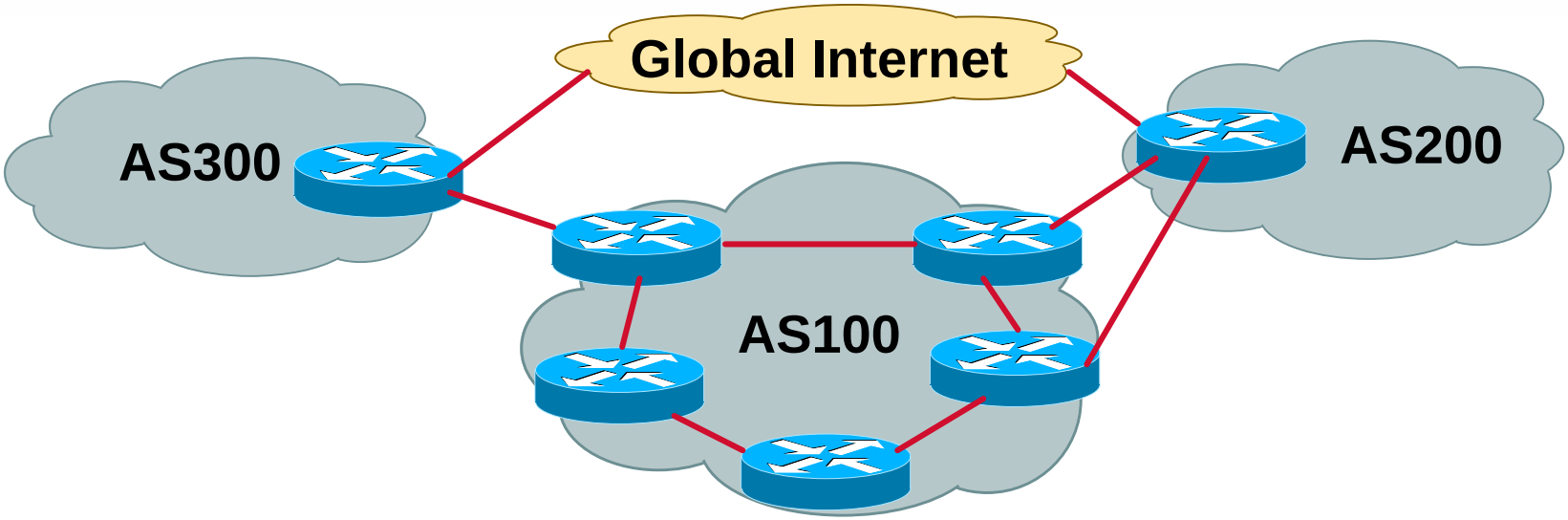
- No need for BGP
- Point static default to upstream ISP
- Upstream ISP advertises stub network
- Policy confined within upstream ISP's policy

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

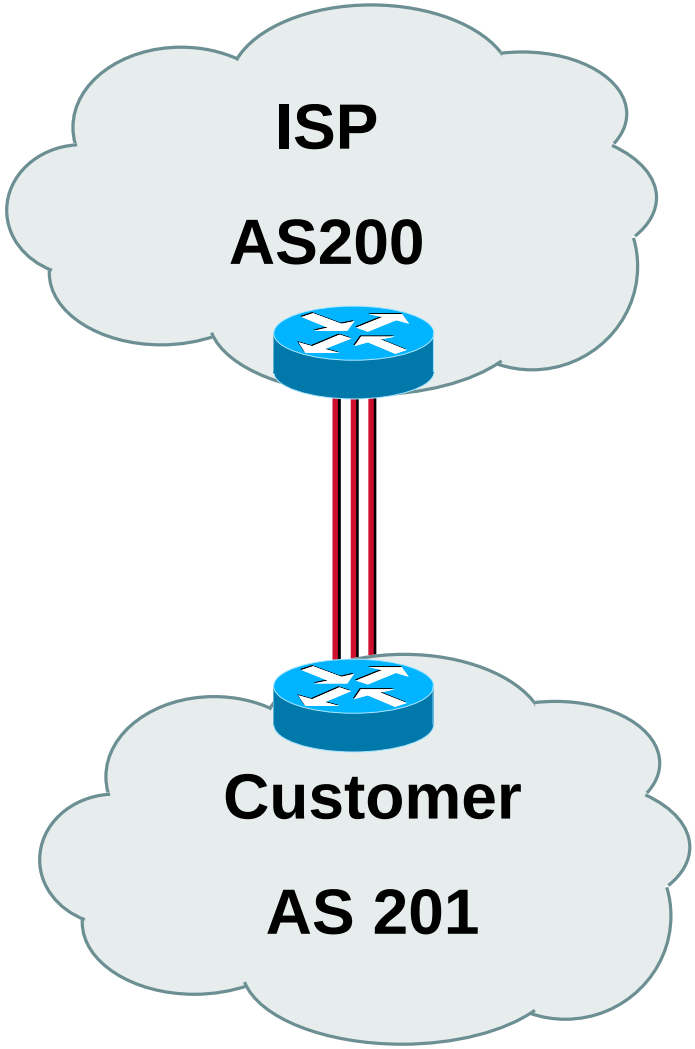
Multi-Homed Network



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

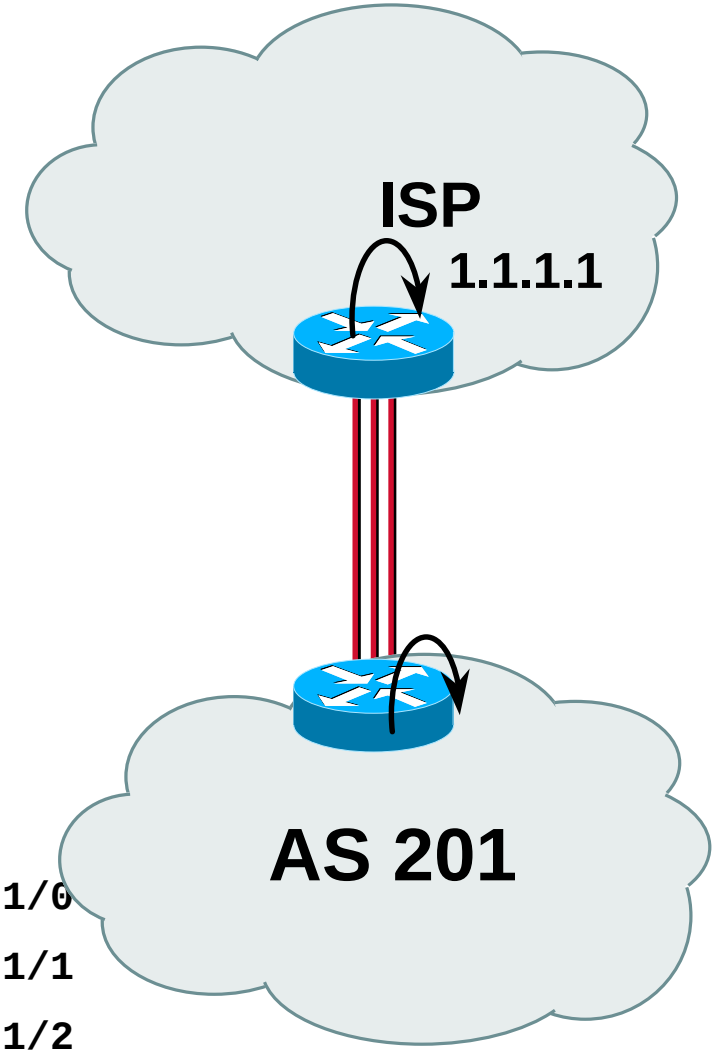
Multiple Sessions to an ISP

- **Several IOS options**
  - **ebgp multihop**
  - **bgp multipath**
  - **cef loadsharing**
  - **bgp attribute manipulation**



Multiple Sessions to an ISP ebgp multihop

- eBGP to loopback addresses
- eBGP prefixes learned with loopback address as next hop

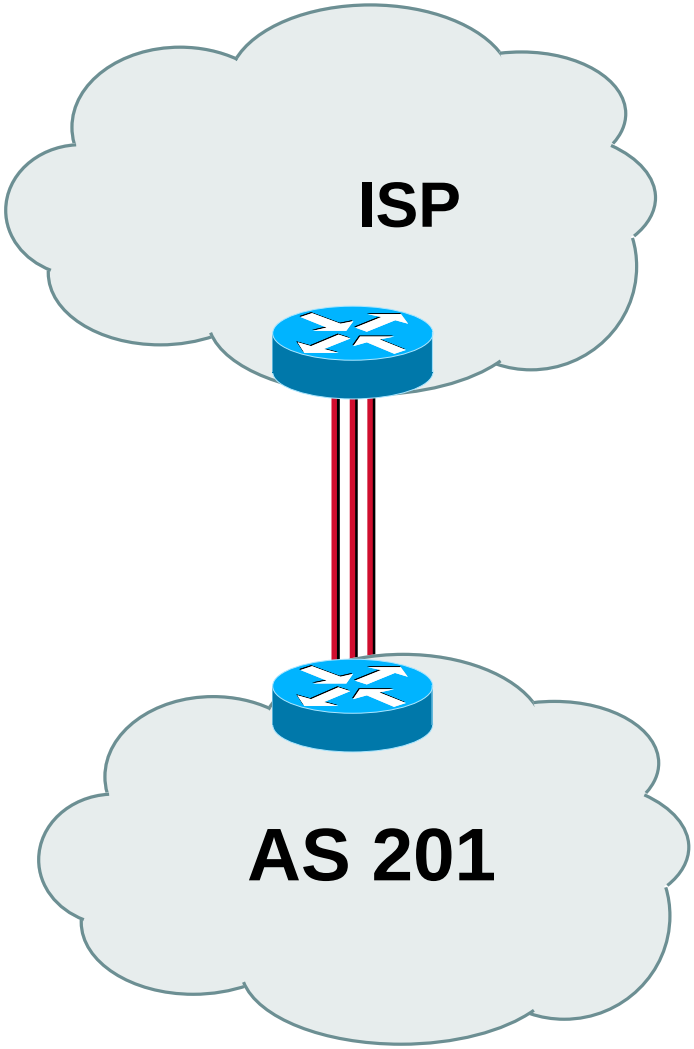



```
router bgp 201
 neighbor 1.1.1.1 remote-as 200
 neighbor 1.1.1.1 ebgp-multihop 5
ip route 1.1.1.1 255.255.255.255 serial 1/0
ip route 1.1.1.1 255.255.255.255 serial 1/1
ip route 1.1.1.1 255.255.255.255 serial 1/2
```

Multiple Sessions to an ISP 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
```



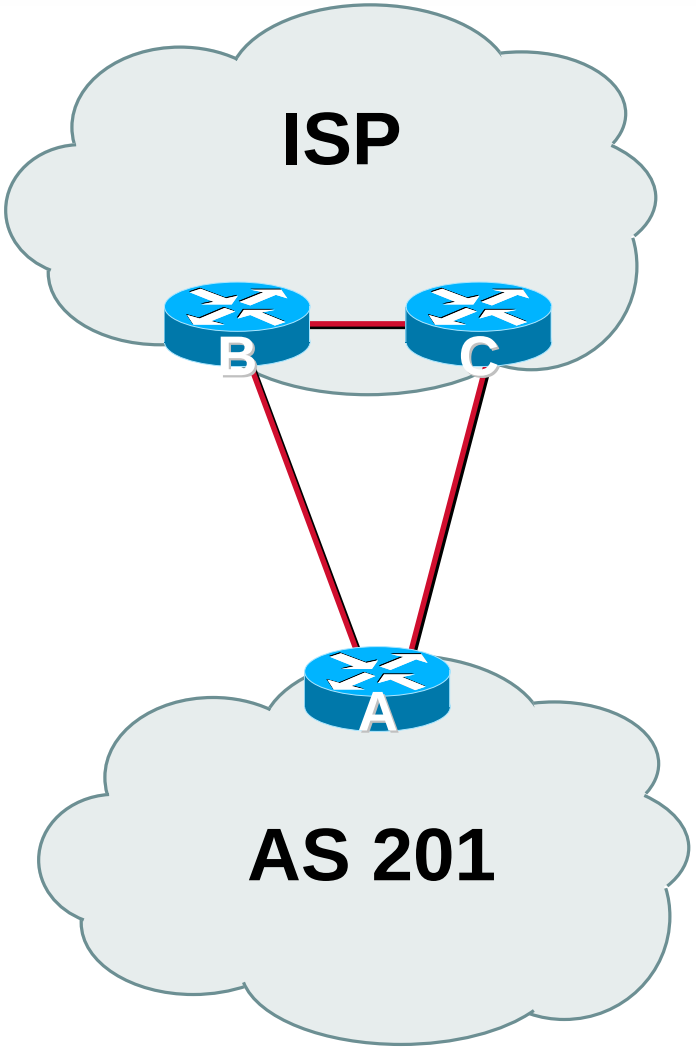
Multiple Sessions to an ISP

- **Use eBGP multi-path to install multiple paths in IP table**

```
router bgp 201
 maximum-path <1-6>
```

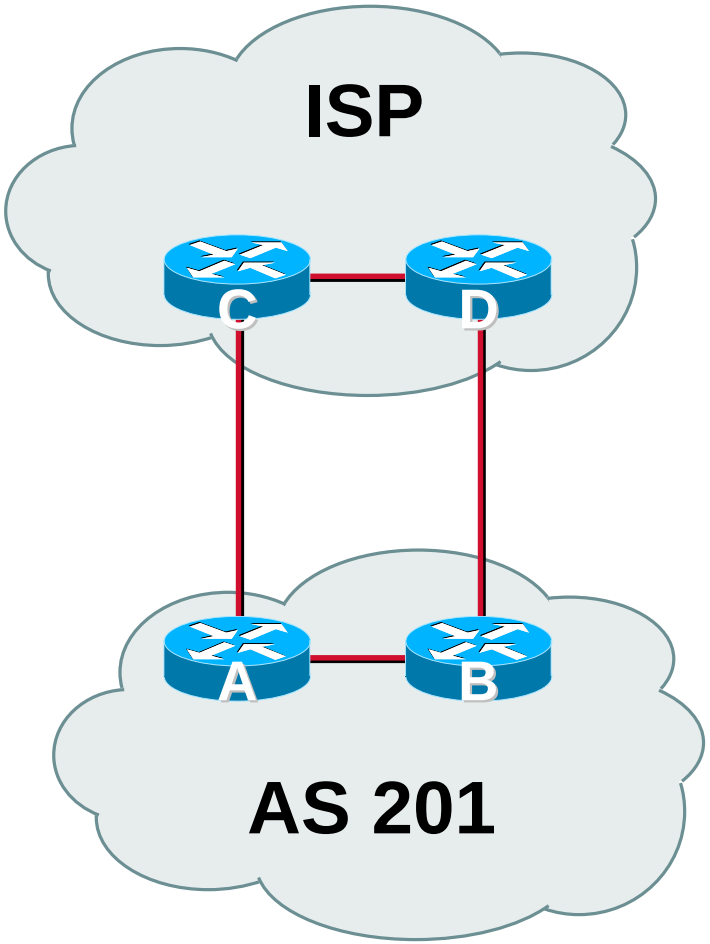
- **Load share over the alternate paths**

  per destination loadsharing



Multiple Sessions to an ISP

- **Simplest scheme is to use defaults**
- **Learn/advertise prefixes for better control**

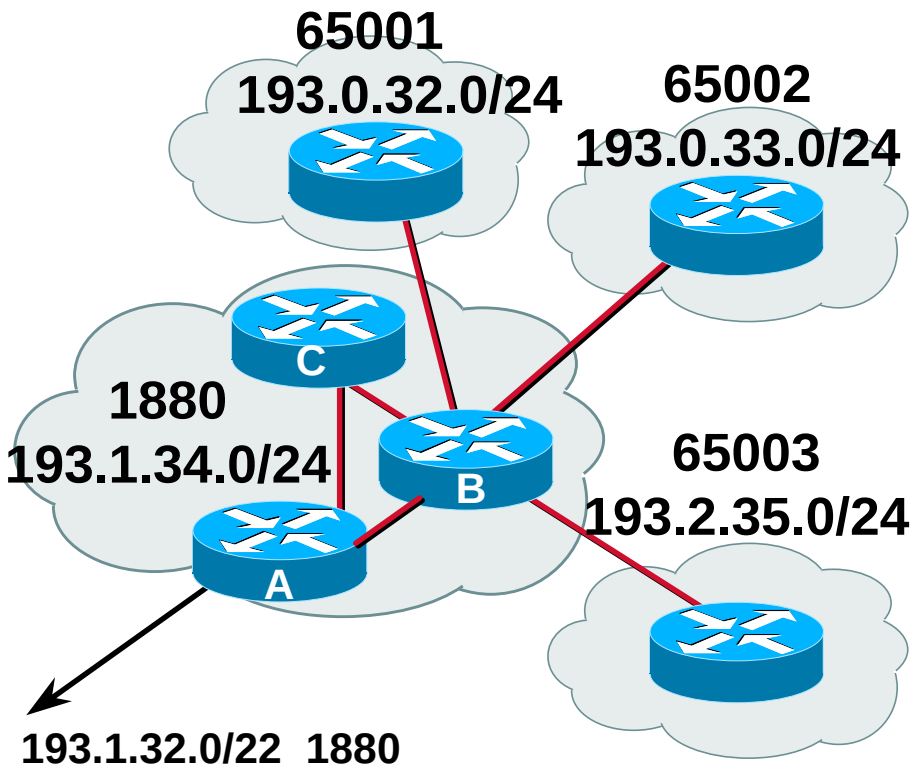


Multiple Sessions to ISPs

- Planning and some work required to achieve load sharing
  - 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

Private-AS

- **Applications**
  - **ISP with single-homed customers**
  - **Enterprise network with several regions and connections to the Internet only in the core**
  - **Within a BGP Confederation**



Private-AS Removal

- **neighbor x.x.x.x remove-private-AS**
- **Rules:**

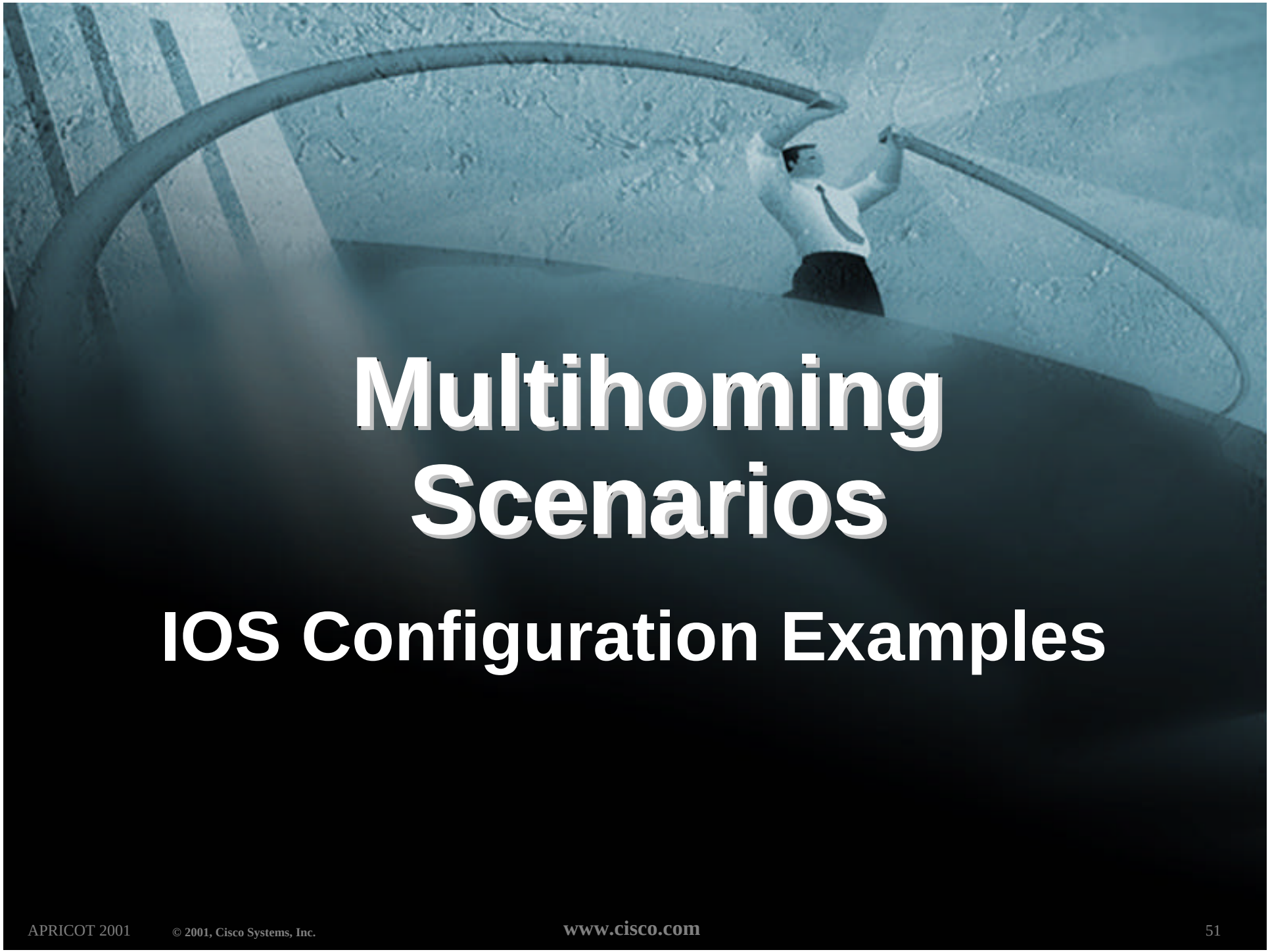
  available for eBGP neighbors only

  if the update has AS_PATH made up of private-AS numbers, the private-AS will be dropped

  if the AS_PATH includes private and public AS numbers, private AS number will not be removed…it is a configuration error!

  if AS_PATH contains the AS number of the eBGP neighbor, the private-AS numbers will not be removed

  if used with confederations, it will work as long as the private AS numbers are after the confederation portion of the AS_PATH

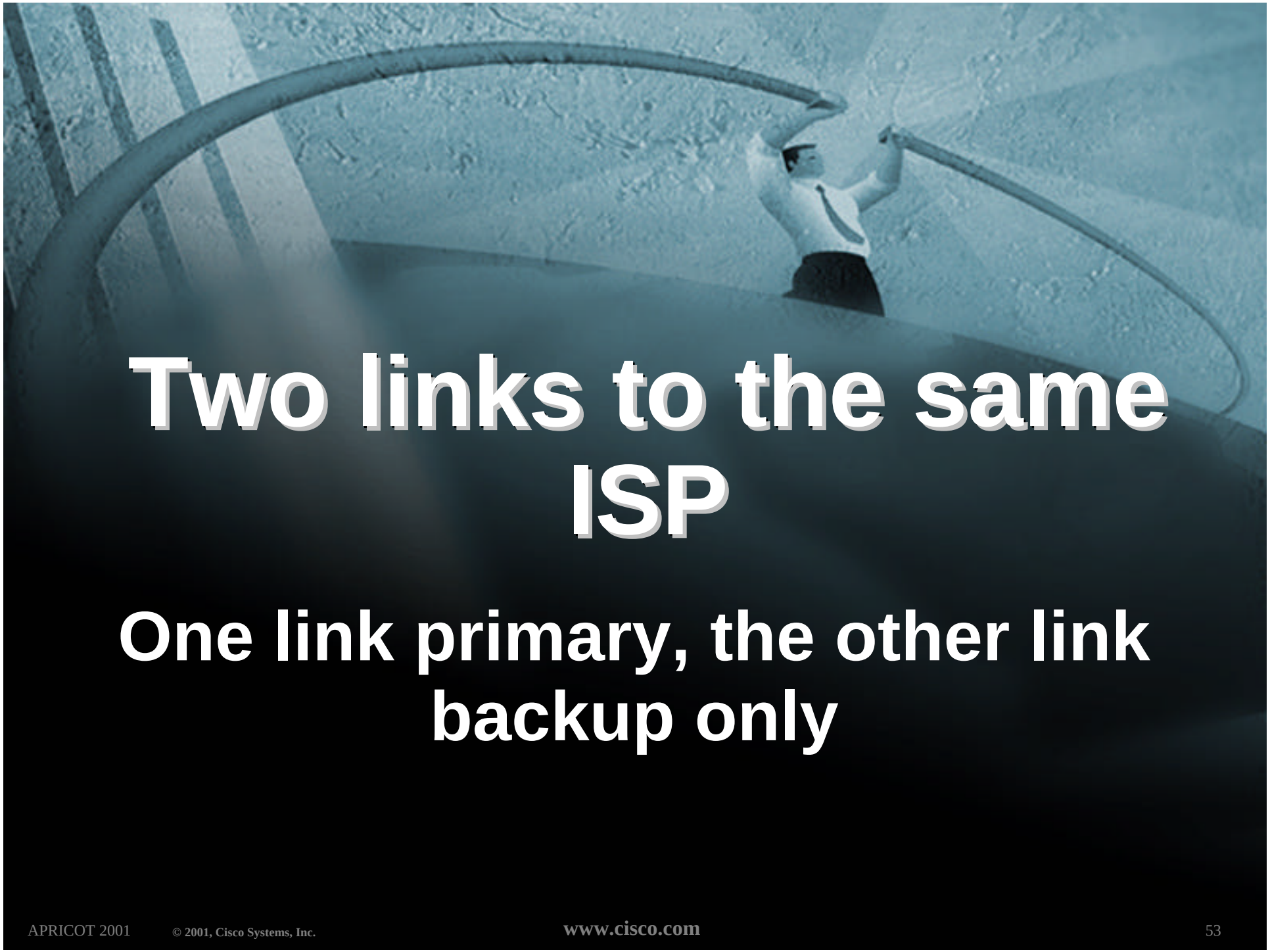


Multihoming Scenarios

IOS Configuration Examples

Multihoming Scenarios

- To the Same ISP
  - One link primary, other as backup
  - Loadsharing
  - RFC2270 multihoming
- To different ISPs
  - One link primary, other as backup
  - Loadsharing
- Communities for multihoming



Two links to the same ISP

One link primary, the other link backup only

Two links to the same ISP

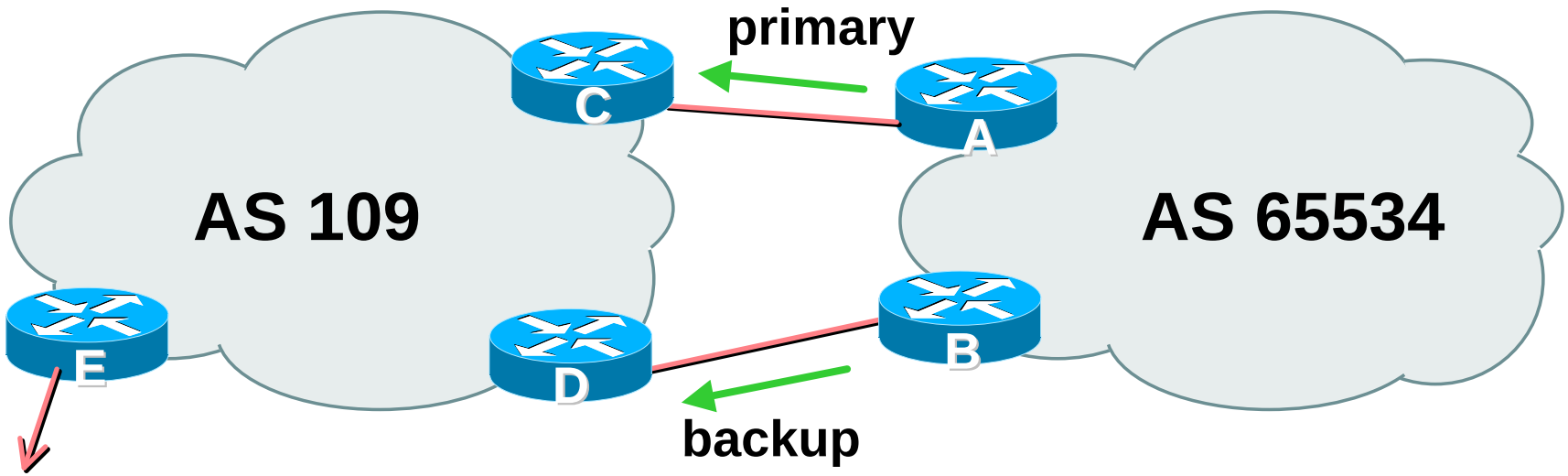
- Can use BGP for this to aid loadsharing

  use a private AS (ASN > 64511)

- upstream ISP proxy aggregates

  in other words, announces only your address block to the Internet (as would be done if you had one statically routed connection)

Two links to the same ISP



- AS109 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 makes standard announcement

  backup link increases metric on outbound, and reduces local-pref on inbound

- **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 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 109
 neighbor 222.222.10.2 description RouterC
 neighbor 222.222.10.2 prefix-list aggregate out
 neighbor 222.222.10.2 prefix-list default in
!
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
```

