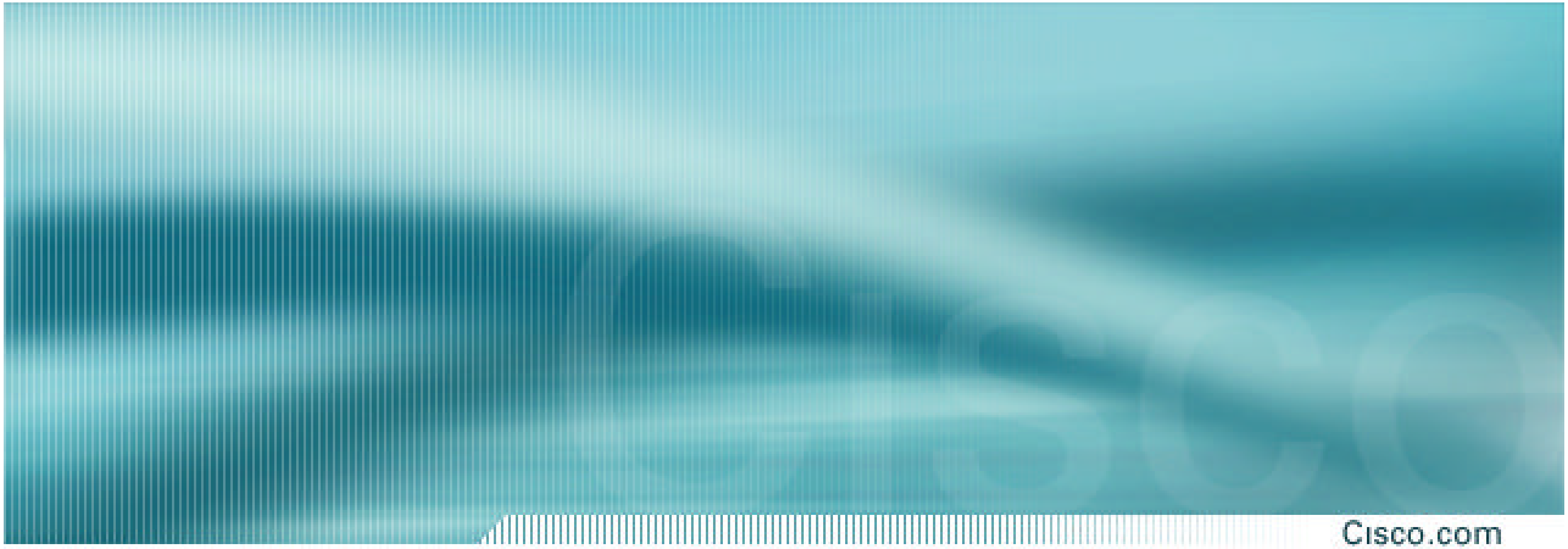


BGP and the Internet

Multihoming

Why Multihome?

- Redundancy

  One connection to internet means the network is dependent on:

  - Local router (configuration, software, hardware)

  - WAN media (physical failure, carrier failure)

  - Upstream Service Provider (configuration, software, hardware)

Why Multihome?

- **Reliability**

  Business critical applications demand continuous availability

  Lack of redundancy implies lack of reliability implies loss of revenue

Why Multihome?

- Supplier Diversity
  - Many businesses demand supplier diversity as a matter of course
  - Internet connection from two or more suppliers
    - With two or more diverse WAN paths
    - With two or more exit points
    - With two or more international connections
    - Two of everything

Why Multihome?

- Not really a reason, but oft quoted…
- Leverage:
  - Playing one ISP off against the other for:
    - Service Quality
    - Service Offerings
    - Availability

Why Multihome?

- **Summary:**
  - Multihoming is easy to demand as requirement of any operation
  - But what does it really mean:
    - In real life?
    - For the network?
    - For the Internet?
  - And how do we do it?

Multihoming Definition

- More than one link external to the local network
  - two or more links to the same ISP
  - two or more links to different ISPs
- Usually two external facing routers
  - one router gives link and provider redundancy only

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 (RIR)

  APNIC, ARIN, LACNIC, RIPE NCC

- Necessary when you have links to more than one ISP or an exchange point
- 16 bit integer, ranging from 1 to 65534

  Zero and 65535 are reserved

  64512 through 65534 are called Private ASNs

Private-AS – Application

- Applications

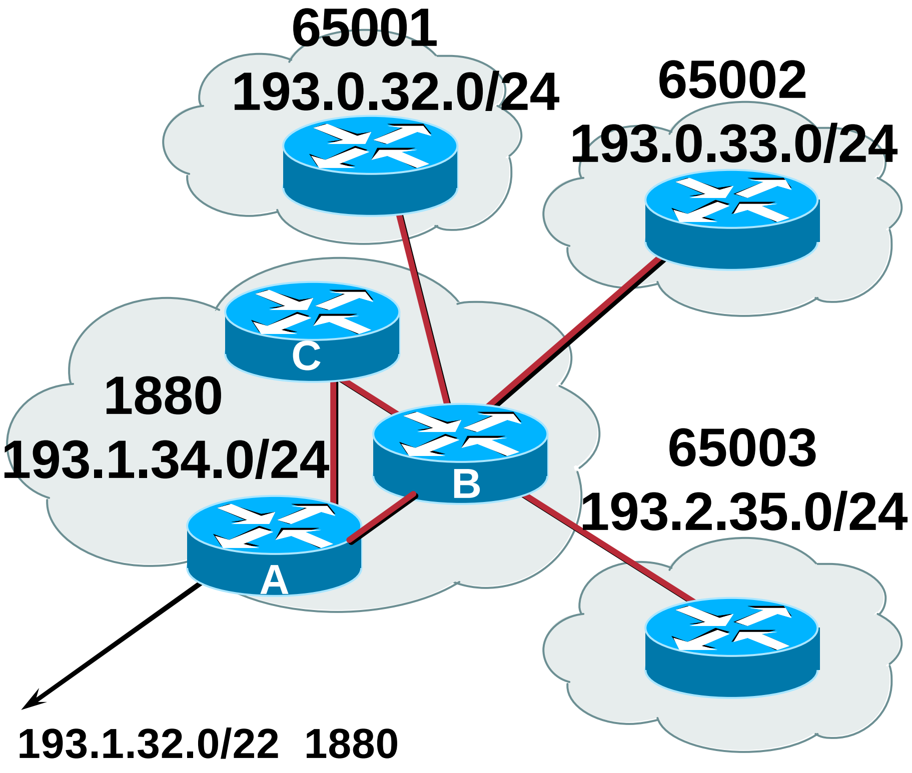
  An ISP with customers multihomed on their backbone (RFC2270)

  -or-

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

  -or-

  Within a BGP Confederation



Private-AS – removal

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

  Include configuration to remove private ASNs in the eBGP template

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

  They should not be leaked to the public Internet

- Cisco IOS

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

Configuring Policy

- **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

- **Route-maps can be used for filtering, but this is more “advanced” configuration**

Policy Tools

- **Local preference**

  outbound traffic flows

- **Metric (MED)**

  inbound traffic flows (local scope)

- **AS-PATH prepend**

  inbound traffic flows (Internet scope)

- **Communities**

  specific inter-provider peering

Originating Prefixes: Assumptions

- MUST announce assigned address block to Internet
- MAY also announce subprefixes – reachability is not guaranteed
- Current RIR minimum allocation is /20

  Several ISPs filter RIR blocks on this boundary

  Several ISPs filter the rest of address space according to the IANA assignments

  This activity is called “Net Police” by some

Originating Prefixes

- RIRs publish their minimum allocation sizes:

APNIC:	www.apnic.net/db/min-alloc.html
ARIN:	ww1.arin.net/statistics/index.html#cidr
LACNIC:	*unknown*
RIPE NCC:	www.ripe.net/ripe/docs/smallest-alloc-sizes.html

- IANA publishes the address space it has assigned to end-sites and allocated to the RIRs:

  www.iana.org/assignments/ipv4-address-space

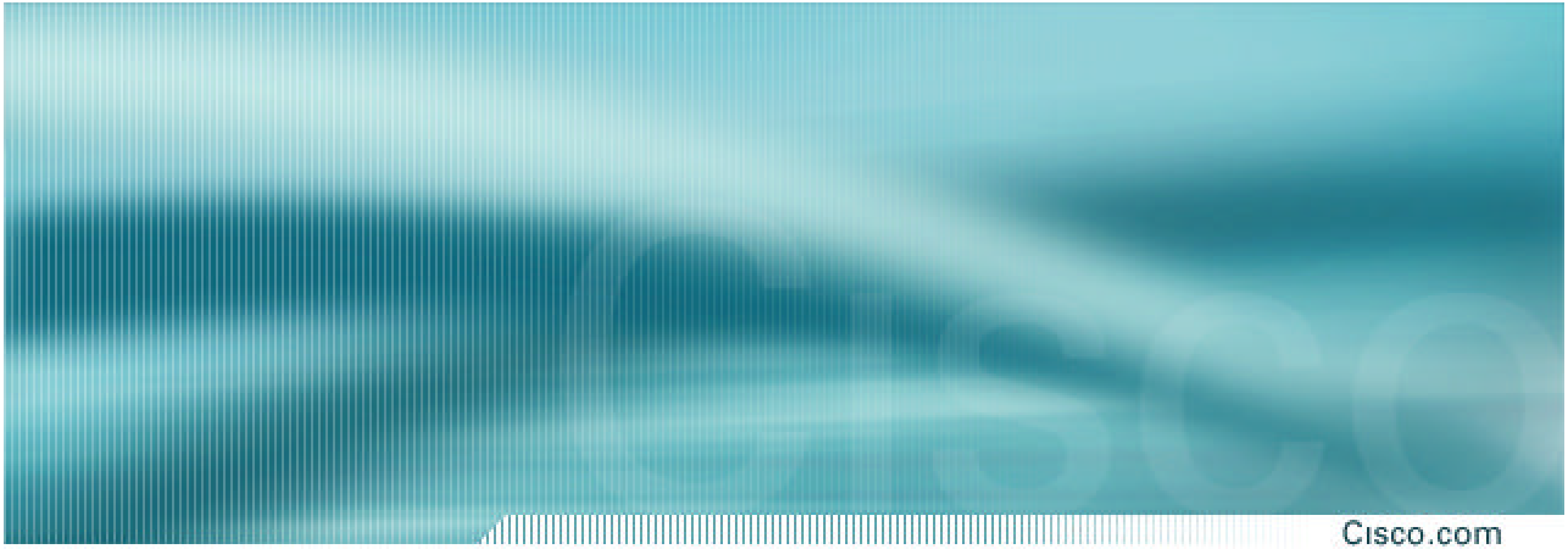
- Several ISPs use this published information to filter prefixes on:

