Multihoming: Practical Deployment

ISP Workshops

Last updated 5th May 2022

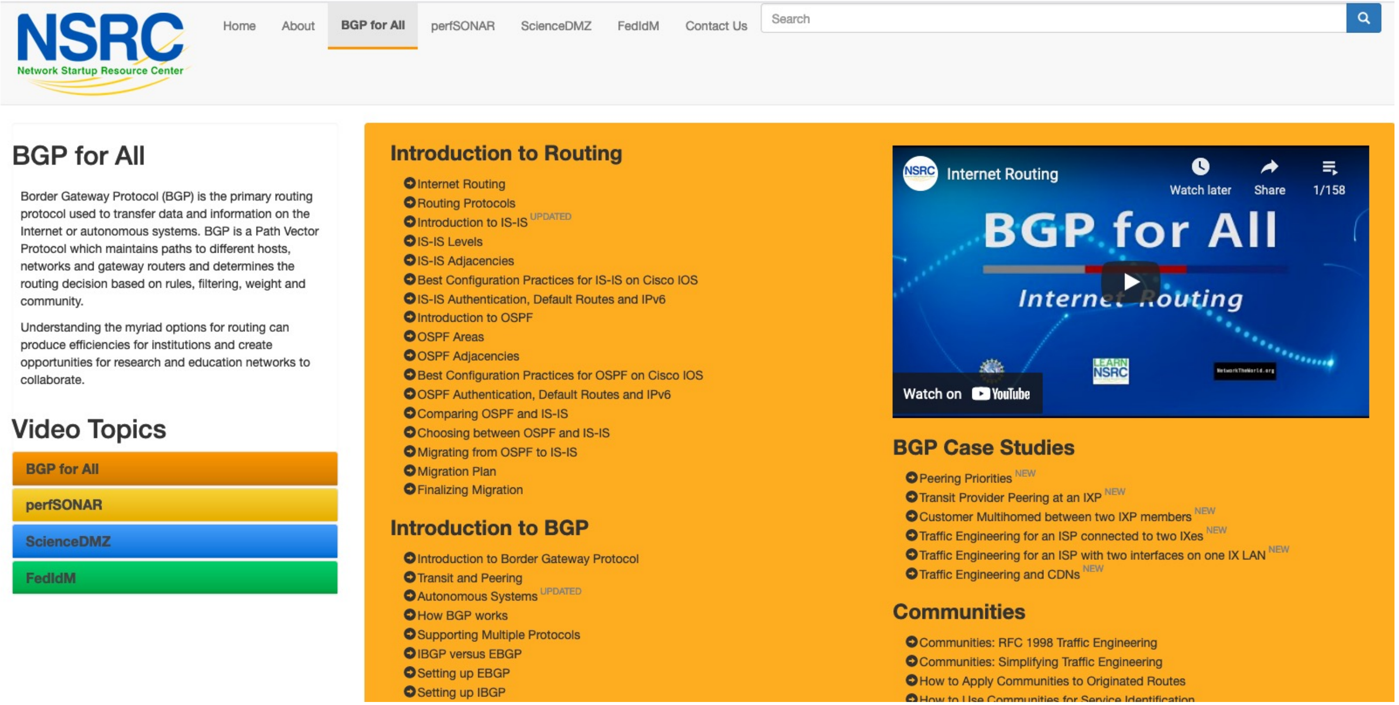
Acknowledgements

- This material was developed by Philip Smith with the support of the Network Startup Resource Center
- Use of these materials is encouraged as long as the source is fully acknowledged and this notice remains in place
- Bug fixes and improvements are welcomed
  - Please email *workshop (at) bgp4all.com*

Philip Smith

BGP Videos

- NSRC has produced a library of BGP presentations, recorded on video, for the whole community to use
  - https://learn.nsrc.org/bgp



BGP for All

Border Gateway Protocol (BGP) is the primary routing protocol used to transfer data and information on the Internet or autonomous systems. BGP is a Path Vector Protocol which maintains paths to different hosts, networks and gateway routers and determines the routing decision based on rules, filtering, weight and community.

Understanding the myriad options for routing can produce efficiencies for institutions and create opportunities for research and education networks to collaborate.

Video Topics

- BGP for All
- perfSONAR
- ScienceDMZ
- FedIdM

Introduction to Routing

- Internet Routing
- Routing Protocols
- Introduction to IS-IS UPDATED
- IS-IS Levels
- IS-IS Adjacencies
- Best Configuration Practices for IS-IS on Cisco IOS
- IS-IS Authentication, Default Routes and IPv6
- Introduction to OSPF
- OSPF Areas
- OSPF Adjacencies
- Best Configuration Practices for OSPF on Cisco IOS
- OSPF Authentication, Default Routes and IPv6
- Comparing OSPF and IS-IS
- Choosing between OSPF and IS-IS
- Migrating from OSPF to IS-IS
- Migration Plan
- Finalizing Migration

Introduction to BGP

- Introduction to Border Gateway Protocol
- Transit and Peering
- Autonomous Systems UPDATED
- How BGP works
- Supporting Multiple Protocols
- IBGP versus EBGP
- Setting up EBGP
- Setting up IBGP

BGP Case Studies

- Peering Priorities NEW
- Transit Provider Peering at an IXP NEW
- Customer Multihomed between two IXP members NEW
- Traffic Engineering for an ISP connected to two IXes NEW
- Traffic Engineering for an ISP with two interfaces on one IX LAN NEW
- Traffic Engineering and CDNs NEW

Communities

- Communities: RFC 1998 Traffic Engineering
- Communities: Simplifying Traffic Engineering
- How to Apply Communities to Originated Routes
- How to Use Communities for Service Identification

Agenda

- Background and Requirements
- The next steps
- 1st link primary, 2nd link backup
- Load share between both links
- End-site network configuration

Background

- Previous Multihoming presentations cover the technology and techniques for:
  - Setting up Multihoming
  - Carrying out traffic engineering to improve load balancing
- Real-world – where to begin?
  - What resources are needed?
  - What equipment is needed?
  - What is required of upstream providers?
- What does a multihoming end-site need to do?

Resource Requirements: IP Address Space

- Entity requires its own independent IPv4 and IPv6 address space
  - Operators do not allow their delegated address space to be routed via other providers (contractual)
  - Operators do not even have IPv4 address space to give to customers now!
- IPv4 address space is very limited
  - Depending on Regional Internet Registry policies, this may be as little as a /24 of IPv4 for ***new*** members
    - With luck, a /23 (two /24s) might be available
- IPv6 is plentiful
  - Some RIR multihoming policies allow for a /48 for a multihoming organisation
    - But traffic engineering isn't possible
  - Better to acquire two /48s or a whole /32 (for organisations operating a network)

Resource Requirements: IP Address Space

- Note well:
  - Traffic engineering is not easy with IPv4 /24 and IPv6 /48
    - These prefixes cannot be subdivided in to smaller pieces and still have global connectivity
    - Advice: obtain two /24s and two /48s, minimum
  - If the RIR can provide an IPv4 /23, it may not be made up of contiguous /24s!
    - Technically this isn't a problem
- Obtaining IPv4 address space requires membership of the RIR
  - (IPv6 and an AS Number come as part of the package for all RIR members)

Resource Requirements: AS Number

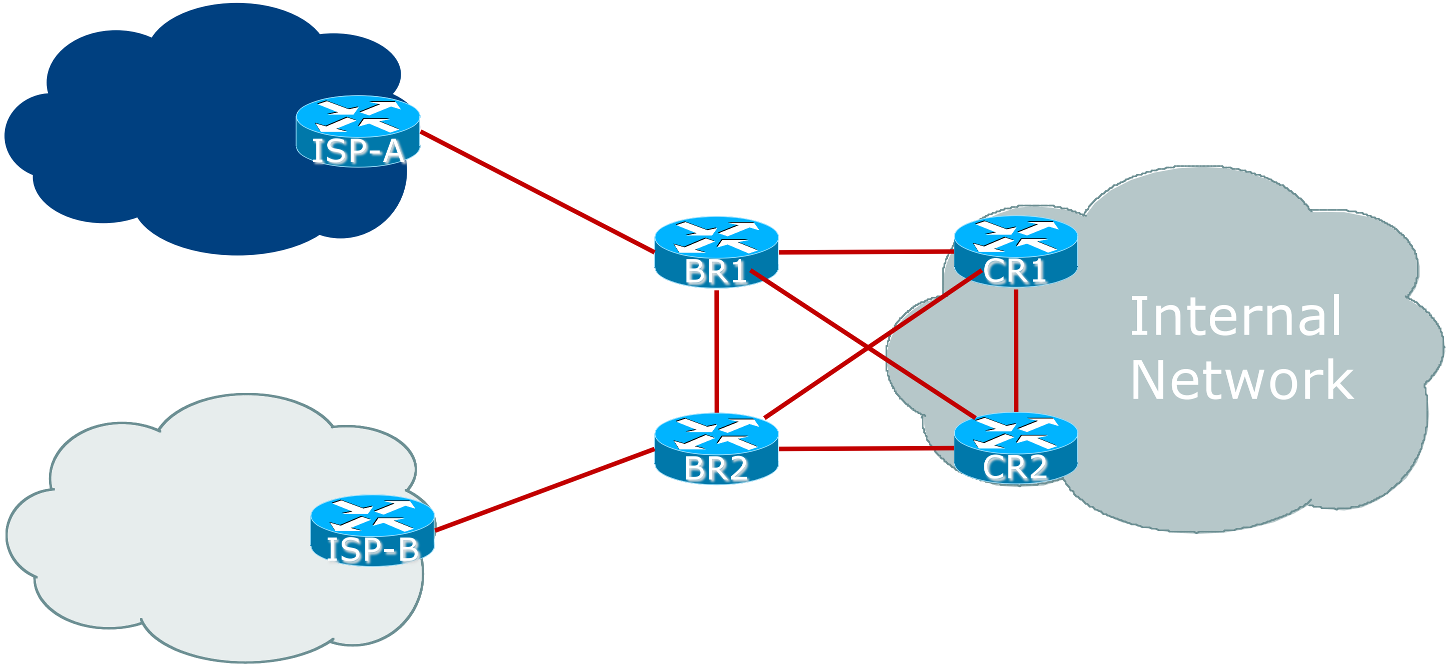
- A public AS number is required by BGP to implement multihoming
  - Private AS numbers theoretically can be used
    - Technically and operationally challenging
    - No advantage since the public AS Number pool is vast
- AS Numbers are available from the Regional Internet Registry as part of the entity’s membership
  - To apply, simply list the two ASes which will be the multihoming partners (i.e. upstream providers)
  - Can also be obtained by one of the upstream providers on behalf of the entity multihoming
    - But if entity already joining RIR to get IP address space, best practice to obtain AS number directly from the RIR

Equipment requirements

- Two border routers are required
  - Multihoming can be done with just one, but then there won’t be any router redundancy
    - Hardware or software failure means outage until repair or replacement
- Routers need:
  - To be able to support BGP
  - To be able to handle the capacity of the link
  - One external interface (for connection to upstream)
  - Two or more internal interfaces
    - Common today for border routers to have four ethernet ports (one used external facing, the other three internal facing)

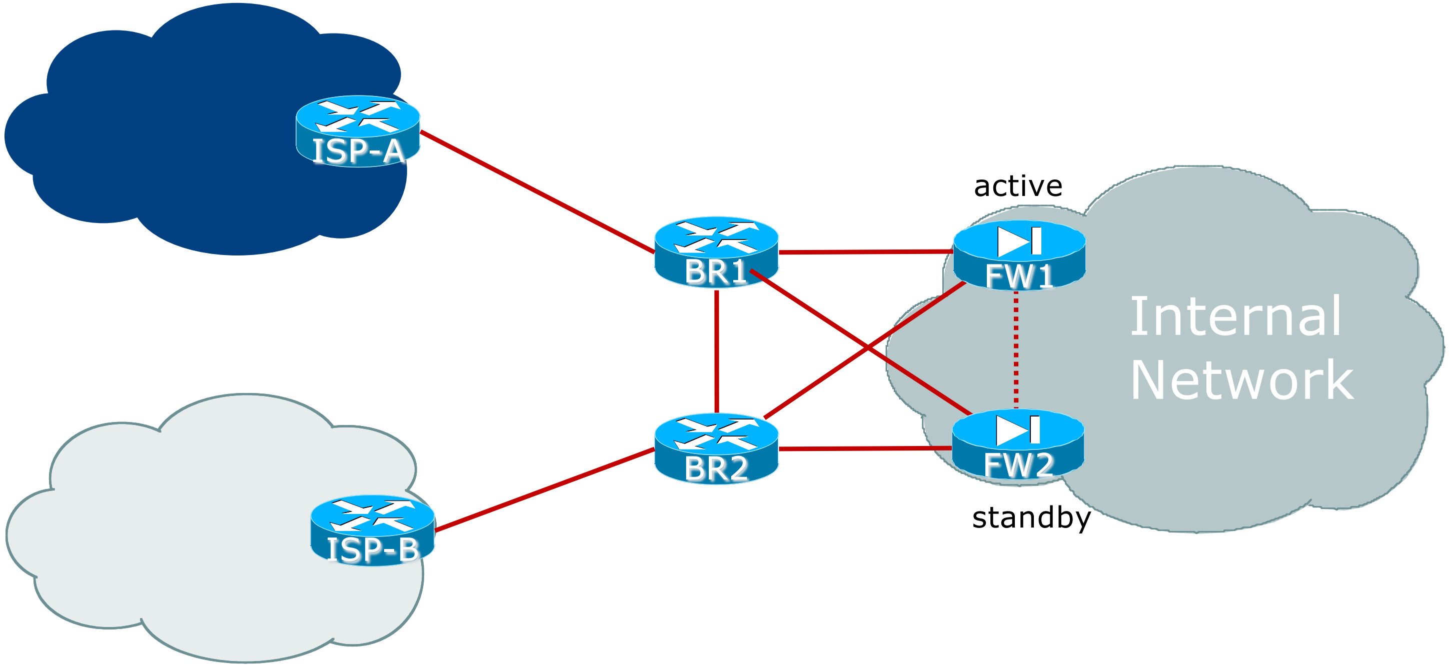
Equipment Requirements

- Scenario One:
  - Border routers to upstream, Core routers host internal network



Equipment Requirements

- Scenario Two:
  - Border routers to upstream, Firewalls host internal network



Equipment Requirements

- Note the redundancy built into the design
  - One border router fails – traffic continues via the other
  - One core router/firewall fails – traffic continues via the other
  - One upstream provider fails – traffic continues via the other
- Border Router function:
  - EBGP with upstream
    - Originating IP address blocks
  - IBGP and OSPF/IS-IS with core devices
  - Traffic engineering via BGP
  - Initial protection of the core network with packet filters
  - Details later!