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

- **Router B Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.6 remote-as 109
 neighbor 222.222.10.6 description RouterD
 neighbor 222.222.10.6 prefix-list aggregate out
 neighbor 222.222.10.6 route-map routerD-out out
 neighbor 222.222.10.6 prefix-list default in
 neighbor 222.222.10.6 route-map routerD-in in
!
..next slide
```

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

```
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
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 109
 neighbor 222.222.10.1 remote-as 65534
 neighbor 222.222.10.1 default-originate
 neighbor 222.222.10.1 prefix-list Customer in
 neighbor 222.222.10.1 prefix-list default out
!
ip prefix-list Customer permit 221.10.0.0/19
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 109
 neighbor 222.222.10.5 remote-as 65534
 neighbor 222.222.10.5 default-originate
 neighbor 222.222.10.5 prefix-list Customer in
 neighbor 222.222.10.5 prefix-list default out
!
ip prefix-list Customer permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
```

Two links to the same ISP

- **Router E is AS109 border router**
  - **removes prefixes in the private AS from external announcements**
  - **implements the proxy aggregation for the customer prefixes**

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

- **Router E Configuration**

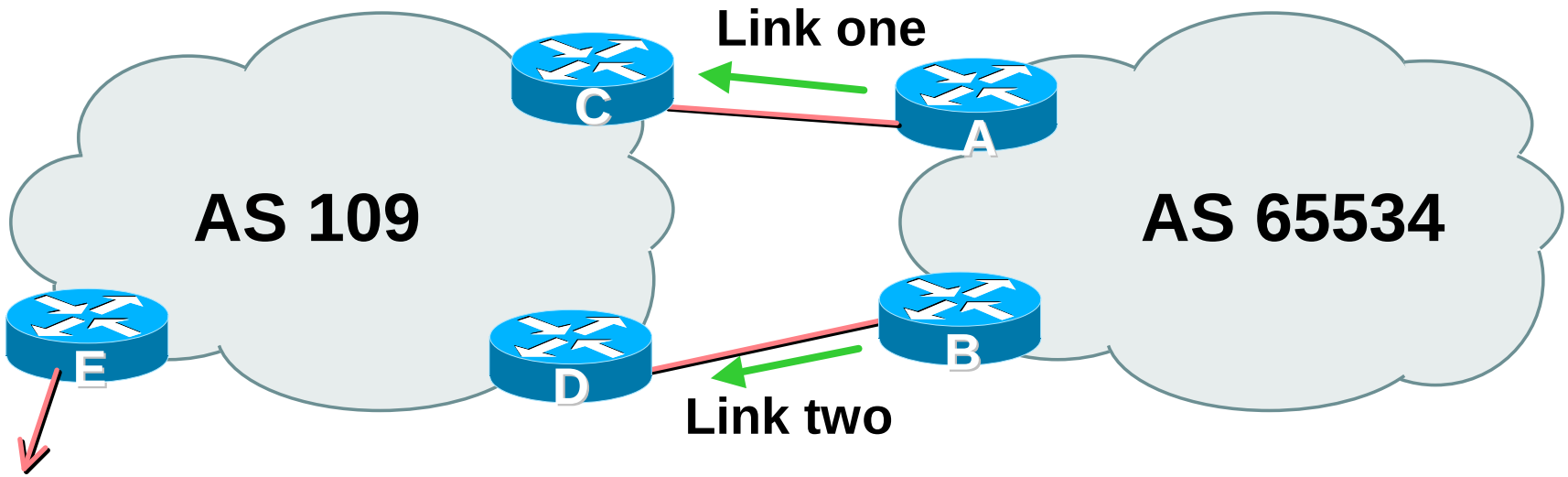
```
router bgp 109
 neighbor 222.222.10.17 remote-as 110
 neighbor 222.222.10.17 remove-private-AS
 neighbor 222.222.10.17 prefix-list Customer out
!
ip prefix-list Customer permit 221.10.0.0/19
```

- **Private AS still visible inside AS109**

Two links to the same ISP

With Redundancy and Loadsharing

Two links to the same ISP (with redundancy)



- AS109 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

Two links to the same ISP

- **Router A Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.0.0 mask 255.255.240.0
 neighbor 222.222.10.2 remote-as 109
 neighbor 222.222.10.2 prefix-list routerC out
 neighbor 222.222.10.2 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerC permit 221.10.0.0/20
ip prefix-list routerC permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.240.0 null0
ip route 221.10.0.0 255.255.224.0 null0
```

Two links to the same ISP

- **Router B Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.16.0 mask 255.255.240.0
 neighbor 222.222.10.6 remote-as 109
 neighbor 222.222.10.6 prefix-list routerD out
 neighbor 222.222.10.6 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerD permit 221.10.16.0/20
ip prefix-list routerD permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.224.0 null0
ip route 221.10.16.0 255.255.240.0 null0
```

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
```

Two links to the same ISP

- **Router C Configuration**

```
router bgp 109
 neighbor 222.222.10.1 remote-as 65534
 neighbor 222.222.10.1 default-originate
 neighbor 222.222.10.1 prefix-list Customer in
 neighbor 222.222.10.1 prefix-list default out
!
ip prefix-list Customer permit 221.10.0.0/19 le 20
ip prefix-list default permit 0.0.0.0/0
```

- **Router C only allows in /19 and /20 prefixes from customer block**

Two links to the same ISP

- **Router D Configuration**

```
router bgp 109
 neighbor 222.222.10.5 remote-as 65534
 neighbor 222.222.10.5 default-originate
 neighbor 222.222.10.5 prefix-list Customer in
 neighbor 222.222.10.5 prefix-list default out
!
ip prefix-list Customer permit 221.10.0.0/19 le 20
ip prefix-list default permit 0.0.0.0/0
```

- **Router D only allows in /19 and /20 prefixes from customer block**

Two links to the same ISP

- **Router E is AS109 border router**

  removes subprefixes in the private AS from external announcements

  removes the private AS from external announcement of the customer /19

Two links to the same ISP (with redundancy)

- **Router E Configuration**

```
router bgp 109
 neighbor 222.222.10.17 remote-as 110
 neighbor 222.222.10.17 remove-private-AS
 neighbor 222.222.10.17 prefix-list Customer out
!
ip prefix-list Customer permit 221.10.0.0/19
```

- **Private AS still visible inside AS109**

Loadsharing to the same ISP

- Loadsharing configuration is only on customer router
- Upstream ISP has to

  remove customer subprefixes from external announcements

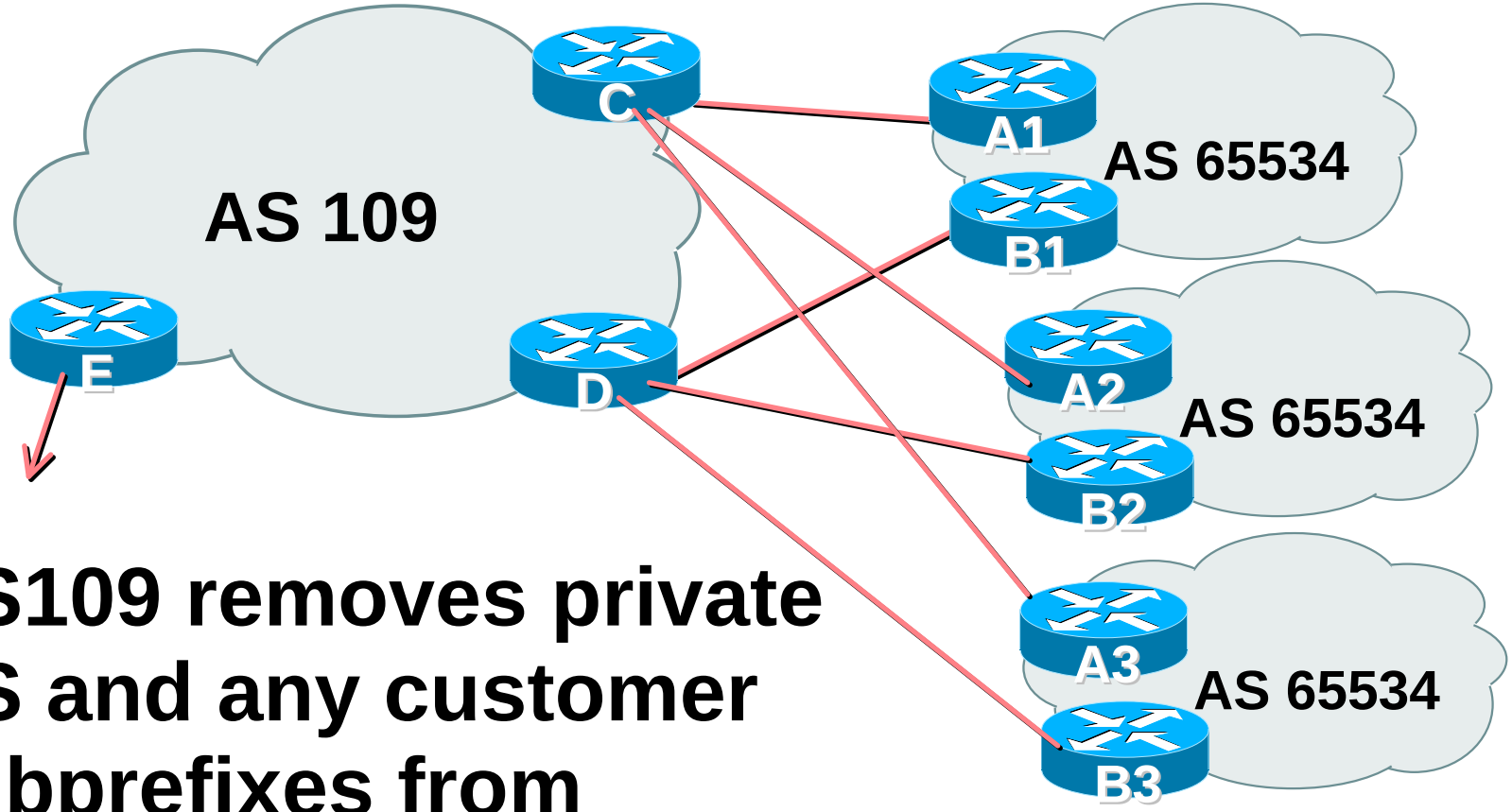
  remove private AS from external announcements
- Could also use BGP communities

Two links to the same ISP

Multiple Dualhomed Customers

(RFC2270)

Multiple Dualhomed Customers (RFC2270)



- AS109 removes private AS and any customer subprefixes from Internet announcement

Multiple Dualhomed Customers

- 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 A*n* and B*n* configuration same as Router A and B previously

Two links to the same ISP

- **Router A1 Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.0.0 mask 255.255.240.0
 neighbor 222.222.10.2 remote-as 109
 neighbor 222.222.10.2 prefix-list routerC out
 neighbor 222.222.10.2 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerC permit 221.10.0.0/20
ip prefix-list routerC permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.240.0 null0
ip route 221.10.0.0 255.255.224.0 null0
```

Two links to the same ISP

- **Router B1 Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.16.0 mask 255.255.240.0
 neighbor 222.222.10.6 remote-as 109
 neighbor 222.222.10.6 prefix-list routerD out
 neighbor 222.222.10.6 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerD permit 221.10.16.0/20
ip prefix-list routerD permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.224.0 null0
ip route 221.10.16.0 255.255.240.0 null0
```

Multiple Dualhomed Customers

- **Router C Configuration**

```
router bgp 109
 neighbor bgp-customers peer-group
 neighbor bgp-customers remote-as 65534
 neighbor bgp-customers default-originate
 neighbor bgp-customers prefix-list default out
 neighbor 222.222.10.1 peer-group bgp-customers
 neighbor 222.222.10.1 description Customer One
 neighbor 222.222.10.1 prefix-list Customer1 in
 neighbor 222.222.10.9 peer-group bgp-customers
 neighbor 222.222.10.9 description Customer Two
 neighbor 222.222.10.9 prefix-list Customer2 in
```

Multiple Dualhomed Customers

```
 neighbor 222.222.10.17 peer-group bgp-customers
 neighbor 222.222.10.17 description Customer Three
 neighbor 222.222.10.17 prefix-list Customer3 in
!
ip prefix-list Customer1 permit 221.10.0.0/19 le 20
ip prefix-list Customer2 permit 221.16.64.0/19 le 20
ip prefix-list Customer3 permit 221.14.192.0/19 le 20
ip prefix-list default permit 0.0.0.0/0
```

- Router C only allows in /19 and /20 prefixes from customer block

Multiple Dualhomed Customers

- **Router D Configuration**

```
router bgp 109
 neighbor bgp-customers peer-group
 neighbor bgp-customers remote-as 65534
 neighbor bgp-customers default-originate
 neighbor bgp-customers prefix-list default out
 neighbor 222.222.10.5 peer-group bgp-customers
 neighbor 222.222.10.5 description Customer One
 neighbor 222.222.10.5 prefix-list Customer1 in
 neighbor 222.222.10.13 peer-group bgp-customers
 neighbor 222.222.10.13 description Customer Two
 neighbor 222.222.10.13 prefix-list Customer2 in
```

Multiple Dualhomed Customers

```
 neighbor 222.222.10.21 peer-group bgp-customers
 neighbor 222.222.10.21 description Customer Three
 neighbor 222.222.10.21 prefix-list Customer3 in
!
ip prefix-list Customer1 permit 221.10.0.0/19 le 20
ip prefix-list Customer2 permit 221.16.64.0/19 le 20
ip prefix-list Customer3 permit 221.14.192.0/19 le 20
ip prefix-list default permit 0.0.0.0/0
```

- **Router D only allows in /19 and /20 prefixes from customer block**

Multiple Dualhomed Customers

- **Router E Configuration is as previously**

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

```
router bgp 109
 neighbor 222.222.10.17 remote-as 110
 neighbor 222.222.10.17 remove-private-AS
 neighbor 222.222.10.17 prefix-list Customers out
!
ip prefix-list Customers permit 221.10.0.0/19
ip prefix-list Customers permit 221.16.64.0/19
ip prefix-list Customers permit 221.14.192.0/19
```