  What should be routed (from IANA)

  The minimum allocation size from the RIRs

“Net Police” prefix list issues

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

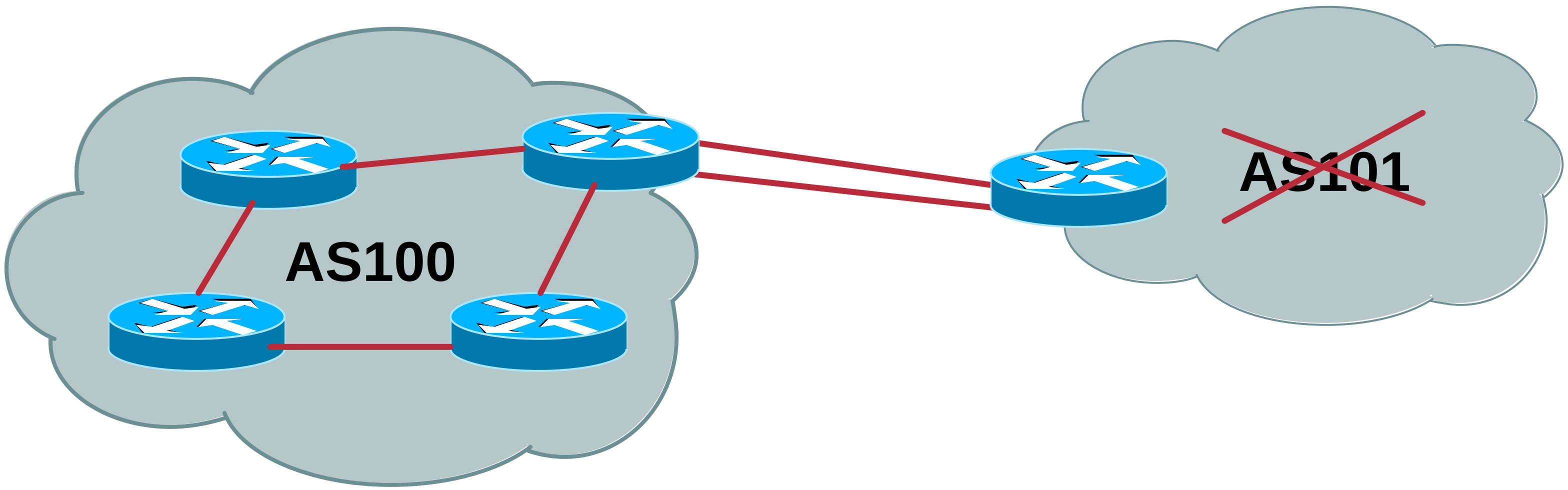


Multihoming Options

Multihoming Scenarios

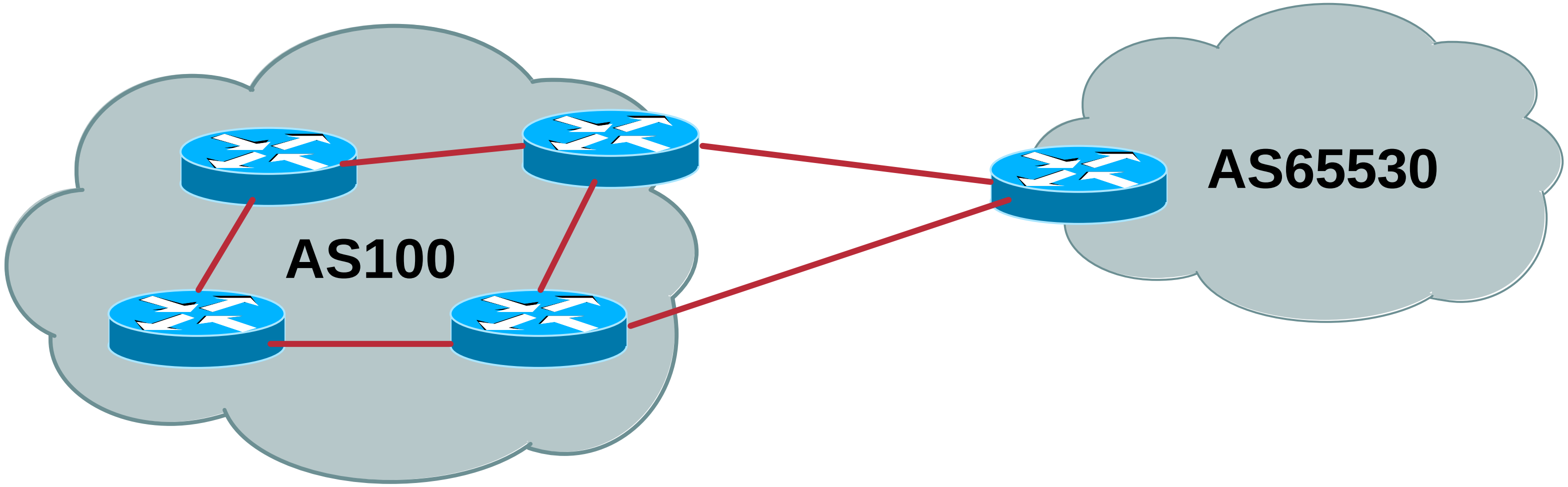
- Stub network
- Multi-homed stub network
- Multi-homed network
- Configuration Options

Stub Network



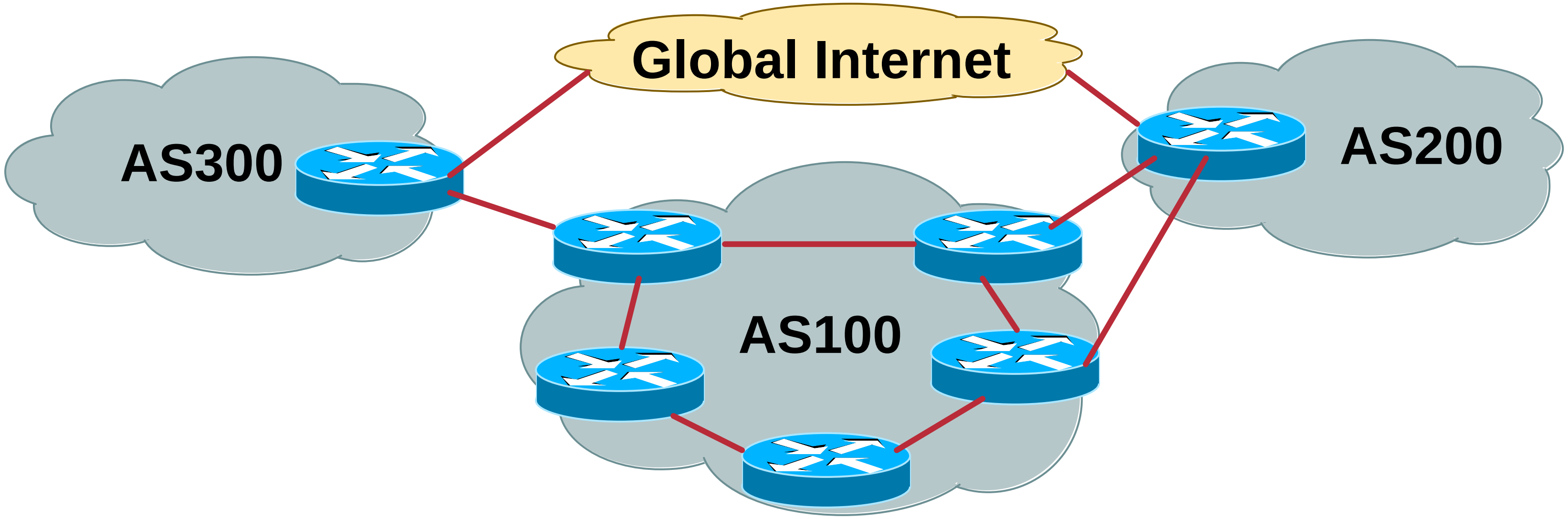
- No need for BGP
- Point static default to upstream ISP
- Upstream ISP advertises stub network
- Policy confined within upstream ISP’s policy

