Border Gateway Protocol Introduction

Scalable Infrastructure Workshop

AfNOG 2012

Border Gateway Protocol (BGP4)

- Part 0: Why use BGP?
- Part 1: Forwarding and Routing (review)
- Part 2: Interior and Exterior Routing
- Part 3: BGP Building Blocks
- Part 4: Configuring BGP
- Part 5: Introducing IPv6

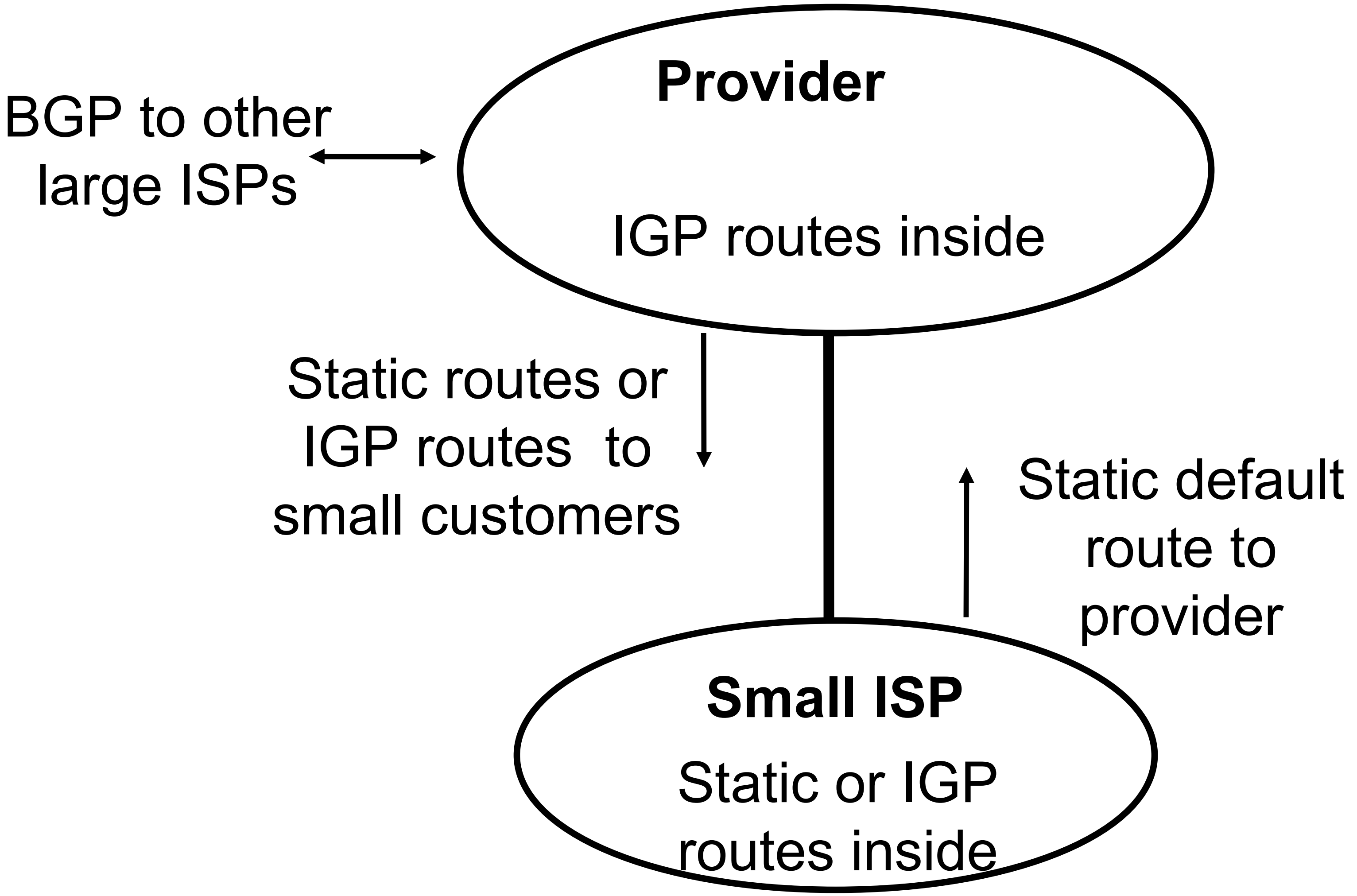
BGP Part 0

Why use BGP?

Consider a typical small ISP

- Local network in one country
- May have multiple POPs in different cities
- Line to Internet
  - International line providing transit connectivity
  - Very, very expensive international line
- Doesn’t yet need BGP