- **Private AS still visible inside AS109**

Multiple Dualhomed Customers

- 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 109
 neighbor 222.222.10.17 remote-as 110
 neighbor 222.222.10.17 prefix-list my-aggregate out
!
ip prefix-list my-aggregate permit 221.8.0.0/13
```

Multihoming Summary

- Use private AS for multihoming to upstream
- Leak subprefixes to upstream only to aid loadsharing
- Upstream router E configuration is uniform across all scenarios

Two links to different ISPs

Two links to different ISPs

- **Use Public ASes**

  or use private AS if agreed with the other ISP

- **Address space comes from**

  both upstreams or

  Regional Internet Registry

- **Configuration concepts very similar**

Two links to different ISPs

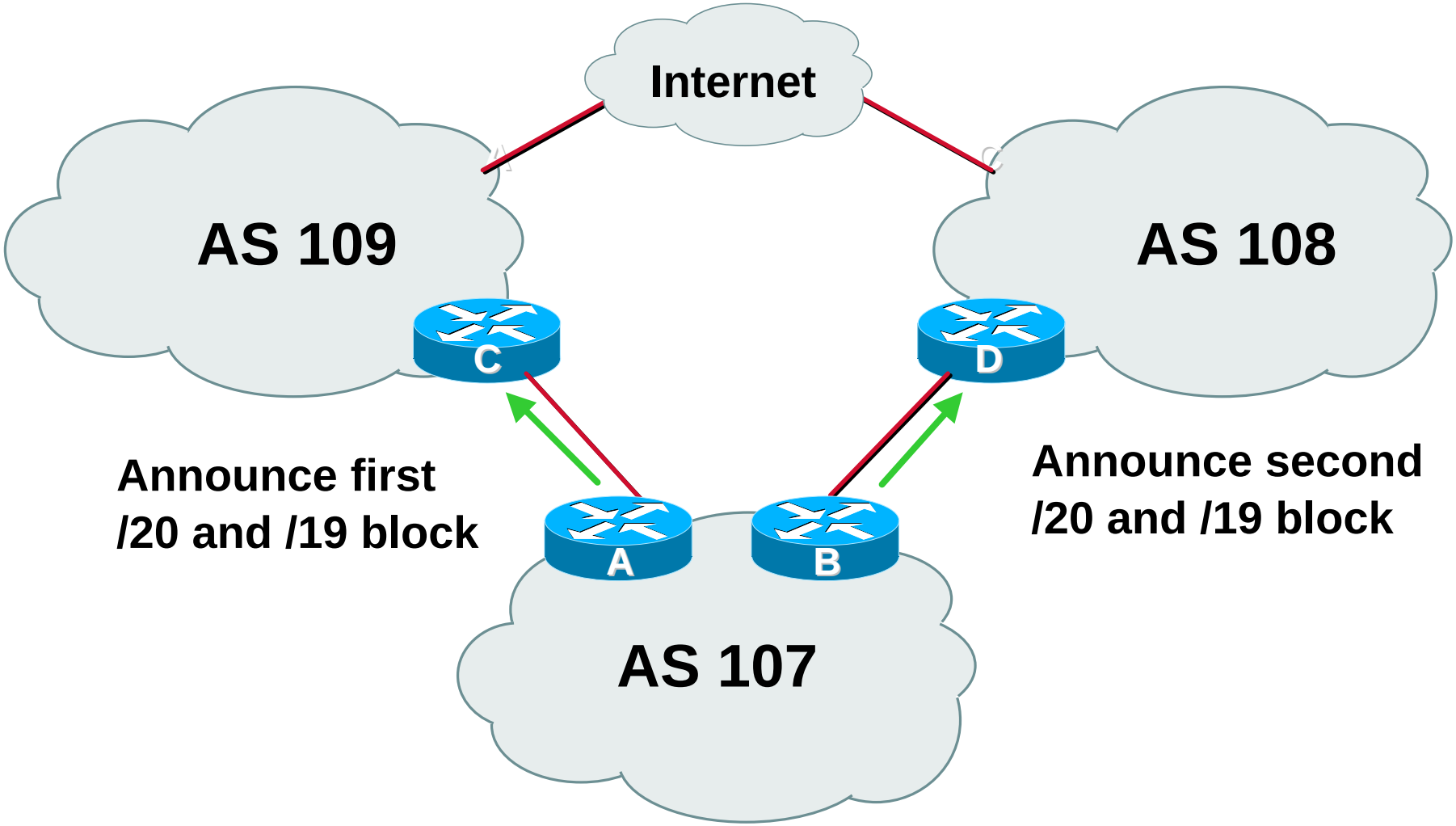
With Redundancy

Two links to different ISPs (with redundancy)

- 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 redundancy)



Two links to different ISPs (with redundancy)

- **Router A Configuration**

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.0.0 mask 255.255.240.0
 neighbor 222.222.10.1 remote-as 109
 neighbor 222.222.10.1 prefix-list firstblock out
 neighbor 222.222.10.1 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
!
ip prefix-list firstblock permit 221.10.0.0/20
ip prefix-list firstblock permit 221.10.0.0/19
```

Two links to different ISPs (with redundancy)

- **Router B Configuration**

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.16.0 mask 255.255.240.0
 neighbor 220.1.5.1 remote-as 108
 neighbor 220.1.5.1 prefix-list secondblock out
 neighbor 220.1.5.1 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
!
ip prefix-list secondblock permit 221.10.16.0/20
ip prefix-list secondblock permit 221.10.0.0/19
```

Two links to different ISPs (with redundancy)

- **Router C Configuration**

```
router bgp 109
 neighbor 221.10.1.1 remote-as 107
 neighbor 221.10.1.1 default-originate
 neighbor 221.10.1.1 prefix-list AS107cust in
 neighbor 221.10.1.1 prefix-list default-out out
!
```

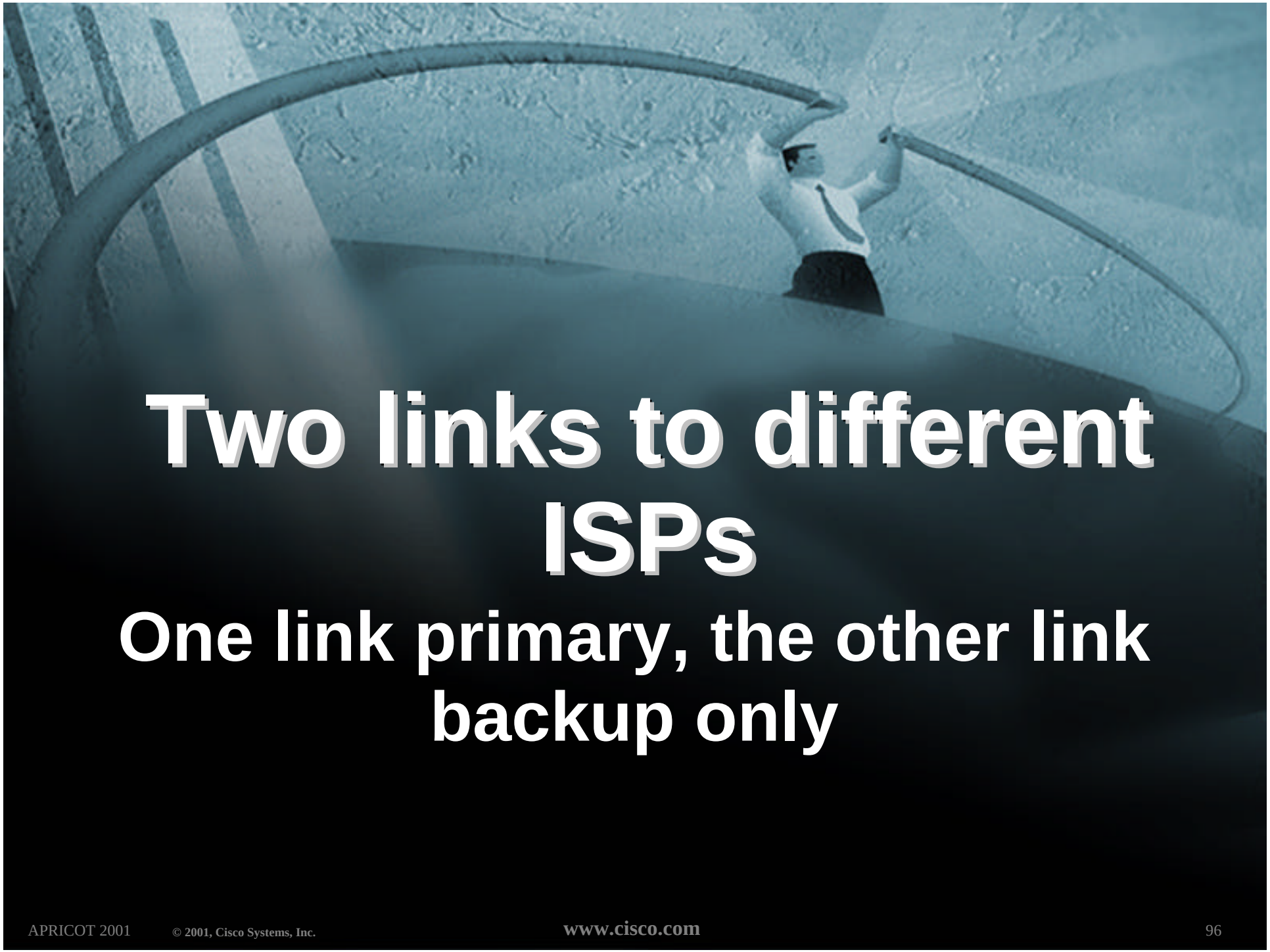
- **Router C only announces default to AS 107**
- **Router C only accepts AS107's prefix block**

Two links to different ISPs (with redundancy)

- **Router D Configuration**

```
router bgp 108
 neighbor 220.1.5.1 remote-as 107
 neighbor 220.1.5.1 default-originate
 neighbor 220.1.5.1 prefix-list AS107cust in
 neighbor 220.1.5.1 prefix-list default-out out
!
```

- **Router D only announces default to AS 107**
- **Router D only accepts AS107's prefix block**



Two links to different ISPs

One link primary, the other link backup only

Two links to different ISPs (one as backup only)

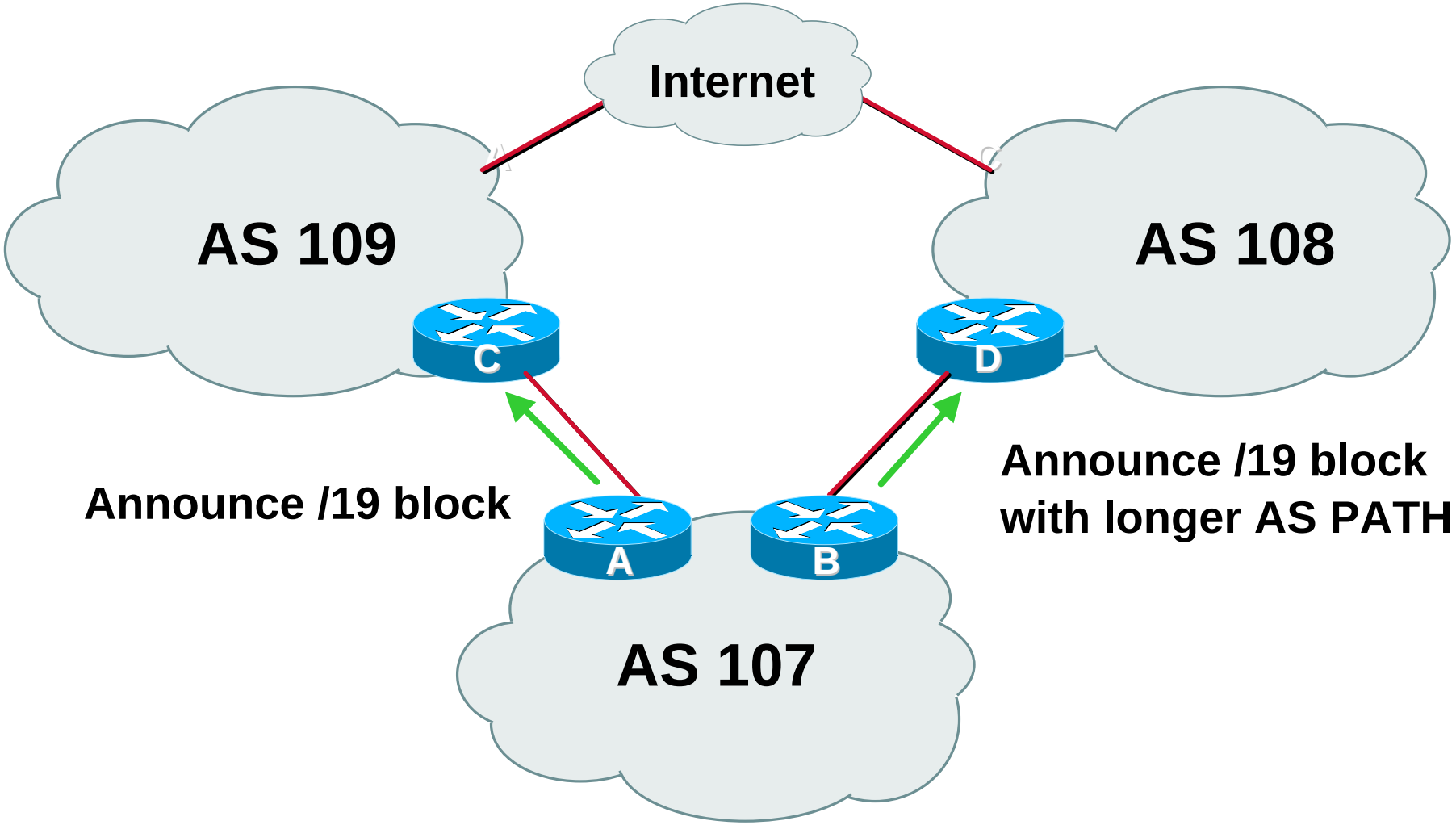
- Announce /19 aggregate on each link

  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)



Two links to different ISPs (one as backup only)

- **Router A Configuration**

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 109
 neighbor 222.222.10.1 prefix-list aggregate out
 neighbor 222.222.10.1 prefix-list default in
!
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
```

Two links to different ISPs (one as backup only)

- **Router B Configuration**

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 neighbor 220.1.5.1 remote-as 108
 neighbor 220.1.5.1 prefix-list aggregate out
 neighbor 220.1.5.1 route-map routerD-out out
 neighbor 220.1.5.1 prefix-list default in
 neighbor 220.1.5.1 route-map routerD-in in
!
..next slide
```

Two links to different ISPs (one as backup only)

- **Router B Configuration**

```
!
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
route-map routerD-out permit 10
 set as-path prepend 107 107 107
!
route-map routerD-in permit 10
 set local-preference 80
```

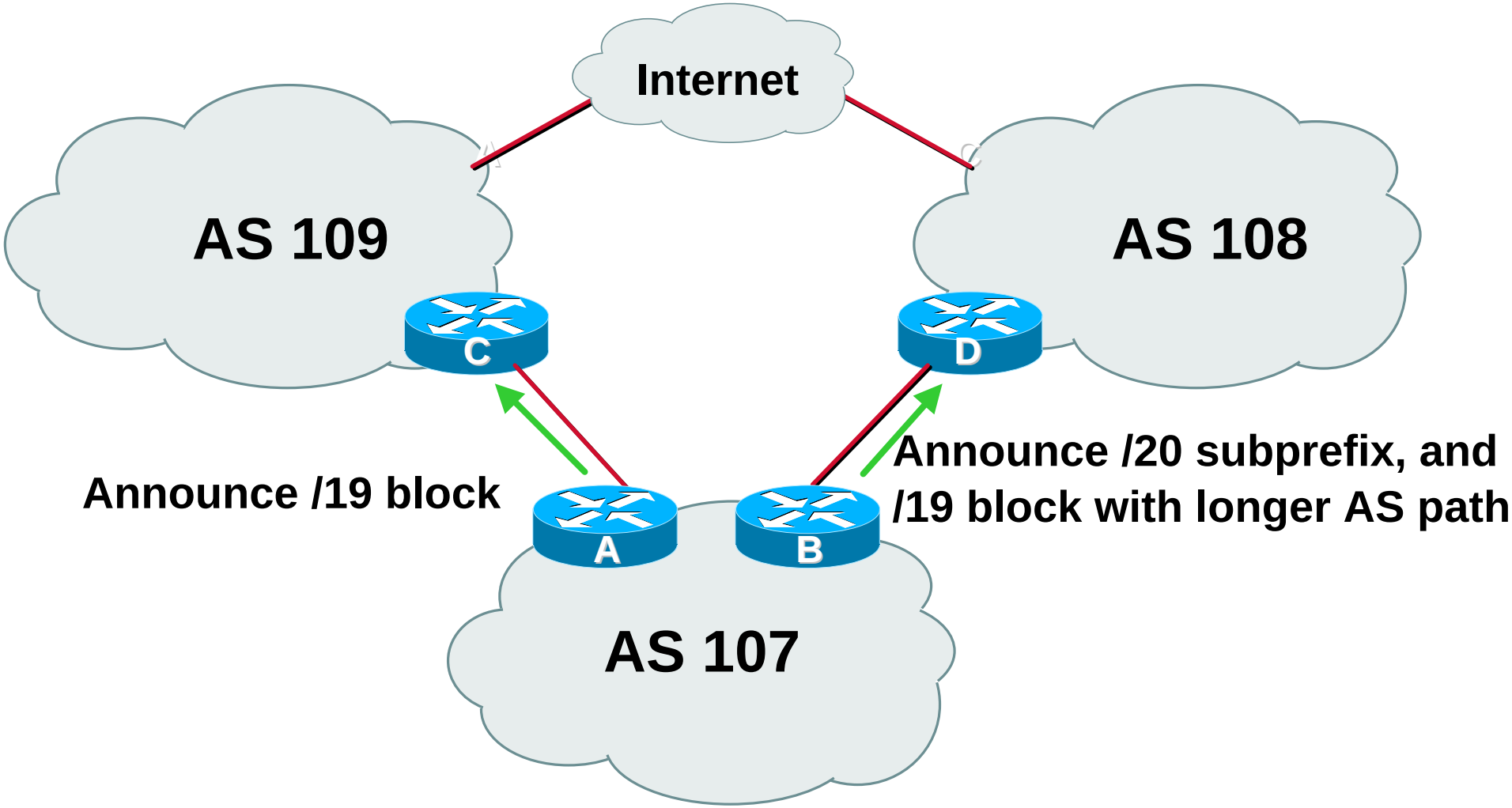
Two links to different ISPs

More Controlled Loadsharing

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



Loadsharing with different ISPs

- **Router A Configuration**

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 109
 neighbor 222.222.10.1 prefix-list default in
 neighbor 222.222.10.1 prefix-list aggregate out
!
ip prefix-list aggregate permit 221.10.0.0/19
```

Loadsharing with different ISPs

- **Router B Configuration**

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.16.0 mask 255.255.240.0
 neighbor 220.1.5.1 remote-as 108
 neighbor 220.1.5.1 prefix-list default in
 neighbor 220.1.5.1 prefix-list subblocks out
 neighbor 220.1.5.1 route-map routerD out
!
..next slide..
```

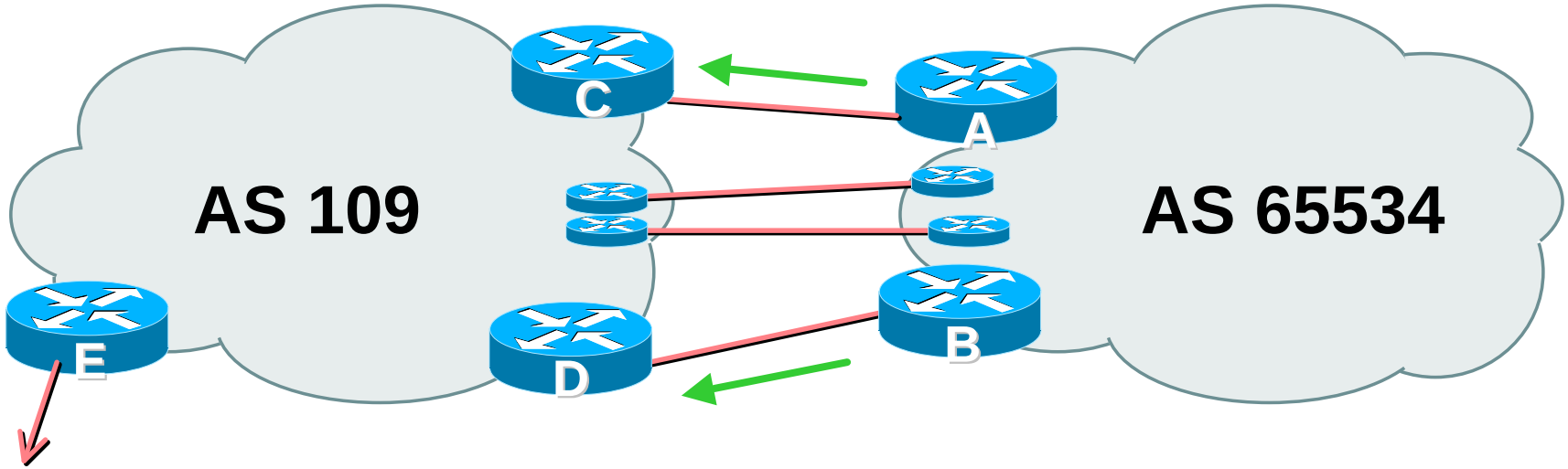
Loadsharing with different ISPs