Selecting an Upstream

- Must have different transit arrangements from each other
  - Otherwise entity only gains localised redundancy
    - If the upstream providers' shared transit fails, the advantages of multihoming are lost
    - Still provides protection against local problems (link outages, upstream outages)
  - Make sure that upstream providers specify who their transits are
    - And check on RouteViews or Hurricane Electric's BGP page:
      - https://routeviews.org or https://bgp.he.net
    - Being at the same Internet Exchange Point is fine!
    - Having private peering with each other is fine too!

Selecting an Upstream

- Make sure the upstream and its transit providers:
  - Serve the region you are interested in
  - Have ready access to the major content providers (direct peering)
    - Or host content caches from the major content providers
- Makes no sense if transit provider of upstream is in a different continent as it will adversely affect your customer and end-user experience
- When multihoming, goals are:
  - Redundancy
  - Minimum latency to critical content
  - Maximum bandwidth to critical content

Upstream Provider Requirements

- Must be willing to support a BGP customer!
  - Many are not (usually citing complexity)
- Must be able to support a BGP customer!
  - References for existing customers help
  - Check on RouteViews or Hurricane Electric’s BGP page for existing BGP customers
  - BGP customer hotline (separate from standard access customer)
- Must be able to update peering policies with their transit providers, peers, and any IXPs they are members of
  - Community policies (published or otherwise)
  - Direct NOC to NOC contact
- Must implement the Mutually Agreed Norms for Routing Security principles (https://manrs.org) in their network

Upstream Provider Requirements

- Other desirable features of an upstream:
  - Support for Route Origin Validation
    - They check for and drop invalid prefixes
  - Provision of DDoS mitigation tools
    - Support for Remotely Triggered Black Hole (RTBH) Filtering
    - Support for customer use of the RTBH BGP community
  - Support for BGP communities for traffic engineering
    - Saves phoning their NOC every time changes are needed
  - Direct access to their NOC
    - BGP customer has more sophisticated needs than a fixed link customer

Agenda

- Background and Requirements
- The next steps
- 1st link primary, 2nd link backup
- Load share between both links
- End-site network configuration

What now?

- Status:
  - Have obtained IPv4 and IPv6 address space from the RIR
  - Have obtained an AS number from the RIR
  - Have procured two suitable border routers
  - Have selected two upstream providers

RPKI: Signing ROAs

- When IPv4 and IPv6 address blocks are delegated, and the AS Number assigned, sign the ROAs
  - ROA stands for Route Origin Authorisation
  - A digital signature stating that a specific AS is authorised to originate a specific address block
  - Document this in your standard operational procedures
    - Don't forget to update the ROA if there are changes in address block size or origin AS
- How to sign ROAs?
  - Available via your RIR portal
  - Usually need to set up two factor authentication first

RPKI: Signing ROAs

- A typical ROA would look like this:

Prefix	10.10.0.0/16
Max-Length	/18
Origin-AS	AS65534

- There can be more than one ROA per address block
  - Allows the operator to originate prefixes from more than one AS
  - Caters for changes in routing policy or prefix origin
  - (Allows your upstream to originate your address block from their AS until you are ready with your BGP)

Creating ROAs – Important Notes

- Always create ROAs for the aggregate and the individual subnets being routed in BGP
- Examples:
  - If creating a ROA for 10.10.0.0/16 **and** “max prefix” length is set to /16
    - There will only be a valid ROA for 10.10.0.0/16
    - If a subnet of 10.10.0.0/16 is originated, it will be state Invalid
  - If creating a ROA for 10.1.32.0/23 **and** “max prefix” length is set to /23
    - There will only be a valid ROA for 10.1.32.0/23
    - If 10.1.32.0/24 or 10.1.33.0/24 is originated, these will be state Invalid
  - If creating a ROA for 10.1.32.0/24 **and** “max prefix” length is set to /24
    - There will only be a valid ROA for 10.1.32.0/24
    - If 10.1.32.0/22 is originated, it will be state NotFound

Internet Routing Registry: Route Object

- A route object documents which AS number is originating the listed route
  - Superseded by a ROA
  - In fact, most RIRs now automatically create a route object in their IRR for each ROA that is signed
- Required by many major transit providers
  - They build their customer and peer filter based on the route-objects listed in the IRR
  - Referring to at least the 5 RIR routing registries and the RADB
  - Some operators run their own instance of the IRR as well
    - May require their customers to place a Route Object there (if not using the 5 RIR or RADB versions of the IRR)

Route Object: Examples

```
route:          100.64.0.0/24
descr:          ENTERPRISE-BLOCK
country:        ZZ
notify:         noc@yy.zz
mnt-by:         MAINT-ZZ-ENTERPRISE
origin:         AS64500
last-modified:  2018-09-18T09:37:40Z
source:         IRR
```

This declares that AS64500 is the origin of 100.64.0.0/24

```
route6:         2001:DB8:F:/48
descr:          ENTERPRISE-V6BLOCK
origin:         AS64500
notify:         noc@yy.zz
mnt-by:         MAINT-ZZ-ENTERPRISE
last-modified:  2010-07-21T03:46:02Z
source:         IRR
```

This declares that AS64500 is the origin of 2001:DB8:F::/48

AS Object: Purpose

- Documents peering policy with other Autonomous Systems
  - Lists network information
  - Lists contact information
  - Lists routes announced to neighbouring autonomous systems
  - Lists routes accepted from neighbouring autonomous systems
- Some operators pay close attention to what is contained in the AS Object
  - Some configure their border router BGP policy based on what is listed in the AS Object

AS Object: Example

```
aut-num:        AS64500
as-name:        ENTERPRISE-AS
descr:          Enterprise Network
country:        ZZ
import:         from AS64505  action pref=100;     accept ANY
export:         to AS64505    announce AS64500
import:         from AS64510  action pref=100;     accept ANY
export:         to AS64510    announce AS64500
<snip>
admin-c:        ENO1-ZZ
tech-c:         ENO1-ZZ
notify:         noc@yy.zz
mnt-by:         RIR-HM
mnt-lower:      MAINT-ZZ-ENTERPRISE
mnt-routes:     MAINT-ZZ-ENTERPRISE
last-modified:  2019-06-09T22:40:10Z
source:         IRR
```

Examples of inbound and outbound policies – RPSL

Internet Routing Registry: Summary

- Route Object
  - Essential to have one
  - These days usually created when a ROA is signed
- AS Object
  - Not essential, but useful and informative
  - Shows operator’s peering policy
    - And the ASNs connected to it

Multihoming decisions

- One upstream primary, the other upstream backup?
  - Leaves one link mostly unused
  - (not really recommended)

- Load balance between two upstreams
  - More common, to take advantage of all available capacity

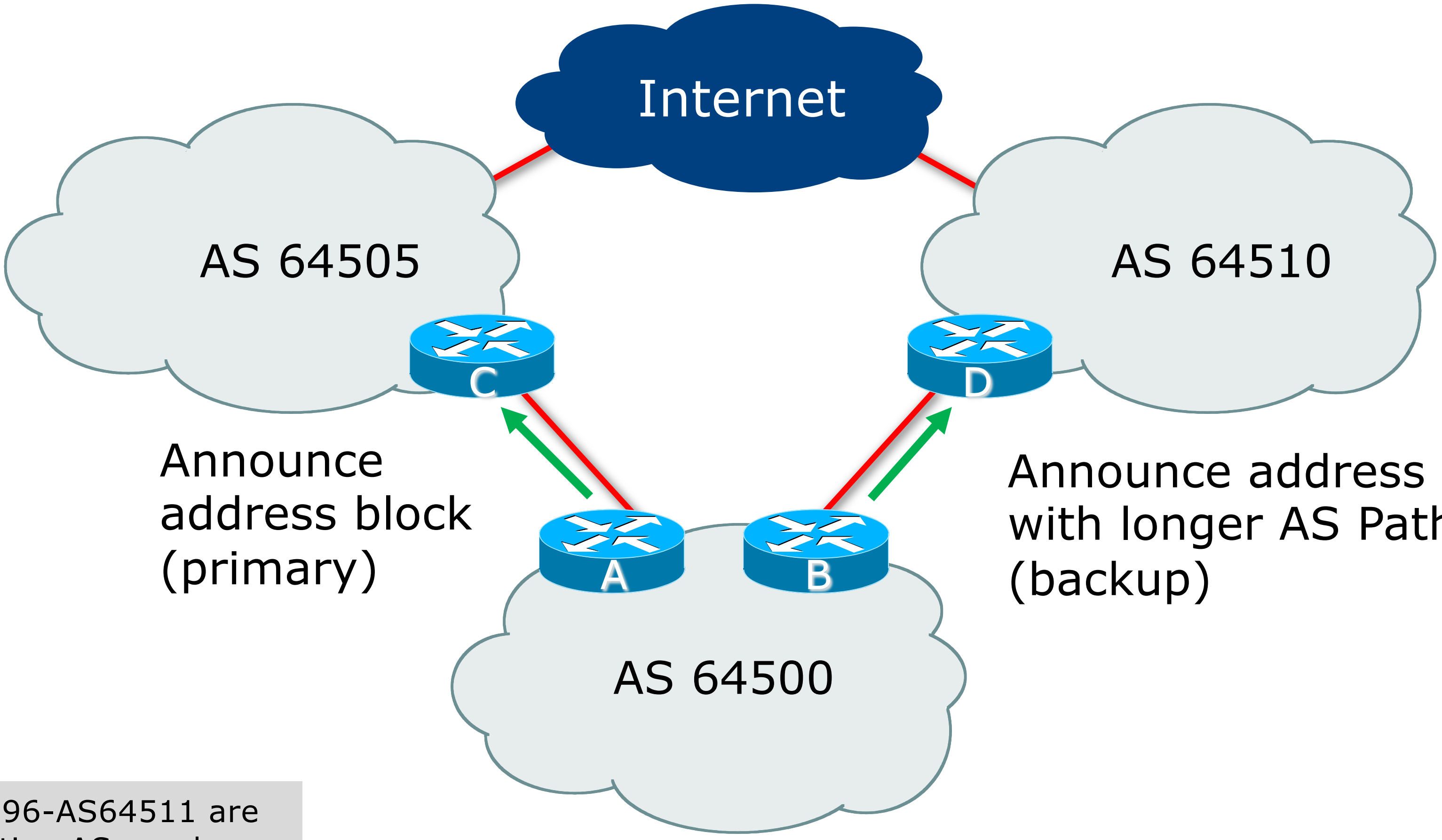
Agenda

- Background and Requirements
- The next steps
- 1st link primary, 2nd link backup
- Load share between both links
- End-site network configuration

1st link primary, 2nd link backup

- Entity has IPv4 /24 and IPv6 /48
- Basic principle:
  - Outbound announcements:
    - Make standard announcement of /24 and /48 on the link to the primary provider
    - Prepend the announcement of /24 and /48 on the link to the backup provider
      - Two or three prepends is enough!
  - Inbound:
    - Only need default route from both upstreams
    - Mark default route from backup provider with low local-preference

1st link primary, 2nd link backup

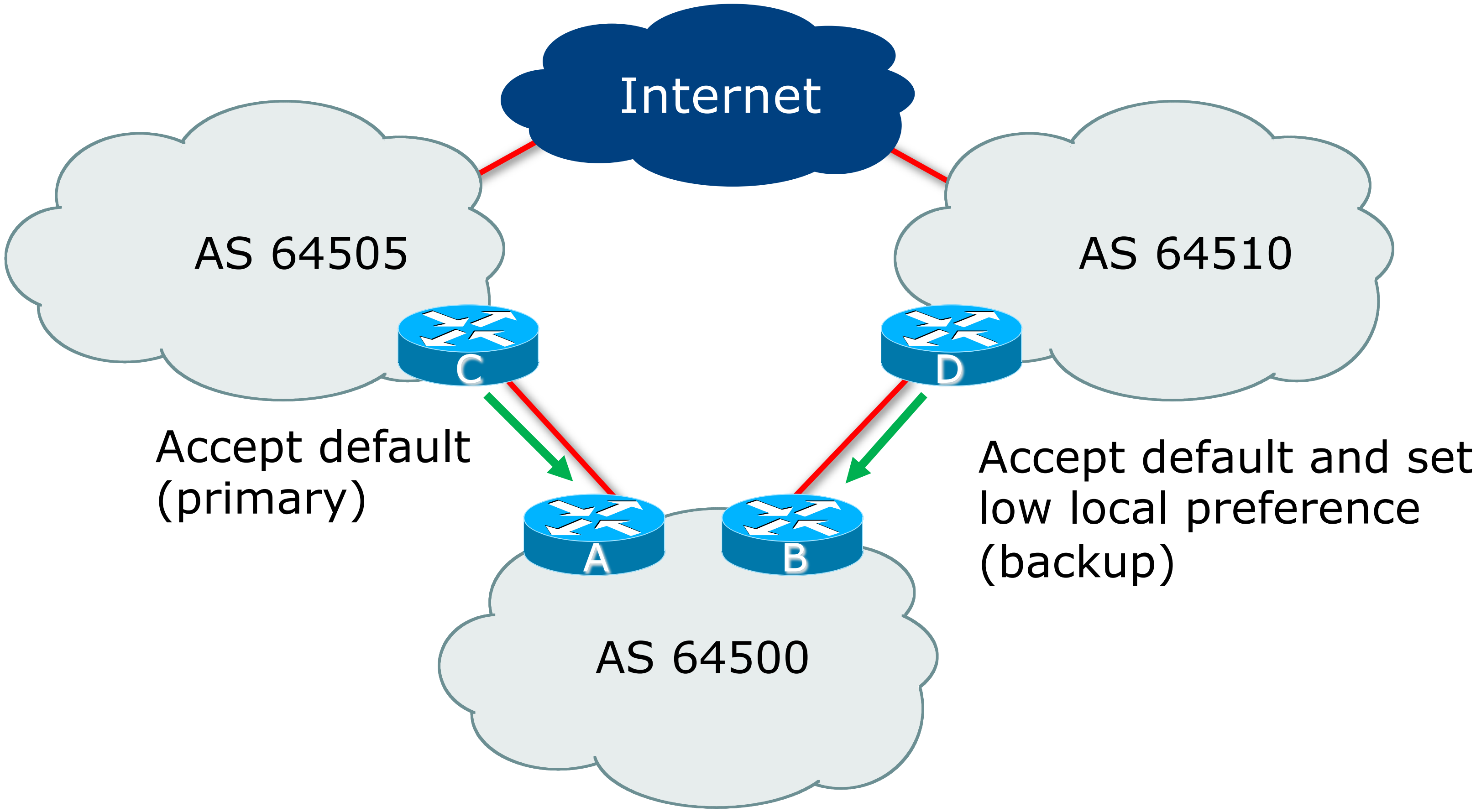


Announce address block (primary)