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

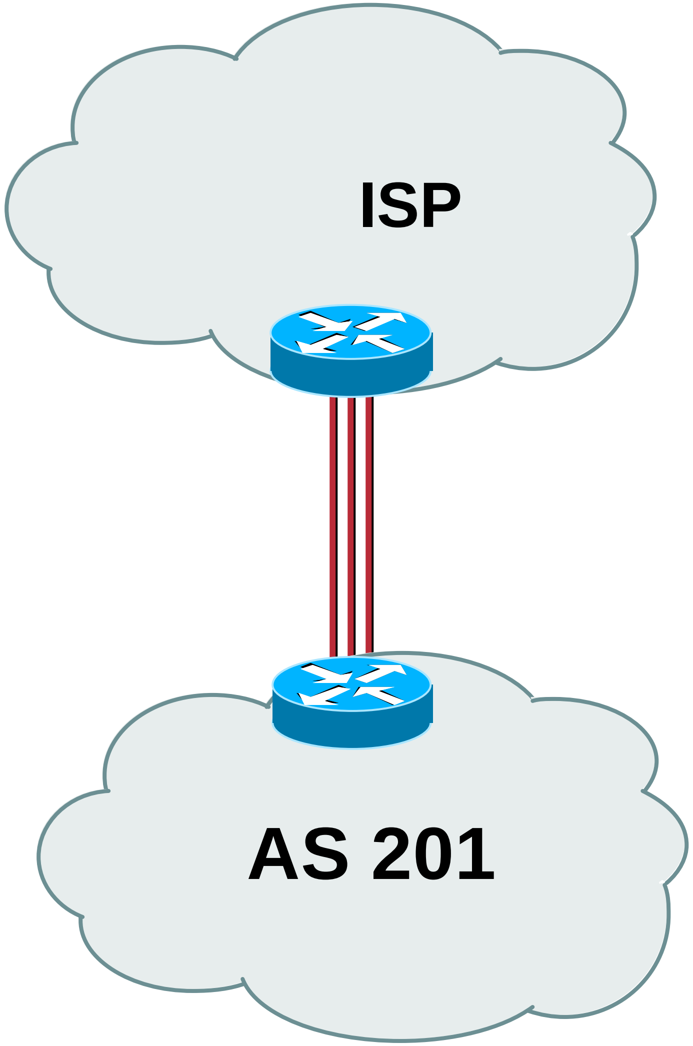
- Several options

  ebgp multihop

  bgp multipath

  cef loadsharing

  bgp attribute manipulation



Multiple Sessions to an ISP – Example One

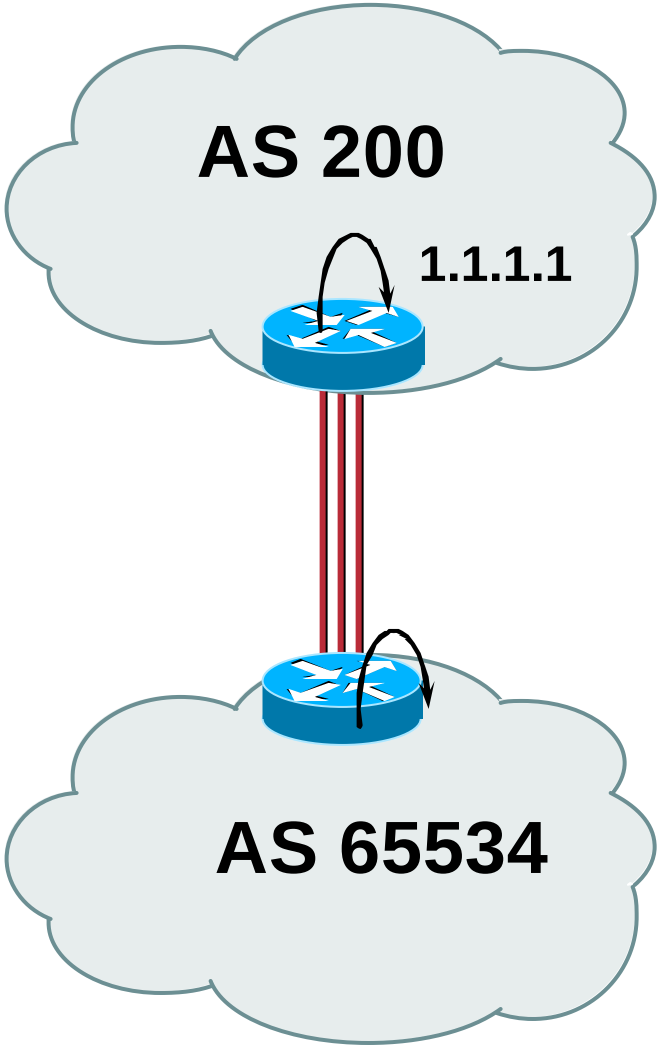
- Use eBGP multihop

    eBGP to loopback addresses

    eBGP prefixes learned with loopback address as next hop

- Cisco IOS

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



Multiple Sessions to an ISP – Example One

- Try and avoid use of ebgp-multihop unless:

  It's absolutely necessary –or–

  Loadsharing across multiple links

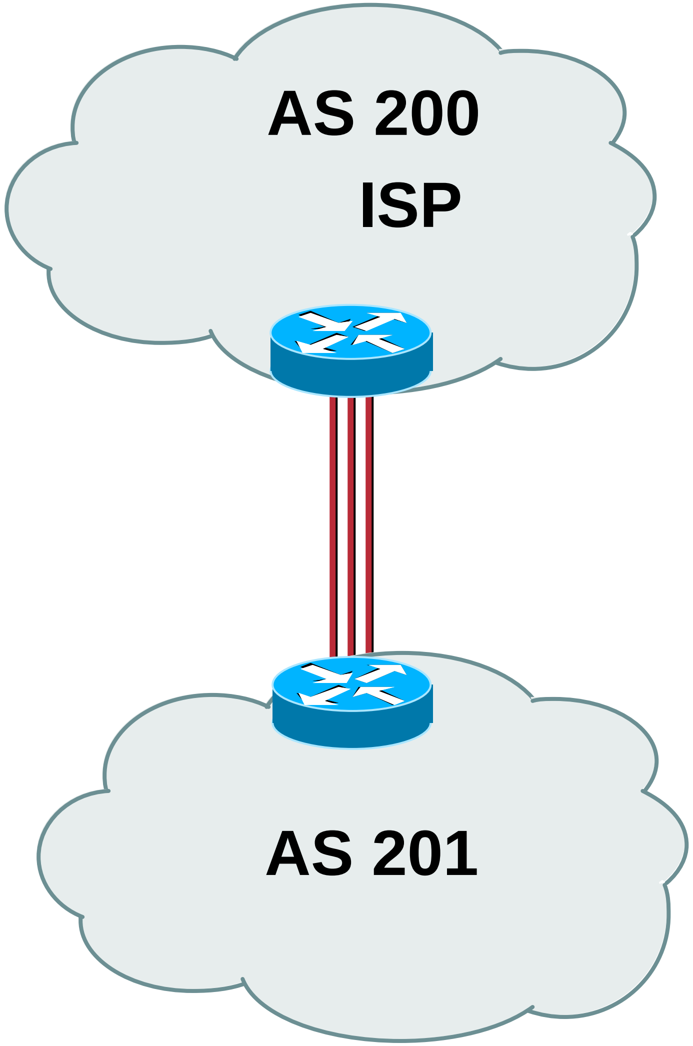
- Many ISPs discourage its use, for example:

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

Multiple Sessions to an ISP
bgp multi path

- Three BGP sessions required
- limit of 6 parallel paths

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



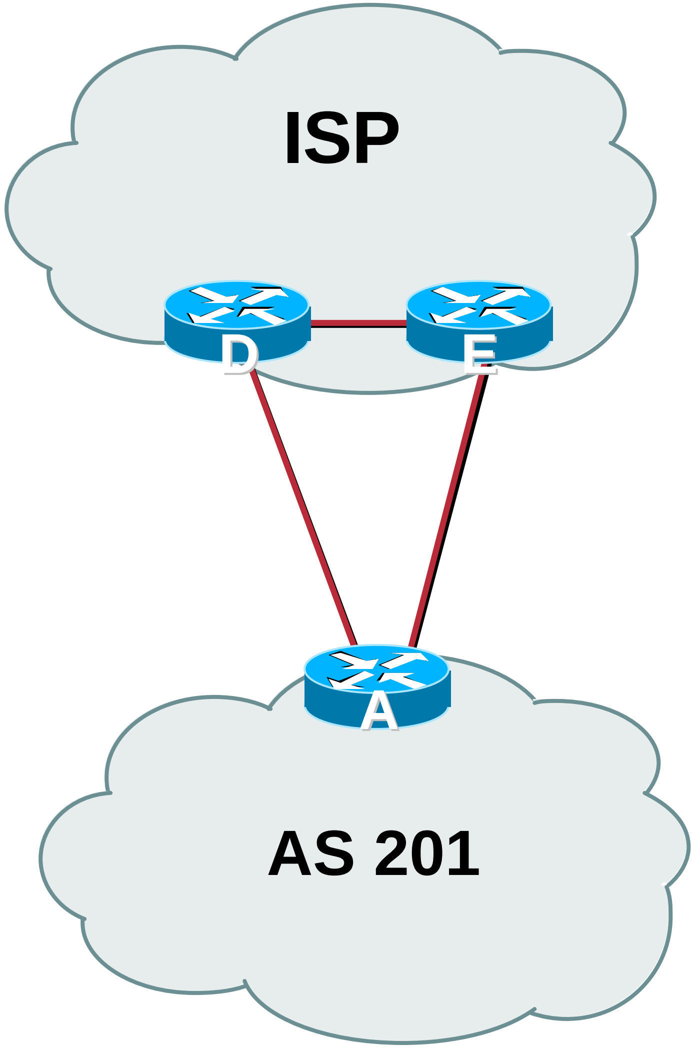
Multiple Sessions to an ISP

- **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

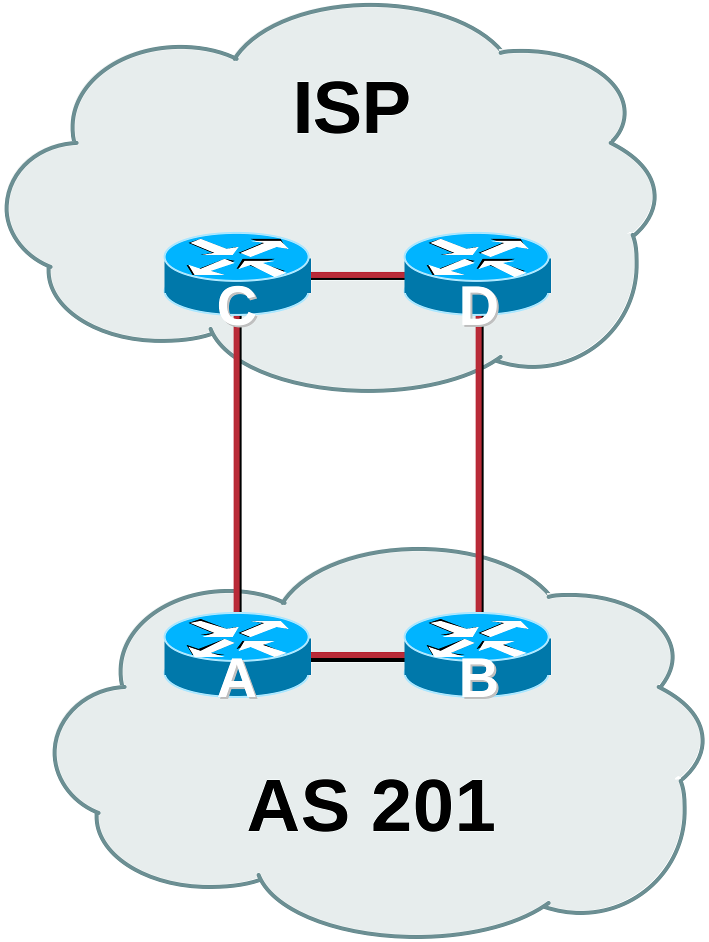
- Simplest scheme is to use defaults
- Learn/advertise prefixes for better control
- Planning and some work required to achieve loadsharing

    Point default towards one ISP

    Learn selected prefixes from second ISP

    Modify the number of prefixes learnt to achieve acceptable load sharing

- No magic solution



Preparing the network

Before we begin…

Preparing the Network
Initial Assumptions

- **The network is not running any BGP at the moment**

  single statically routed connection to upstream ISP

- **The network is not running any IGP at all**

  Static default and routes through the network to do "routing"

- **If multihoming to two different ISPs, AS number has been applied for and received**