```
route-map routerD permit 10
 match ip address prefix-list aggregate
 set as-path prepend 107 107
route-map routerD permit 20
!
ip prefix-list subblocks permit 221.10.0.0/19 le 20
ip prefix-list aggregate permit 221.10.0.0/19
```

Loadsharing Using Communities

4 links - Private AS

Private AS



- **AS109 removes private AS and any customer subprefixes from Internet announcement**

Private AS

- Announce /19 aggregate on each link
- Split /19 and announce as four /21s, 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
  - use the no-export community for subprefixes

Private AS

- **Router A Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.0.0 mask 255.255.248.0
 neighbor 222.222.10.2 remote-as 109
 neighbor 222.222.10.2 send-community
 neighbor 222.222.10.2 prefix-list subblocks1 out
 neighbor 222.222.10.2 route-map routerC-out out
 neighbor 222.222.10.2 prefix-list default in
!
..next slide
```

Private AS

```
ip prefix-list subblocks1 permit 221.10.0.0/19
ip prefix-list subblocks1 permit 221.10.0.0/21
!
ip prefix-list firstblock permit 221.10.0.0/21
ip prefix-list default permit 0.0.0.0/0
!
route-map routerC-out permit 10
 match ip address prefix-list firstblock
 set community no-export
route-map routerC-out permit 20
```

Private AS

- **Router B Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 network 221.10.24.0 mask 255.255.248.0
 neighbor 222.222.20.2 remote-as 109
 neighbor 222.222.20.2 send-community
 neighbor 222.222.20.2 prefix-list subblocks2 out
 neighbor 222.222.20.2 route-map routerD-out out
 neighbor 222.222.20.2 prefix-list default in
!
..next slide
```

Private AS

```
ip prefix-list subblocks2 permit 221.10.0.0/19
ip prefix-list subblocks2 permit 221.10.24.0/21
!
ip prefix-list secondblock permit 221.10.24.0/21
ip prefix-list default permit 0.0.0.0/0
!
route-map routerD-out permit 10
 match ip address prefix-list secondblock
 set community no-export
route-map routerD-out permit 20
```

Private AS

- **Router E Configuration**

```
router bgp 109
 neighbor 222.222.10.17 remote-as 110
 neigbhor 222.222.10.17 remove-private-AS
!
```

- **Router E removes the private AS from external announcements**
- **Router E automatically removes subprefixes with no-export community set**
- **Private AS still visible inside AS109**

Private AS

- Router C and D configuration is as previously
- AS109 routers will not advertise prefixes marked with community no-export to other ASes
- AS109 routers still need to filter the private AS
- Only a single /19 prefix is announced to the Internet - no routing table bloat! :-)

Public AS

- **If customer has public AS, configuration is the same**

  Don’t need remove-private-AS configuration command

Enterprise Multihoming

Enterprise Multihoming

- Common scenario in Internet today
- More and more non-SPs multihoming for:
  - service provider redundancy
  - link redundancy
- Issues on Internet today:
  - Routing Table size accelerating
  - more and more /24 prefixes appearing in Internet Routing Table
  - ASN consumption accelerating

Enterprise Multihoming

- **The following example**

  could apply to smaller ISPs who don't yet have their own address block

  requires BGP but a private AS (ASN >64511) can and should be used

  is good for the health of the Internet

Medium/Large ISP Multihoming

- **ISPs should obtain their own address block and ASN**
  - **Get it from RIR**
  - **Makes multihoming easier**
  - **Makes changing upstreams easier**
  - **Does not cause so much fragmentation in Internet Routing Table**

Enterprise Multihoming

Example

Enterprise Multihoming

- **Common situation is enterprise multihoming**
  - **address space used by enterprise comes from both upstream ISPs**
  - **multihoming and loadsharing more difficult**
  - **want to avoid leaking subprefixes of upstream provider address space when possible**
  - **require provider redundancy (not just link redundancy)**

Enterprise Multihoming

- **Address space from upstream should match link bandwidth to upstream, e.g.**

  ISP1 ® Enterprise = 256kbps ® /22

  ISP2 ® Enterprise = 128kbps ® /23

  assumes address space is uniformly distributed across network

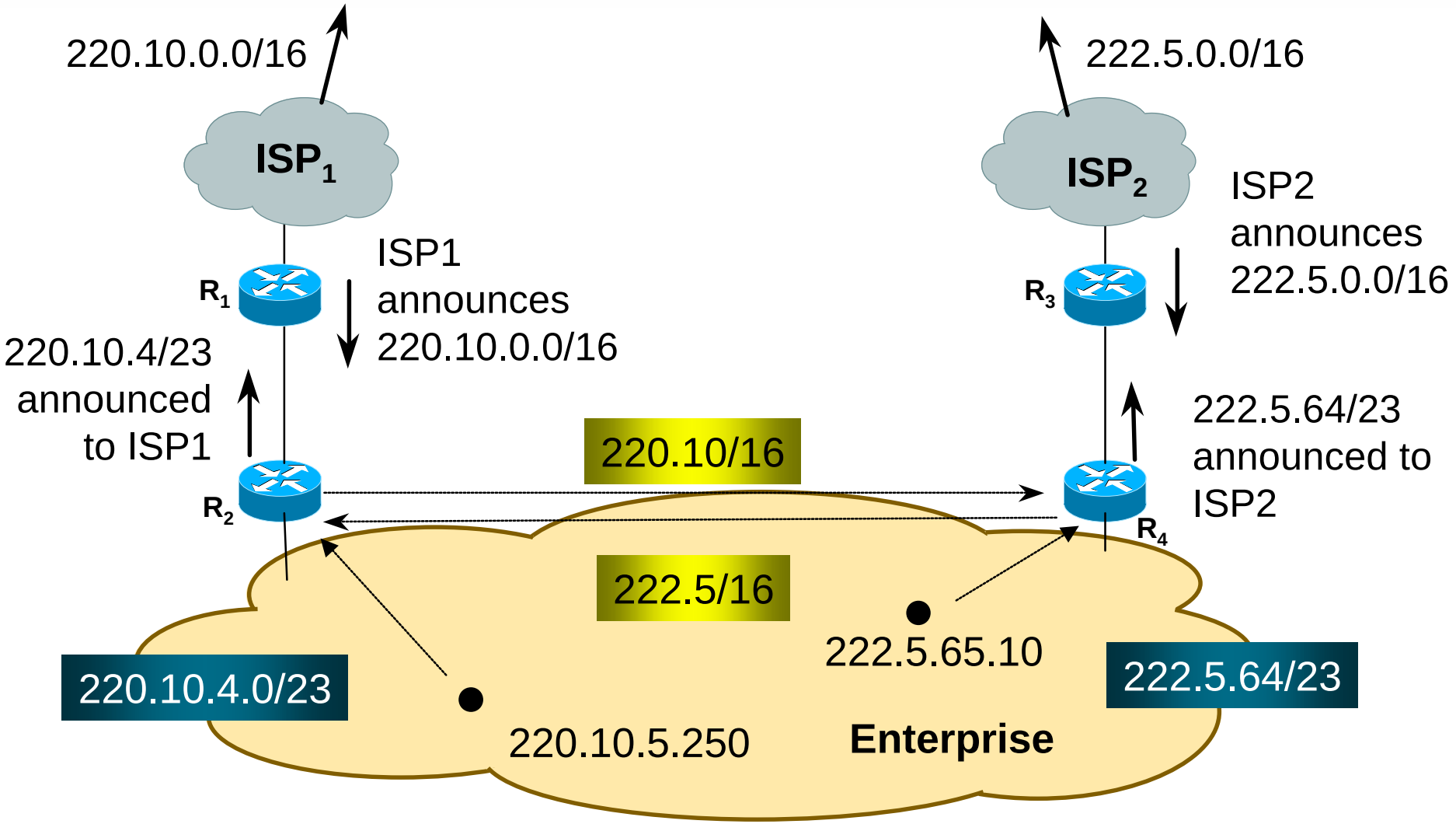
  assumes that there is a requirement for 3x /23 in the Enterprise backbone

- **Next example assumes equal bandwidth links from Enterprise to ISP1 and ISP2**

Enterprise Multihoming Conditional Advertisement

- **New conditional advertisement feature in Cisco IOS**
  - **loadsharing under normal conditions**
  - **subprefixes only announced in failure scenarios**
  - **requires upstreams to announce only one prefix to enterprise border network**

Steady State



Steady State

- ISP1 has 220.10.0.0/16 address block
- ISP2 has 222.5.0.0/16 address block
- Enterprise customer multihomes
  - upstreams don't announce subprefixes
  - can use private AS (ASN>64511)
  - R2 and R4 originate default in their IGP
    - outbound traffic uses nearest exit (IGP metrics)

Steady State

- **Router2 configuration:**

```
router bgp 65534
 network 220.10.4.0 mask 255.255.254.0
 network 222.5.64.0 mask 255.255.254.0
 neighbor <R1> remote-as 150
 neighbor <R1> prefix-list isp1-in in
 neighbor <R1> prefix-list isp1-out out
 neighbor <R1> advertise-map isp2-sb non-exist-map isp2-bb
 neighbor <R4> remote-as 65534
 neighbor <R4> update-source loopback 0
!
ip route 220.10.4.0 255.255.254.0 null0 250
..next slide
```

Steady State

```
ip route 222.5.64.0 255.255.254.0 null0 250
!
ip prefix-list isp1-out permit 220.10.4.0/23
ip prefix-list isp2-out permit 222.5.64.0/23
!
ip prefix-list isp1-in permit 220.10.0.0/16
ip prefix-list isp2-in permit 222.5.0.0/16
!
route-map isp2-sb permit 10
 match ip address prefix-list isp2-out
!
route-map isp2-bb permit 10
 match ip address prefix-list isp2-in
!
```

Steady State

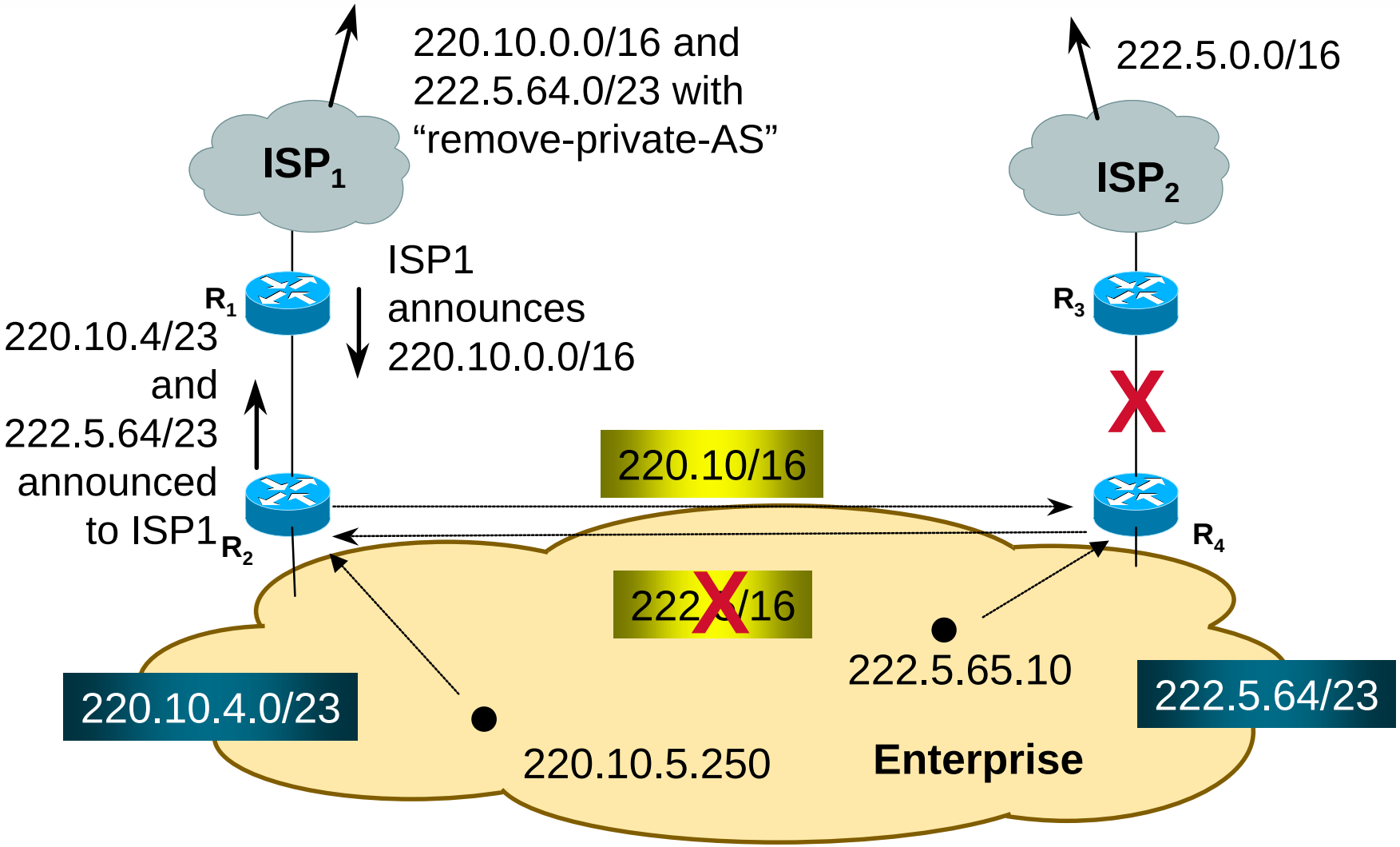
- **Router2 peers iBGP with Router4**

  hears ISP2's /16 prefix
- **Router2 peers eBGP with Router1**

  hears ISP1's /16 prefix only

  announces 220.10.4.0/23 only

Link Failure



Link Failure

- **Peering between Router 4 and Router3 (ISP2) goes down**
  - **222.5.0.0/16 prefix withdrawn**
- **Conditional advertisement process activated**
  - **Router2 starts to announce 222.5.64.0/23 to Router1**
- **Connectivity for Enterprise maintained**

Enterprise Multihoming

- **Conditional advertisement useful when address space comes from both upstreams**
  - **no subprefixes leaked to Internet unless in failure situation**
- **Alternative backup mechanism would be to leak /23 prefixes with longer AS path**
  - **routing table bloat, reachability issues**

What goes in the Internet Routing Registry?

- ISP1 and ISP2 obviously put their own address blocks as route objects in the IRR
- ISP1 will put the ISP1 subprefix which Enterprise will announce into the IRR with origin-as of ISP2
- ISP2 will put the ISP2 subprefix which Enterprise will announce into the IRR with origin-as of ISP1
- No inconsistent origin AS, no “problem”

BGP and the Internet

Advanced Community Usage

Cisco Systems

RFC1998

- Informational RFC
- Describes how to implement loadsharing and backup on multiple inter-AS links
  - BGP communities used to determine local preference in upstream's network
- Gives control to the customer
- Simplifies upstream's configuration
  - simplifies network operation!

RFC1998

- **Community values defined to have particular meanings:**

ASx:100	**set local pref 100**	**preferred route**
ASx:90	**set local pref 90**	**backup route if dualhomed on ASx**
ASx:80	**set local pref 80**	**main link is to another ISP with same AS path length**
ASx:70	**set local pref 70**	**main link is to another ISP**

RFC1998

- **Sample Customer Router Configuration**

```
router bgp 107
 neighbor x.x.x.x remote-as 109
 neighbor x.x.x.x description Backup ISP
 neighbor x.x.x.x route-map config-community out
 neighbor x.x.x.x send-community
!
ip as-path access-list 20 permit ^$
ip as-path access-list 20 deny .*
!
route-map config-community permit 10
 match as-path 20
 set community 109:90
```

RFC1998

- **Sample ISP Router Configuration**

```
! Homed to another ISP
ip community-list 70 permit 109:70
! Homed to another ISP with equal ASPATH length
ip community-list 80 permit 109:80
! Customer backup routes
ip community-list 90 permit 109:90
!
route-map set-customer-local-pref permit 10
 match community 70
 set local-preference 70
```

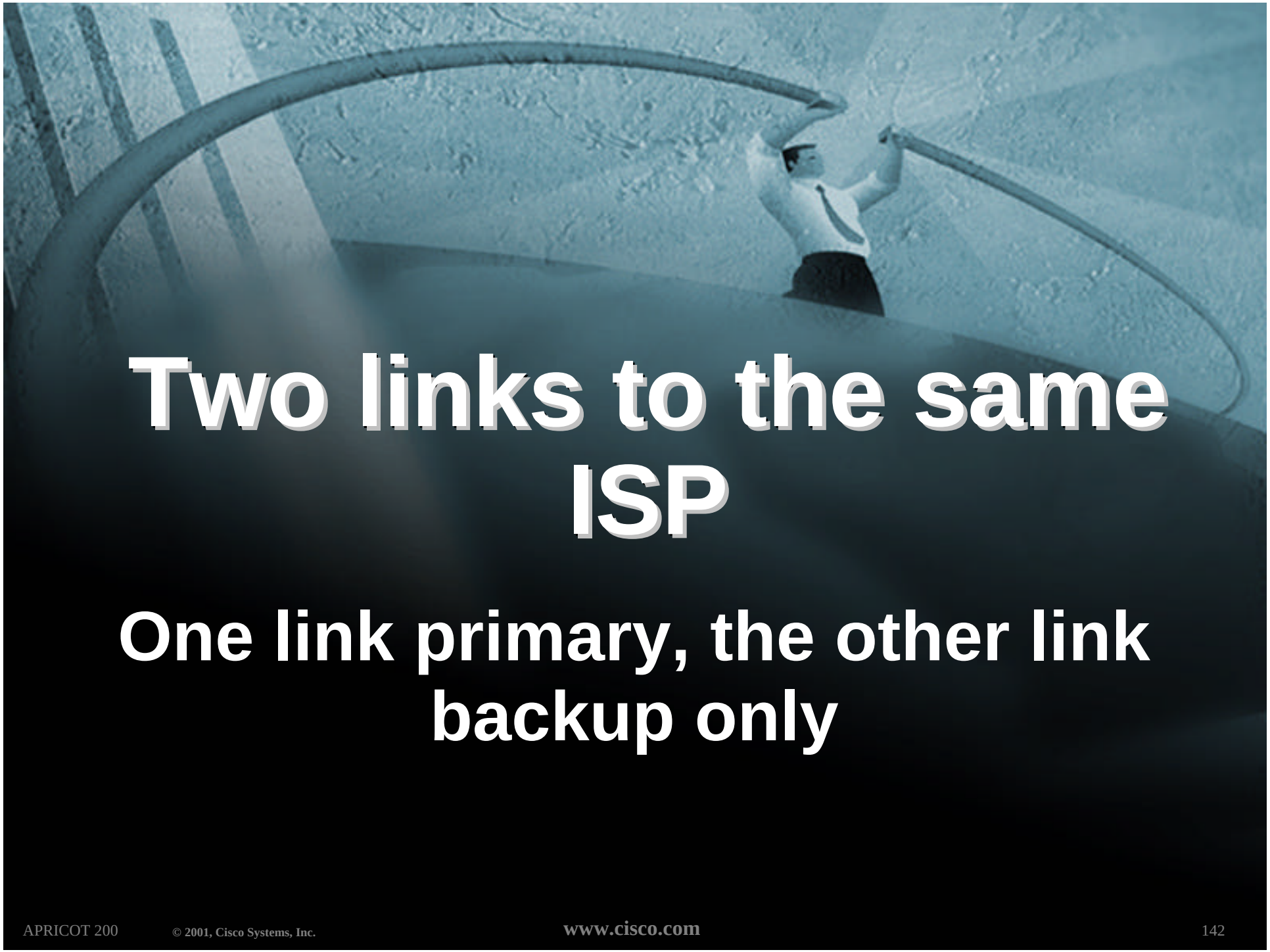
RFC1998

- **Sample ISP Router Configuration**

```
route-map set-customer-local-pref permit 20
 match community 80
 set local-preference 80
!
route-map set-customer-local-pref permit 30
 match community 90
 set local-preference 90
!
route-map set-customer-local-pref permit 40
 set local-preference 100
```

RFC1998

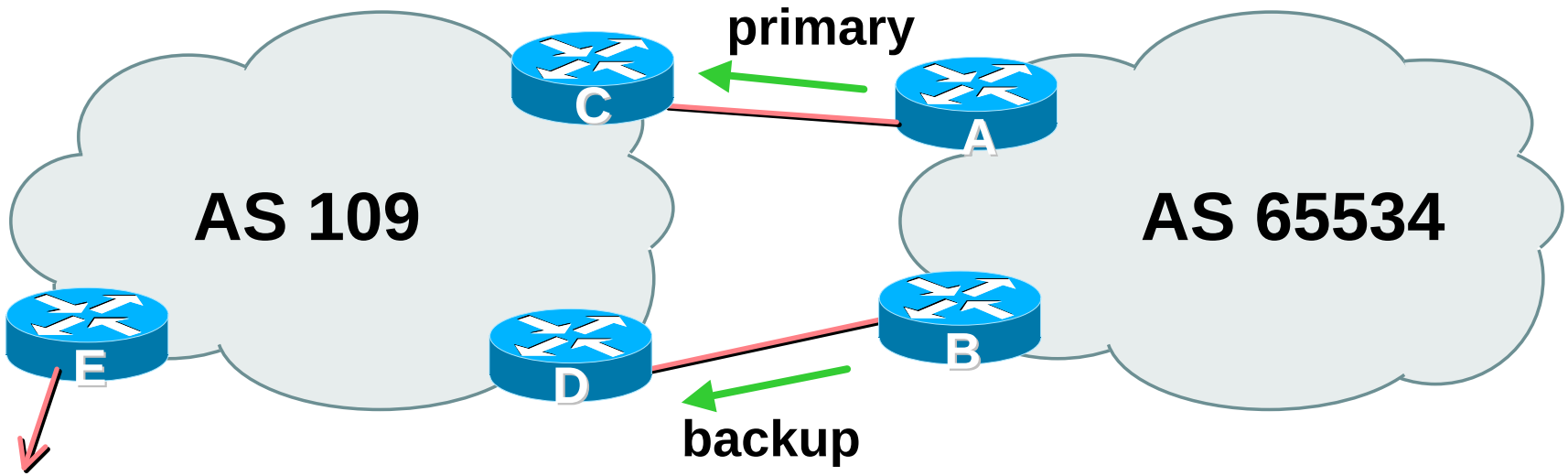
- **Supporting RFC1998**
  - **many ISPs do, more should**
  - **check AS object in the Internet Routing Registry**
  - **if you do, insert comment in AS object in the IRR**



Two links to the same ISP

One link primary, the other link backup only

Two links to the same ISP



- **AS109 proxy aggregates for AS 65534**

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

- Announce /19 aggregate on each link

  primary link makes standard announcement

  backup link sends community

- 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 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 109
 neighbor 222.222.10.2 description RouterC
 neighbor 222.222.10.2 prefix-list aggregate out
 neighbor 222.222.10.2 prefix-list default in
!
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
```