Announce address block with longer AS Path (backup)

NB: AS64496-AS64511 are documentation AS numbers

1st link primary, 2nd link backup



Accept default
(primary)

Accept default and set
low local preference
(backup)

1st link primary, 2nd link backup

- Router A Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  neighbor 100.66.10.1 remote-as 64505
  neighbor 100.66.10.1 prefix-list AGGREGATE out
  neighbor 100.66.10.1 prefix-list DEFAULT in
  neighbor 100.66.10.1 activate
!
ip prefix-list AGGREGATE permit 100.64.0.0/24
ip prefix-list DEFAULT permit 0.0.0.0/0
!
ip route 100.64.0.0 255.255.255.0 null0
```

1st link primary, 2nd link backup

- Router B Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  neighbor 100.67.5.1 remote-as 64510
  neighbor 100.67.5.1 route-map AS64510-PREPEND out
  neighbor 100.67.5.1 route-map LP-LOW in
  neighbor 100.67.5.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
!
...next slide...
```

1st link primary, 2nd link backup

```
ip prefix-list AGGREGATE permit 100.64.0.0/24
ip prefix-list DEFAULT permit 0.0.0.0/0
!
route-map AS64510-PREPEND permit 10
 description Three prepends to AS64510
 match ip address prefix-list AGGREGATE
 set as-path prepend 64500 64500 64500
!
route-map AS64510-PREPEND deny 20
!
route-map LP-LOW permit 10
 description All routes local pref 80
 match ip address prefix-list DEFAULT
 set local-preference 80
!
route-map LP-LOW deny 20
```

1st link primary, 2nd link backup

- Not a common situation as most sites tend to prefer using whatever capacity they have
  - (Useful when two competing ISPs agree to provide mutual backup to each other)
- But it shows the basic concepts of using local-prefs and AS-path prepends for engineering traffic in the chosen direction

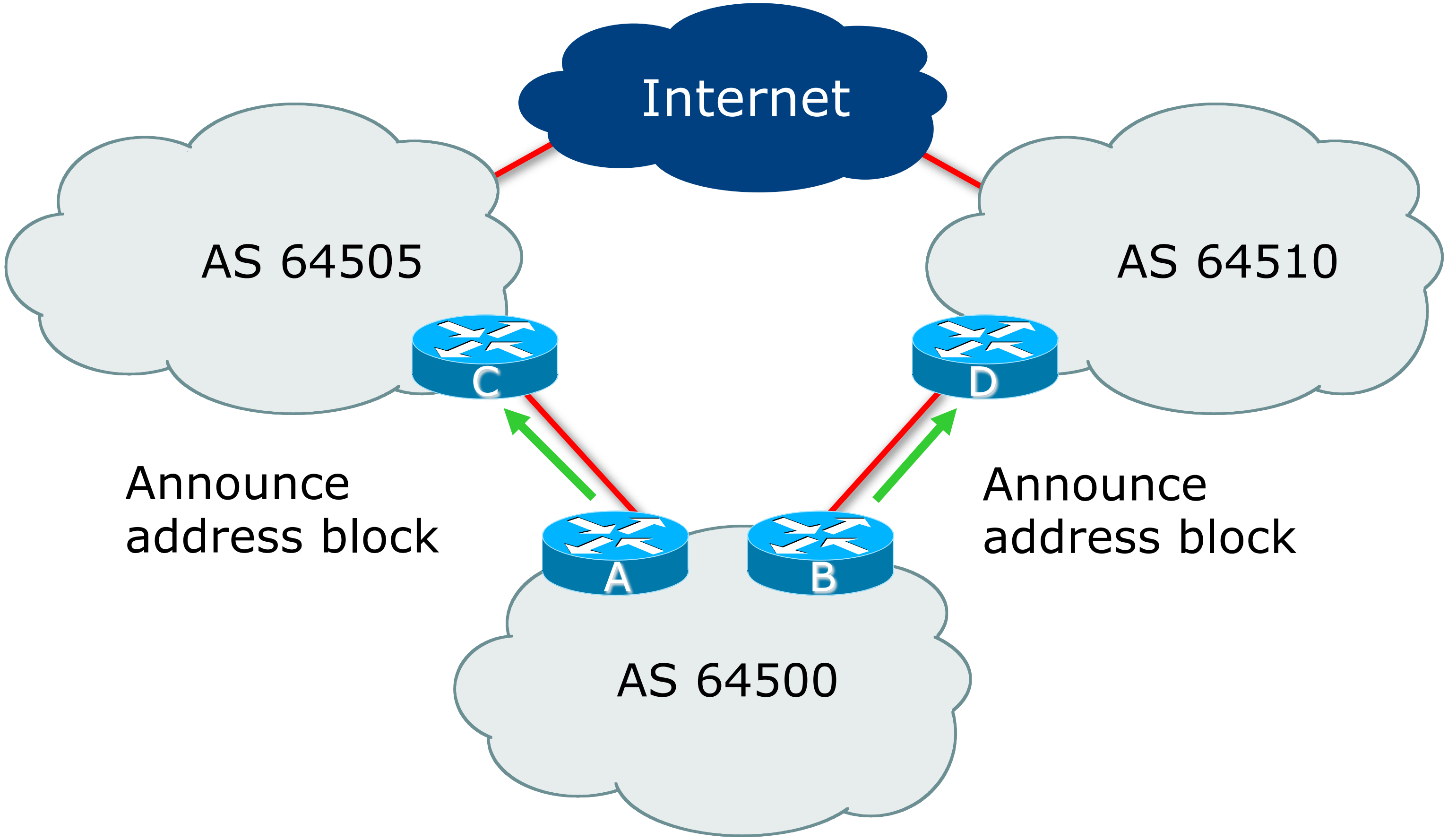
Agenda

- Background and Requirements
- The next steps
- 1st link primary, 2nd link backup
- Load share between both links
  - Option 1
  - Option 2
  - Option 3
  - Option 4
- End-site network configuration

Loadsharing between two upstreams: Option 1

- Entity has IPv4 /24 and IPv6 /48
  - Challenging to load balance 😩
- Basic principle:
  - Outbound announcements:
    - Make standard announcement of /24 and /48 on the link to the first provider
    - Make standard announcement of /24 and /48 on the link to the second provider
  - Inbound:
    - Accept default route from upstreams

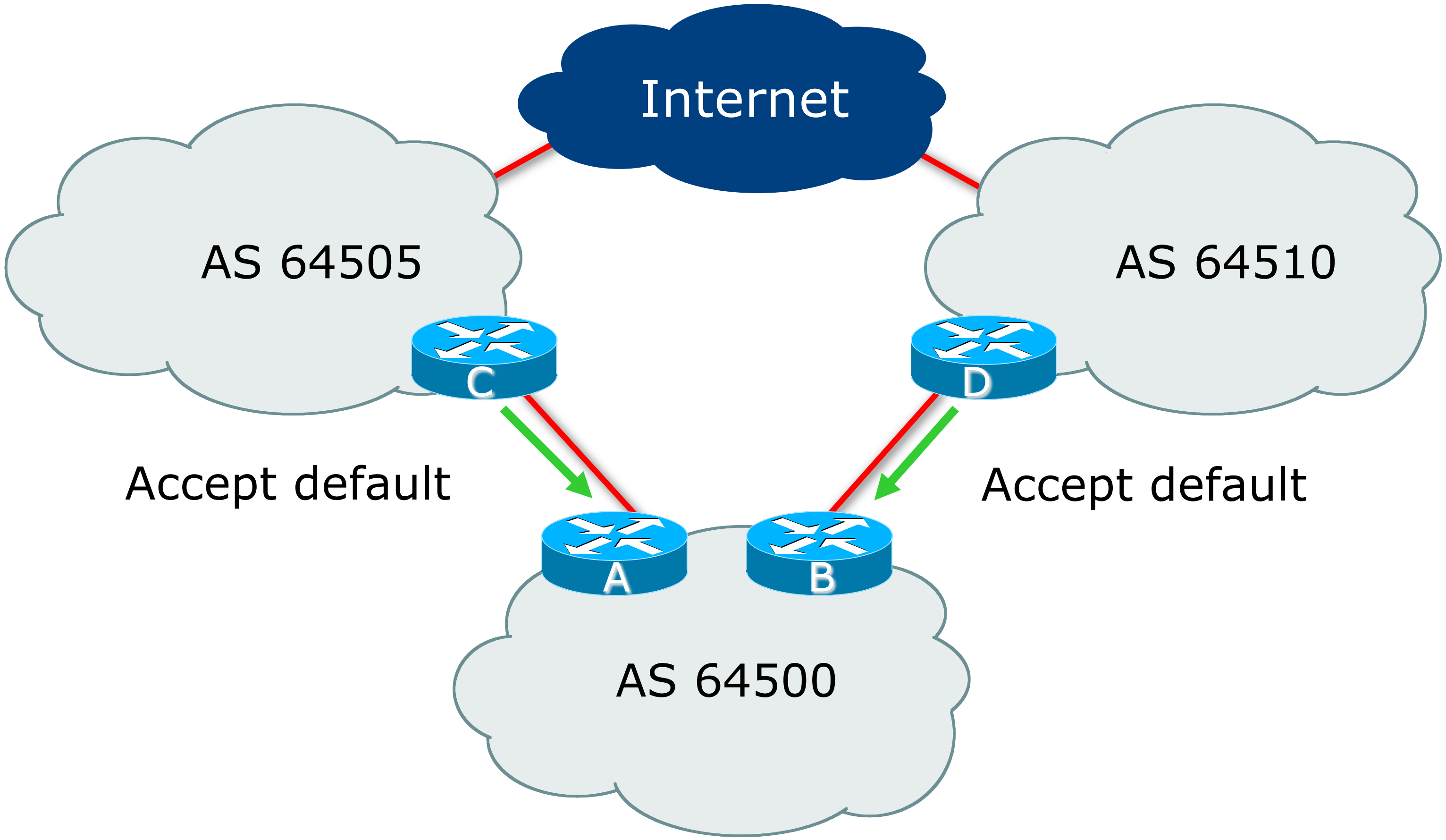
Loadsharing between two upstreams: Option 1



Announce address block

Announce address block

Loadsharing between two upstreams: Option 1



Loadsharing between two upstreams: Option 1

- Router A Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  neighbor 100.66.10.1 remote-as 64505
  neighbor 100.66.10.1 prefix-list AGGREGATE out
  neighbor 100.66.10.1 prefix-list DEFAULT in
  neighbor 100.66.10.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
!
ip prefix-list DEFAULT permit 0.0.0.0/0
ip prefix-list AGGREGATE permit 100.64.0.0/24
```

Loadsharing between two upstreams: Option 1

- Router B Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  neighbor 100.67.5.1 remote-as 64510
  neighbor 100.67.5.1 prefix-list AGGREGATE out
  neighbor 100.67.5.1 prefix-list DEFAULT in
  neighbor 100.67.5.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
!
ip prefix-list DEFAULT permit 0.0.0.0/0
ip prefix-list AGGREGATE permit 100.64.0.0/24
```

Loadsharing between two upstreams: Option 1

- Problems:
  - No load balancing of outbound traffic
    - BGP has only one best path
  - Which upstream used for outbound traffic?
    - Depends on router implementation
      - Might be the first to bring up BGP session
      - Might be the lowest neighbour IP address
  - Load balancing of inbound traffic is non-deterministic
    - Relies on the AS-PATH length from "the Internet", i.e. content that the entity's users want to access
      - Shortest AS-PATH wins, meaning one upstream may carry all incoming traffic

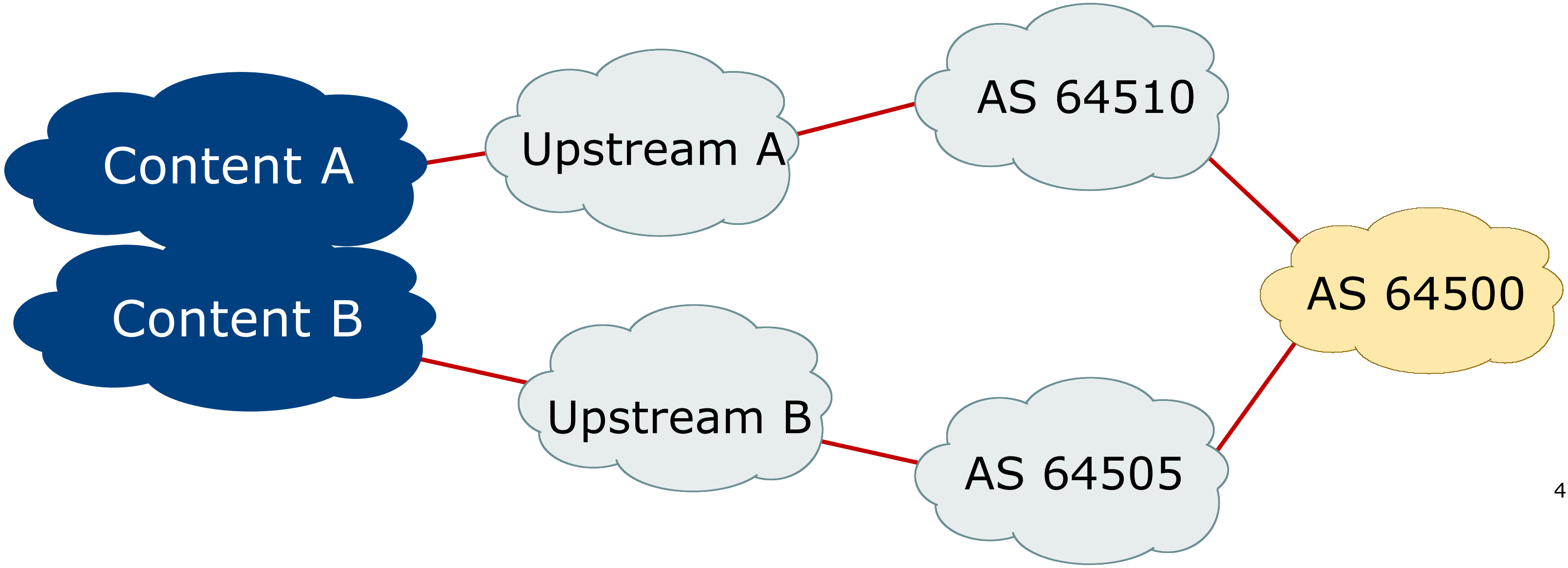
Loadsharing between two upstreams: Option 1