Preparing the Network
First Step: IGP

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

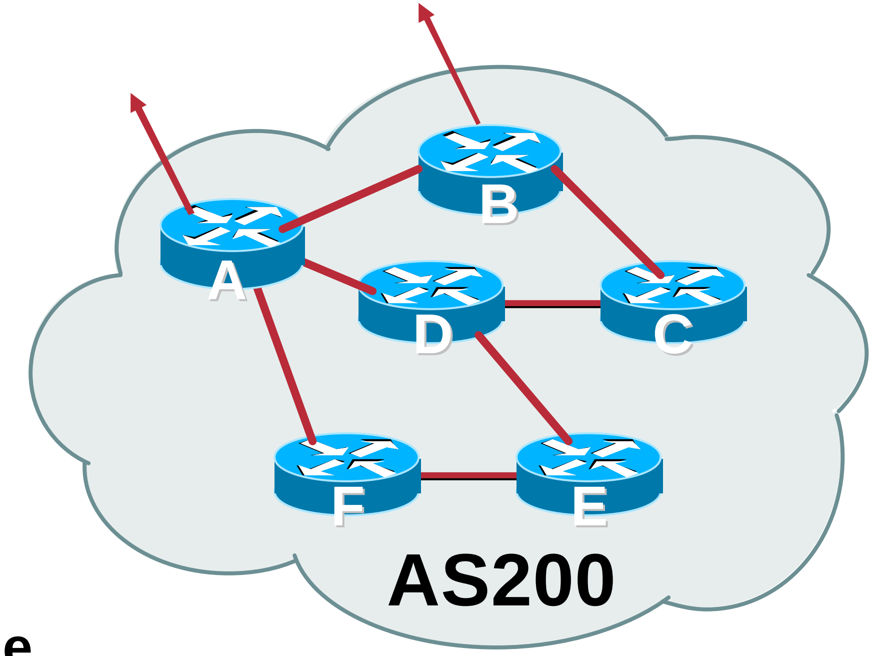
Preparing the Network
Second Step: iBGP

- **Second step is to configure the local network to use iBGP**
- **iBGP can run on**

  all routers, or

  a subset of routers, or

  just on the upstream edge
- ***iBGP must run on all routers which are in the transit path between external connections***



Preparing the Network
Second Step: iBGP (Transit Path)

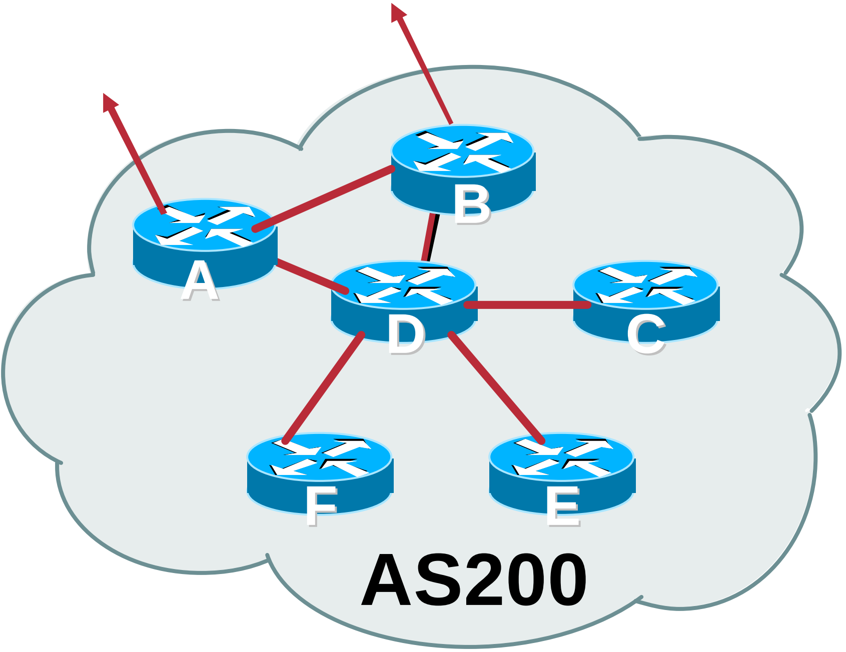
- ***iBGP must run on all routers which are in the transit path between external connections***

- **Routers C, E and F are not in the transit path**

  Static routes or IGP will suffice

- **Router D is in the transit path**

  Will need to be in iBGP mesh, otherwise routing loops will result



Preparing the Network Layers

- Typical SP networks have three layers:

  Core – the backbone, usually the transit path

  Distribution – the middle, PoP aggregation layer

  Aggregation – the edge, the devices connecting customers

Preparing the Network Aggregation Layer

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

Preparing the Network Distribution Layer

- **Usually runs iBGP**

  Partial or full routing (as with aggregation layer)

- **But does not have to run iBGP**

  IGP is then used to carry customer prefixes (does not scale)

  IGP is used to determine nearest exit

- **Networks which plan to grow large should deploy iBGP from day one**

  Migration at a later date is extra work

  No extra overhead in deploying iBGP, indeed IGP benefits

Preparing the Network
Core Layer

- Core of network is usually the transit path
- iBGP necessary between core devices

  Full routes or partial routes:

    Transit ISPs carry full routes in core

    Edge ISPs carry partial routes only

- Core layer includes AS border routers

Preparing the Network
iBGP Implementation

- Decide on:

  Best iBGP policy (full vs partial route mix)

  iBGP scaling technique (communities, route-reflectors, peer-groups)

- Then deploy iBGP:

  Step 1: Introduce iBGP (making sure that iBGP distance is greater than IGP distance)

  Step 2: Install customer prefixes into iBGP

  Step 3: Make iBGP distance less than IGP

  Check! Does the network still work?

  Step 4: Withdraw customer prefixes from the IGP

  Step 5: Deployment of eBGP follows

Preparing the Network Configuration – Before BGP

```
interface serial 0/0
 ip address 221.10.0.1 255.255.255.252
!
interface serial 0/1
 ip address 221.10.0.5 255.255.255.252
!
router ospf 100
 redistribute connected subnets       ! Point-to-point link
 redistribute static subnets          ! Customer networks
!
ip route 221.10.24.0 255.255.252.0 serial 0/0
ip route 221.10.28.0 255.255.254.0 serial 0/1
```

Preparing the Network Configuration – Steps 1 & 2

```
interface serial 0/0
 ip address 221.10.0.1 255.255.255.252
!
interface serial 0/1
 ip address 221.10.0.5 255.255.255.252
!
router ospf 100
 redistribute connected subnets           ! point-to-point links
 redistribute static subnets              ! customer nets into OSPF
!
router bgp 100
 neighbor 221.10.1.2 remote-as 100
 neighbor 221.10.1.2 description iBGP with Router2
 ...
 network 221.10.24.0 mask 255.255.252.0
 network 221.10.28.0 mask 255.255.254.0
 distance bgp 200 200 200
!
ip route 221.10.24.0 255.255.252.0 serial 0/0
ip route 221.10.28.0 255.255.254.0 serial 0/1
```

Preparing the Network Configuration – Steps 3 & 4

```
interface serial 0/0
 ip address 221.10.0.1 255.255.255.252
!
interface serial 0/1
 ip address 221.10.0.5 255.255.255.252
!
router ospf 100
 redistribute connected subnets           ! point-to-point links
!
router bgp 100
 neighbor 221.10.1.2 remote-as 100
 neighbor 221.10.1.2 description iBGP with Router2
 ...
 network 221.10.24.0 mask 255.255.252.0
 network 221.10.28.0 mask 255.255.254.0
 distance bgp 200 200 200
!
ip route 221.10.24.0 255.255.252.0 serial 0/0
ip route 221.10.28.0 255.255.254.0 serial 0/1
```

Preparing the Network
Configuration Summary

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

Two links to the same ISP

Basic – No Redundancy

Two links to the same ISP

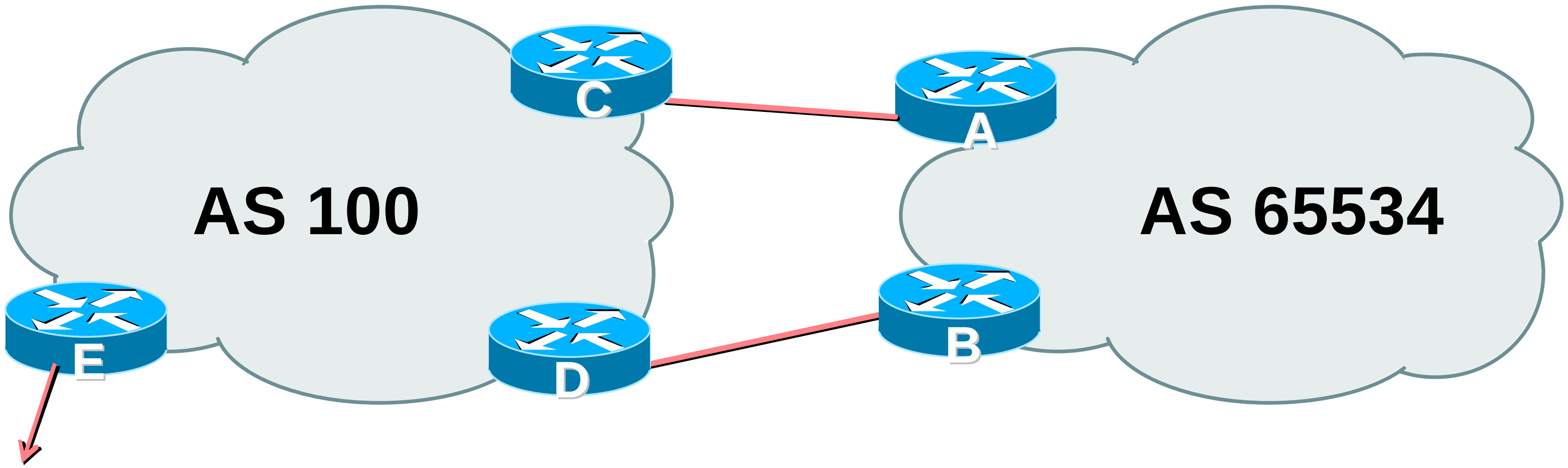
- **Can use BGP for this to aid loadsharing**

  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



- AS100 proxy aggregates for AS 65534

Two links to the same ISP

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

  basic inbound loadsharing

- Example has no practical use, but demonstrates the principles

Two links to the same ISP

- Router A Configuration

```
router bgp 65534
 network 221.10.0.0 mask 255.255.240.0
 network 221.10.16.0 mask 255.255.240.0
 neighbor 222.222.10.2 remote-as 100
 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 route 221.10.0.0 255.255.240.0 null0
ip route 221.10.16.0 255.255.240.0 null0
```