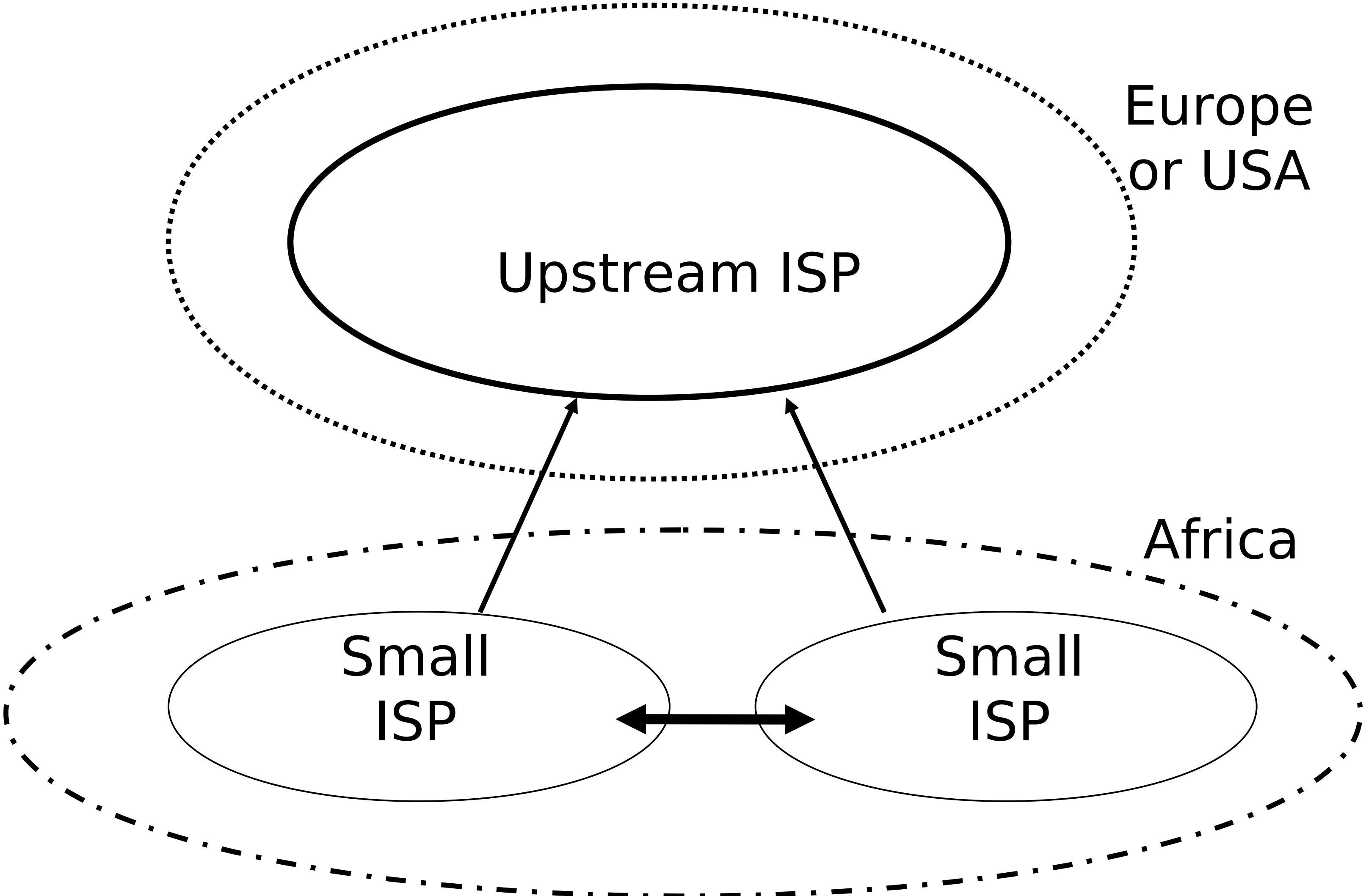
Small ISP with one upstream provider


BGP to other large ISPs
Provider
IGP routes inside
Static routes or IGP routes to small customers
Static default route to provider
Small ISP
Static or IGP routes inside

What happens with other ISPs in the same country

- Similar setup
- Traffic between you and them goes over
  - Your expensive line
  - Their expensive line
- Traffic can be significant
  - Your customers want to talk to their customers
  - Same language/culture
  - Local email, discussion lists, web sites

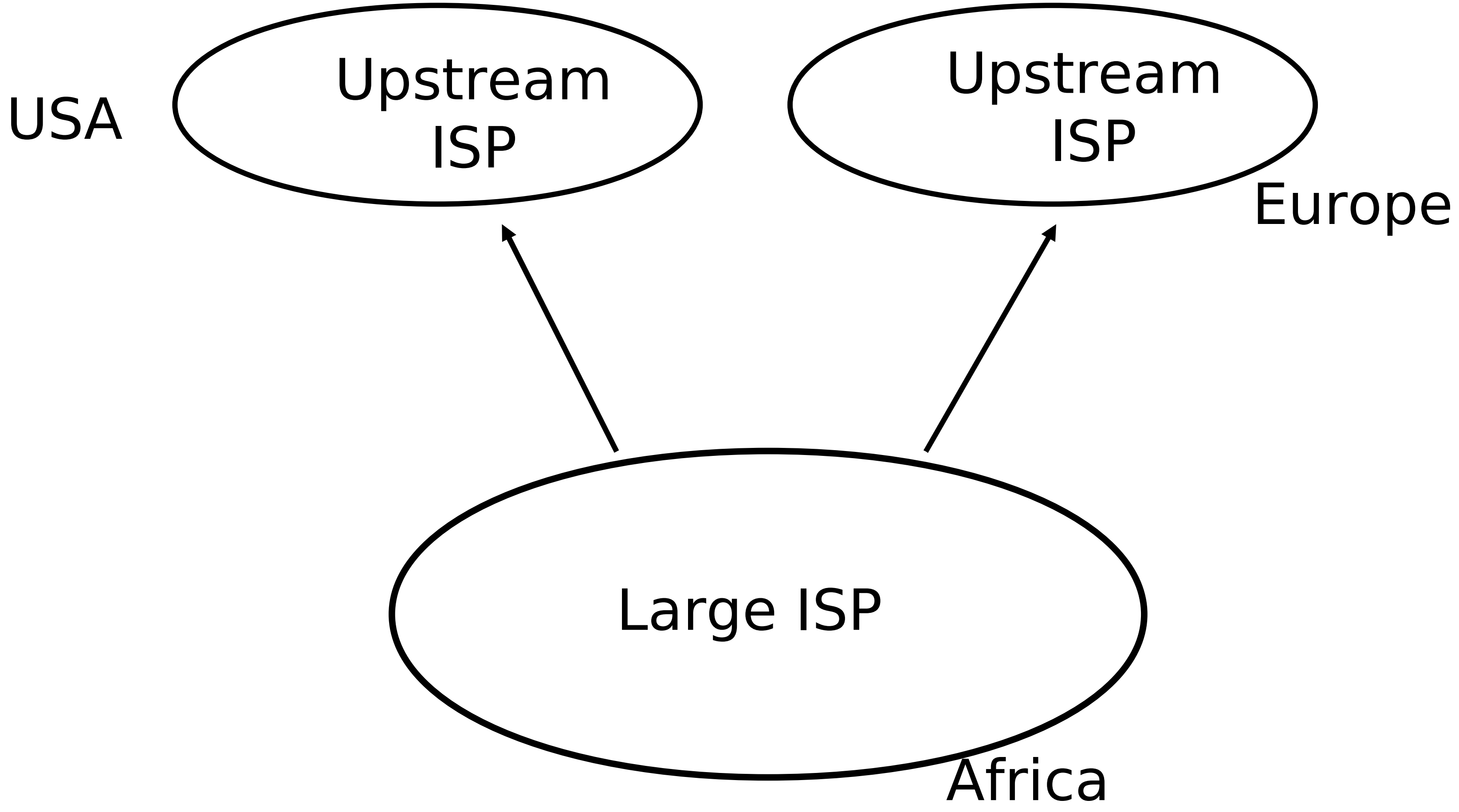
Keeping Local Traffic Local



Consider a larger ISP with multiple upstreams

- Large ISP multi-homes to two or more upstream providers
  - multiple connections
  - to achieve:
    - redundancy
    - connection diversity
    - increased speeds
  - Use BGP to choose a different upstream for different destination addresses

A Large ISP with more than one upstream provider



Terminology: “Policy”

- Where do you want your traffic to go?
  - It is difficult to get what you want, but you can try
- Control of how you accept and send routing updates to neighbours
  - Prefer cheaper connections
  - Prefer connections with better latency
  - Load-sharing, etc

“Policy” (continued)

- Implementing policy:
  - Accepting routes from some ISPs and not others
  - Sending some routes to some ISPs and not to others
  - Preferring routes from some ISPs over those from other ISPs

“Policy” Implementation

- You want to use a local line to talk to the customers of other local ISPs
  - local peering
- You do not want other local ISPs to use your expensive international lines
  - no free transit!
- So you need some sort of control over routing policies
- BGP can do this

Terminology: “Peering” and “Transit”

- **Peering**: getting connectivity to the network of other ISPs
  - … and just that network, no other networks
  - Usually at zero cost (zero-settlement)
- **Transit**: getting connectivity though the other ISP to other ISP networks
  - … getting connectivity to rest of world (or part thereof)
  - Usually with a monetary cost (customer-provider relationship)

Terminology: “Aggregation”

- Combining of several smaller blocks of address space into a larger block
- For example:
  - 192.168.4.0/24 and 192.168.5.0/24 are contiguous address blocks
  - They can be combined and represented as 192.168.4.0/23…
  - …with no loss of information!

“Aggregation” (continued)

- Useful because it hides detailed information about the local network:
  - The outside world needs to know about the range of addresses in use
  - The outside world does **not** need to know about the small pieces of address space used by different customers inside your network

“Aggregation” (continued)

- Aggregation is very necessary when using BGP to “talk” to the Internet
- Saves memory and CPU for routing calculations
- Saves bandwidth for BGP announcements
- Some providers may filter (ignore) de-aggregated route announcements

- But deaggregation may be useful for traffic engineering

Summary:
Why do I need BGP?