- Option 1 is no good at all:
  - Cannot subdivide the IPv4 /24 or the IPv6 /48 to help with inbound load balancing
  - Cannot load balance outbound traffic between two default routes

- Fixes:
  - Obtain two IPv4 /24s and two IPv6 /48s
    - This is examined in Option 2 following
  - Request upstream providers to send "some routes"
    - This is examined in Option 3 & 4 following

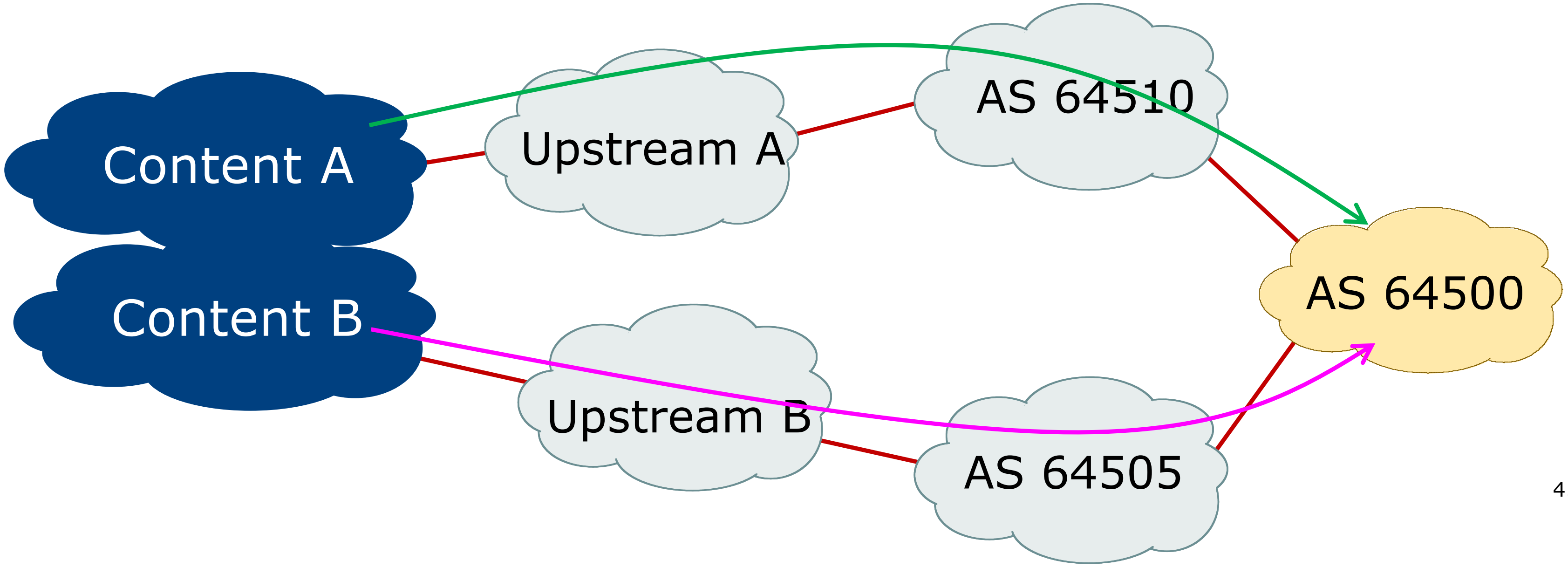
Loadsharing between two upstreams: Option 1

- Previous example handles this situation:
  - Equal path lengths from various content to the multihoming entity
  - Content A should traverse Upstream A to get to AS64500
  - Content B should traverse Upstream B to get to AS64500



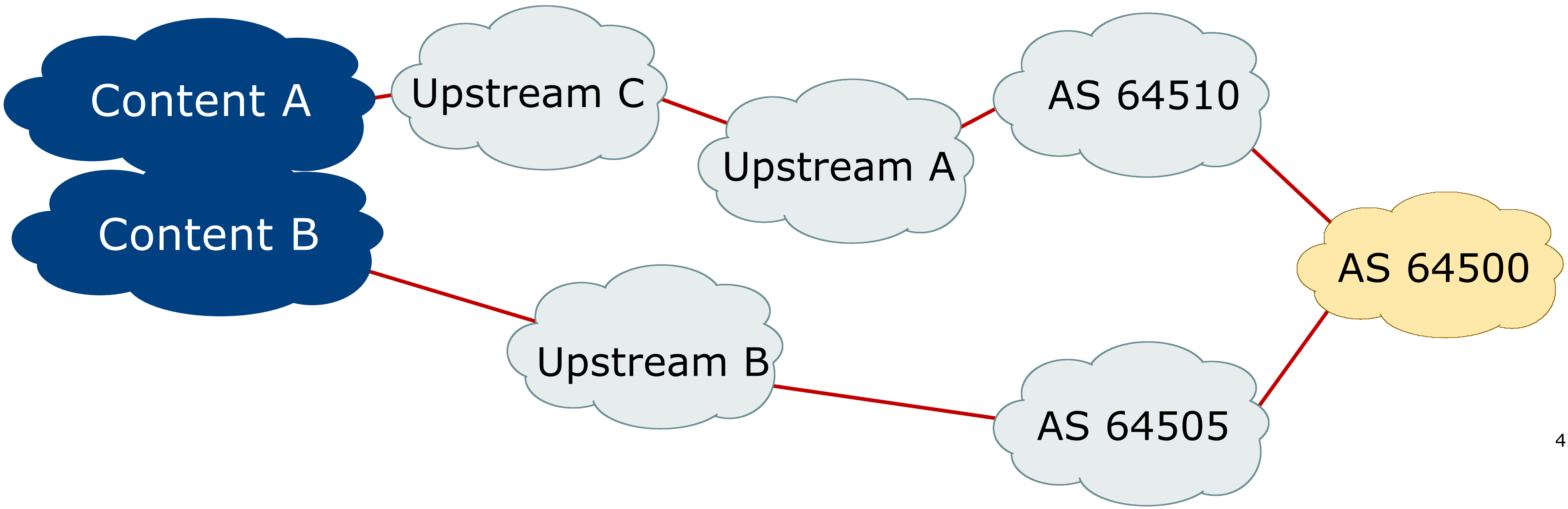
Loadsharing between two upstreams: Option 1

- Previous example handles this situation:
  - Equal path lengths from various content to the multihoming entity
  - Content A should traverse Upstream A to get to AS64500
  - Content B should traverse Upstream B to get to AS64500



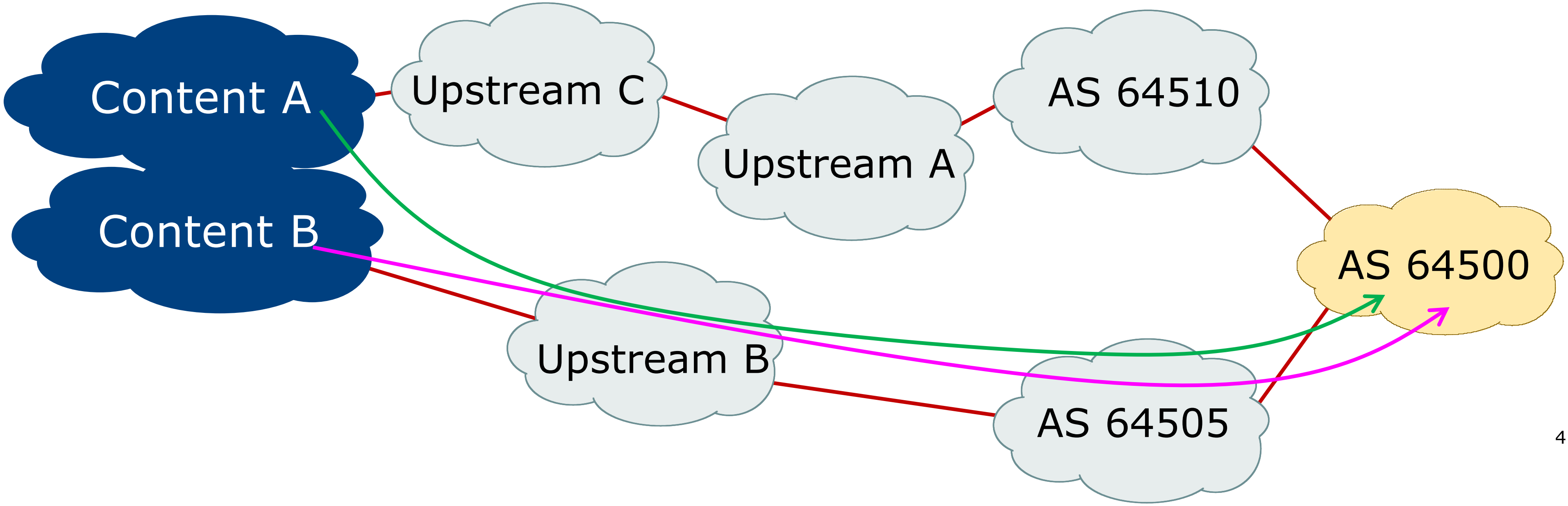
Loadsharing between two upstreams: Option 1

- What about unequal path lengths from content to multihoming entity?



Loadsharing between two upstreams: Option 1

- What about unequal path lengths from content to multihoming entity?
  - Need to prepend announcement from AS64500 to make effective path length the same