Two links to the same ISP

- **Router B Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.240.0
 network 221.10.16.0 mask 255.255.240.0
 neighbor 222.222.10.6 remote-as 100
 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 route 221.10.0.0 255.255.240.0 null0
ip route 221.10.16.0 255.255.240.0 null0
```

Two links to the same ISP

- Router C Configuration

```
router bgp 100
 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 221.10.0.0/20
ip prefix-list default permit 0.0.0.0/0
```

Two links to the same ISP

- Router D Configuration

```
router bgp 100
 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 221.10.16.0/20
ip prefix-list default permit 0.0.0.0/0
```

Two links to the same ISP

- Router E is AS100 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

- Router E Configuration

```
router bgp 100
 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 route 221.10.0.0 255.255.224.0 null0
!
ip as-path access-list 1 deny ^65534$
ip as-path access-list 1 permit ^$
```

- Private AS still visible inside AS100

Two links to the same ISP

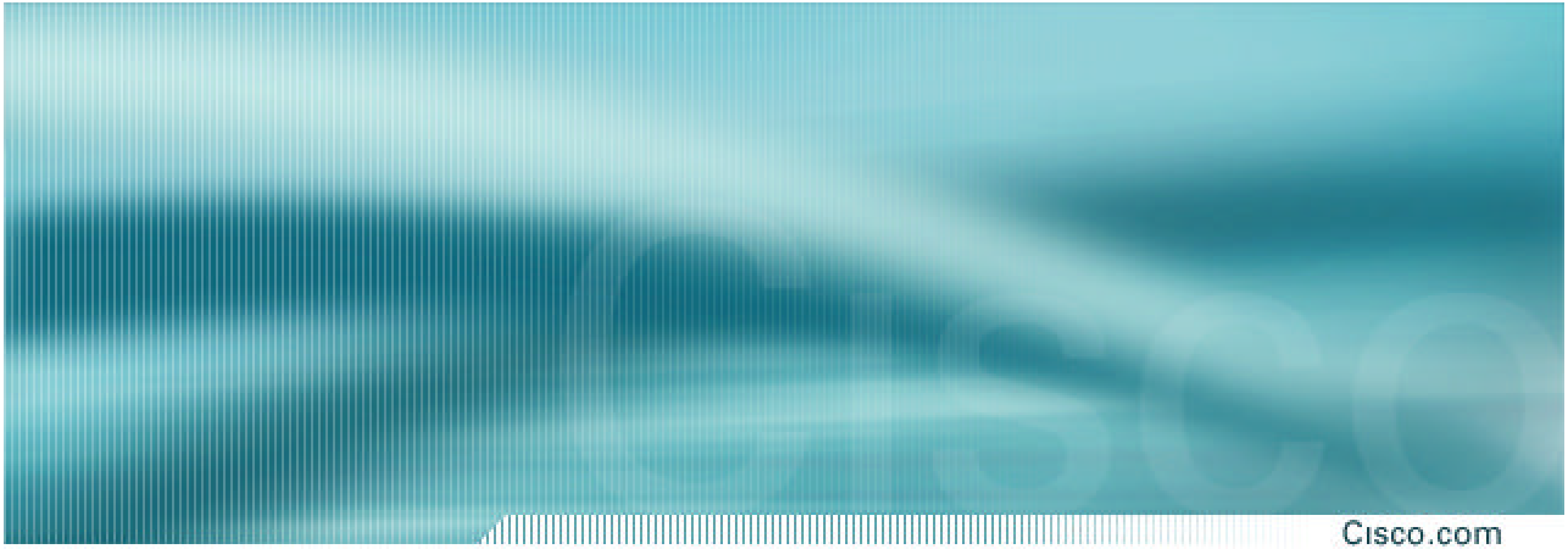
- **Big Problem**:

  no backup in case of link failure

- /19 address block not announced
- AS Path filtering “awkward”

  easier to use bgp command

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



Two links to the same ISP

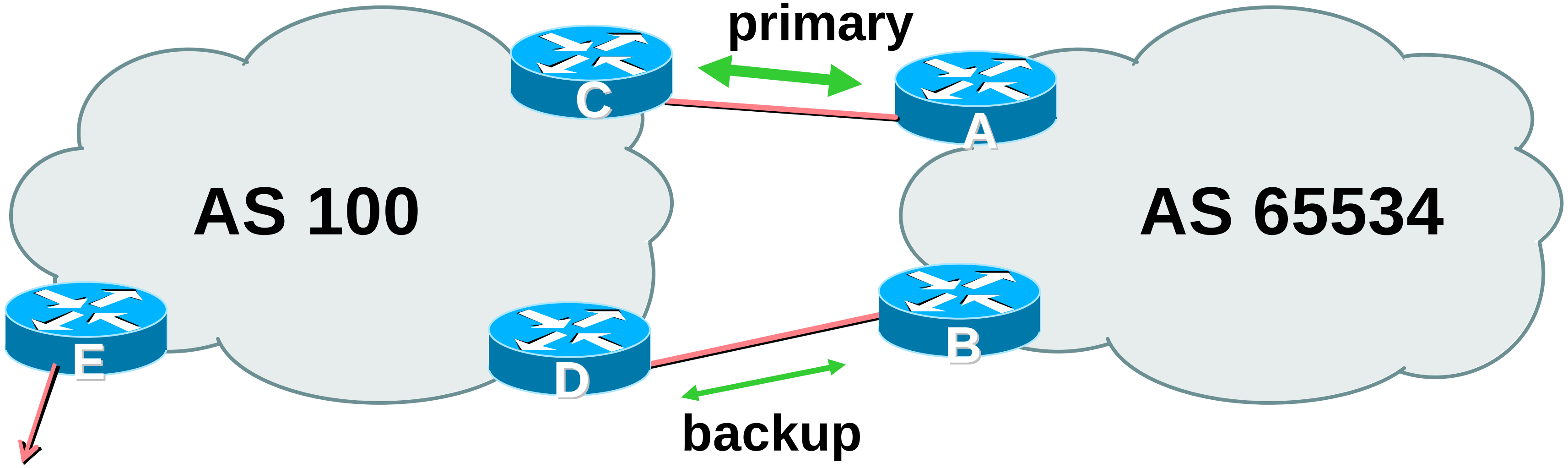
One link primary, the other link backup only

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

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

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

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



- **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:

  Outbound – announce /19 unaltered

  Inbound – receive default route

  backup link:

  Outbound – announce /19 with increased metric

  Inbound – received default, and reduce local preference

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

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

- **Router A Configuration**

```
router bgp 65534
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.2 remote-as 100
 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 100
 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 100
 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 100
 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 (one as backup only)

- **Router E Configuration**

```
router bgp 100
 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
```

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

Two links to the same ISP

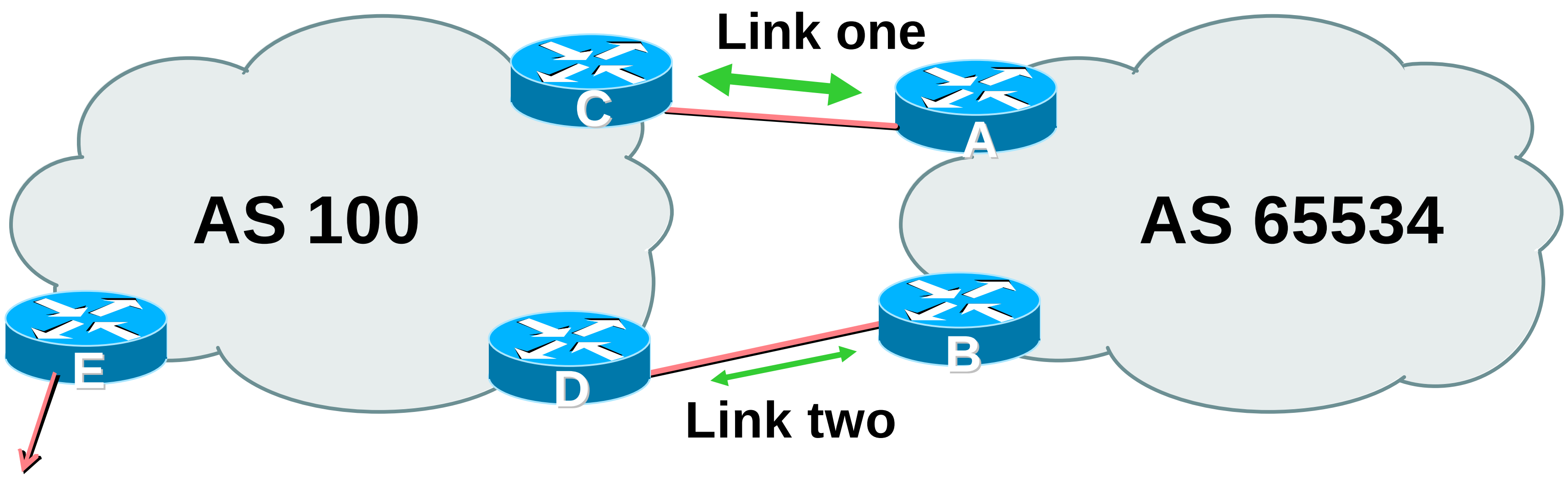
With Redundancy and Loadsharing

Loadsharing to the same ISP

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

  Unequal capacity circuits requires more refinement – see later

Loadsharing to the same ISP (with redundancy)



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

Loadsharing to the same ISP (with redundancy)

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

  basic inbound loadsharing

  assumes equal circuit capacity and even spread of traffic across address block

- Vary the split until "perfect" loadsharing achieved
- Accept the default from upstream

  basic outbound loadsharing by nearest exit

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

Loadsharing to the same ISP (with redundancy)

- 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 100
 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
```

Loadsharing to the same ISP (with redundancy)

- 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 100
 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
(with redundancy)

- **Default route for outbound traffic?**

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

  e.g. on router A:

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

Loadsharing to the same ISP (with redundancy)

- Router C Configuration

```
router bgp 100
 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

Loadsharing to the same ISP (with redundancy)

- Router D Configuration

```
router bgp 100
 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

Loadsharing to the same ISP (with redundancy)

- Router E is AS100 border router

  removes subprefixes in the private AS from external announcements

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

Loadsharing to the same ISP (with redundancy)

- Router E Configuration

```
router bgp 100
 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 AS100

Loadsharing to the same ISP (with redundancy)

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

  remove customer subprefixes from external announcements

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

Two links to the same ISP

Multiple Dualhomed Customers

(RFC2270)

Multiple Dualhomed Customers (RFC2270)

- Unusual for an ISP just to have one dualhomed customer

  Valid/valuable service offering for an ISP with multiple PoPs

  Better for ISP than having customer multihome with another provider!