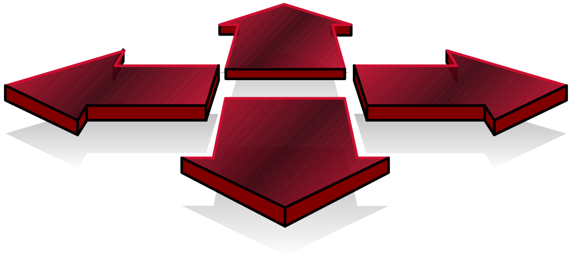
- Multi-homing – connecting to multiple providers
  - upstream providers
  - local networks – regional peering to get local traffic
- Policy discrimination
  - controlling how traffic flows
  - do not accidentally provide transit to non-customers

BGP Part 1

Forwarding and Routing

Routing versus Forwarding

- Routing = building maps and giving directions
- Forwarding = moving packets between interfaces according to the "directions"



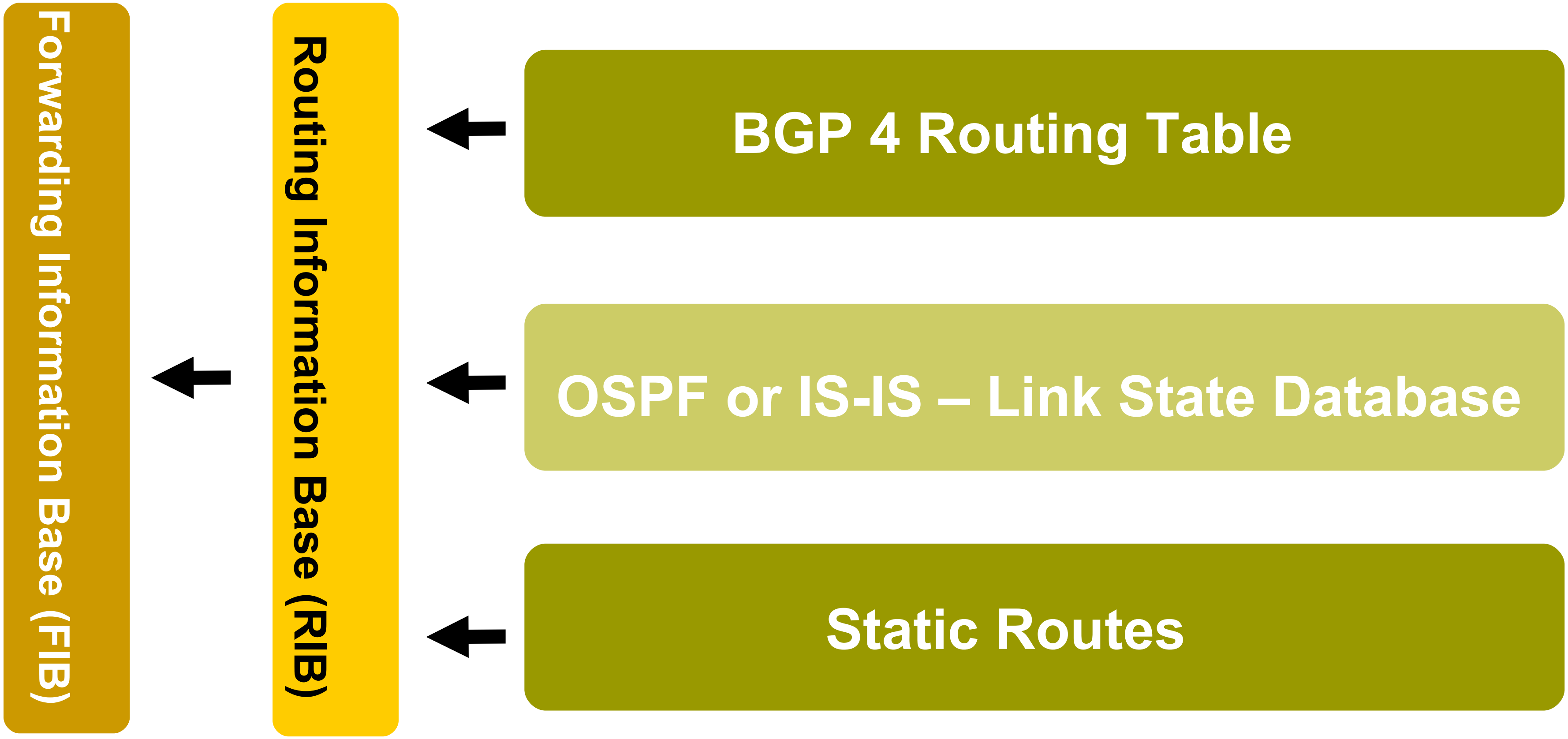
Routing Table/RIB

- Routing table is managed by a routing protocol (e.g. OSPF or BGP)
- Often called the RIB – Routing Information Base
- Each routing protocol has its own way of managing its own routing tables
- Each routing protocol has a way of exchanging information between routers using the same protocol

Forwarding Table/FIB

- Forwarding table determines how packets are sent through the router
- Often called the FIB – Forwarding Information Base
- Made from routing table built by routing protocols
  - Best routes from routing tables are installed
- Performs the lookup to find next-hop and outgoing interface
- Switches the packet with new encapsulation as per the outgoing interface

Routing Tables Feed the Forwarding Table



Forwarding Information Base (FIB)

Routing Information Base (RIB)

BGP 4 Routing Table

OSPF or IS-IS – Link State Database

Static Routes

IP Routing

- Each router or host makes its own routing decisions
- Sending machine does not have to determine the entire path to the destination
- Sending machine just determines the next-hop along the path (based on destination IP address)
  - This process is repeated until the destination is reached, or there's an error
- Forwarding table is consulted (at each hop) to determine the next-hop