Loadsharing between two upstreams: Option 1

- Router A Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  neighbor 100.66.10.1 remote-as 64505
  neighbor 100.66.10.1 route-map AS64505-prepend out
  neighbor 100.66.10.1 prefix-list DEFAULT in
  neighbor 100.66.10.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
!
ip prefix-list DEFAULT permit 0.0.0.0/0
ip prefix-list AGGREGATE permit 100.64.0.0/24
!
route-map AS64505-prepend permit 10
 match ip address prefix-list AGGREGATE
 set as-path prepend 64500
```

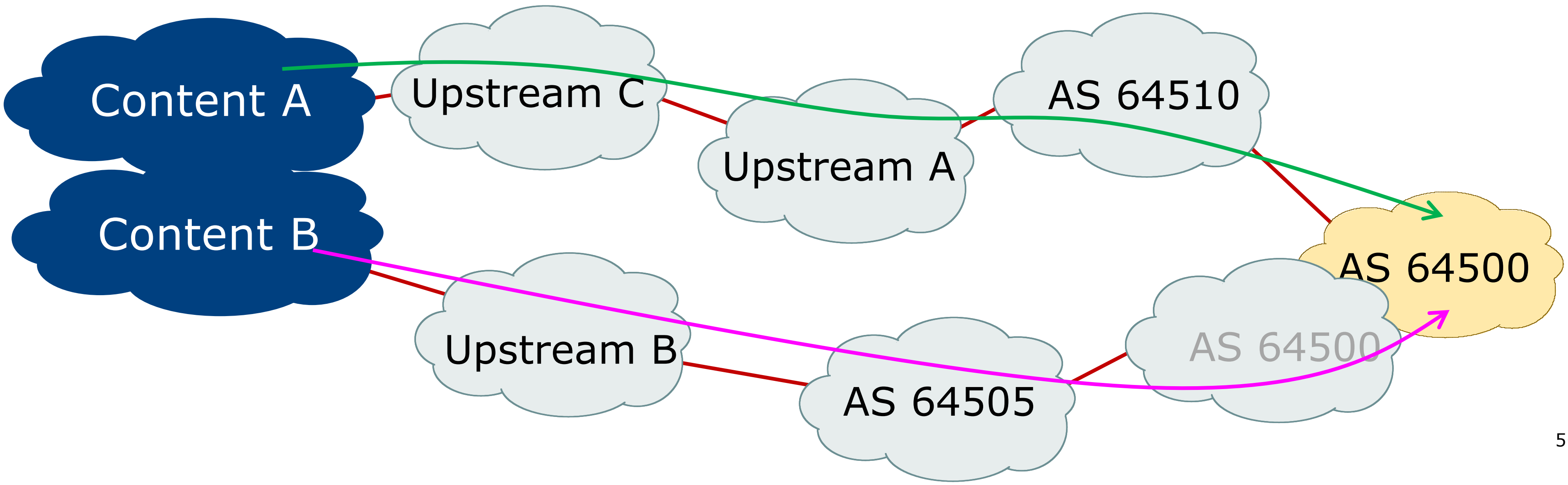
Loadsharing between two upstreams: Option 1

- Result:
  - The path length from Content B to AS64500 is now the same as the path length from Content A to AS64500
  - So we should be back to some sort of load balancing for incoming traffic
    - Should be okay given most Internet end-sites are downloaders
  - (Outgoing is still following default route – not optimised at all)

- Comments:
  - Traffic engineering with a single IPv4 /24 and a single IPv6 /48 is hard

Loadsharing between two upstreams: Option 1

- Prepending announcement from AS64500 to make effective path length from perspective of Content A and Content B the same



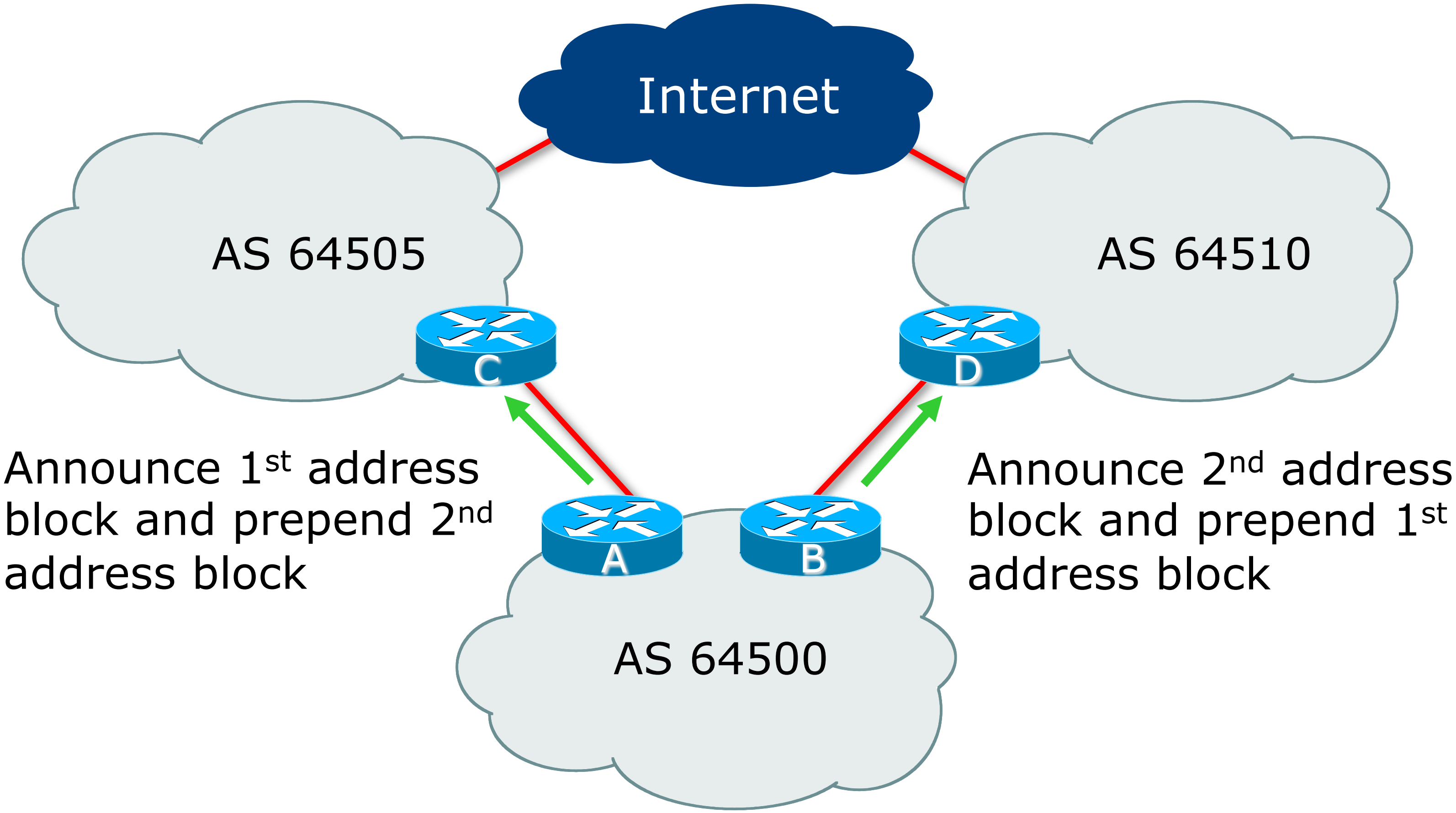
Agenda

- Background and Requirements
- The next steps
- 1st link primary, 2nd link backup
- Load share between both links
  - Option 1
  - Option 2
  - Option 3
  - Option 4
- End-site network configuration

Loadsharing between two upstreams: Option 2

- Entity has two IPv4 /24s and two IPv6 /48s
  - Easier to load balance 😁
  - One /24 and /48 is used on the first link
  - The other /24 and /48 is used on the second link
- Basic principle:
  - Outbound announcements:
    - Make standard announcement of first /24 and /48 and prepend the second /24 and /48 on the link to the first provider
    - Make standard announcement of second /24 and /48 and prepend the first /24 and /48 on the link to the second provider
  - Inbound:
    - Accept default route from both upstreams

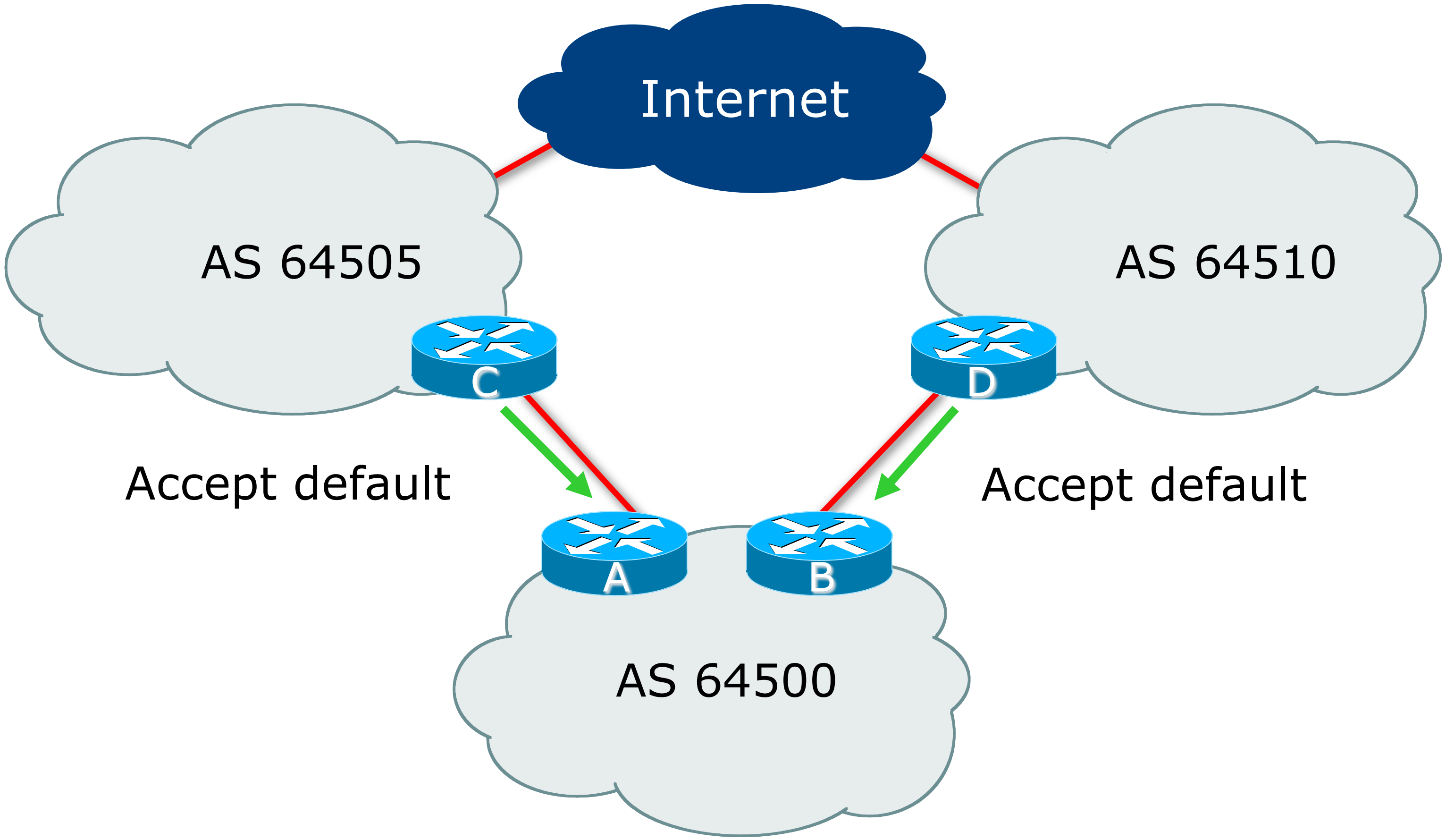
Loadsharing between two upstreams: Option 2



Announce 1st address block and prepend 2nd address block

Announce 2nd address block and prepend 1st address block

Loadsharing between two upstreams: Option 2



Loadsharing between two upstreams: Option 2

- Router A Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  network 100.64.15.0 mask 255.255.255.0
  neighbor 100.66.10.1 remote-as 64505
  neighbor 100.66.10.1 route-map AS64505-OUT out
  neighbor 100.66.10.1 prefix-list DEFAULT in
  neighbor 100.66.10.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
ip route 100.64.15.0 255.255.255.0 null0
!
...continued...
```

Loadsharing between two upstreams: Option 2

```
ip prefix-list DEFAULT permit 0.0.0.0/0
ip prefix-list FIRST24 permit 100.64.0.0/24
ip prefix-list SECOND24 permit 100.64.15.0/24
!
route-map AS64505-OUT permit 10
 description 1st /24 untouched
 match ip address prefix-list FIRST24
!
route-map AS64505-OUT permit 20
 description 2nd /24 three prepends to AS64505
 match ip address prefix-list SECOND24
 set as-path prepend 64500 64500 64500
!
route-map AS64505-OUT deny 30
 description Drop everything else
!
```

Loadsharing between two upstreams: Option 2

- Router B Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  network 100.64.15.0 mask 255.255.255.0
  neighbor 100.67.5.1 remote-as 64510
  neighbor 100.67.5.1 route-map AS64510-OUT out
  neighbor 100.67.5.1 prefix-list DEFAULT in
  neighbor 100.67.5.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
ip route 100.64.15.0 255.255.255.0 null0
!
...continued...
```

Loadsharing between two upstreams: Option 2