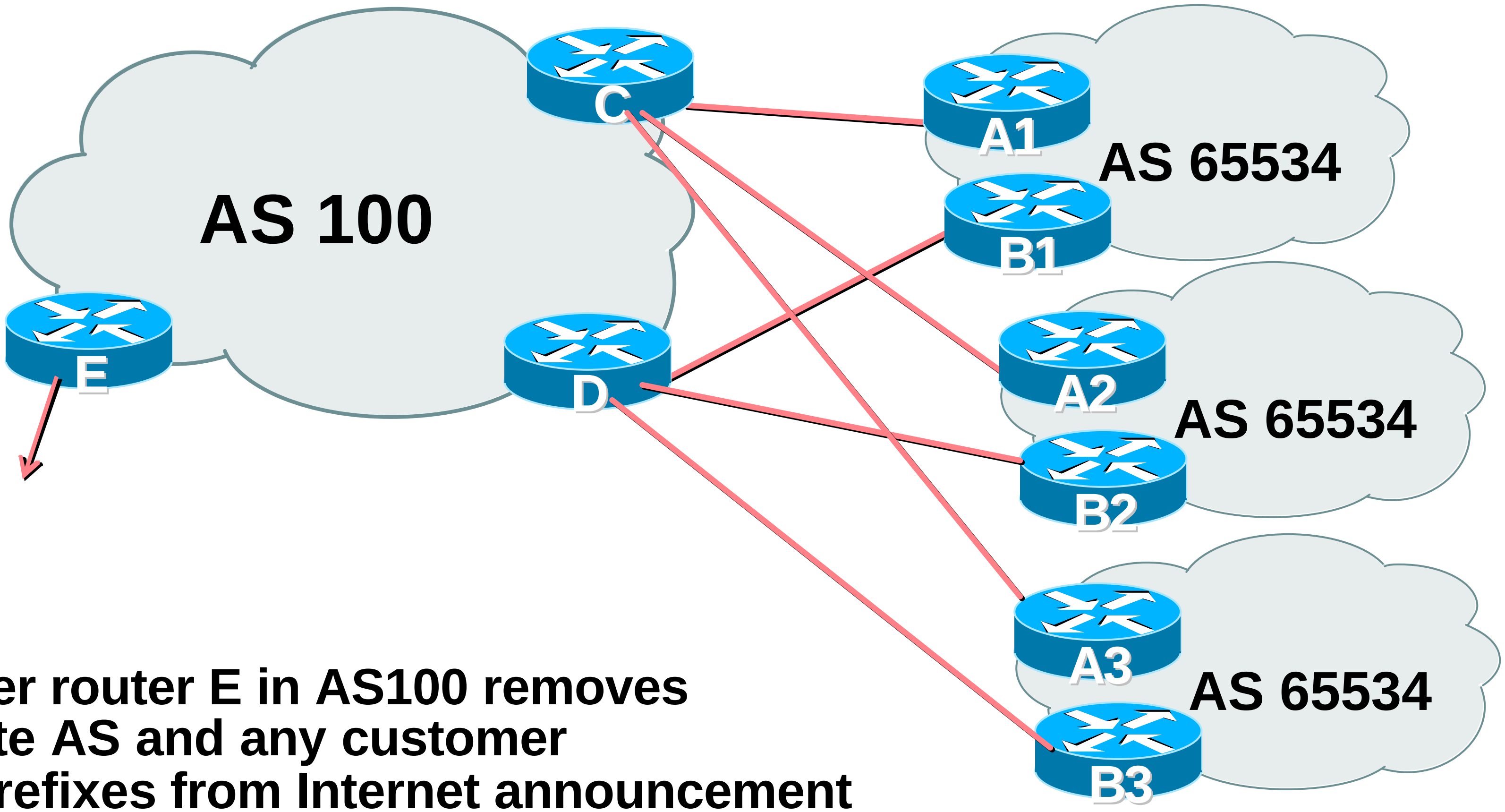
- Look at scaling the configuration

  Þ Simplifying the configuration

  Using templates, peer-groups, etc

  Every customer has the same configuration (basically)

Multiple Dualhomed Customers (RFC2270)



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

Multiple Dualhomed Customers (RFC2270)

- **Customer announcements as per previous example**

- **Use the *same* private AS for each customer**

    documented in RFC2270

    address space is not overlapping

    each customer hears default only

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

Multiple Dualhomed Customers (RFC2270)

- **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 100
 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
```

Multiple Dualhomed Customers (RFC2270)

- 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 100
 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 (RFC2270)

- **Router C Configuration**

```
router bgp 100
 neighbor bgp-customers peer-group
 neighbor bgp-customers remote-as 65534
 neighbor bgp-customers default-originate
 neighbor bgp-customers prefix-list default out
 neighbor 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 (RFC2270)

```
 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 (RFC2270)

- **Router D Configuration**

```
router bgp 100
 neighbor bgp-customers peer-group
 neighbor bgp-customers remote-as 65534
 neighbor bgp-customers default-originate
 neighbor bgp-customers prefix-list default out
 neighbor 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 (RFC2270)

```
 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 (RFC2270)

- **Router E Configuration**

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

```
router bgp 100
 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 AS100**

Multiple Dualhomed Customers (RFC2270)

- **If customers’ prefixes come from ISP’s address block**

  do NOT announce them to the Internet

  announce ISP aggregate only

- **Router E configuration:**

```
router bgp 100
 neighbor 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 a Public AS

    Or use private AS if agreed with the other ISP

    But some people don’t like the “inconsistent-AS” which results from use of a private-AS

- Address space comes from

    both upstreams or

    Regional Internet Registry

- Configuration concepts very similar

Inconsistent-AS?

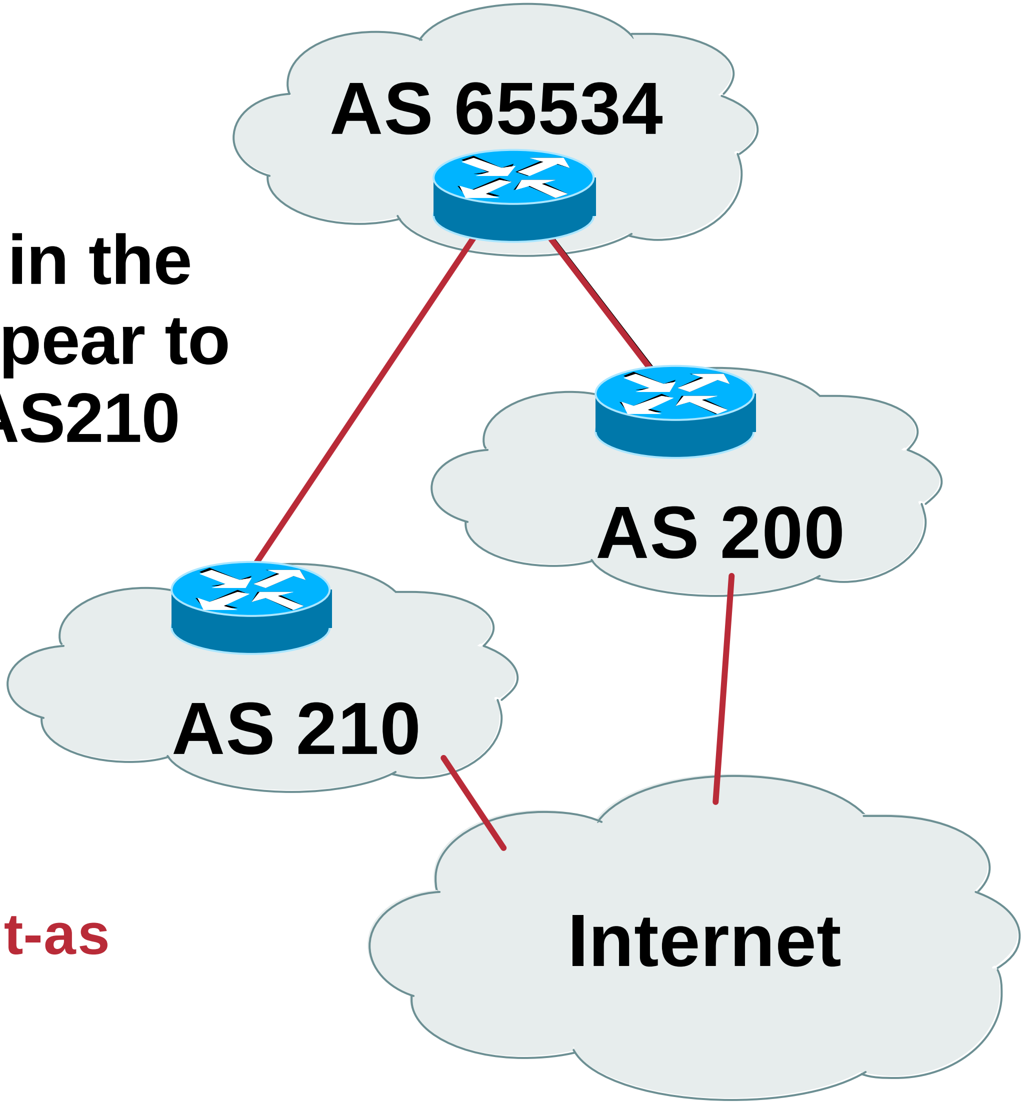
- **Viewing the prefixes originated by AS65534 in the Internet shows they appear to be originated by both AS210 and AS200**

  This is NOT bad

  Nor is it illegal

- **IOS command is**

  show ip bgp inconsistent-as



Two links to different ISPs

Basic – No Redundancy

Two links to different ISPs (no redundancy)

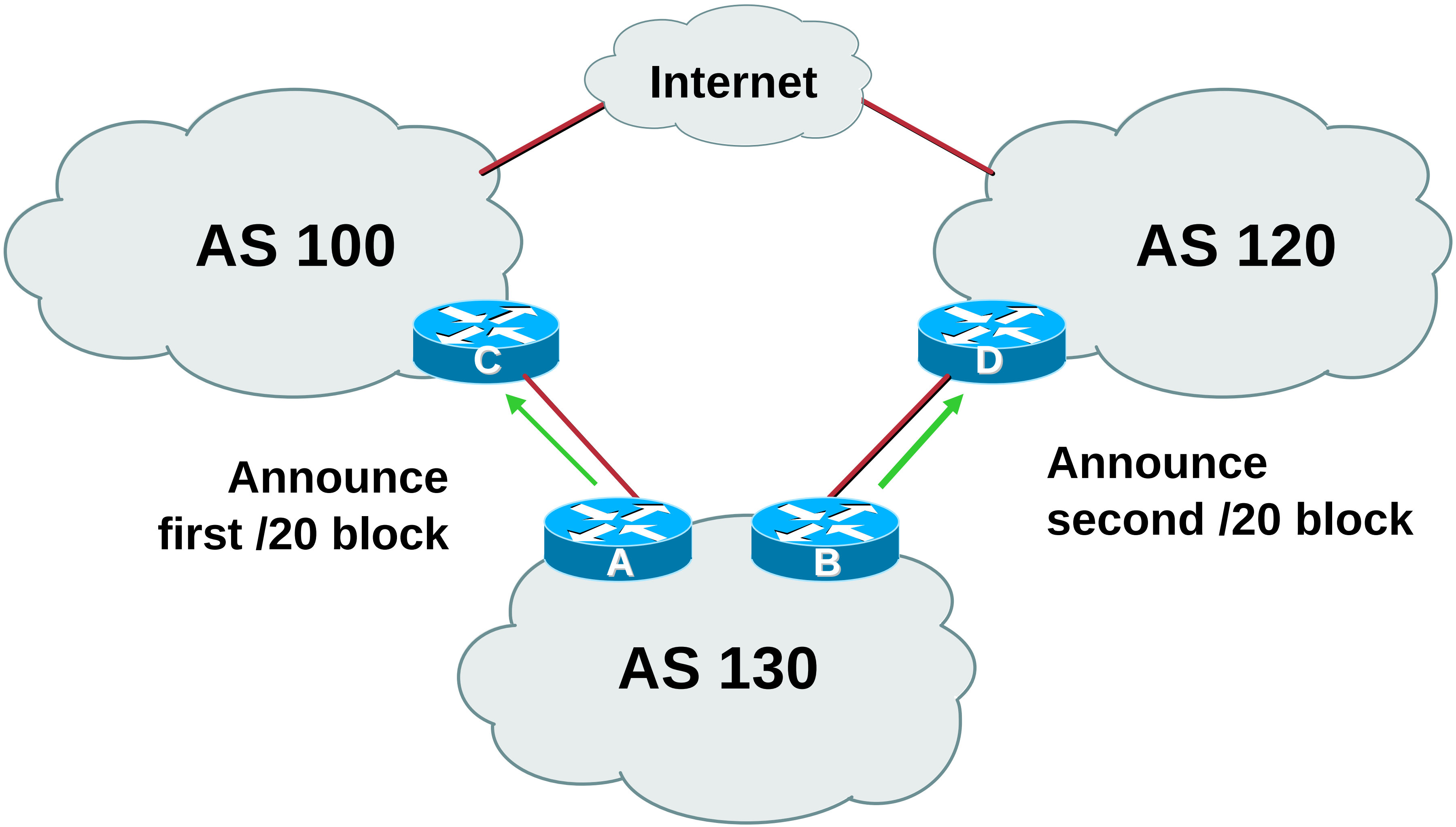
- **Example for PI space**

  ISP network, or large enterprise site

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

  basic inbound loadsharing

Two links to different ISPs (no redundancy)



Two links to different ISPs (no redundancy)

- Router A Configuration