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

- **Router B Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.6 remote-as 109
 neighbor 222.222.10.6 description RouterD
 neighbor 222.222.10.6 send-community
 neighbor 222.222.10.6 prefix-list aggregate out
 neighbor 222.222.10.6 route-map routerD-out out
 neighbor 222.222.10.6 prefix-list default in
 neighbor 222.222.10.6 route-map routerD-in in
!
..next slide
```

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

```
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
route-map routerD-out permit 10
 match ip address prefix-list aggregate
 set community 109:90
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 109
 neighbor 222.222.10.1 remote-as 65534
 neighbor 222.222.10.1 default-originate
 neighbor 222.222.10.1 prefix-list Customer in
 neighbor 222.222.10.1 prefix-list default out
!
ip prefix-list Customer permit 221.10.0.0/19
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 109
 neighbor 222.222.10.5 remote-as 65534
 neighbor 222.222.10.5 default-originate
 neighbor 222.222.10.5 prefix-list Customer in
 neighbor 222.222.10.5 route-map bgp-cust-in in
 neighbor 222.222.10.5 prefix-list default out
!
ip prefix-list Customer permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
..next slide
```

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

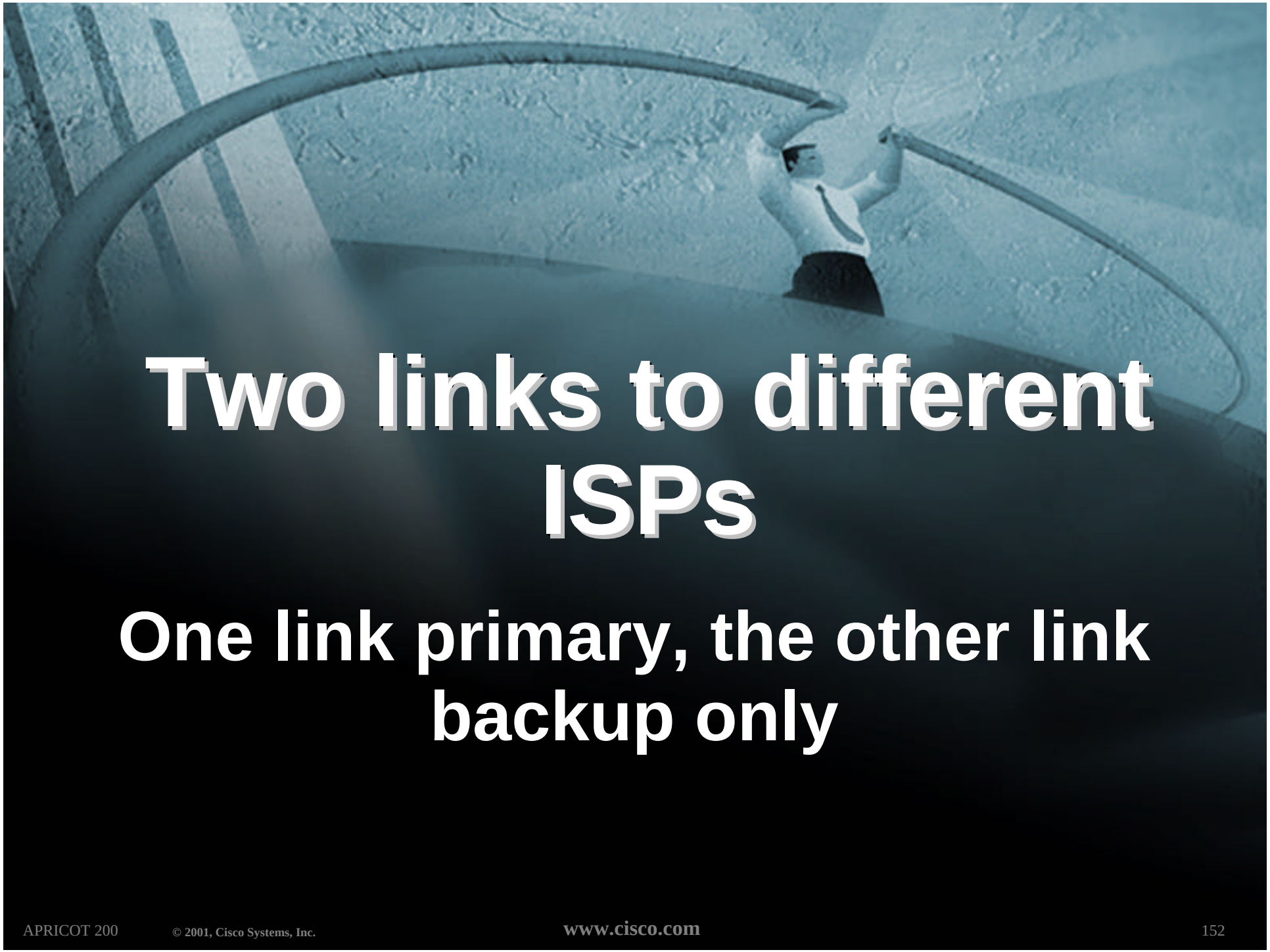
```
  ip prefix-list Customer permit 221.10.0.0/19
  ip prefix-list default permit 0.0.0.0/0
  !
  ip community-list 90 permit 109:90
  !
<snip>
  route-map bgp-cust-in permit 30
   match community 90
   set local-preference 90
  route-map bgp-cust-in permit 40
   set local-preference 100
```

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

- **Router E Configuration**

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.17 remote-as 110
 neighbor 222.222.10.17 filter-list 1 out
!
ip as-path access-list 1 deny ^(65534_)+$
ip as-path access-list 1 permit ^$
ip route 221.10.0.0 255.255.224.0 null0
```

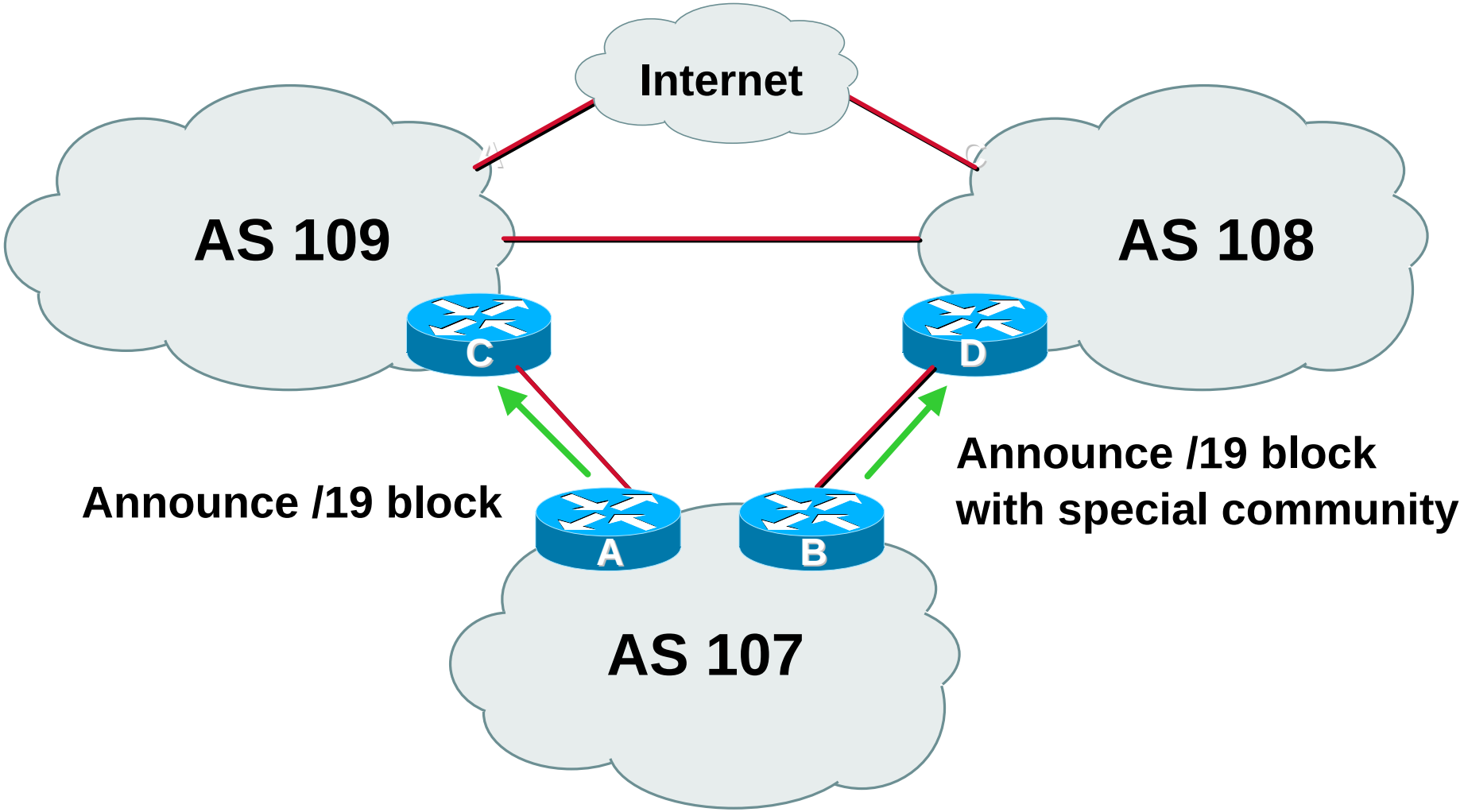
- **Router E removes prefixes in the private AS from external announcements**
- **Private AS still visible inside AS109**



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

  main link sends community 109:100 - this sets local pref in AS109 to 100

  backup link sends community 108:80 - this sets local pref in AS108 to 80

- 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)

- Note that this assumes that AS109 and AS108 are interconnected
- If they are not, AS path length "stuffing" has to be used too

  but that can also be done using communities on a per community basis

Two links to different ISPs (one as backup only)

- **Router A Configuration**

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 109
 neighbor 222.222.10.1 prefix-list aggregate out
 neighbor 222.222.10.1 route-map routerC-out out
 neighbor 222.222.10.1 prefix-list default in
!
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
route-map routerC-out permit 10
 set community 109:100
```

Two links to different ISPs (one as backup only)

- **Router B Configuration**

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 neighbor 220.1.5.1 remote-as 108
 neighbor 220.1.5.1 prefix-list aggregate out
 neighbor 220.1.5.1 route-map routerD-out out
 neighbor 220.1.5.1 prefix-list default in
 neighbor 220.1.5.1 route-map routerD-in in
..next slide
```

Two links to different ISPs (one as backup only)

```
ip prefix-list aggregate permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
!
route-map routerD-out permit 10
 set community 108:80
!
route-map routerD-in permit 10
 set local-preference 80
```

Two links to different ISPs (one as backup only)

- **Router D**

  sees path from router B with community 108:80 set - sets local preference to 80

  sees path from peering with AS109 - default local preference is 100

  local-pref comes before AS Path length

  highest local-pref wins

  traffic for AS107 is sent to AS109

Two links to different ISPs (one as backup only)

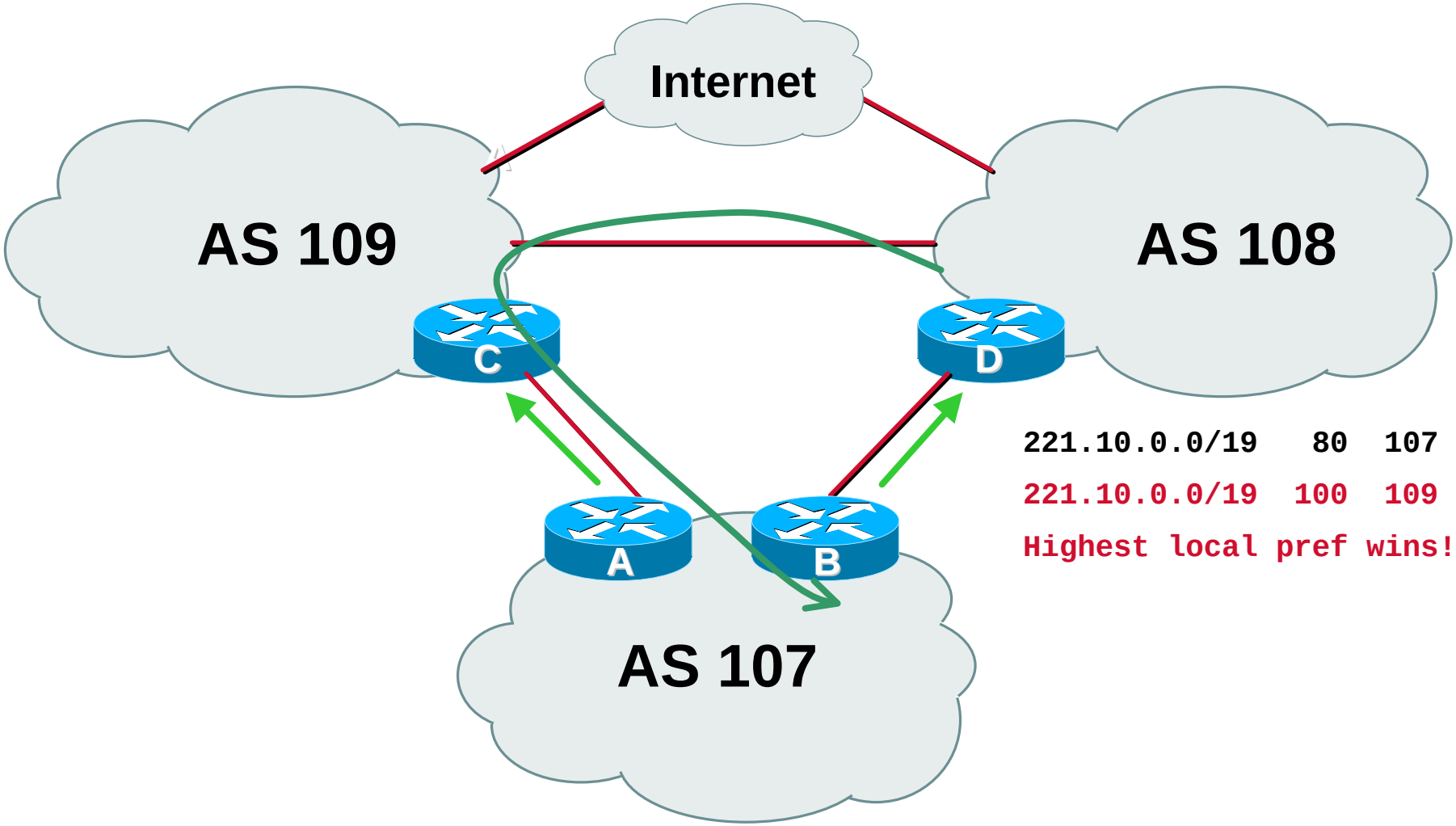
- **Router D**

  Only requires RFC1998 configuration

  no per customer configuration

  scalability!

Two links to different ISPs (one as backup only)




```
221.10.0.0/19    80  107
221.10.0.0/19   100  109 107
Highest local pref wins!
```

Two links to different ISPs (one as backup only)

- If AS107 wants to make the link to AS108 the main link

  sends community 108:100 to router C

  sends community 109:80 to router B

- AS108 and AS109 NOC intervention not required

Additional Community usage

Additional Communities

- RFC1998 is okay for “simple” multihomed customers
  - assumes that upstreams are interconnected
- There are other additions to RFC1998 used by many ISPs

More Community Definitions

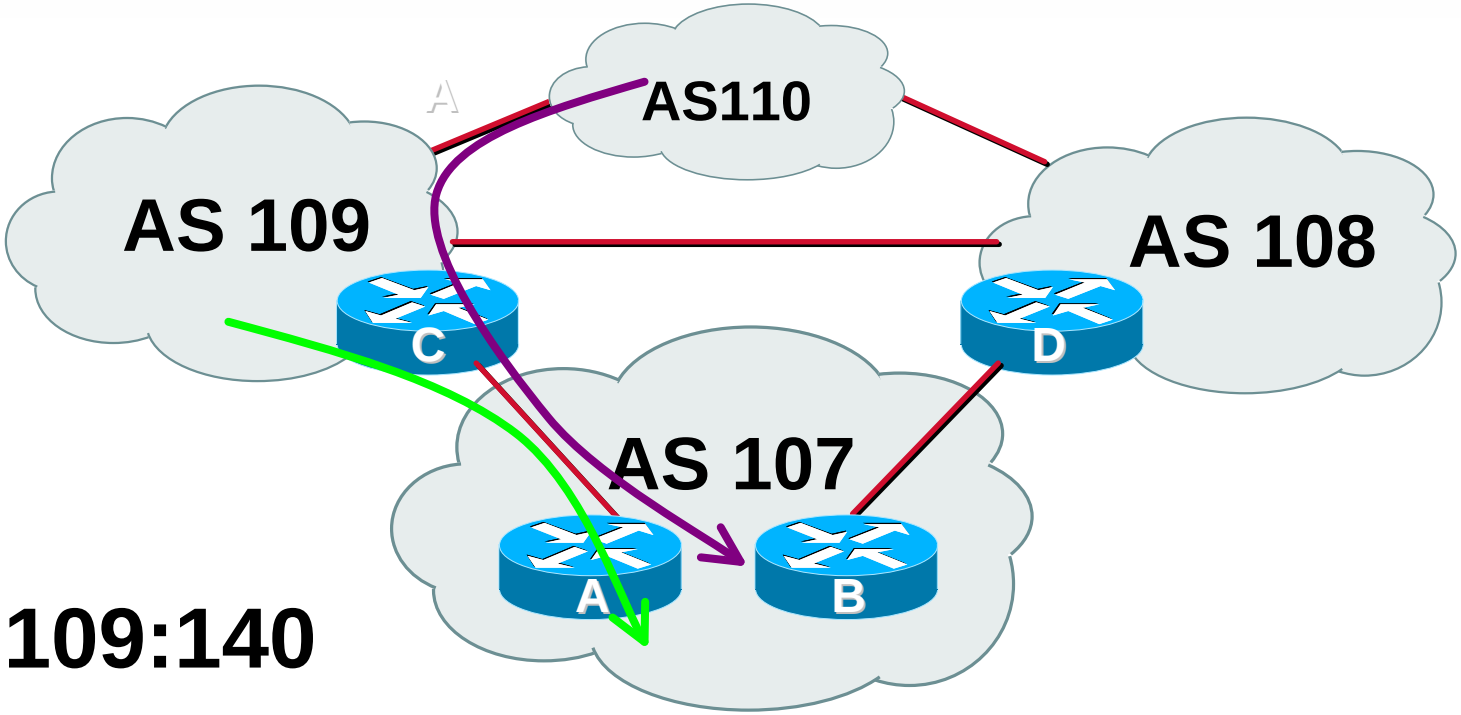
ASx:140	set local pref 140	set local pref high on upstreams
ASx:130	set local pref 130	set local pref low on upstreams
ASx:120	set local pref 120	more preferred (opposite to ASx:80)

<RFC1998 definitions>

ASx:60	set local pref 60	ASx:90 but add 2 times AS PATH
ASx:50	set local pref 50	don't announce to any peer
ASx:40	set local pref 40	and set local pref high on upstreams
ASx:30	set local pref 30	and set local pref low on upstreams

(and variations on this theme depending on local conditions, e.g. IXPs, domestic vs. international transit, etc.)