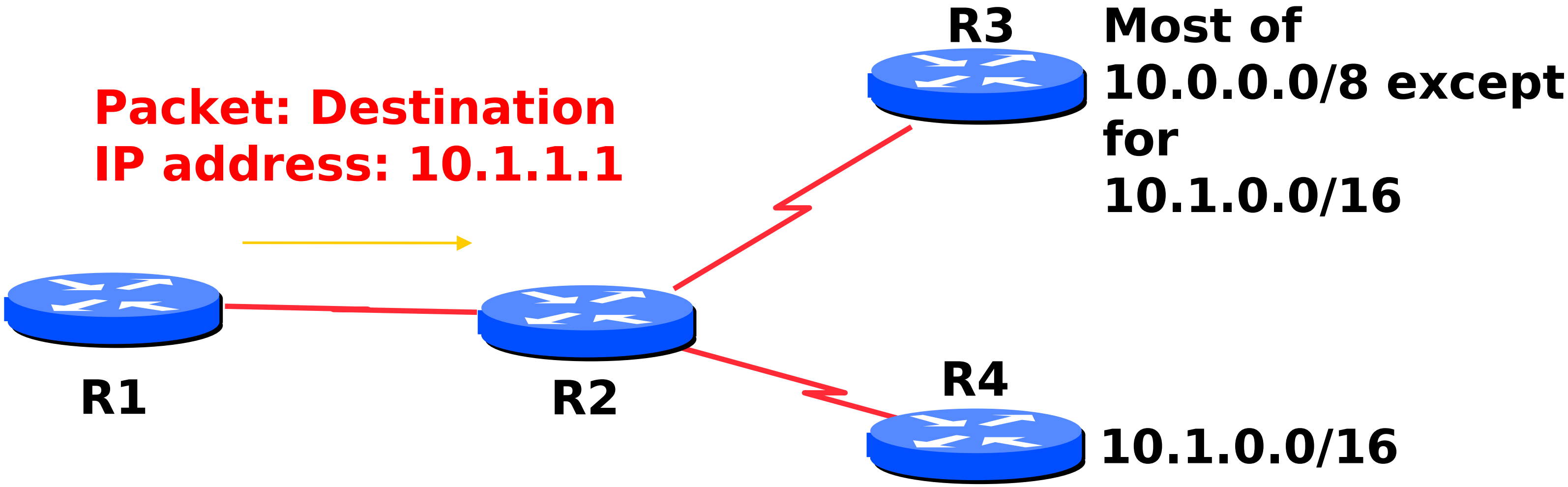
IP Routing

- Classless routing
  - route entries include
    - destination
    - next-hop
    - mask (prefix-length) indicating size of address space described by the entry
- Longest match
  - for a given destination, find longest prefix match in the routing table
  - example: destination is 35.35.66.42
    - routing table entries are 35.0.0.0/8, 35.35.64.0/19 and 0.0.0.0/0
    - All these routes match, but the /19 is the longest match

IP routing

- Default route
  - where to send packets if there is no entry for the destination in the routing table
  - most machines have a single default route
  - often referred to as a default gateway
  - 0.0.0.0/0
    - matches all possible destinations, but is usually not the longest match

IP route lookup: Longest match routing



R3

Most of 10.0.0.0/8 except for 10.1.0.0/16

Packet: Destination IP address: 10.1.1.1

R1

R2

R4

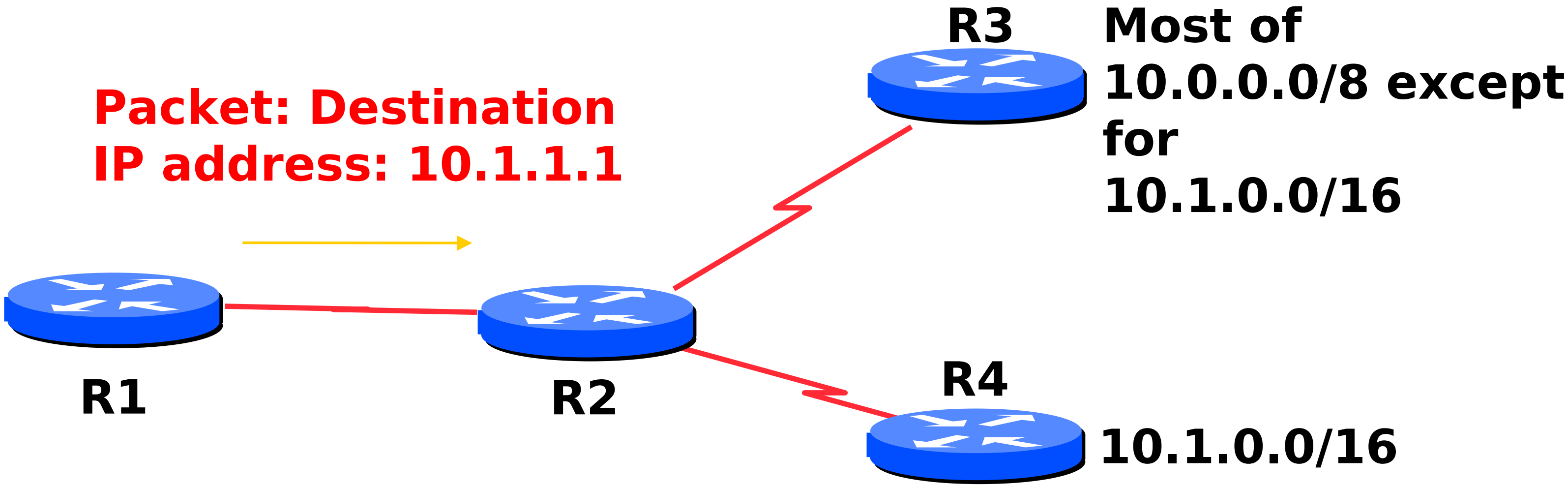
10.1.0.0/16

Based on destination IP address

R2’s IP forwarding table

10.0.0.0/8 → R3
10.1.0.0/16 → R4
20.0.0.0/8 → R5
0.0.0.0/0 → R1

IP route lookup: Longest match routing



R3

Most of 10.0.0.0/8 except for 10.1.0.0/16

Packet: Destination IP address: 10.1.1.1

R1

R2

R4

10.1.0.0/16

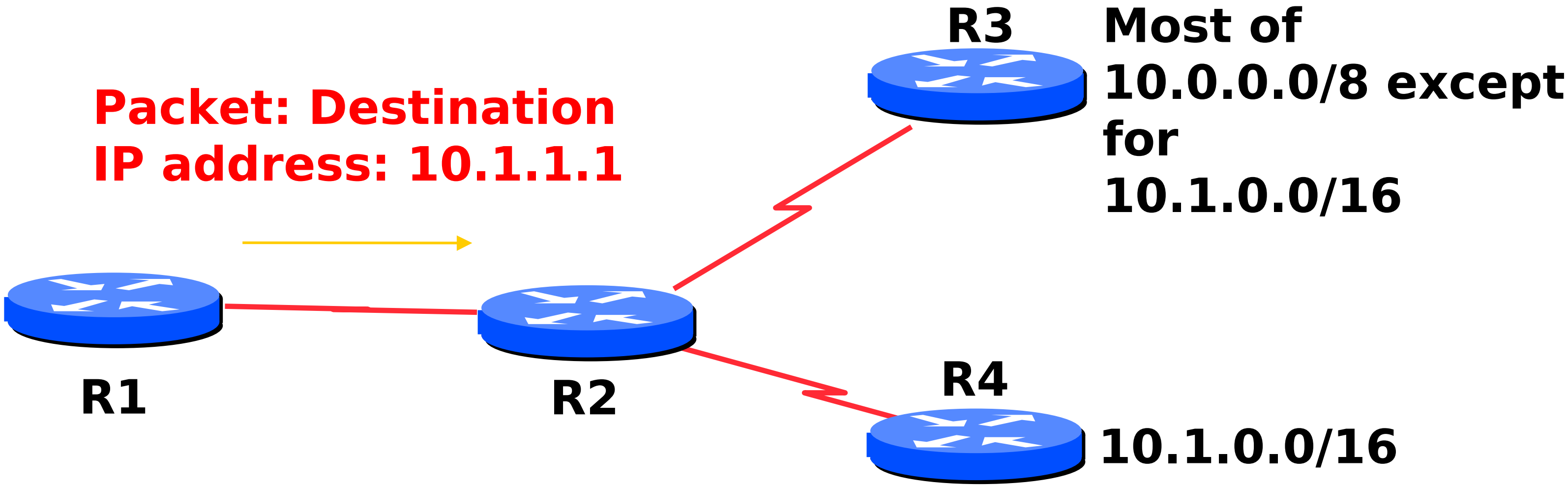
Based on destination IP address

R2’s IP forwarding table

10.0.0.0/8 → R3
10.1.0.0/16 → R4
20.0.0.0/8 → R5
0.0.0.0/0 → R1

10.1.1.1 & FF.00.00.00
vs.
10.0.0.0 & FF.00.00.00
Match! (length 8)