```
ip prefix-list DEFAULT permit 0.0.0.0/0
ip prefix-list FIRST24 permit 100.64.0.0/24
ip prefix-list SECOND24 permit 100.64.15.0/24
!
route-map AS64510-OUT permit 10
 description 1st /24 three prepends to AS64510
 match ip address prefix-list FIRST24
 set as-path prepend 64500 64500 64500
!
route-map AS64510-OUT permit 20
 description 2nd /24 untouched
 match ip address prefix-list SECOND24
!
route-map AS64510-OUT deny 30
 description Drop everything else
!
```

Loadsharing between two upstreams: Option 2

- Inbound traffic flow:
  - All traffic for first /24 and first /48 comes in through AS64505 upstream
  - All traffic for second /24 and second /48 comes in through AS64510 upstream
  - Loadbalancing? Not really:
    - Entity needs to implement addressing plan to equally use IP address space across both blocks, keeping in mind the traffic levels
    - But this is all that can be done with small address space
    - Only extra tuning available is to adjust AS-Path prepend on primary and backup paths
- Outbound traffic flow:
  - No change from Option 1

Agenda

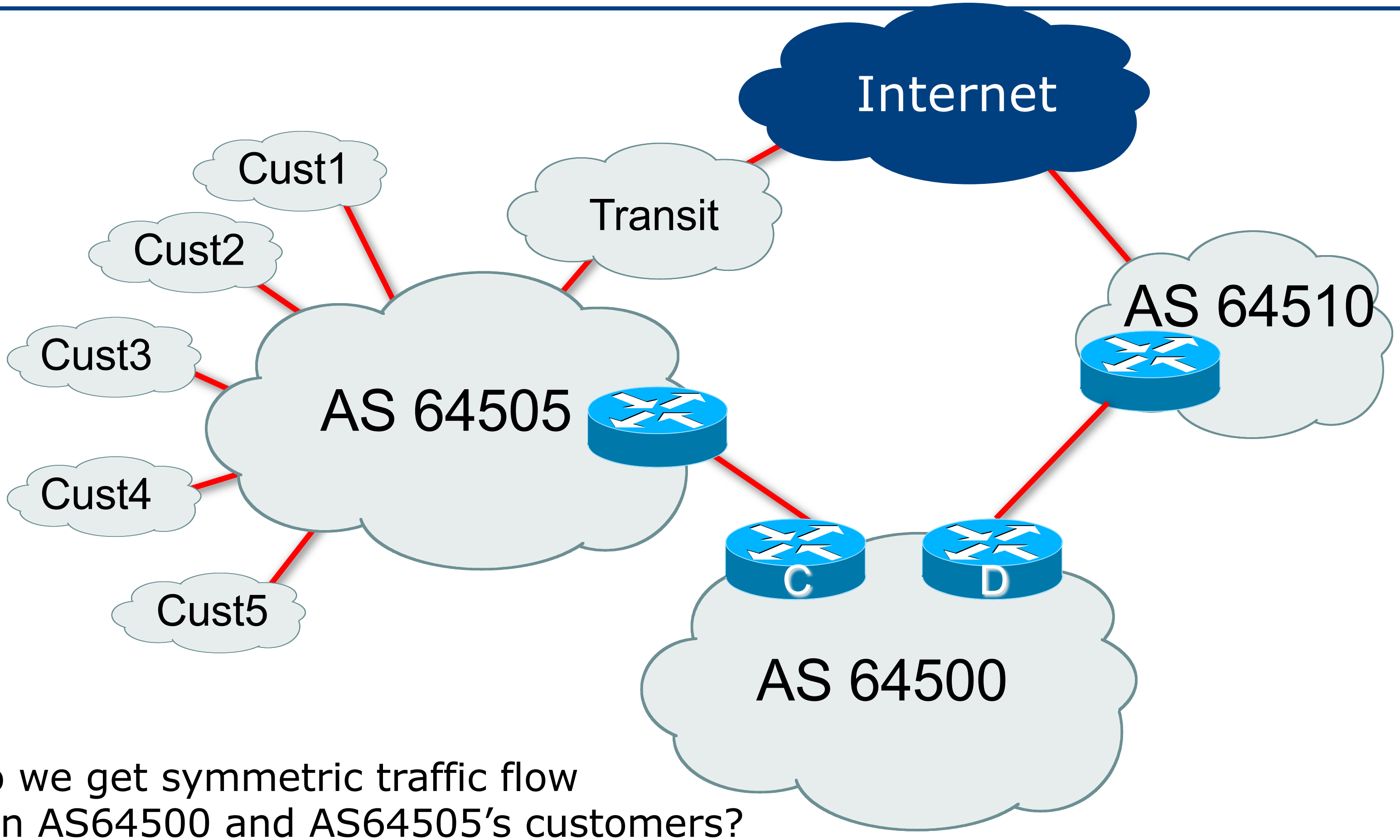
- Background and Requirements
- The next steps
- 1st link primary, 2nd link backup
- Load share between both links
  - Option 1
  - Option 2
  - Option 3
  - Option 4
- End-site network configuration

Loadsharing between two upstreams: Option 3

- Entity has two IPv4 /24s and two IPv6 /48s
  - Easier to load balance 😁
- Basic principle:
  - Outbound announcements unchanged from Option 2
  - Inbound:
    - Ask upstreams for the default route, their aggregates, and all customer originated provider independent address space (Option 3a)

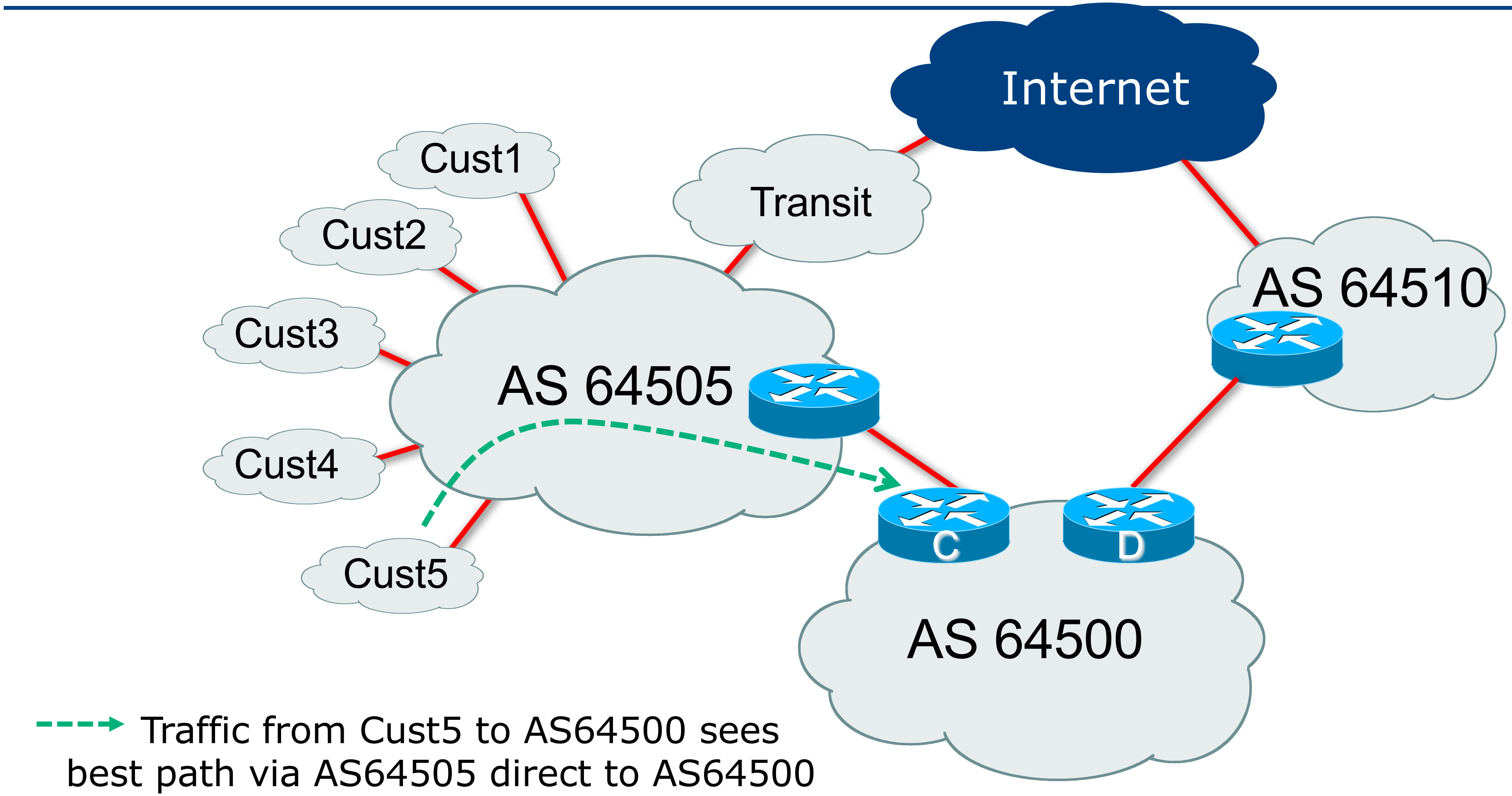
      OR
    - Ask upstreams for the default and the global BGP table (Option 3b)

Loadsharing with different ASes



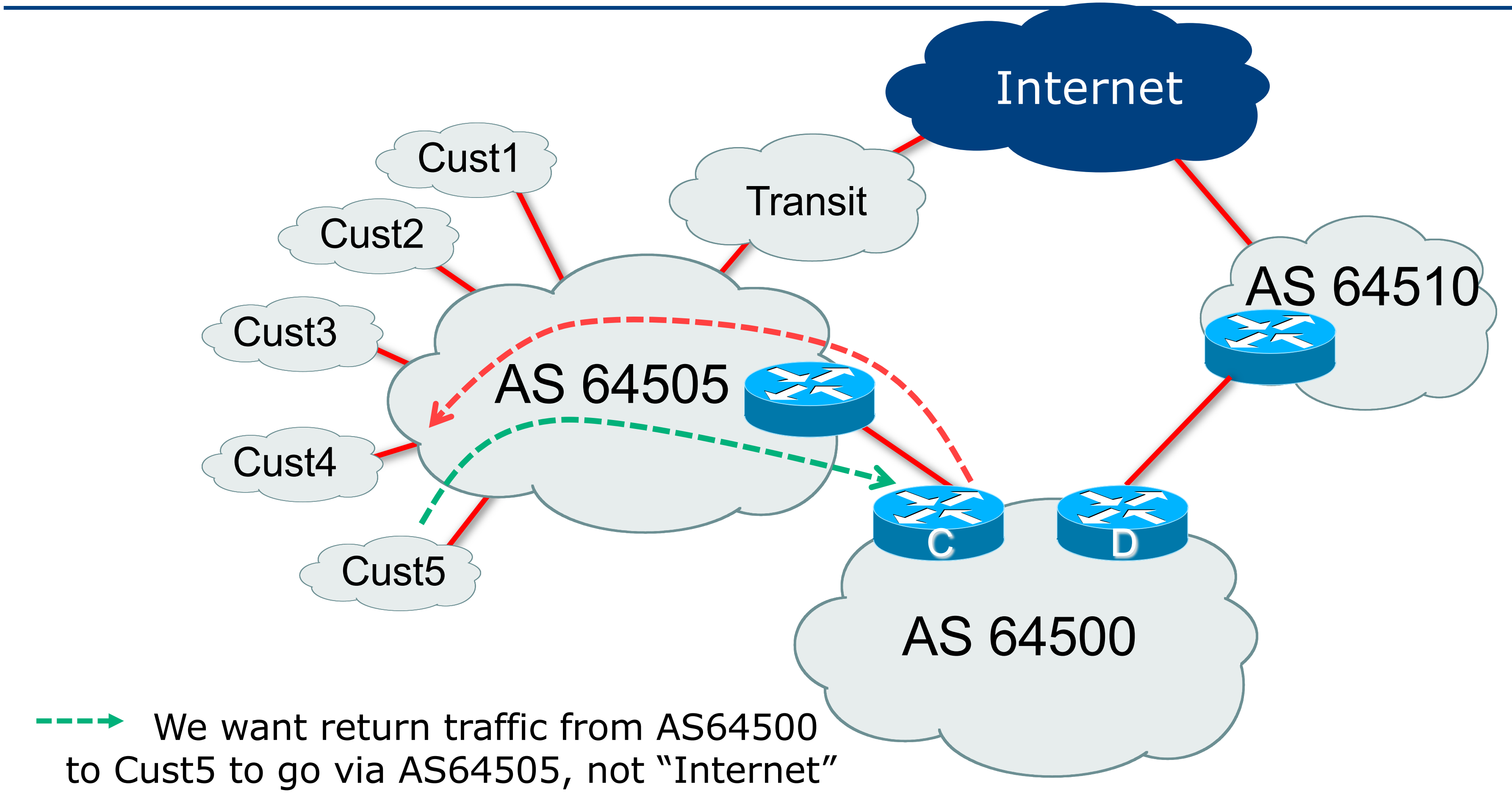
How do we get symmetric traffic flow
between AS64500 and AS64505's customers?

Loadsharing with different ASes



Traffic from Cust5 to AS64500 sees best path via AS64505 direct to AS64500

Loadsharing with different ASes

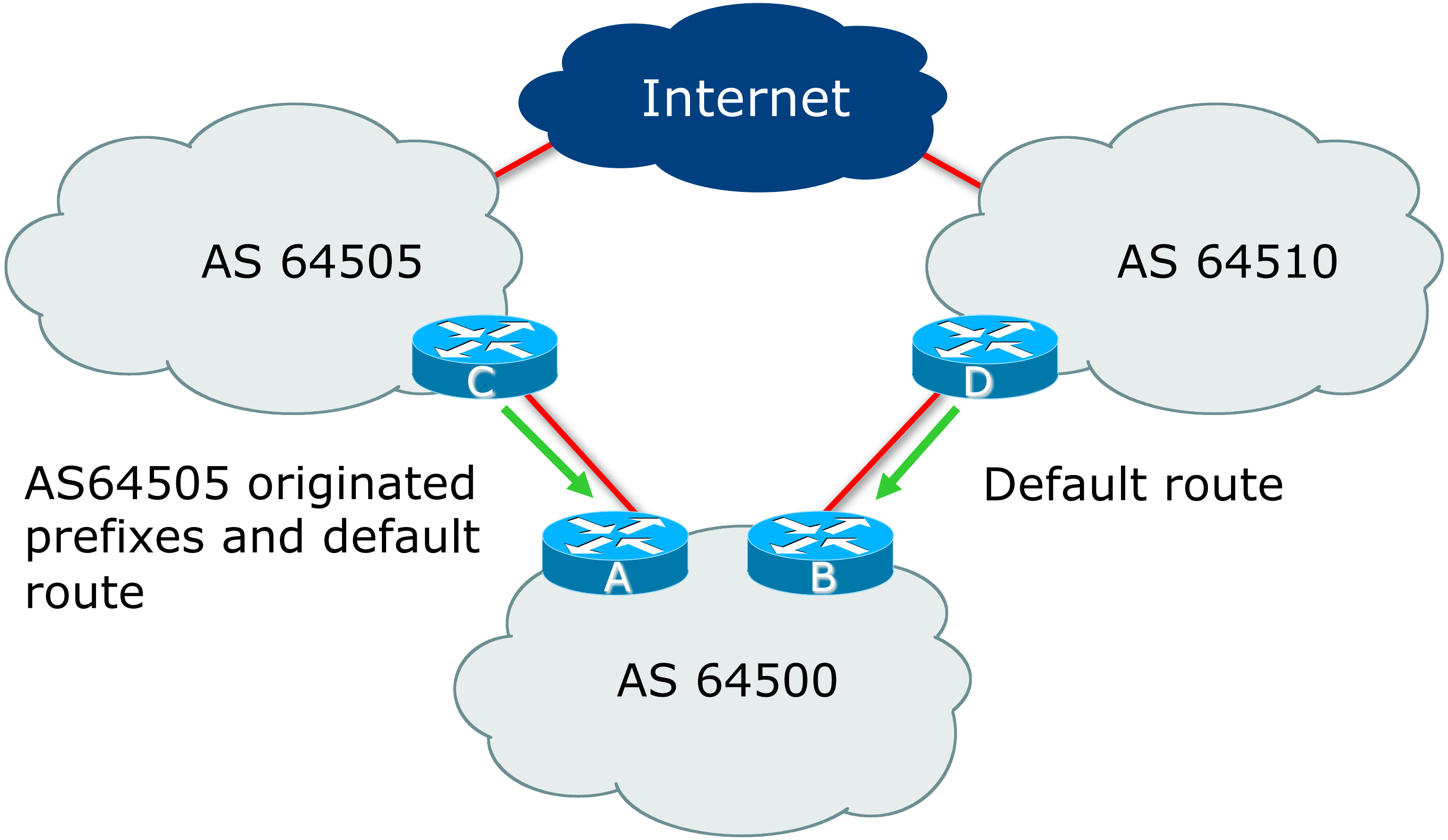


We want return traffic from AS64500 to Cust5 to go via AS64505, not "Internet"

Loadsharing between two upstreams: Option 3a

- First the theory:
  - AS64500 announces address space to AS64505
    - Best path from AS64505 to AS64500 is over the direct link
    - Therefore the return path for traffic needs to be over the direct link too
    - And so AS64505 needs to send their originated aggregates to AS64500
      - This includes any customers using AS64505's address space
    - Default route also sent – gets low local-preference
  - Result:
    - Traffic from AS64505 and its customers goes directly to AS64500
    - Traffic from AS64500 goes directly to AS64505 and its customers
    - The path to AS64510 is used for everything else
    - Adjust to suit (see Option 3b)

Loadsharing between two upstreams: Option 3a



AS64505 originated prefixes and default route

Default route

Loadsharing between two upstreams: Option 3a

- Router A Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  network 100.64.15.0 mask 255.255.255.0
  neighbor 100.66.10.1 remote-as 64505
  neighbor 100.66.10.1 route-map AS64505-OUT out
  neighbor 100.66.10.1 route-map AS64505-IN in
  neighbor 100.66.10.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
ip route 100.64.15.0 255.255.255.0 null0
!
...continued...
```

Loadsharing between two upstreams: Option 3a

```
ip prefix-list DEFAULT permit 0.0.0.0/0
!
ip as-path access-list 1 permit ^64505$
!
route-map AS64505-IN permit 10
 description Accept default
 match ip address prefix-list DEFAULT
 set local-preference 80
!
route-map AS64505-IN permit 20
 description Accept AS64505 originated routes
 match as-path access-list 1
!
route-map AS64505-IN deny 30
 description Drop everything else
!
```