```
router bgp 130
 network 221.10.0.0 mask 255.255.240.0
 neighbor 222.222.10.1 remote-as 100
 neighbor 222.222.10.1 prefix-list routerC out
 neighbor 222.222.10.1 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerC permit 221.10.0.0/20
```

Two links to different ISPs (no redundancy)

- Router B Configuration

```
router bgp 130
 network 221.10.16.0 mask 255.255.240.0
 neighbor 220.1.5.1 remote-as 120
 neighbor 220.1.5.1 prefix-list routerD out
 neighbor 220.1.5.1 prefix-list default in
!
ip prefix-list default permit 0.0.0.0/0
ip prefix-list routerD permit 221.10.16.0/20
```

Two links to different ISPs (no redundancy)

- **Router C Configuration**

```
router bgp 100
 neighbor 221.10.1.1 remote-as 130
 neighbor 221.10.1.1 default-originate
 neighbor 221.10.1.1 prefix-list AS130cust in
 neighbor 221.10.1.1 prefix-list default-out out
!
```

- **Router C only announces default to AS 130**
- **Router C only accepts AS130’s prefix block**

Two links to different ISPs (no redundancy)

- Router D Configuration

```
router bgp 120
 neighbor 220.1.5.1 remote-as 130
 neighbor 220.1.5.1 default-originate
 neighbor 220.1.5.1 prefix-list AS130cust in
 neighbor 220.1.5.1 prefix-list default-out out
!
```

- Router D only announces default to AS 130
- Router D only accepts AS130’s prefix block

Two links to different ISPs (no redundancy)

- **Big Problem**:

  no backup in case of link failure

- /19 address block not announced

Two links to different ISPs

One link primary, the other link backup only

Two links to different ISPs (one as backup only)

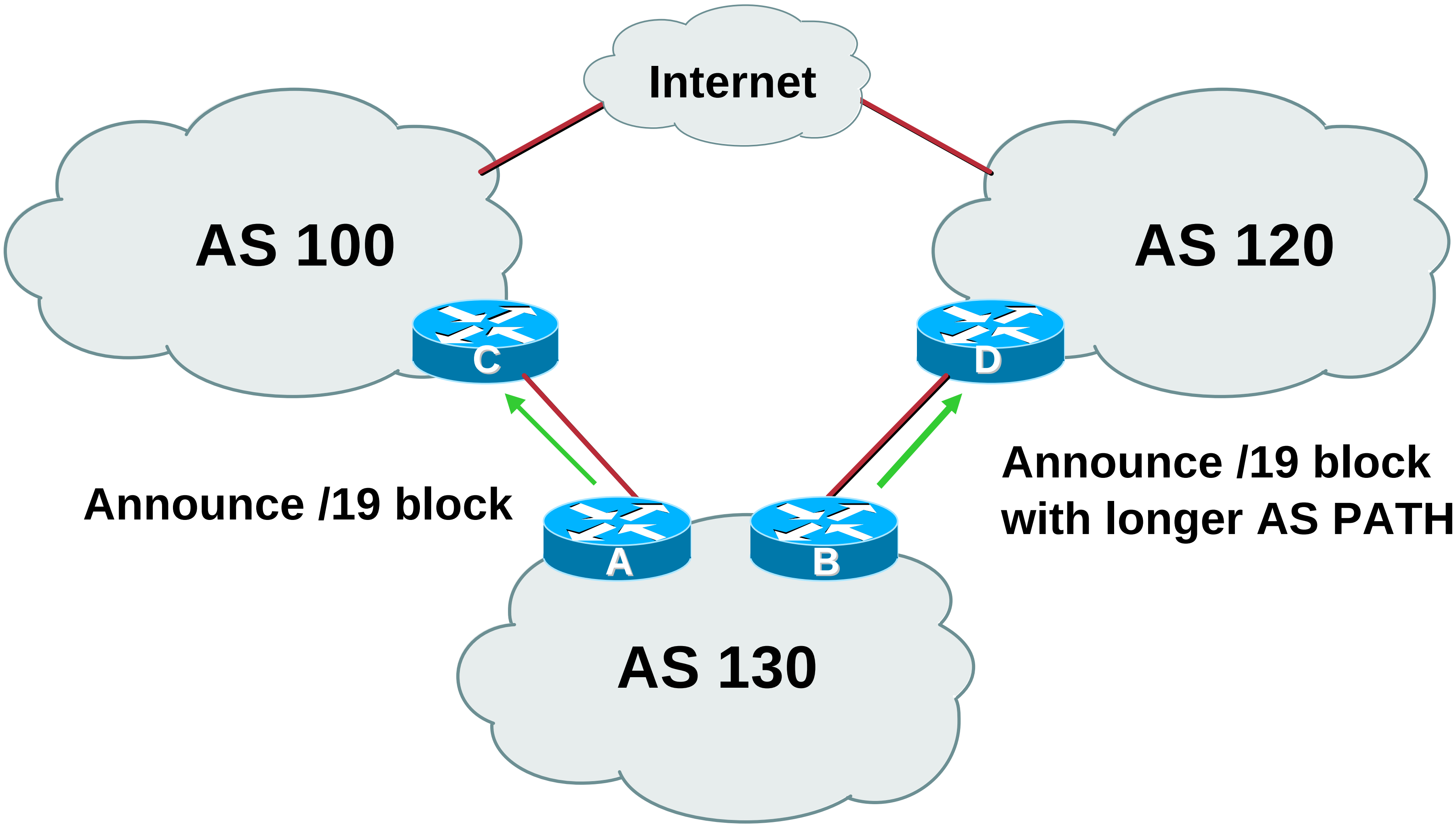
- Announce /19 aggregate on each link

  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)



Announce /19 block

Announce /19 block with longer AS PATH

Two links to different ISPs (one as backup only)

- **Router A Configuration**

```
router bgp 130
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 100
 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
```

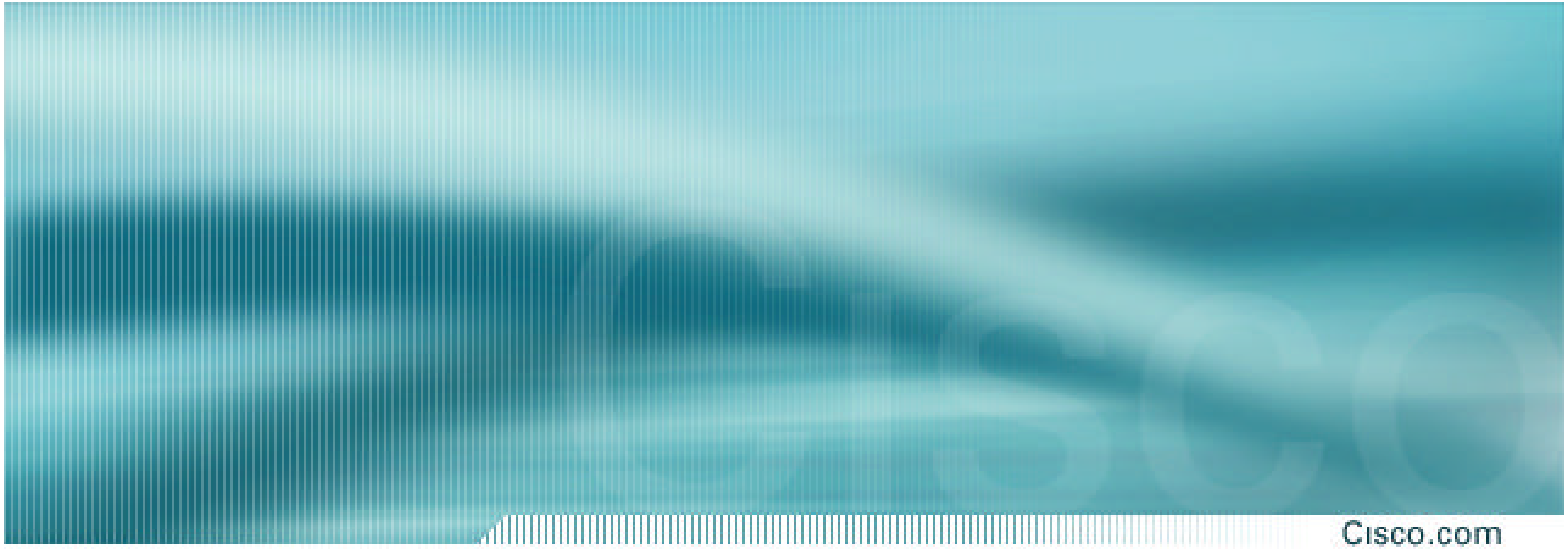
Two links to different ISPs (one as backup only)

- **Router B Configuration**