IP route lookup:
Longest match routing



Packet: Destination IP address: 10.1.1.1

Most of 10.0.0.0/8 except for 10.1.0.0/16

10.1.0.0/16

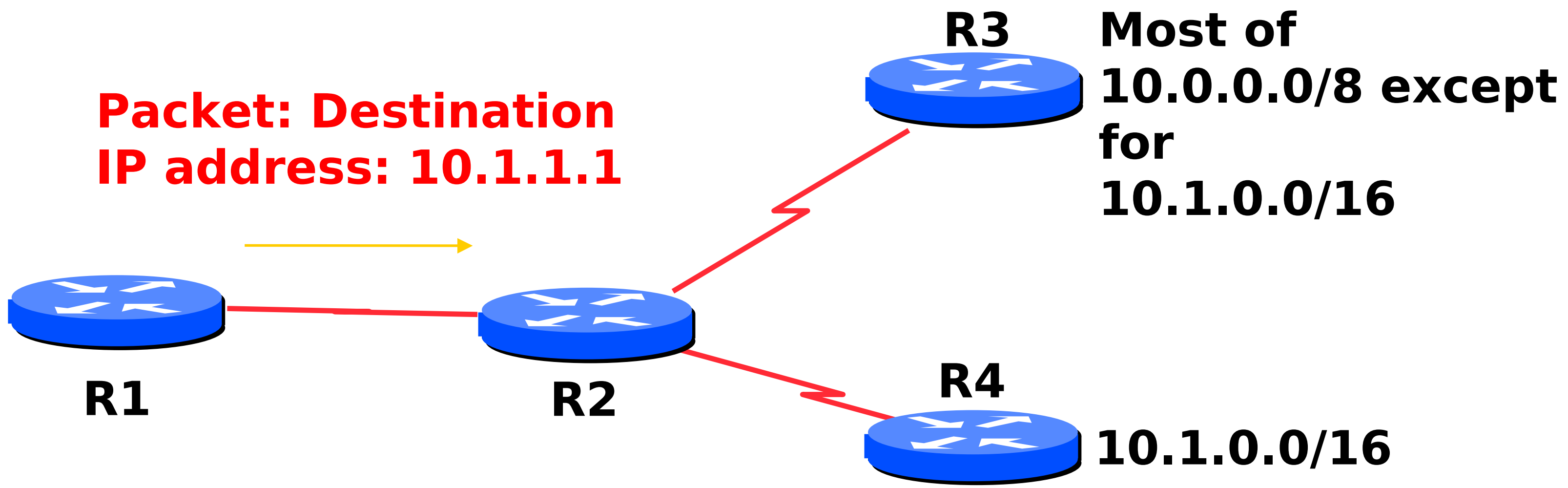
Based on destination IP address

R2's IP forwarding table

10.0.0.0/8 → R3
10.1.0.0/16 → R4
20.0.0.0/8 → R5
0.0.0.0/0 → R1

10.1.1.1 & FF.FF.00.00
vs.
10.1.0.0 & FF.FF.00.00
Match! (length 16)

IP route lookup: Longest match routing


R3
Most of 10.0.0.0/8 except for 10.1.0.0/16
Packet: Destination IP address: 10.1.1.1
R1
R2
R4
10.1.0.0/16

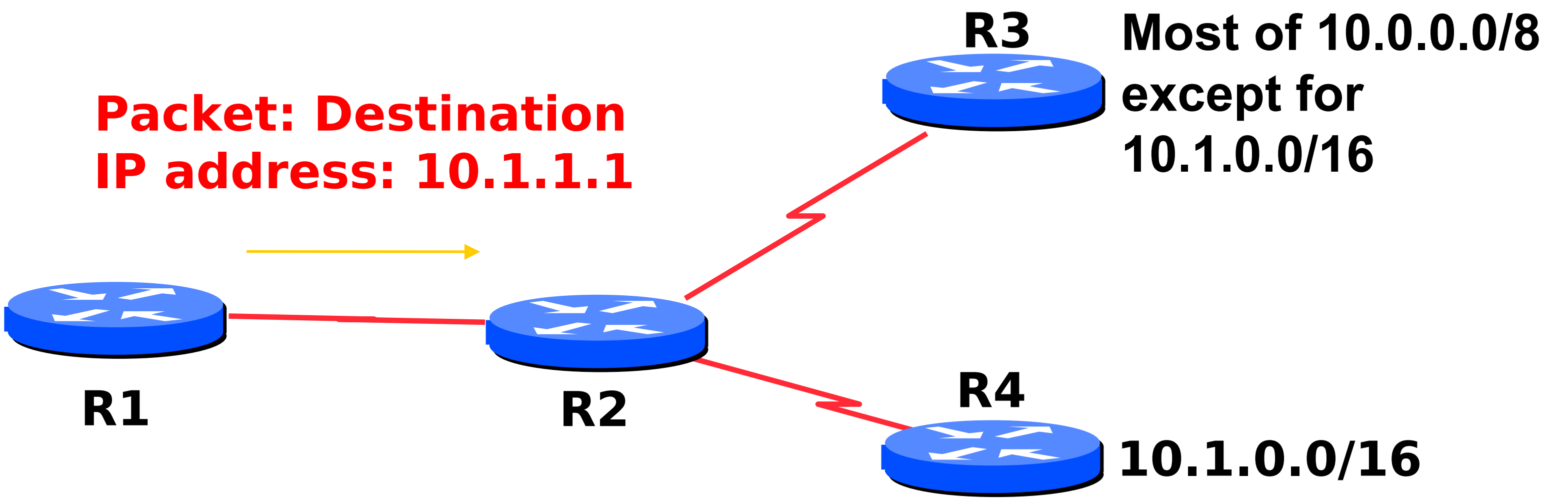
Based on destination IP address

R2’s IP forwarding table

10.0.0.0/8 → R3
10.1.0.0/16 → R4
20.0.0.0/8 → R5
0.0.0.0/0 → R1

10.1.1.1 & FF.00.00.00
vs.
20.0.0.0 & FF.00.00.00
No Match!