Examples

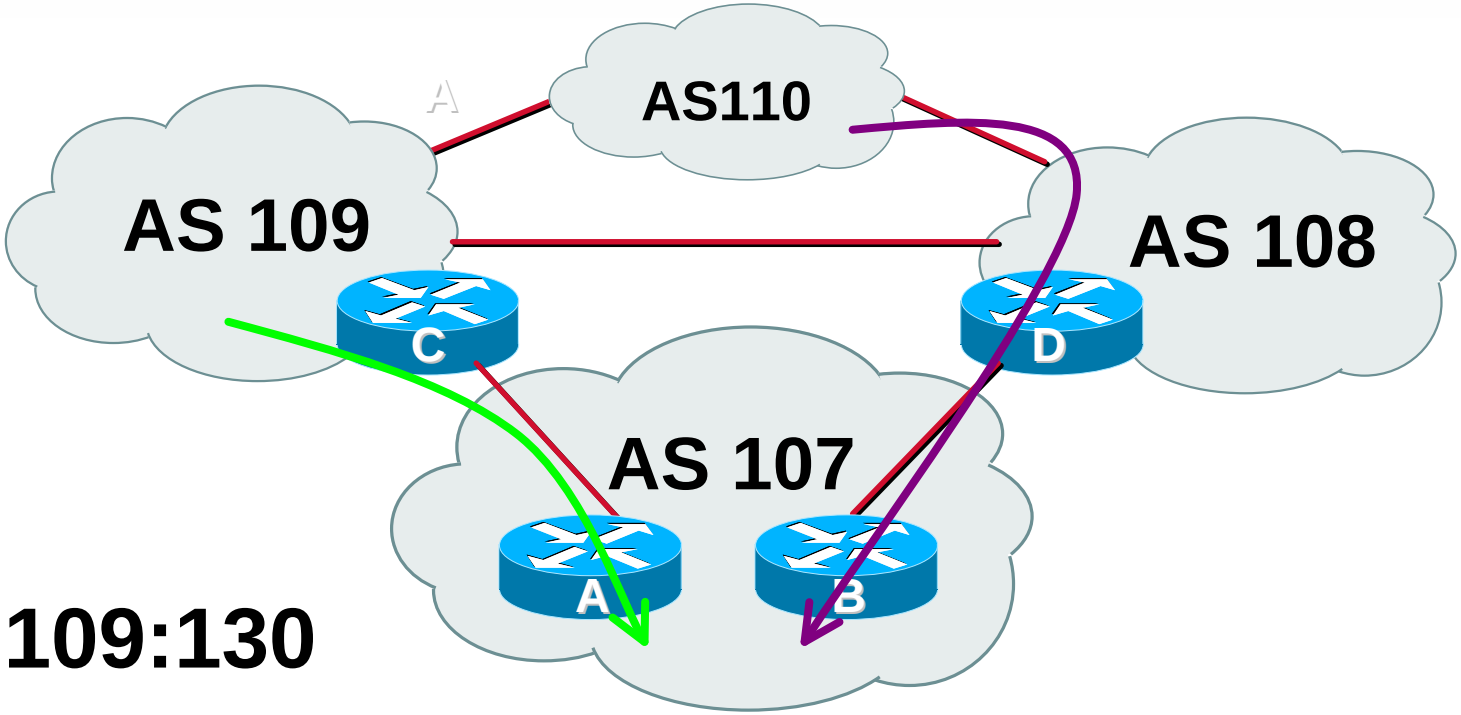


- **109:140**

  traffic in AS109 comes directly to you

  traffic in AS110 sent to AS109 rather than best path

Examples



- **109:130**

  traffic in AS109 comes directly to you

  traffic in AS110 sent to AS108 rather than best path

Examples

- **109:60**

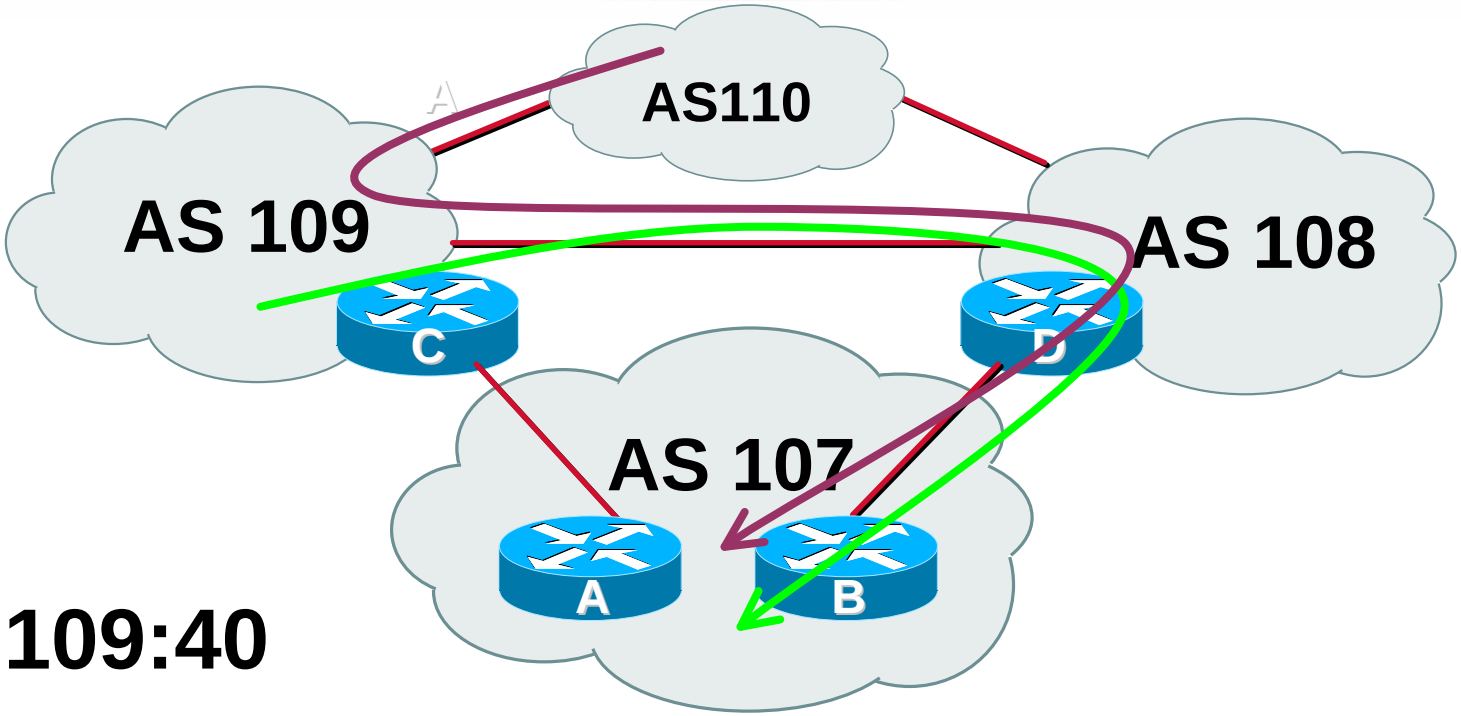
  set local pref low in AS109

  prepend any announcements to peers of AS109 with 109_109 - AS109 is my backup transit AS

- **109:50**

  don’t announce to any peer - used when you have good local connections to AS109 and better long distance via AS108

Examples

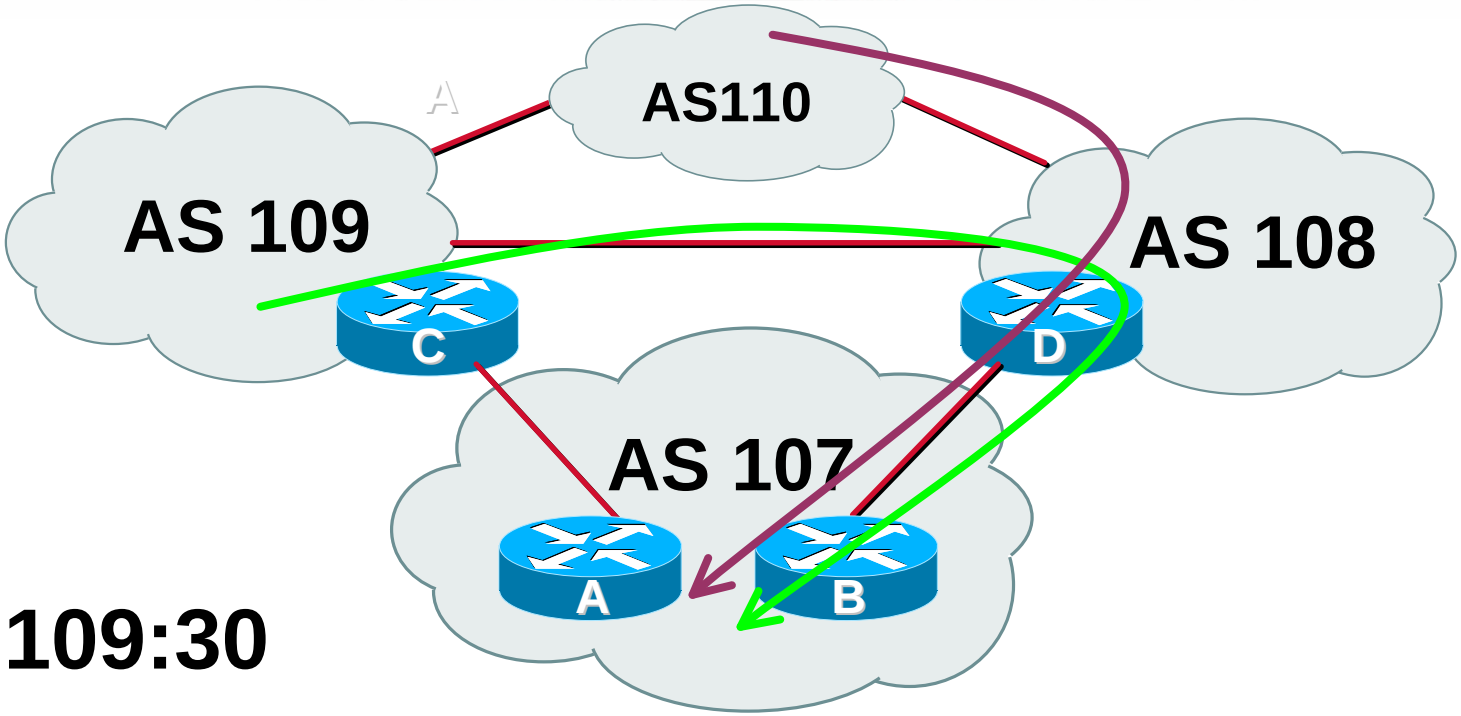


- **109:40**

  traffic in AS109 sent to AS108

  traffic in AS110 sent to AS109 rather than best path

Examples



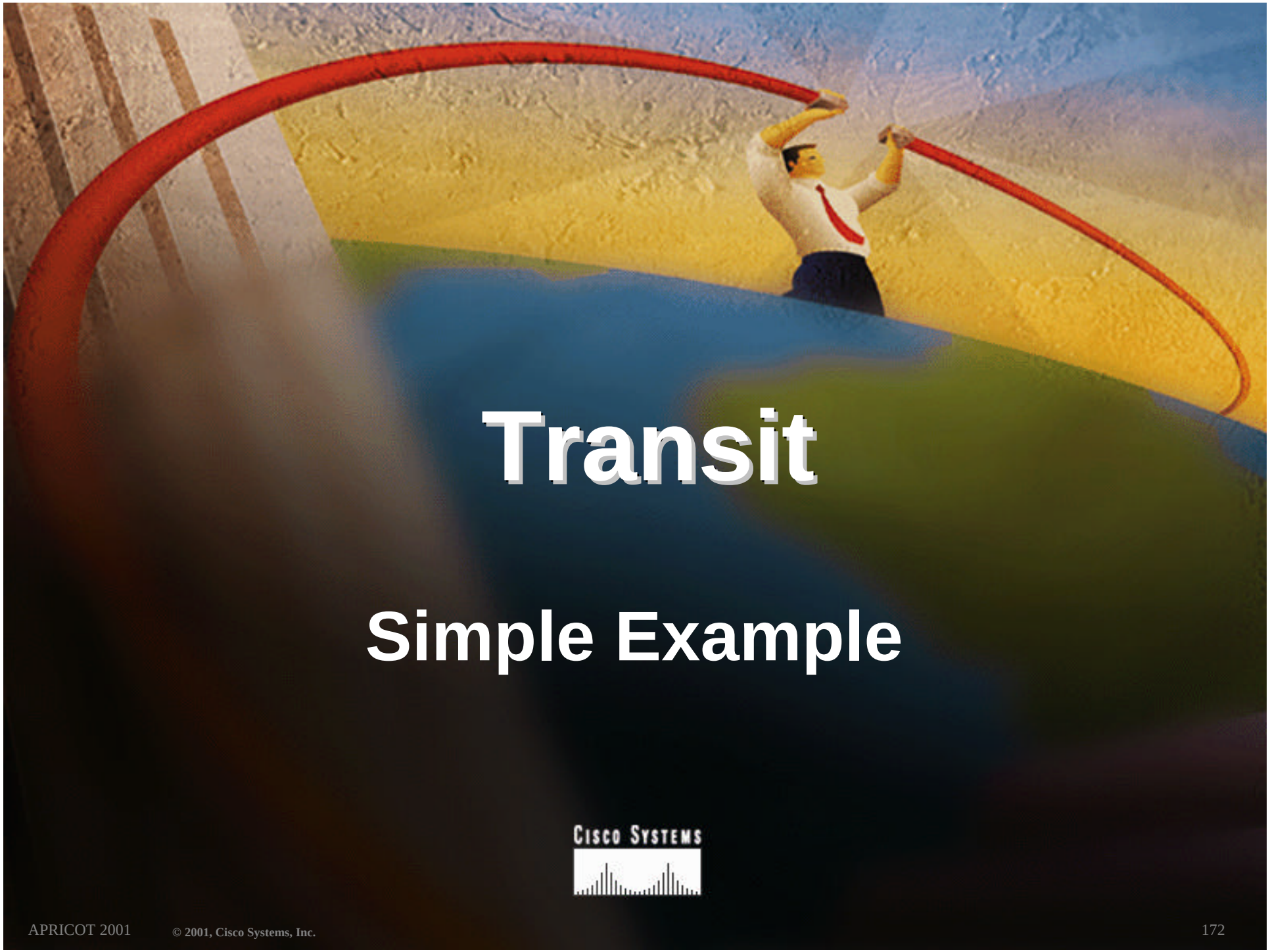
- **109:30**

  traffic in AS109 sent to AS108

  traffic in AS110 sent to AS108 rather than best path

Communities

- **Communities are fun! ☺**
- **And they are extremely powerful tools**
- **Think about community policies, e.g. like these above**
- **Supporting extensive community usage makes customer configuration easy**
- ***Watch out for routing loops!***



Definition

- **Transit - carrying traffic across a network, usually for a fee**

traffic and prefixes originating from one AS are carried across one or more intermediate ASes to reach their destination AS

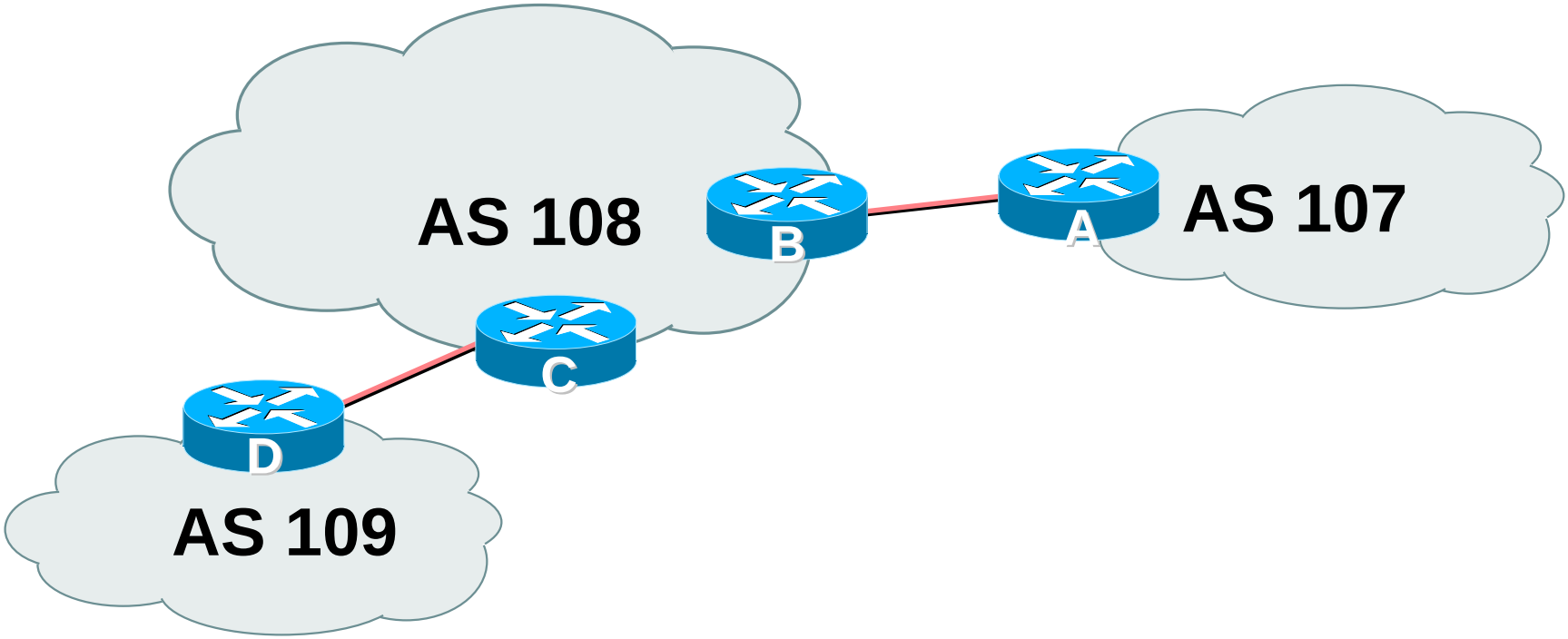
ISP Transit Issues

- Only announce default to your BGP customers unless they need more prefixes
- Only accept the prefixes which your customer is entitled to originate
- If your customer hasn't told you he is providing transit, don't accept anything else

ISP Transit Example

- **AS107 and AS109 are stub/customer ASes of AS108**
  - **they may have their own peerings with other ASes**
  - **minimal routing table desired**
  - **minimum complexity required**

ISP Transit



- **AS108 is transit provider between AS107 and AS109**

ISP Transit

- **Router A Configuration**

```
router bgp 107
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 108
 neighbor 222.222.10.2 prefix-list upstream out
 neighbor 222.222.10.2 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list upstream permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.224.0 null0
```

ISP Transit

- **Router B Configuration**

```
router bgp 108
 neighbor 222.222.10.1 remote-as 107
 neighbor 222.222.10.1 default-originate
 neighbor 222.222.10.1 prefix-list Customer107 in
 neighbor 222.222.10.1 prefix-list default out
!
ip prefix-list Customer107 permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
```

- **Router B announces default to Router A, only accepts customer /19**

ISP Transit

- **Router C Configuration**

```
router bgp 108
 neighbor 222.222.20.1 remote-as 109
 neighbor 222.222.20.1 default-originate
 neighbor 222.222.20.1 prefix-list Customer109 in
 neighbor 222.222.20.1 prefix-list default out
!
ip prefix-list Customer109 permit 219.0.0.0/19
ip prefix-list default permit 0.0.0.0/0
```

- **Router C announces default to Router D, only accepts customer /19**

ISP Transit

- **Router D Configuration**

```
router bgp 109
 network 219.0.0.0 mask 255.255.224.0
 neighbor 222.222.20.2 remote-as 108
 neighbor 222.222.20.2 prefix-list upstream out
 neighbor 222.222.20.2 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list upstream permit 219.0.0.0/19
!
ip route 219.0.0.0 255.255.224.0 null0
```

ISP Transit

- **This is simple case:**
  - **if AS107 or AS109 get another address block, it requires AS108 and their own filters to be changed**
    - **some ISP transit provider are better skilled at doing this than others!**
  - **May not scale if they are frequently adding new prefixes**

Exchange Points

Simple Example

Definition

- **Internet Exchange Point – common or neutral interconnect location where several ASes exchange routing information and traffic**

Exchange Point Example

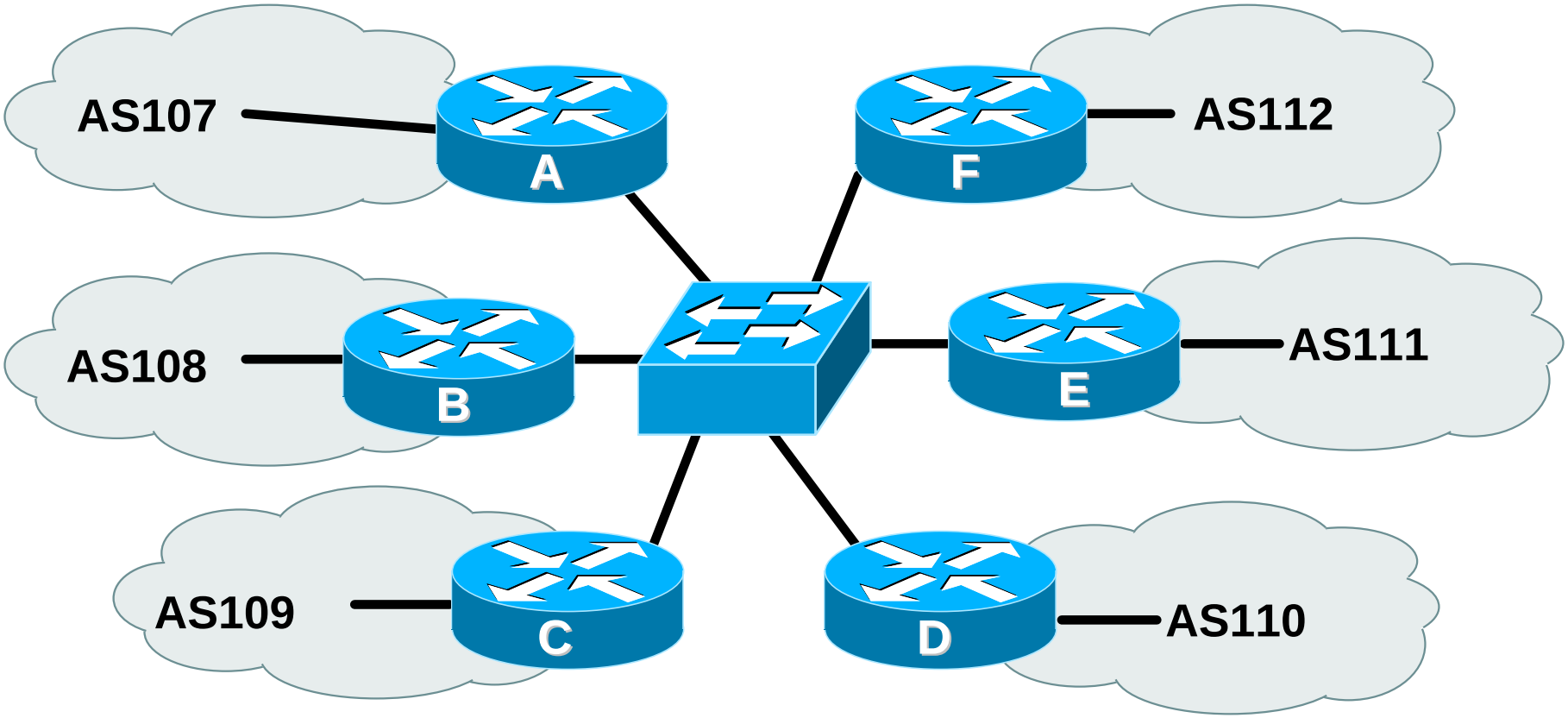
- **Exchange point with 6 ASes present**