```
router bgp 130
 network 221.10.0.0 mask 255.255.224.0
 neighbor 220.1.5.1 remote-as 120
 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
!
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 130 130 130
!
route-map routerD-in permit 10
 set local-preference 80
```

Two links to different ISPs
(one as backup only)

- Not a common situation as most sites tend to prefer using whatever capacity they have
- But it shows the basic concepts of using local-prefs and AS-path prepends for engineering traffic in the chosen direction



Two links to different ISPs

With Redundancy

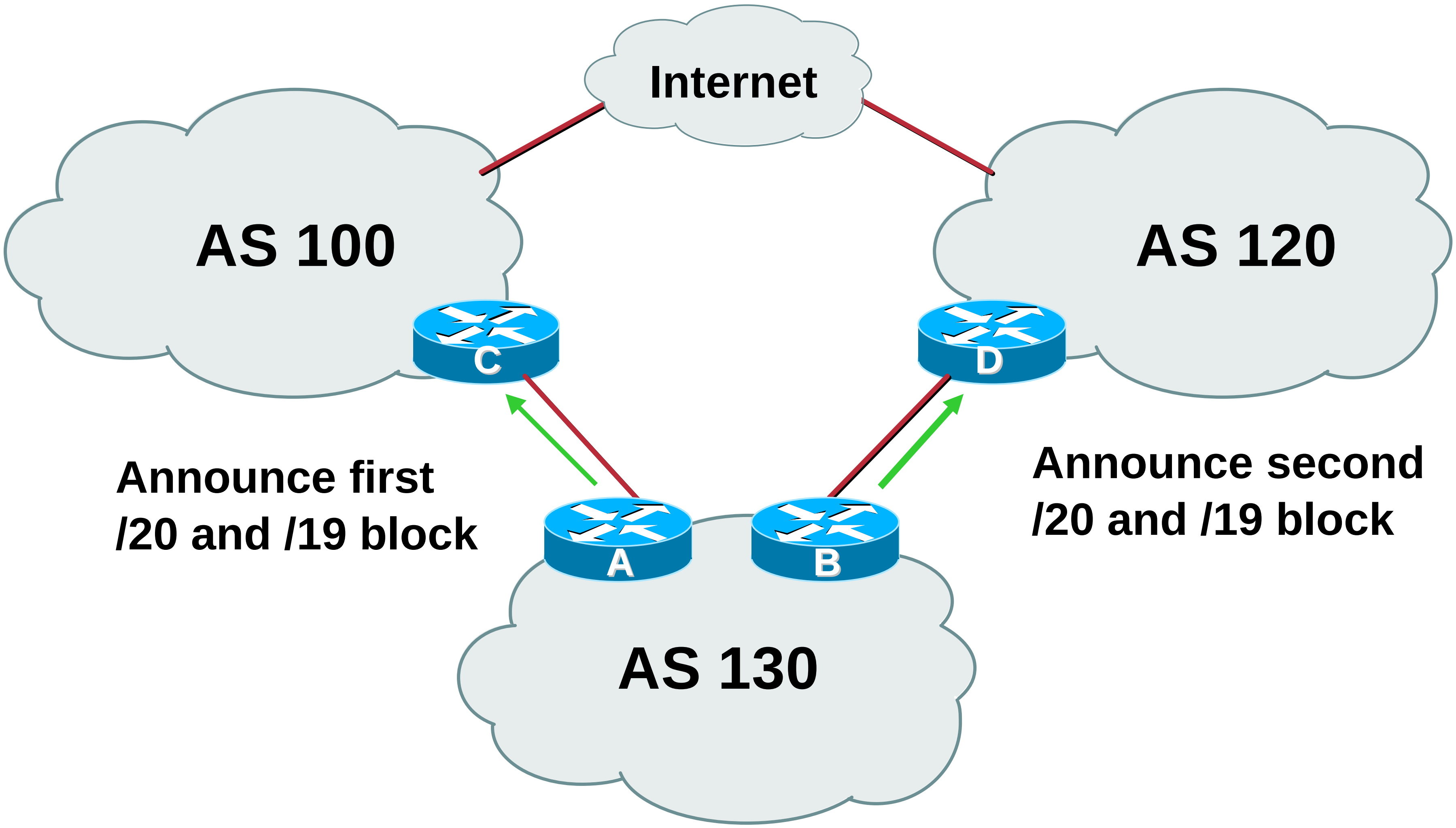
Two links to different ISPs (with redundancy)

- **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)



Announce first /20 and /19 block

Announce second /20 and /19 block

Two links to different ISPs (with redundancy)

- Router A Configuration

```
router bgp 130
 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 100
 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 130
 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 120
 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 loadsharing)

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

  Start with a simple concept

  And build on it…!

Two links to different ISPs

More Controlled Loadsharing

Loadsharing with different ISPs

- **Announce /19 aggregate on each link**

  On first link, announce /19 as normal

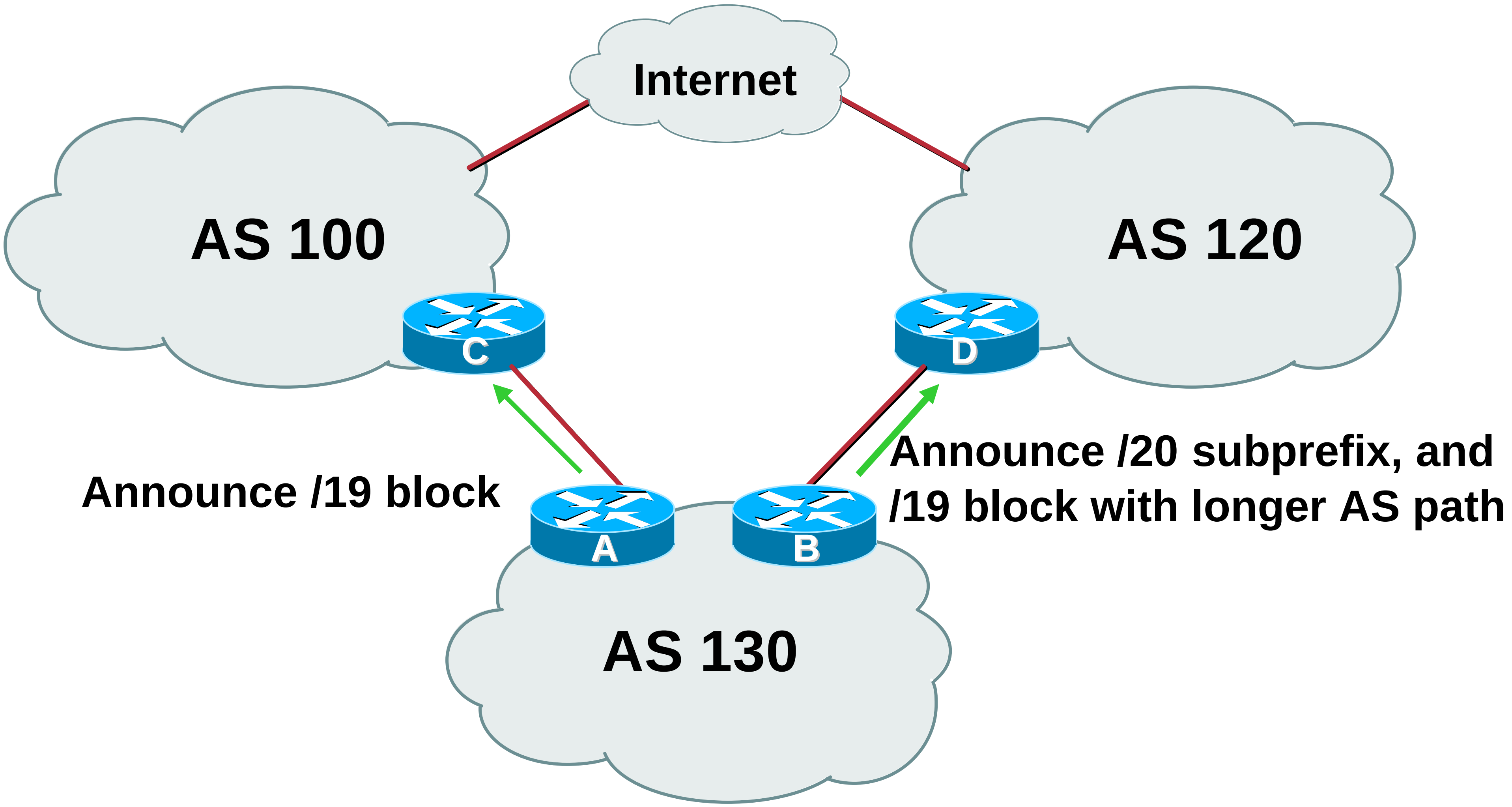
  On second link, announce /19 with longer AS PATH, and announce one /20 subprefix

  - **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



Announce /19 block

Announce /20 subprefix, and /19 block with longer AS path

Loadsharing with different ISPs

- Router A Configuration

```
router bgp 130
 network 221.10.0.0 mask 255.255.224.0
 neighbor 222.222.10.1 remote-as 100
 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 130
 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 120
 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
!
route-map routerD permit 10
 match ip address prefix-list aggregate
 set as-path prepend 130 130
route-map routerD permit 20
!
ip prefix-list subblocks permit 221.10.0.0/19 le 20
ip prefix-list aggregate permit 221.10.0.0/19
```

Loadsharing with different ISPs

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

BGP and the Internet

Multihoming