IP route lookup: Longest match routing



Packet: Destination IP address: 10.1.1.1

Most of 10.0.0.0/8 except for 10.1.0.0/16

10.1.0.0/16

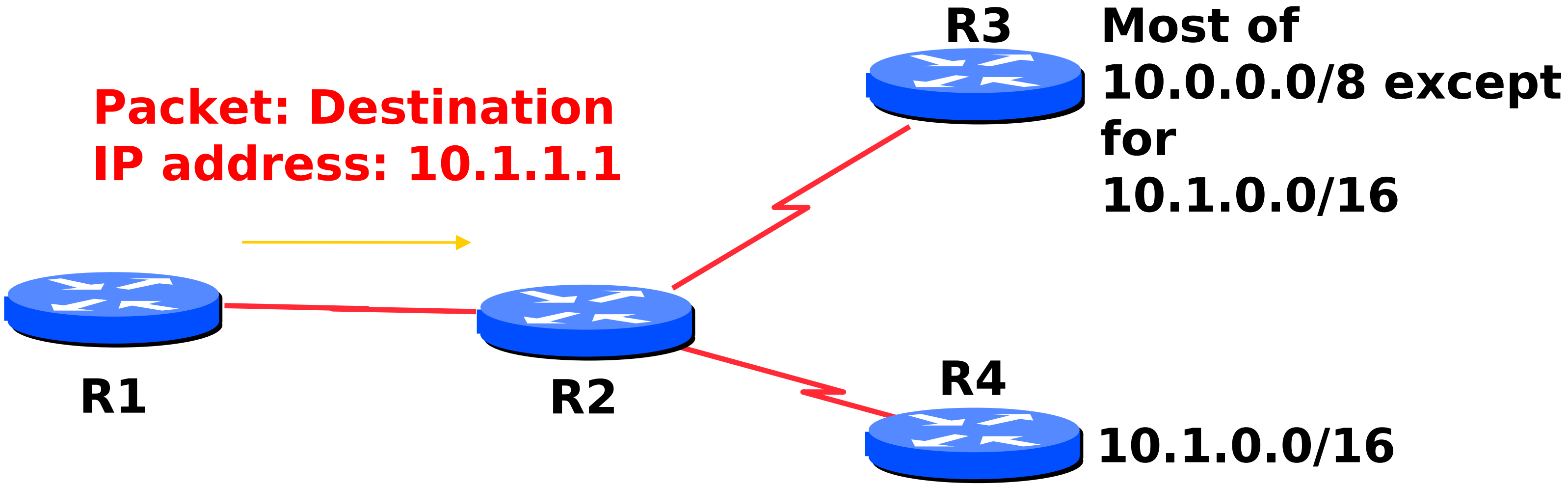
Based on destination IP address

R2’s IP forwarding table

10.0.0.0/8 → R3
10.1.0.0/16 → R4
20.0.0.0/8 → R5
0.0.0.0/0 → R1

10.1.1.1 & 00.00.00.00
vs.
0.0.0.0 & 00.00.00.00
Match! (length 0)

IP route lookup: Longest match routing



Packet: Destination IP address: 10.1.1.1

Most of 10.0.0.0/8 except for 10.1.0.0/16

10.1.0.0/16

Based on destination IP address

R2's IP forwarding table

10.0.0.0/8 → R3
10.1.0.0/16 → R4
20.0.0.0/8 → R5
0.0.0.0/0 → R1

This is the longest matching prefix (length 16). "R2" will send the packet to "R4".

IP route lookup: Longest match routing

- Most specific/longest match always wins!!
  - Many people forget this, even experienced ISP engineers
- Default route is 0.0.0.0/0
  - Can handle it using the normal longest match algorithm
  - Matches everything. Always the shortest match.

Static vs. Dynamic routing

- Static routes
  - Set up by administrator
  - Changes need to be made by administrator
  - Only good for small sites and star topologies
  - Bad for every other topology type
- Dynamic routes
  - Provided by routing protocols
  - Changes are made automatically
  - Good for network topologies which have redundant links (most!)

Dynamic Routing

- Routers compute routing tables dynamically based on information provided by other routers in the network
- Routers communicate topology to each other via different protocols
- Routers then compute one or more next hops for each destination – trying to calculate the most optimal path
- Automatically repairs damage by choosing an alternative route (if there is one)

BGP Part 2

Interior and Exterior Routing

Interior vs. Exterior Routing Protocols

- Interior gateway protocol (IGP)
  - Automatic neighbour discovery
  - Under control of a single organisation
  - Generally trust your IGP routers
  - Routes go to all IGP routers
  - Usually not filtered

- Exterior gateway protocol (EGP)
  - Specifically configured peers
  - Connecting with outside networks
  - Neighbours are not trusted
  - Set administrative boundaries
  - Filters based on policy

IGP

- Interior Gateway Protocol
- Within a network/autonomous system
- Carries information about internal prefixes
- Examples – OSPF, ISIS, EIGRP, RIP

EGP

- Exterior Gateway Protocol
- Used to convey routing information between networks/ASes
- De-coupled from the IGP
- Current EGP is BGP4

Why Do We Need an EGP?

- Scaling to large network
  - Hierarchy
  - Limit scope of failure
- Define administrative boundary
- Policy
  - Control reachability to prefixes

Scalability and policy issues

- Just getting direct line is not enough
- Need to work out how to do routing
  - Need to get local traffic between ISP’s/peers
  - Need to make sure the peer ISP doesn’t use us for transit
  - Need to control what networks to announce, what network announcements to accept to upstreams and peers

Scalability: Not using static routes

- **`ip route their_net their_gw`**
- Does not scale
- Millions of networks around the world

Scalability: Not using IGP (OSPF/ISIS)

- Serious operational consequences:
  - If the other ISP has a routing problem, you will have problems too
  - Your network prefixes could end up in the other ISP’s network — and vice-versa
  - Very hard to filter routes so that we don’t inadvertently give transit

Using BGP instead

- BGP = Border Gateway Protocol
- BGP is an **exterior** routing protocol
- Focus on routing **policy**, not topology
- BGP can make ‘groups’ of networks (Autonomous Systems)
- Good route filtering capabilities
- Ability to isolate from others' problems

Border Gateway Protocol

- A Routing Protocol used to exchange routing information between networks
  - exterior gateway protocol
- Described in RFC4271
  - RFC4276 gives an implementation report on BGP-4
  - RFC4277 describes operational experiences using BGP-4
- The Autonomous System is BGP’s fundamental operating unit
  - It is used to uniquely identify networks with a common routing policy

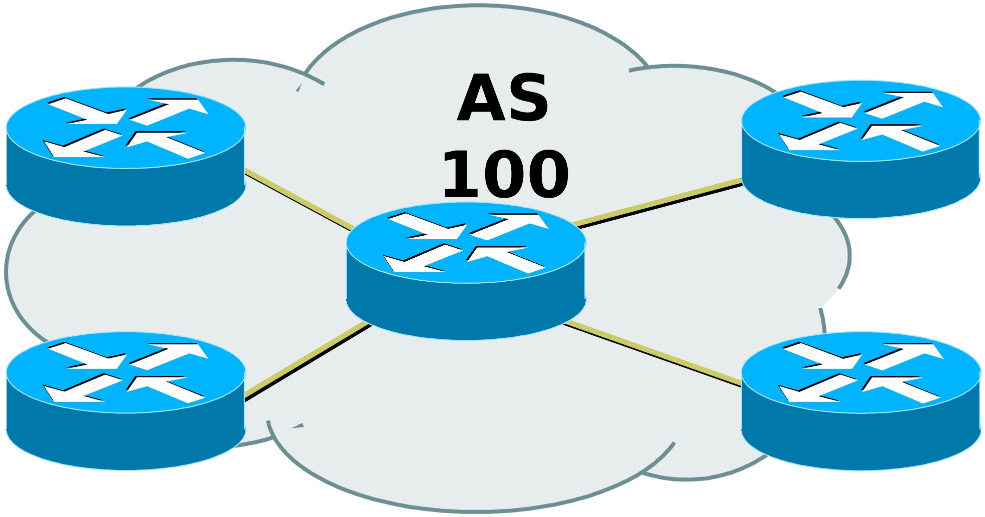
BGP Part 3

BGP Building Blocks

BGP Building Blocks

- Autonomous System (AS)
- Types of Routes
- IGP/EGP
- DMZ
- Policy
- Egress
- Ingress

Autonomous System (AS)



- Collection of networks with same policy
- Single routing protocol
- Usually under single administrative control
- IGP to provide internal connectivity

Autonomous System (AS)

- Autonomous systems is a misnomer
  - Not much to do with freedom, independence, …
- Just a handle for a group of networks that is under the same administrative control
- Identified by an AS number

Autonomous System (AS)