  Layer 2 - ethernet switch

- **Each ISP peers with the other**

  NO transit across the IXP allowed

Exchange Point



each of these represents a border router in a different autonomous system

Exchange Point Router A configuration

```
interface fastethernet 0/0
 description Exchange Point LAN
 ip address 220.5.10.2 mask 255.255.255.224
 ip verify unicast reverse-path
 no ip directed-broadcast
 no ip proxy-arp
 no ip redirects
!
router bgp 107
 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 myprefixes out
```

..next slide

Exchange Point

```
neighbor 220.5.10.2 remote-as 108
neighbor 222.5.10.2 peer-group ixp-peers
neighbor 222.5.10.2 prefix-list peer108 in
neighbor 220.5.10.3 remote-as 109
neighbor 222.5.10.3 peer-group ixp-peers
neighbor 222.5.10.3 prefix-list peer109 in
neighbor 220.5.10.4 remote-as 110
neighbor 222.5.10.4 peer-group ixp-peers
neighbor 222.5.10.4 prefix-list peer110 in
neighbor 220.5.10.5 remote-as 111
neighbor 222.5.10.5 peer-group ixp-peers
neighbor 222.5.10.5 prefix-list peer111 in
neighbor 220.5.10.6 remote-as 112
neighbor 222.5.10.6 peer-group ixp-peers
neighbor 222.5.10.6 prefix-list peer112 in
```

Exchange Point

```
!
ip route 221.10.0.0 255.255.224.0 null0
!
ip prefix-list myprefixes permit 221.10.0.0/19
ip prefix-list peer108 permit 222.0.0.0/19
ip prefix-list peer109 permit 222.30.0.0/19
ip prefix-list peer110 permit 222.12.0.0/19
ip prefix-list peer111 permit 222.18.128.0/19
ip prefix-list peer112 permit 222.1.32.0/19
!
```

Exchange Point

- Configuration of the other routers in the AS is similar in concept
- Notice inbound and outbound prefix filters

  outbound announces myprefixes only

  inbound accepts peer prefixes only

Exchange Point

- **Ethernet port configuration**

  use *ip verify unicast reverse-path*

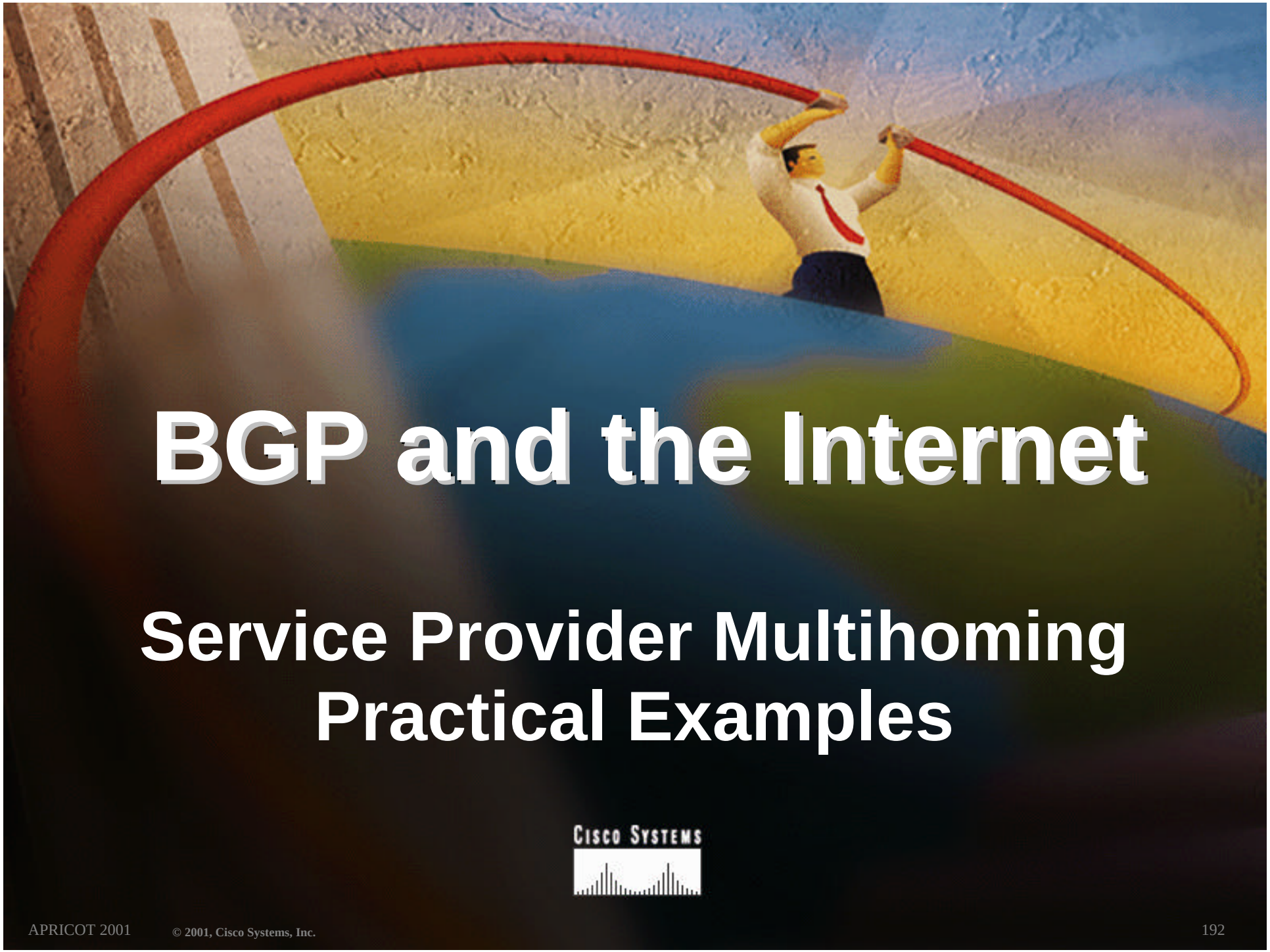
  helps prevent “stealing of bandwidth”

- **IXP border router must NOT carry prefixes with origin outside local AS and IXP participant ASes**

  helps prevent “stealing of bandwidth”

Exchange Point

- **Issues:**
  - **AS107 needs to know all the prefixes its peers are announcing**
  - **New prefixes requires the prefix-lists to be updated**
- **Alternative solutions**
  - **Use the Internet Routing Registry to build prefix list**
  - **Use AS Path filters (could be risky)**



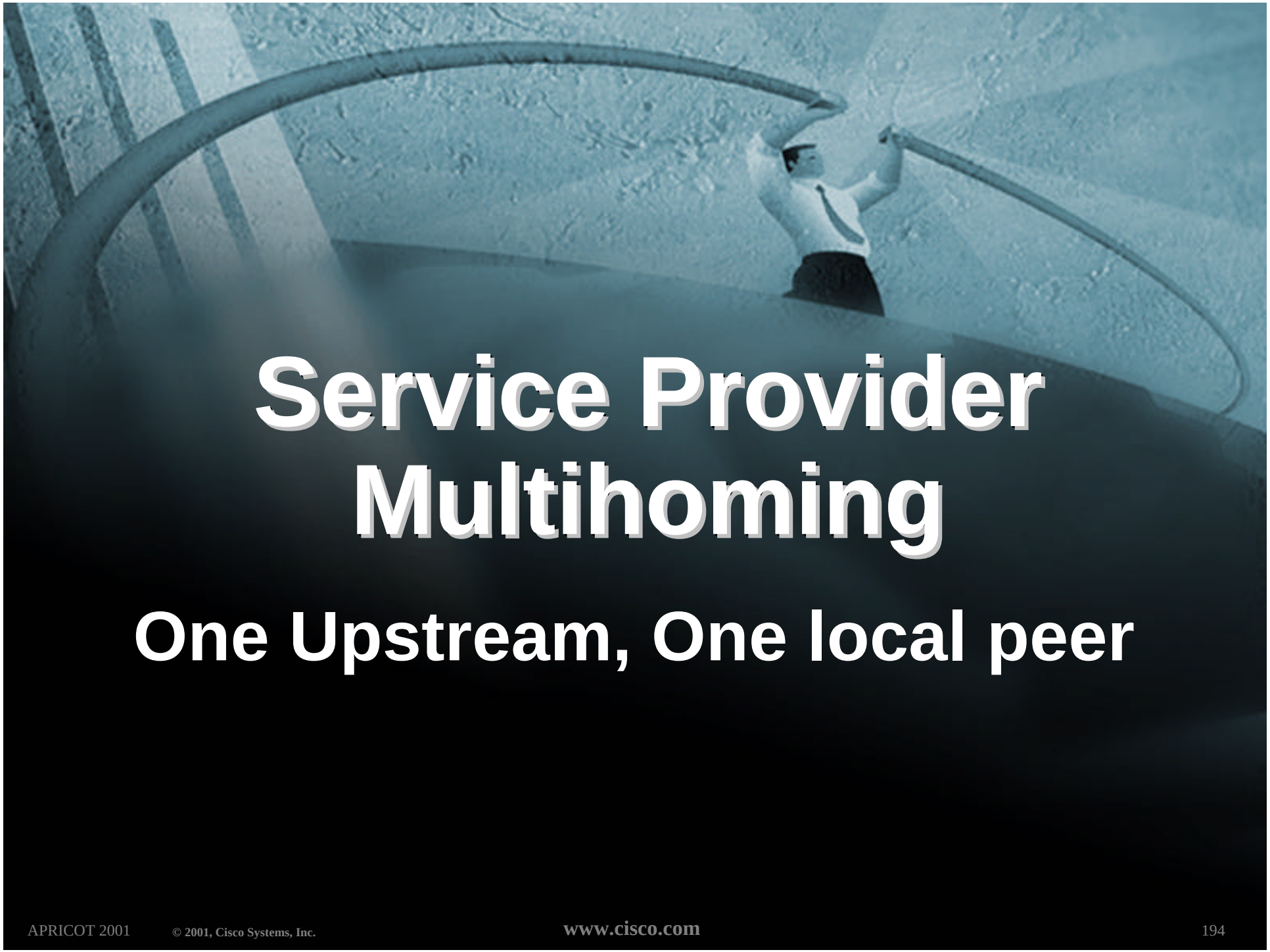
BGP and the Internet

Service Provider Multihoming
Practical Examples

Cisco Systems

Service Provider Multihoming

- **Examples**
  - **One upstream, one local peer**
  - **One upstream, local exchange point**
  - **Two upstreams, one local peer**
  - **Two upstreams, one local and one regional peer**
  - **US and regional upstreams, with local peers**
  - **IDC Multihoming**
- **All examples require BGP**



Service Provider Multihoming

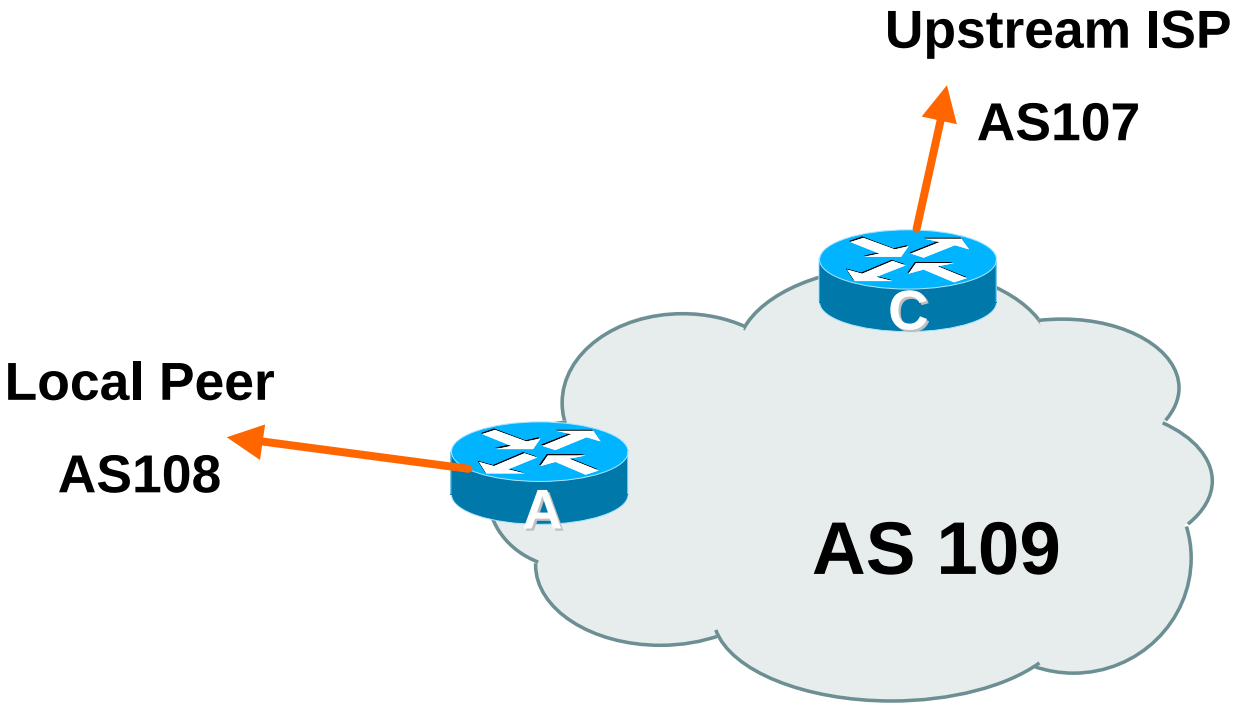
One Upstream, One local peer

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
- Border routers talk iBGP with each other

One Upstream, One Local Peer



One Upstream, One Local Peer

- **Router A Configuration**

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 108
 neighbor 222.222.10.2 prefix-list my-block out
 neighbor 222.222.10.2 prefix-list AS108-peer in
!
ip prefix-list AS108-peer permit 222.5.16.0/19
ip prefix-list AS108-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 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 108
 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 ^(108_)+$
!
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 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 107
 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
- Local traffic goes to and from local peer, everything else goes to upstream
- Routers A and C minimum requirements
  - 2500/2600 with standard RAM

Service Provider Multihoming

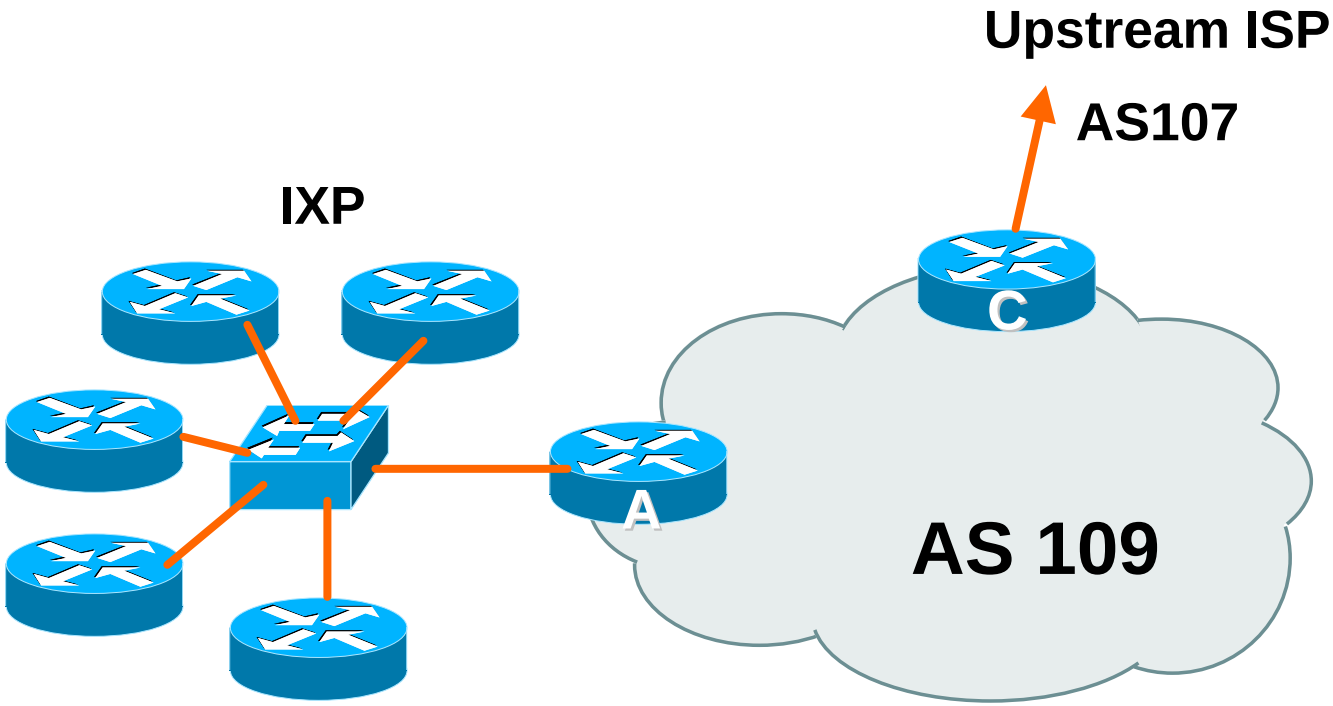
One Upstream, Local Exchange Point

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



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 109
 network 221.10.0.0 mask 255.255.224.0
..next slide
```

One Upstream, Local Exchange Point

```
 neighbor ixp-peers peer-group
 neighbor ixp-peers soft-reconfiguration in
 neighbor ixp-peers prefix-list my-block out
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
..next slide
```

One Upstream, Local Exchange Point