Loadsharing between two upstreams: Option 3a

- Router B Configuration

Same as earlier

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  network 100.64.15.0 mask 255.255.255.0
  neighbor 100.67.5.1 remote-as 64510
  neighbor 100.67.5.1 route-map AS64510-OUT out
  neighbor 100.67.5.1 prefix-list DEFAULT in
  neighbor 100.67.5.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
ip route 100.64.15.0 255.255.255.0 null0
!
ip prefix-list DEFAULT permit 0.0.0.0/0
...
```

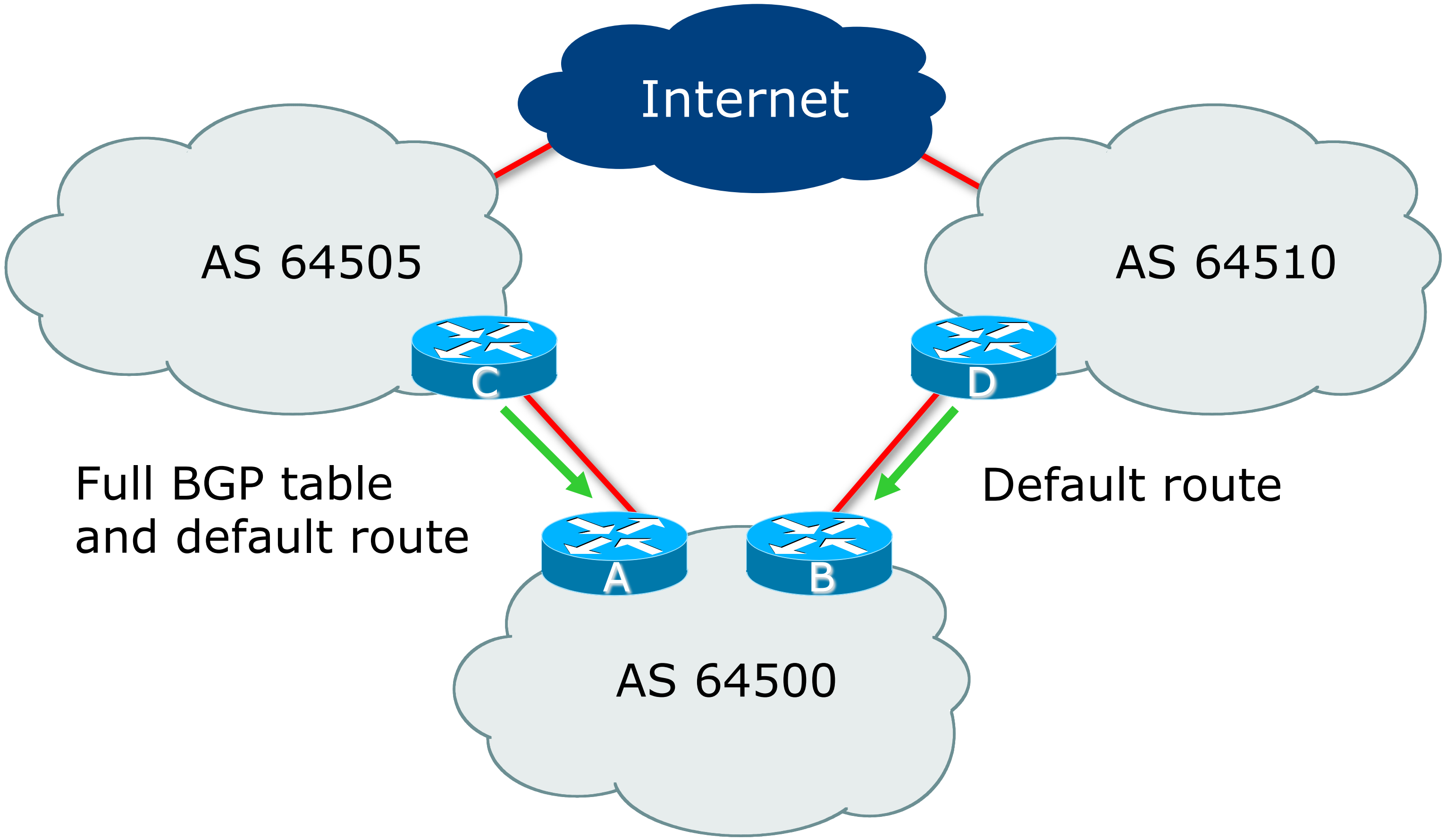
Loadsharing between two upstreams: Option 3a

- AS64500 sees the following routing table entries:
  - Prefixes originated by AS64505 upstream
  - Default route from AS64505 upstream
    - Local preference set to 80 (less than default 100)
  - Default route from AS64510 upstream
- Result:
  - All traffic to AS64505 goes via direct link
  - All traffic to rest of Internet goes via AS64510
- Is this ideal?
  - No, but it’s a start!

Loadsharing between two upstreams: Option 3b

- First the theory:
  - In addition to Option 3a…
  - AS64505 announces address space learned from AS64500 to its customers, peers, and transits
    - Best path from customers, peers (and possibly transits) will be via AS64505 to AS64500
    - Therefore the return path for traffic needs to be over the direct link too
    - And so AS64505 needs to send their customers, peers, and transit originated aggregates to AS64500 too
  - Unlikely for any upstream provider to give this degree of granularity
    - Solution: ask them for the global BGP table
    - But don’t panic: we’ll throw most of it away!

Loadsharing between two upstreams: Option 3b



Loadsharing between two upstreams: Option 3b

- Router A Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  network 100.64.15.0 mask 255.255.255.0
  neighbor 100.66.10.1 remote-as 64505
  neighbor 100.66.10.1 route-map AS64505-OUT out
  neighbor 100.66.10.1 route-map AS64505-IN in
  neighbor 100.66.10.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
ip route 100.64.15.0 255.255.255.0 null0
!
...continued...
```

Loadsharing between two upstreams: Option 3b

```
ip prefix-list DEFAULT permit 0.0.0.0/0
!
ip as-path access-list 1 permit ^64505$
ip as-path access-list 1 permit ^64505_[0-9]+$
!
route-map AS64505-IN permit 10
 description Accept default
 match ip address prefix-list DEFAULT
 set local-preference 80
!
route-map AS64505-IN permit 20
 description Accept AS64505 originated routes
 match as-path access-list 1
!
route-map AS64505-IN deny 30
 description Drop everything else
!
```

Loadsharing between two upstreams: Option 3b

- AS64500 sees the following routing table entries:
  - Prefixes originated by AS64505 upstream
  - Prefixes originated by the immediate AS neighbours of AS64505
  - Default route from AS64505 upstream
    - Local preference set to 80 (less than default 100)
  - Default route from AS64510 upstream
- Result:
  - All traffic to AS64505 and its AS neighbours goes via direct link
  - All traffic to rest of Internet goes via AS64510
- Is this ideal?
  - It’s a lot better than Option 3a!

Loadsharing between two upstreams

- How to progress this further?
  - For more outbound traffic on the link to AS64505, or to have more symmetric traffic flows, allow another AS in the permitted path:

```
ip as-path access-list 1 permit ^64505$
ip as-path access-list 1 permit ^64505_[0-9]+$
ip as-path access-list 1 permit ^64505_[0-9]+_[0-9]+$
```

  - And if that is too much, start excluding ASNs seen in the received paths
    - There are a large number of possible variations here
    - Adjust to suit local needs and local conditions

Loadsharing between two upstreams

- Here is one from a real live network:
  - Next-hop AS suppressed (64501 used instead)
  - Example to give a hint as to the thought process

```
ip as-path access-list 1 permit _64501$
ip as-path access-list 1 permit _64501_[0-9]+$
ip as-path access-list 1 permit _64501_[0-9]+_[0-9]+$
ip as-path access-list 1 deny _4637_
ip as-path access-list 1 deny _1299_
ip as-path access-list 1 deny _6453_
ip as-path access-list 1 deny _3356_
ip as-path access-list 1 deny _3491_
ip as-path access-list 1 permit _16509_
ip as-path access-list 1 permit _2764_[0-9]+$
ip as-path access-list 1 permit _13335_
ip as-path access-list 1 permit _1221_[0-9]+$
ip as-path access-list 1 permit _7474_[0-9]+$
ip as-path access-list 1 permit _7545_[0-9]+$
```

Loadsharing between two upstreams

- AS-PATH access-list looks quite complicated – what does it achieve for the end site?
  - Both upstreams are connected to the same Internet Exchange Point
  - Transit providers of both upstreams have private peering with each other
  - Major content providers peer across the same Internet Exchange Point

- AS-PATH filter, in this example, is successful in ensuring that traffic flow across both upstreams is mostly symmetric
  - Needed a lot of trial and error to make it work well
  - And it is not "fit and forget" – needs regular monitoring

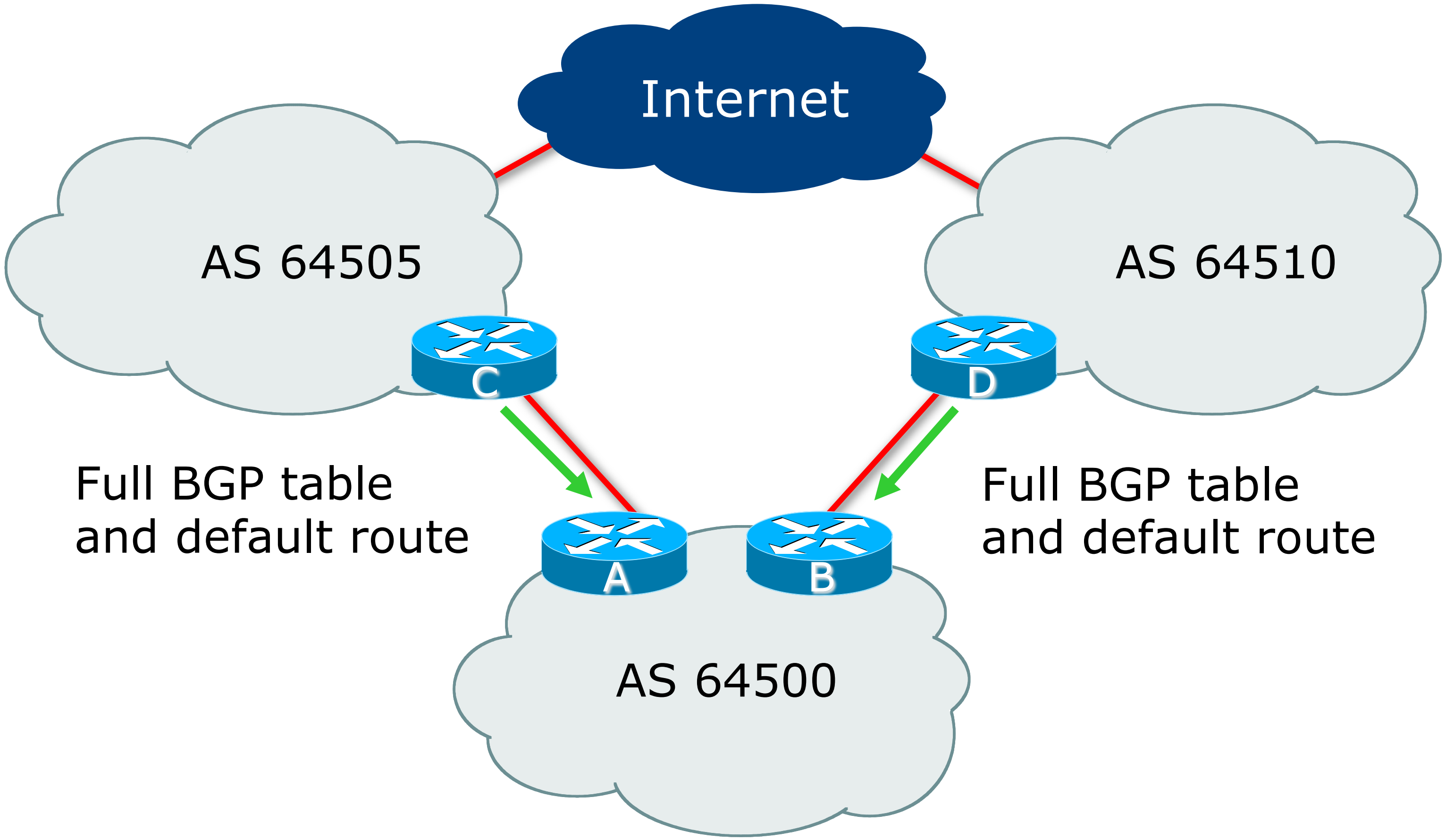
Agenda

- Background and Requirements
- The next steps
- 1st link primary, 2nd link backup
- Load share between both links
  - Option 1
  - Option 2
  - Option 3
  - Option 4
- End-site network configuration

Loadsharing between two upstreams: Option 4

- What about the link to the other upstream
  - That can remain as just a default route
  - Because traffic to them and their adjacent ASNs will simply follow defaults
- However, if the two upstreams have customers multihoming between them, then:
  - Get the full BGP table from AS64510 as well
  - Filter based on AS path, as was done for AS64505
- No right or wrong solution
  - Having confidence to filter based on AS path is the key to making this work
  - And try and keep traffic flows symmetric wherever possible

Loadsharing between two upstreams: Option 4



Loadsharing between two upstreams: Option 4

- Similar configuration for both upstreams:
  - Request full BGP table from both
    - And the default route
  - Don’t panic! We are throwing most of it away
  - AS-PATH filter keeps adjacent ASNs only
    - Adjust the filter to suit your local conditions
  - As for default route:
    - No need to local preference it now – leave default 100
    - Best path will be by lowest router ID (not ideal either)

Loadsharing between two upstreams: Option 4

- Router A Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  network 100.64.15.0 mask 255.255.255.0
  neighbor 100.66.10.1 remote-as 64505
  neighbor 100.66.10.1 route-map AS64505-OUT out
  neighbor 100.66.10.1 route-map AS64505-IN in
  neighbor 100.66.10.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
ip route 100.64.15.0 255.255.255.0 null0
!
...continued...
```

Loadsharing between two upstreams: Option 4