- Identified by ‘AS number’
  - example: AS16907 (ISPKenya)
- Examples:
  - Service provider
  - Multi-homed customers
  - Anyone needing policy discrimination for networks with different routing policies
- Single-homed network (one upstream provider) does not need an AS number
  - Treated like part of upstream AS

Autonomous System Number (ASN)

- Two ranges
  - 0-65535 (original 16-bit range)
  - 65536-4294967295 (32-bit range - RFC4893)
- Usage:
  - 0 and 65535 (reserved)
  - 1-64495 (public Internet)
  - 64496-64511 (documentation - RFC5398)
  - 64512-65534 (private use only)
  - 23456 (represent 32-bit range in 16-bit world)
  - 65536-65551 (documentation - RFC5398)
  - 65552-4294967295 (public Internet)
- 32-bit range representation specified in RFC5396
  - Defines “asplain” (traditional format) as standard notation

Configuring BGP in IOS

- This command enables BGP in IOS for AS100:
  - `router bgp 100`
- For ASNs > 65535, the AS number can be entered in either plain notation, or in dot notation:

```
router bgp 131076
```

or

```
router bgp 2.4
```

- IOS will display ASNs in plain notation by default
  - Dot notation is optional:

```
router bgp 2.4
 bgp asnotation dot
```

Router Support for 4-byte ASNs

- Most vendors now support 4-byte ASNs in their routing software
- A complete list is at:
  - http://as4.cluepon.net

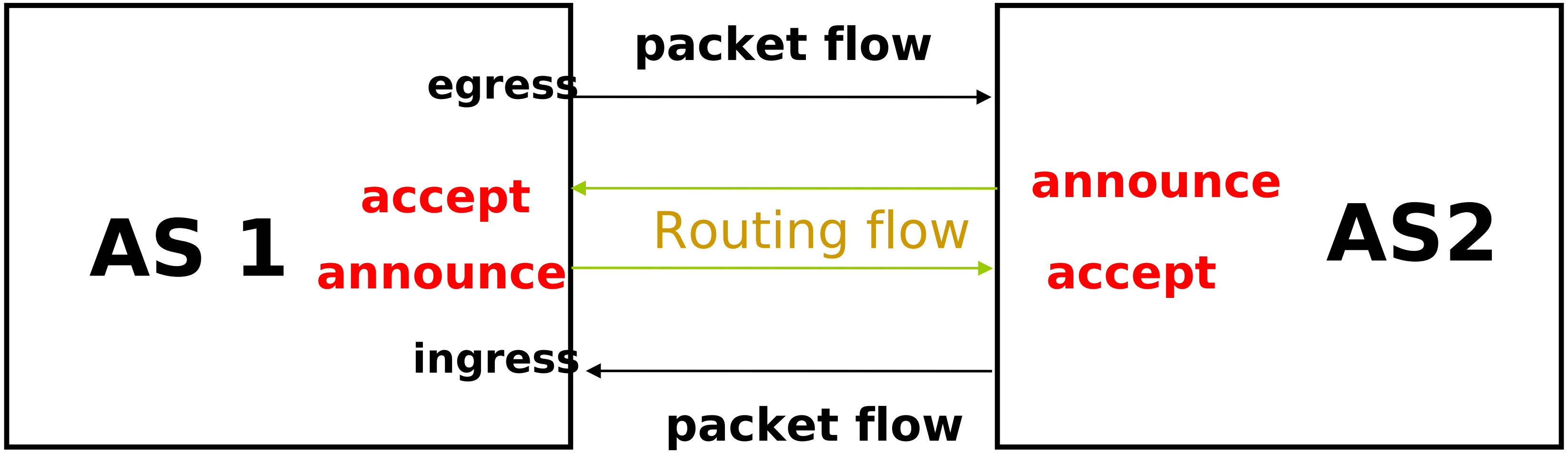
Autonomous System Number (ASN)

- ASNs are distributed by the Regional Internet Registries
  - They are also available from upstream ISPs who are members of one of the RIRs
- Current 16-bit ASN allocations up to 55295 have been made to the RIRs
  - Around 34200 are visible on the Internet (as of 2009)
- The RIRs also have received 1024 32-bit ASNs each
  - 580 have been assigned, but only 100 are visible on the Internet (as of 2009) (early adopters)
- See www.iana.org/assignments/as-numbers

Using AS numbers

- BGP can filter on AS numbers
  - Get all networks of the other ISP using one handle
  - Include future new networks without having to change routing filters
    - AS number for new network will be same
  - Can use AS numbers in filters with regular expressions
- BGP actually does routing computation on IP numbers

Routing flow and packet flow



- For networks in AS1 and AS2 to communicate:
  - AS1 must announce routes to AS2
  - AS2 must accept routes from AS1
  - AS2 must announce routes to AS1
  - AS1 must accept routes from AS2

Egress Traffic

- Packets exiting the network
- Based on:
  - Route availability (what others send you)
  - Route acceptance (what you accept from others)
  - Policy and tuning (what you do with routes from others)
  - Peering and transit agreements