```
 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
 neighbor 220.5.10.6 remote-as 104
 neighbor 222.5.10.6 peer-group ixp-peers
 neighbor 222.5.10.6 prefix-list peer104 in
!
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
…
```

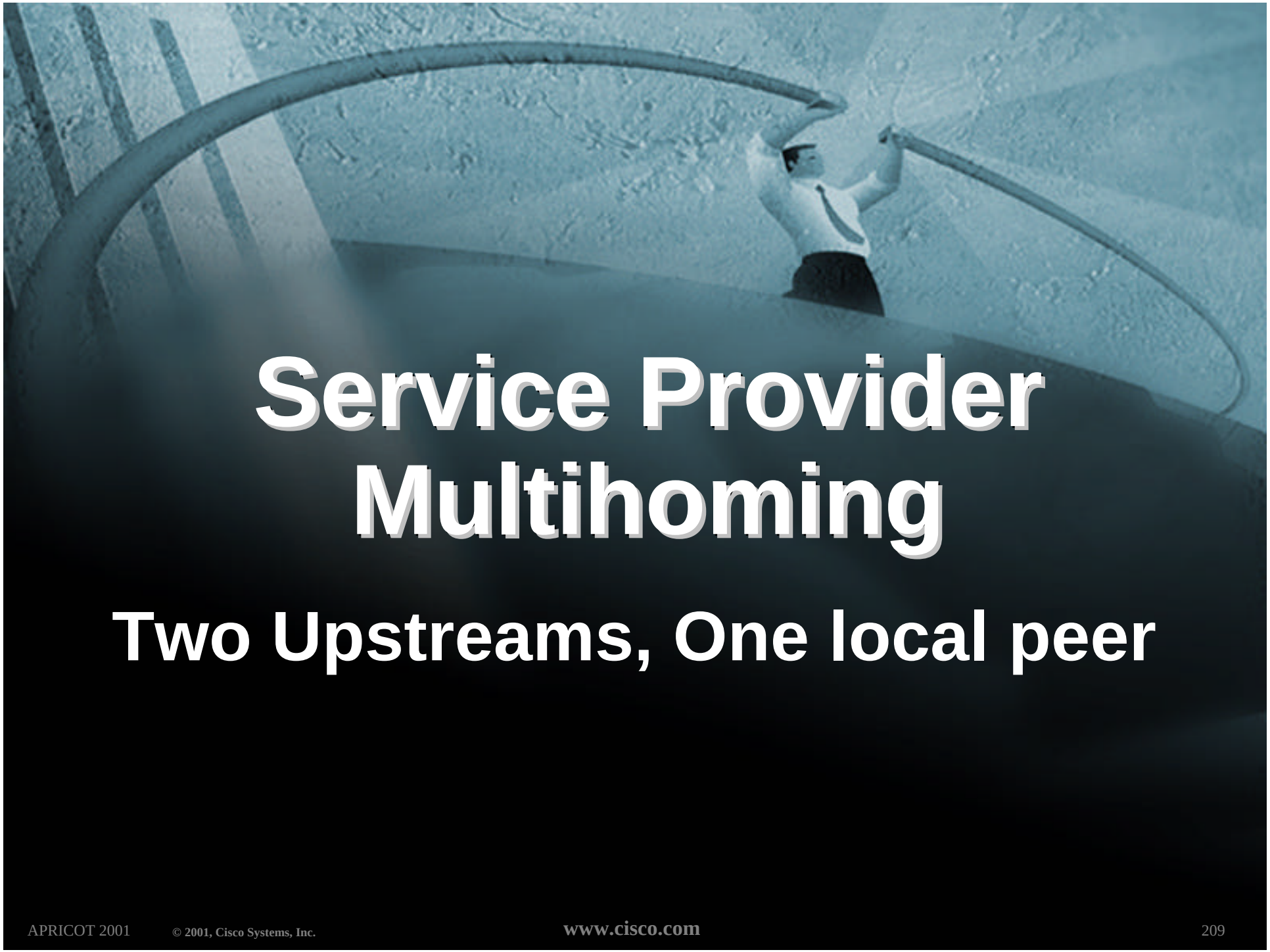
One Upstream, Local Exchange Point

- **Router C Configuration**

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 107
 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
- Minimum requirements
  - Router A: 2500/2600 with large RAM
  - Router C: 2500/2600 with standard RAM



Service Provider Multihoming

Two Upstreams, One local peer

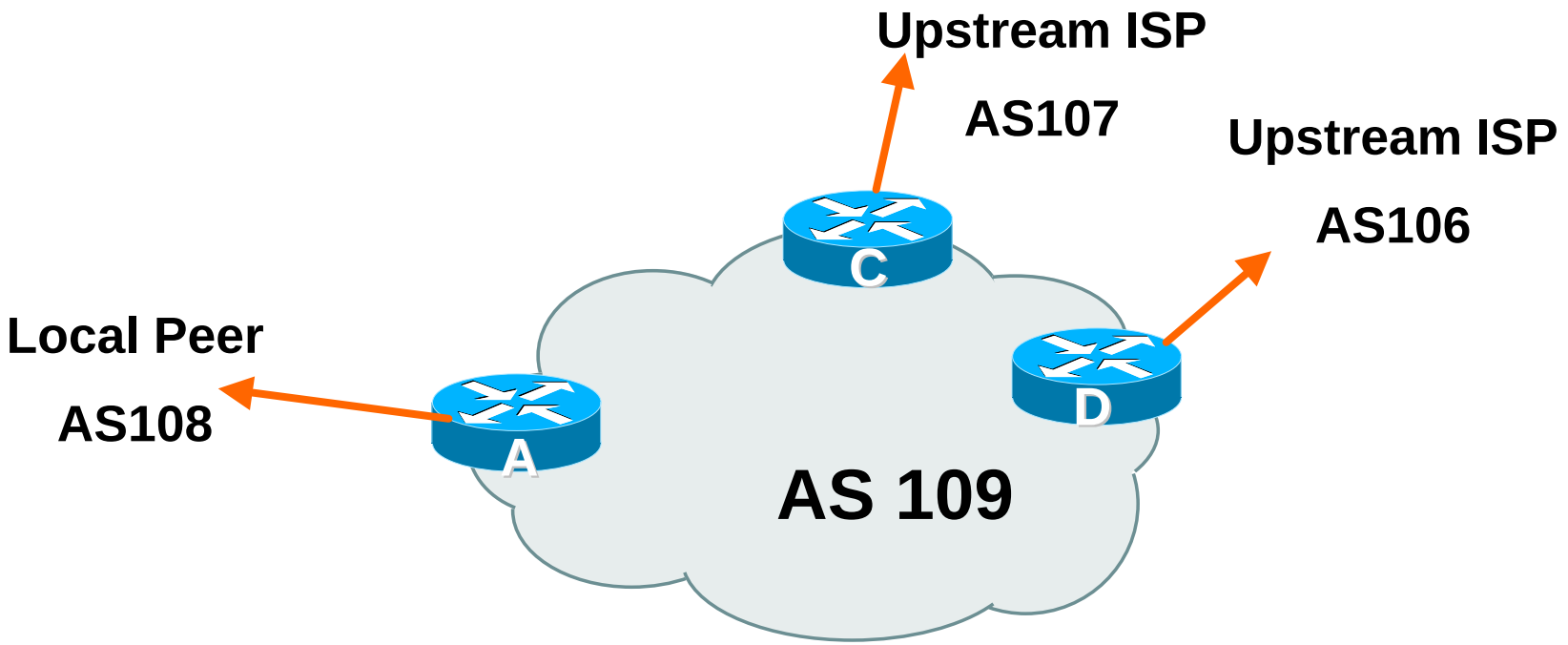
Two Upstreams, One Local Peer

- 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



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 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 107
 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 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.5 remote-as 106
 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!**
  - **Accept default from one upstream and some routes from the other upstream**
    - **Best compromise!**

Two Upstreams, One Local Peer – Full Routes

- **Router C Configuration**

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 107
 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 AS107-loadshare in
!
ip prefix-list my-block permit 221.10.0.0/19
! See earlier presentation for RFC1918 list
!
ip route 221.10.0.0 255.255.224.0 null0
..next slide
```

Two Upstreams, One Local Peer – Full Routes

```
ip as-path access-list 10 permit ^(107_)+$
ip as-path access-list 10 permit ^(107_)+_[0-9]+$
!
route-map AS107-loadshare permit 10
 match ip as-path 10
 set local-preference 120
route-map AS107-loadshare permit 20
 set local-preference 80
!
```

Two Upstreams, One Local Peer – Full Routes

- **Router C configuration:**
  - **Accept full routes from AS107**
  - **Tag prefixes originated by AS107 and AS107's neighbouring ASes with local preference 120**
  - **Remaining prefixes tagged with local preference of 80**
  - **Traffic to those ASes will go over AS107 link**
  - **Traffic to other all other ASes will go over the link to AS106**
- **Router D configuration same as Router C without the route-map**

Two Upstreams, One Local Peer – Full Routes

- **Full routes from upstreams**
  - **Expensive – needs 128Mbytes RAM today**
  - **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 start of tutorial for examples**

Two Upstreams, One Local Peer – Partial Routes

- **Router C Configuration**

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 107
 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
!
ip prefix-list my-block permit 221.10.0.0/19
ip prefix-list default permit 0.0.0.0/0
! See earlier presentation for RFC1918 list
!
ip route 221.10.0.0 255.255.224.0 null0
```

Two Upstreams, One Local Peer – Partial Routes

```
ip as-path access-list 10 permit ^(107_)+$
ip as-path access-list 10 permit ^(107_)+_[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 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.5 remote-as 106
 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 AS107**
    - **(or get them to send less)**
  - **Filter ASNs so only AS107 and AS107's neighbouring ASes are accepted**
  - **Allow default, and set it to local preference 80**
  - **Traffic to those ASes will go over AS107 link**
  - **Traffic to other all other ASes will go over the link to AS106**
  - **If AS106 link fails, backup via AS107 – 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 presentation for examples

Service Provider Multihoming

Two Upstreams, One regional peer, One local peer

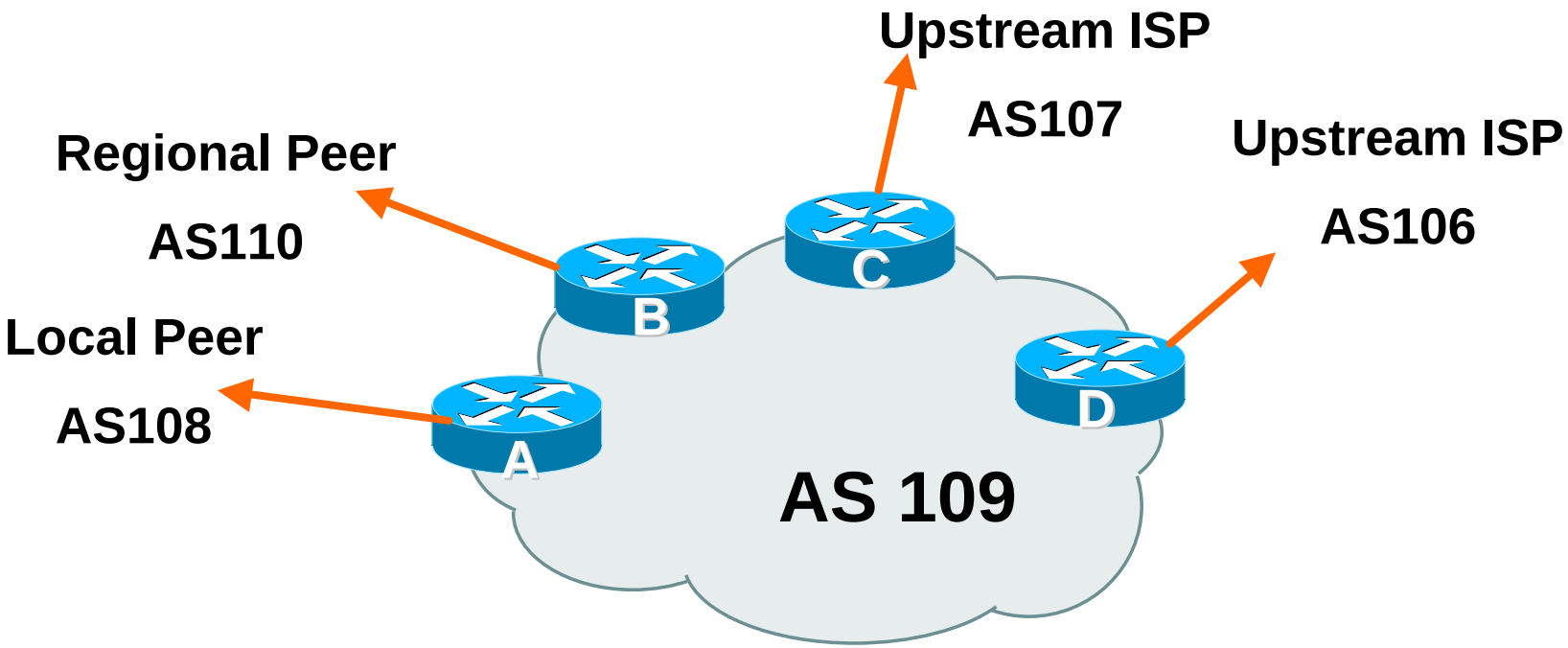
Two Upstreams, One Regional and One Local Peer

- 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
- Accept all routes from regional peer

Two Upstreams, One Regional and One Local Peer



Two Upstreams, one Regional and One Local Peer

- **Router A**
  - **Same routing configuration as in previous examples**
  - **Same hardware configuration**

Two Upstreams, one Regional and One Local Peer

- **Router B –Configuration**

```
router bgp 109
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.5 remote-as 110
 neighbor 222.222.10.5 prefix-list my-block out
 neighbor 222.222.10.5 filter-list 10 in
!
ip as-path access-list 10 permit ^(110_)+$
ip as-path access-list 10 permit ^(110_)+_[0-9]+$
!
ip prefix-list my-block permit 221.10.0.0/19
!
ip route 221.10.0.0 255.255.224.0 null0
```

Two Upstreams, one Regional and One Local Peer

- **Configuration of Router B**
  - **Take local AS from the regional peer**
  - **Also take regional peer's customer and other ASes they give**
  - **Local and regional traffic stays in the region**
  - **The two upstreams use similar configuration to previously, loadsharing as required.**

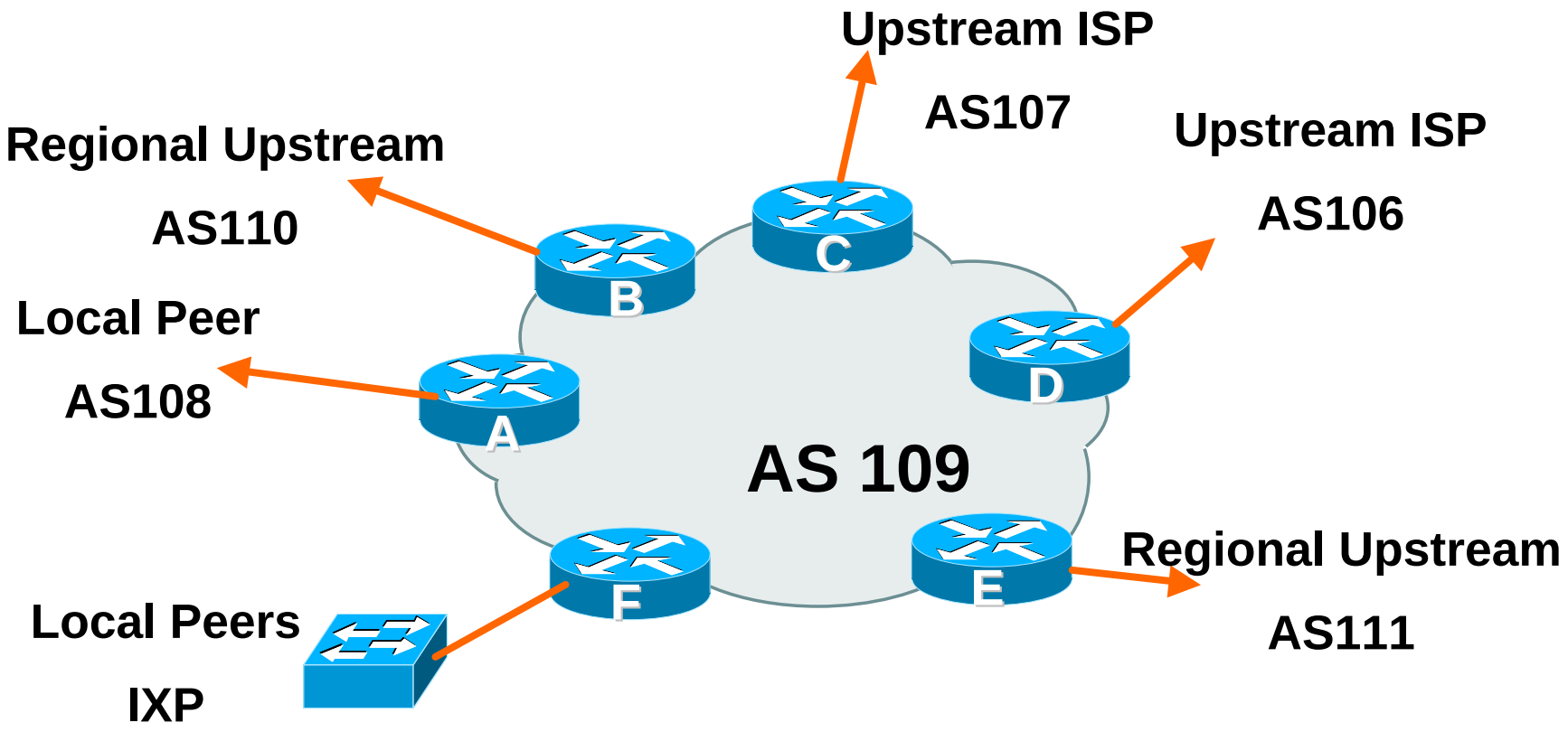
Service Provider Multihoming

Two US upstreams, two regional upstreams, and local peers

US and 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

US and Regional Upstreams, Local Peers



US and 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**
  - **Set >100 local preference on inbound announcements**

US and Regional Upstreams, Local Peers - Detail

- **Router F – local IXP peering**
  - **Accept all (local) routes**
  - **Local traffic stays local**
  - **Use prefix and/or AS-path filters**
  - **Set >100 local preference on inbound announcements**

US and Regional Upstreams, Local Peers - Detail

- **Router B – regional upstream**
  - **They provide transit to Internet, but longer AS path than US Upstreams**
  - **Accept all regional routes from them**
    - **e.g. ^110_[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 US links go down**

US and Regional Upstreams, Local Peers - Detail

- **Router E – regional upstream**
  - **They provide transit to Internet, but longer AS path than US Upstreams**
  - **Accept all regional routes from them**
    - **e.g. ^111_[0-9]+$**
  - **Ask them to send default, or send a network you can use as default**
    - **Set local pref on “default” to 50**
  - **Will provide backup to Internet only when direct US links go down**

US and Regional Upstreams, Local Peers - Detail

- **Router C – first US upstream**
  - **Accept all their customer and AS neighbour routes from them**
    - **e.g. ^107_[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 US upstream goes down**

US and Regional Upstreams, Local Peers - Detail

- **Router D – second US upstream**
  - **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**

US and 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 US upstreams
- To modify loadsharing tweak what is heard from the two regionals and the first US upstream
  - Best way is through modifying the AS-path filter

US and 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 the multihoming presentation for hints and ideas**

US and Regional Upstreams, Local Peers - Questions

- **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

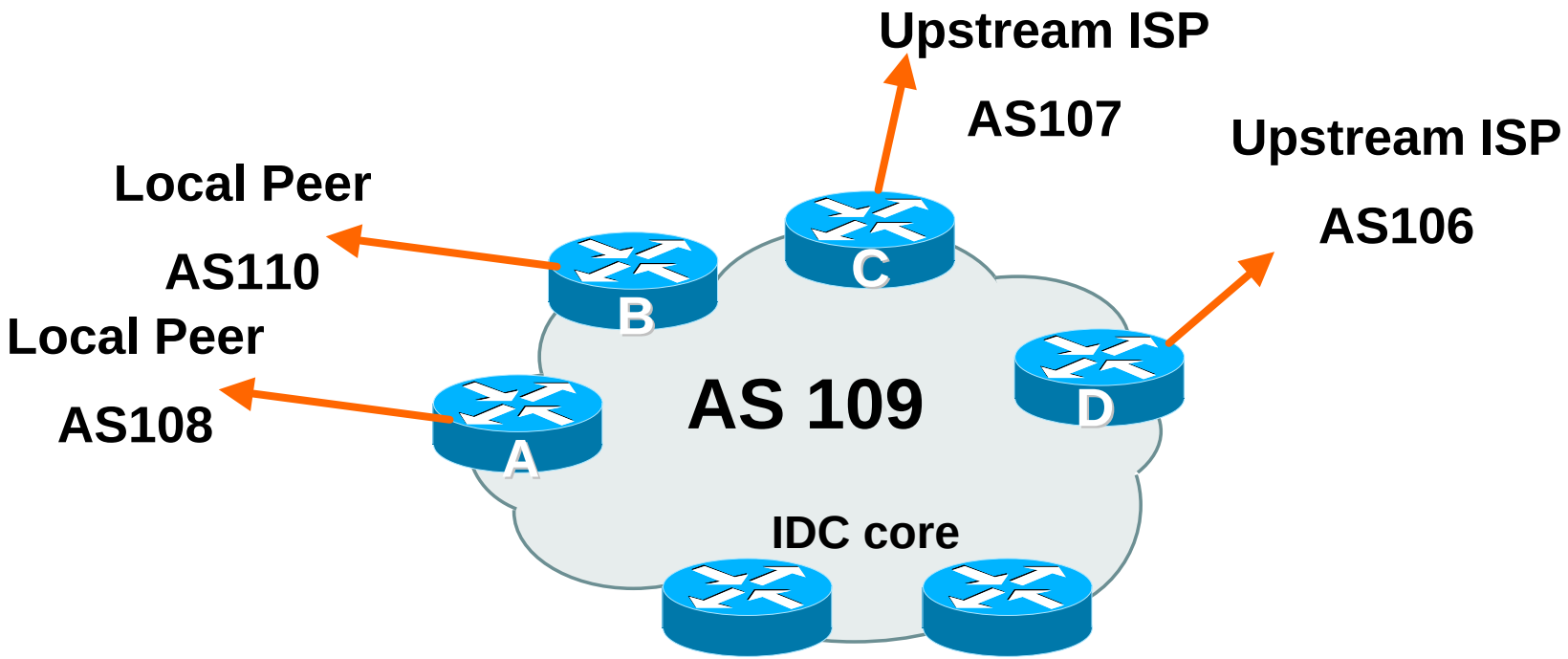
Service Provider Multihoming

IDC Multihoming

IDC Multihoming

- IDCs typically are not RIR members so are not allocated their own address block
- 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
- Example has two upstreams and two local peers

Two Upstreams, Two Local Peers - IDC



Assigned /24 from AS107 and /23 from AS106.

Circuit to AS107 is 2Mbps, circuit to AS106 is 4Mbps

IDC Multihoming

- **Router A and B configuration**

  In: Should accept all routes from AS108 and AS110

  Out: Should announce all address space to AS108 and AS110

  Straightforward

IDC Multihoming

- **Router C configuration**
  - **In: Accept partial routes from AS107**
    - **e.g. ^107_[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

  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**
- **Thought applied to address space**
  - **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!**

Service Provider Multihoming

Configuration Hints

Hints

- Use ISP software – Cisco IOS 12.0S
- Use route-refresh

  If not supported by peer AS, use soft-reconfiguration and make sure router has enough memory

  Don’t forget “clear ip bgp <neigh> *in|out*”
- Use route-flap damping

  Check RIPE-210 for recommended values

More Hints

- **Full routing table means minimum of 128Mbytes memory**
  - **Plan accordingly – routing table is growing exponentially**
- **Prefix-lists to filter prefixes**
- **Filter-lists to filter ASNs**
- **Route-maps to implement policy**
- **Keep it Simple!**

The End!

- **Any Questions?**

- **Please fill in evaluation form**
- **The presentation will be available at**

  http://www.cisco.com/public/cons/isp/documents

- **My contact info:**

  Philip Smith <pfs@cisco.com>