```
ip prefix-list DEFAULT permit 0.0.0.0/0
!
ip as-path access-list 1 permit ^64505$
ip as-path access-list 1 permit ^64505_[0-9]+$
ip as-path access-list 1 permit ^64505_[0-9]+_[0-9]+$
!
route-map AS64505-IN permit 10
 description Accept default
 match ip address prefix-list DEFAULT
!
route-map AS64505-IN permit 20
 description Accept AS64505 originated routes
 match as-path access-list 1
!
route-map AS64505-IN deny 30
 description Drop everything else
!
```

Loadsharing between two upstreams: Option 4

- Router B Configuration

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  network 100.64.15.0 mask 255.255.255.0
  neighbor 100.67.5.1 remote-as 64510
  neighbor 100.67.5.1 route-map AS64510-OUT out
  neighbor 100.67.5.1 route-map AS64510-IN in
  neighbor 100.67.5.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
ip route 100.64.15.0 255.255.255.0 null0
!
...continued...
```

Loadsharing between two upstreams: Option 4

```
ip prefix-list DEFAULT permit 0.0.0.0/0
!
ip as-path access-list 2 permit ^64510$
ip as-path access-list 2 permit ^64510_[0-9]+$
ip as-path access-list 2 permit ^64510_[0-9]+_[0-9]+$
!
route-map AS64510-IN permit 10
 description Accept default
 match ip address prefix-list DEFAULT
!
route-map AS64510-IN permit 20
 description Accept AS64510 originated routes
 match as-path access-list 2
!
route-map AS64510-IN deny 30
 description Drop everything else
!
```

Loadsharing between two upstreams: Option 4

- AS64500 sees the following routing table entries:
  - Prefixes originated by AS64505 and AS64510 upstreams
  - Prefixes originated by the immediate AS neighbours of AS64505 and AS64510
  - Prefixes originated by the AS neighbours of immediate AS neighbours of AS64505 and AS64510
  - Default route from AS64505 and AS64510 upstreams
  - Default route from AS64510 upstream
- Result:
  - All traffic to AS64505, AS64510, their immediate AS neighbours, and the immediate AS neighbours of those neighbours, follows the direct specific path
  - All traffic to rest of Internet follows the default route
- Is this ideal?
  - It can be better than Option 3b in the case where AS64505 and AS64510 have a direct connection with each other (private or bi-lateral/IXP)

Loadsharing between two upstreams: Option 4

- Further improvements:
  - Selecting one upstream as default, and the other as backup
    - Right now, default is by lowest upstream neighbour IP address – not ideal
  - Accept even more routes from each upstream
    - Become less reliant on the default (and lowest neighbour IP address)
  - Etc

Agenda

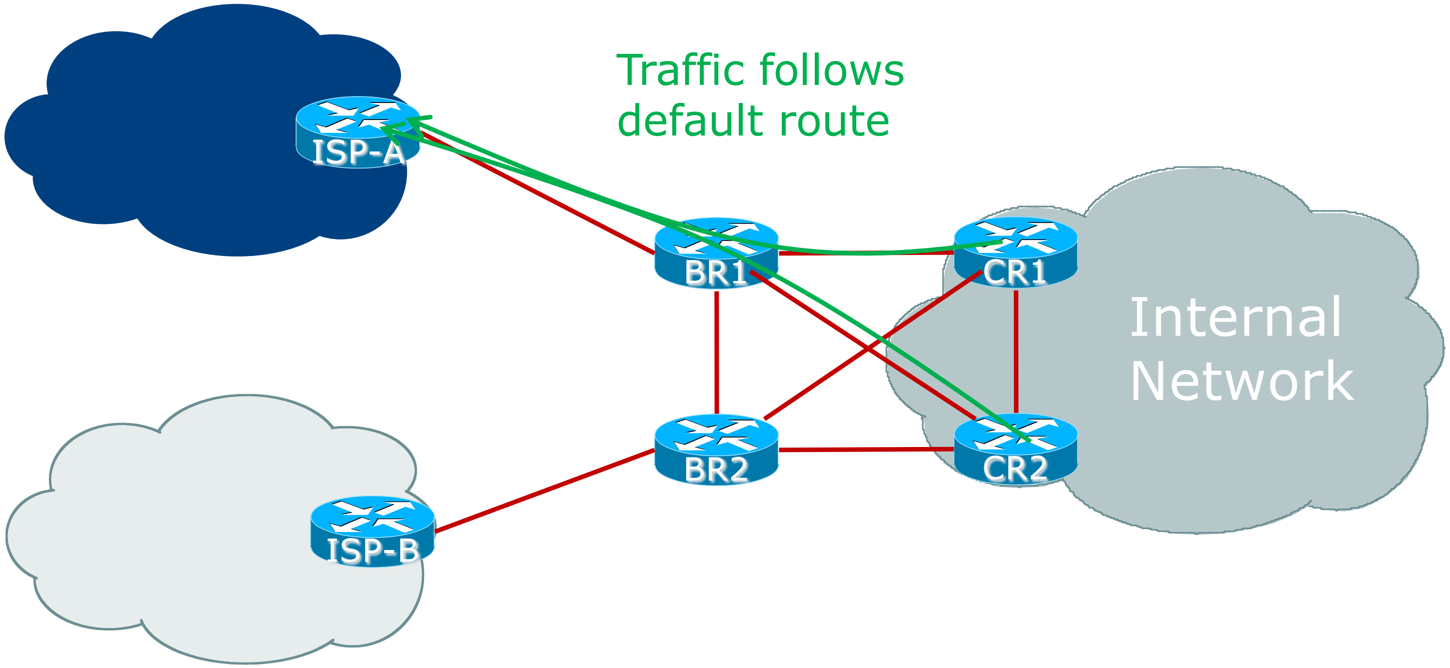
- Background and Requirements
- The next steps
- 1st link primary, 2nd link backup
- Load share between both links
  - Option 1
  - Option 2
  - Option 3
  - Option 4
- End-site network configuration

End-site network configuration

- Up to now, we left end-site network like this:
  - Border routers announce default into core
  - Best path from core to border follows default route
  - Border routers choose best path if more specific information is available
- Is it possible to improve on this?
  - **Yes!**

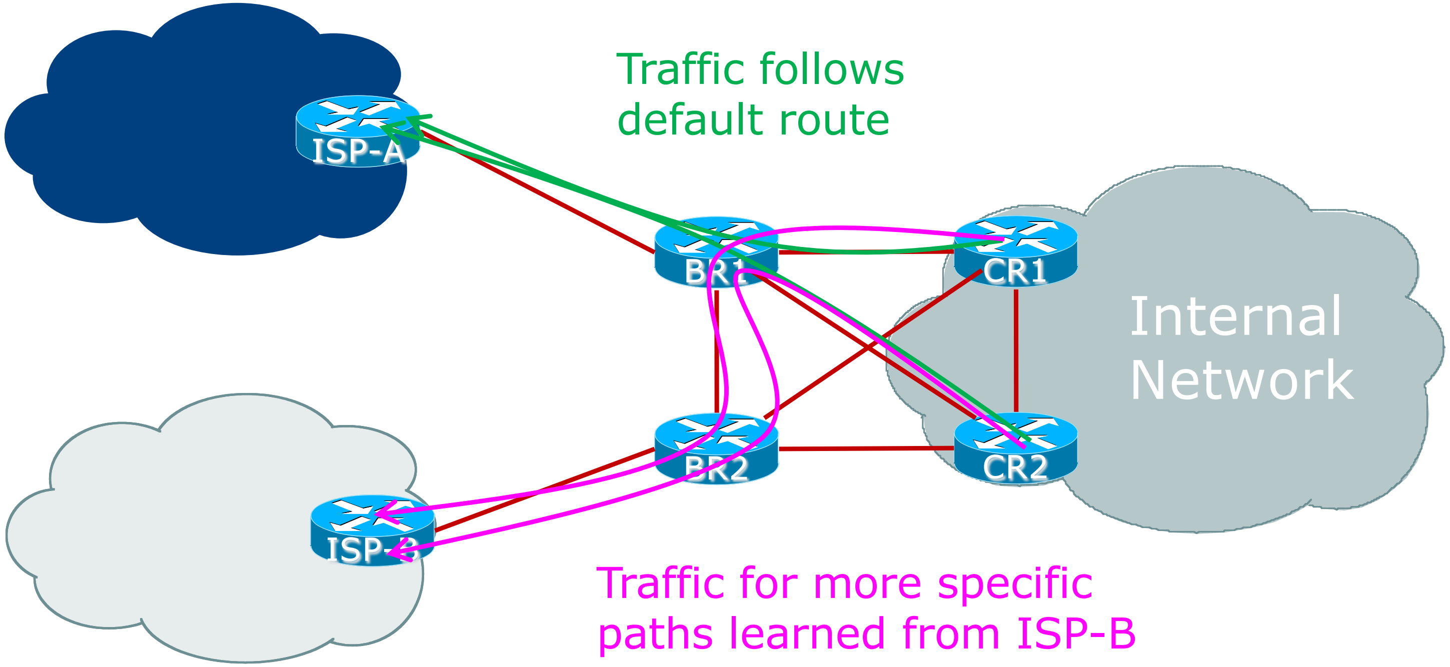
End-site network configuration

- How do the defaults work?
  - If BR1 default is better than BR2 default (according to BGP):



End-site network configuration

- How do the defaults work?
  - If BR1 default better than BR2 default (according to BGP) – most outbound traffic via BR1, unless more specific path learned from BR2



Traffic follows default route

Traffic for more specific paths learned from ISP-B

End-site network configuration

- Notice the CR2 → BR1 → BR2 → ISP-B path for traffic going to specific routes learned from ISP-B
  - We can improve on this
- Would be better to have CR1 and CR2 traffic going directly to BR2 for routes learned from ISP-B
  - To do this:
    - Announce routes by IBGP from border routers to the core routers
    - **Be Careful:** in many enterprises, the core routers are often L3 switches with limited FIB sizes
      - Don’t allow the FIB to overflow otherwise the core devices will behave in very unpredictable ways (and end users will experience random connectivity problems)

End-site network configuration

- Announcing routes to the core:
  - Tag routes learned from upstreams
    - Default gets **`no-advertise`** community
    - Specific accepted routes get an internal community
    - Configure **`maximum-prefix`** on EBGP sessions so that errors don’t break core router FIB
  - Set up IBGP peer-group to allow partial routes from Border to Core
    - Border router as route-reflector, core router as client
  - Originate default route in OSPF/IS-IS
    - No need to carry in IBGP
    - Caters for EBGP session going down and BGP timeout delays

End-site network configuration: BR1

- BR1 BGP Configuration (enhanced from Option 4 earlier)

```
router bgp 64500
 address-family ipv4
  network 100.64.0.0 mask 255.255.255.0
  network 100.64.15.0 mask 255.255.255.0
  neighbor CORE peer-group
  neighbor CORE remote-as 64500
  neighbor CORE route-reflector-client
  neighbor CORE send-community
  neighbor CORE update-source Loopback0
  neighbor 100.64.0.2 remote-as 64500
  neighbor 100.64.0.2 send-community
  neighbor 100.64.0.2 update-source Loopback0
  neighbor 100.64.0.2 description IBGP with BR2
  neighbor 100.64.0.2 activate

...continued...
```

End-site network configuration: BR1

```
  neighbor 100.64.0.3 peer-group CORE
  neighbor 100.64.0.3 description IBGP with CR1
  neighbor 100.64.0.3 activate
  neighbor 100.64.0.4 peer-group CORE
  neighbor 100.64.0.4 description IBGP with CR2
  neighbor 100.64.0.4 activate
  neighbor 100.66.10.1 remote-as 64505
  neighbor 100.66.10.1 description EBGP with ISP-A
  neighbor 100.66.10.1 route-map AS64505-OUT out
  neighbor 100.66.10.1 route-map AS64505-IN in
  neighbor 100.66.10.1 maximum-prefix 3000
  neighbor 100.66.10.1 activate
!
ip route 100.64.0.0 255.255.255.0 null0
ip route 100.64.15.0 255.255.255.0 null0
!

...continued...
```

End-site network configuration: BR1

```
ip prefix-list DEFAULT permit 0.0.0.0/0
!
ip as-path access-list 1 permit ^64505$
ip as-path access-list 1 permit ^64505_[0-9]+$
ip as-path access-list 1 permit ^64505_[0-9]+_[0-9]+$
!
route-map AS64505-IN permit 10
 description Accept default
 match ip address prefix-list DEFAULT
 set community no-advertise
!
route-map AS64505-IN permit 20
 description Accept AS64505 originated routes
 match as-path access-list 1
!
route-map AS64505-IN deny 30
 description Drop everything else
!
```

End-site network configuration: BR1

- BR1 OSPF Configuration
  - **`default-originate`** originates a default within OSPF if it exists in the Global RIB (i.e. from BGP)
  - The **`metric 10`** sets the metric to be 10 on the default route
- Similar concept exists for IS-IS if that is the IGP of choice

```
interface Gigabit 0/0
 description Link to ISP-A
 ip address 100.66.10.2 255.255.255.252
!
interface Gigabit 1/0
 description Link to CR1
 ip address 100.64.0.129 255.255.255.252
 ip ospf network point-to-point
 ip ospf 64500 area 0
!
<similar for Gigabit 2/0 & 3/0 for CR2 and BR2>
!
router ospf 64500
 passive-interface default
 no passive-interface Gigabit 1/0
 no passive-interface Gigabit 2/0
 no passive-interface Gigabit 3/0
 default-information originate metric 10
!
```

End-site network configuration

- Similar configuration applies on BR2
- Configuration Notes:
  - OSPF
    - Can set metric for the default route announced from BR2 to be the same as for BR1, but better to make it a higher value so that it is a backup
    - Can set cost on the BR2-CR2 and BR2-CR1 links to be higher than the BR1-CR1 and BR1-CR2 links so that default route followed is always to BR1 first
  - IBGP
    - As infrastructure scales, putting route filters on the IBGP session to protect core routers with limited FIB sizes is also an option
    - If full BGP table taken from both upstreams, then IBGP filters strongly recommended

Summary

- Presentation has examined:
  - Minimum requirements for multihoming
  - The preparations needed
  - Options available for small end-sites
  - End-site core network configuration suggestions

Multihoming: Practical Deployment

ISP Workshops