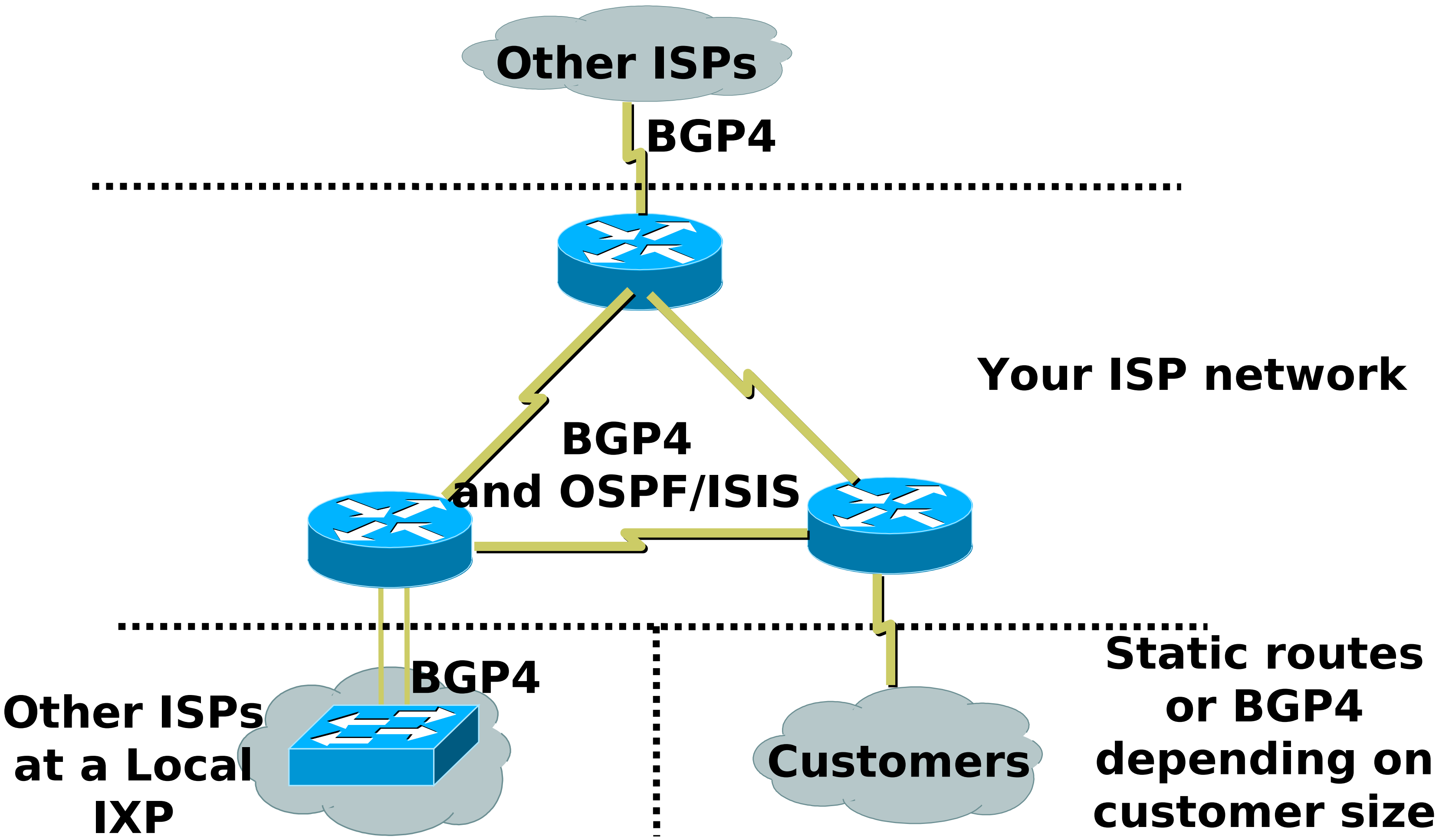
Ingress Traffic

- Packets entering your network
- Ingress traffic depends on:
  - What information you send and to whom
  - Based on your addressing and ASes
  - Based on others’ policy (what they accept from you and what they do with it
  - Difficult for you to influence what others choose to do, so load balancing ingress traffic is difficult

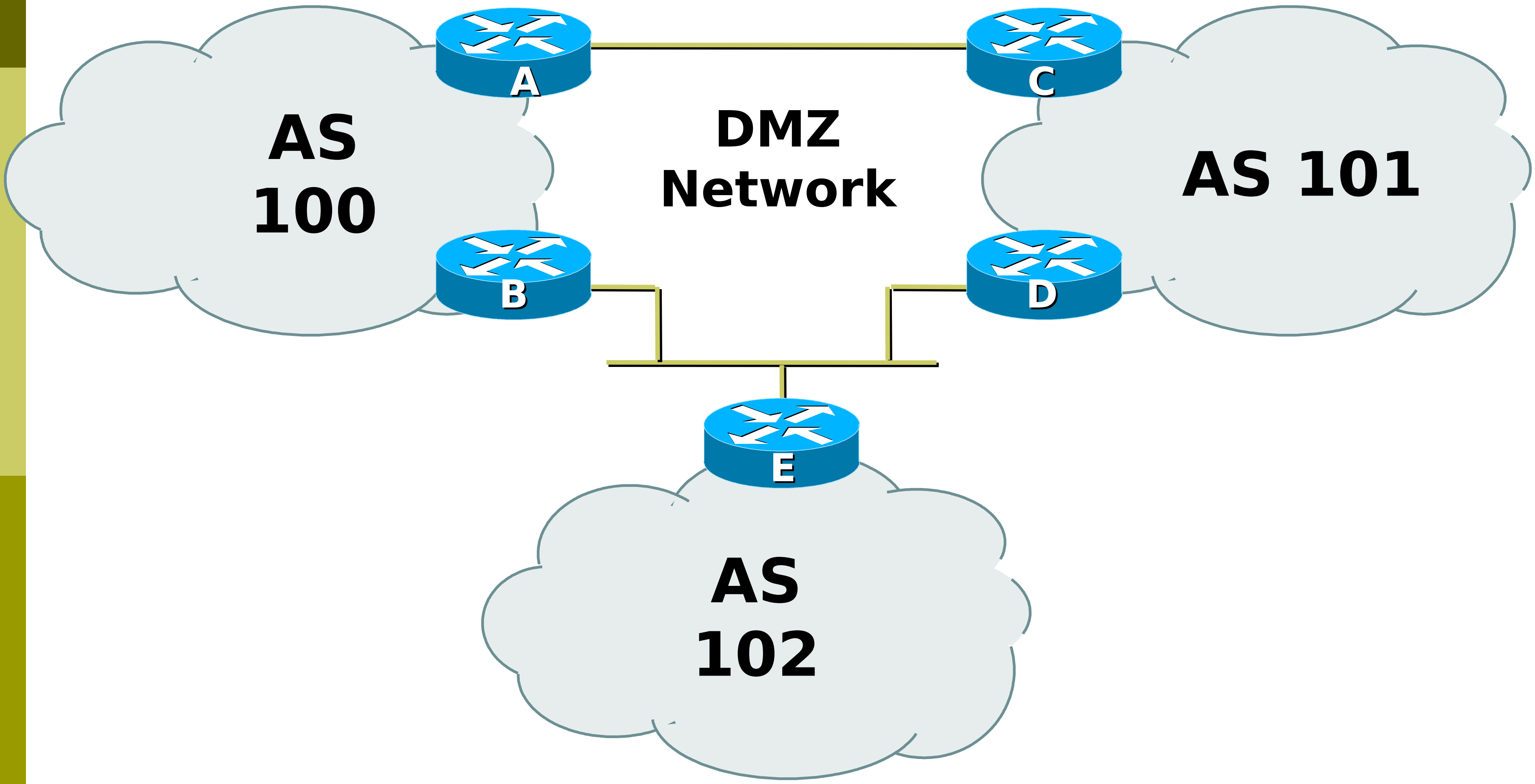
Types of Routes

- Static Routes
  - configured manually
- Connected Routes
  - created automatically when an interface is ‘up’
- Interior Routes
  - Routes within an AS
  - learned via IGP (e.g. OSPF)
- Exterior Routes
  - Routes exterior to AS
  - learned via EGP (e.g. BGP)

Hierarchy of Routing Protocols



DMZ (De-Militarised Zone)



- DMZ network is shared between ASes

Basics of a BGP route

- Seen from output of “show ip bgp”
- Prefix and mask — what IP addresses are we talking about?
  - 192.168.0.0/16 or 192.168.0.0/255.255.0.0
- Origin – How did the route originally get into BGP?
  - “?” — incomplete, “e” — EGP, “i” — IGP
- AS Path – what ASes did the route go through before it got to us?
  - “701 3561 1”

BGP Part 4

Configuring BGP

Basic commands

Getting routes into BGP

Basic BGP commands

- Configuration commands

```
router bgp <AS-number>
 no auto-summary
 no synchronization
 neighbor <ip address> remote-as <as-number>
```

- Show commands

```
show ip bgp summary
show ip bgp neighbors
show ip bgp neighbor <ip address>
```

Configuring BGP with 4-byte ASNs

- If both peers support 4-byte ASNs, configuration is as per previously, just larger numbers
- Old software sees AS23456 instead of the actual 4-byte ASN
- 23456 is visible in the output from “show ip bgp” if the actual ASN is 32-bits
- Also use 23456 in the configuration “neighbor ..... remote-as 23456”

Configuring BGP with 4-byte ASNs

- If one peer only supports 2-byte ASNs, use AS23456 as the transition AS

Transition ASN as Router A cannot configure AS70000 directly

- Router A:

```
router bgp 100
 neighbor 1.1.1.1 remote-as 23456
```

- Router B:

```
router bgp 70000
 neighbor 1.1.1.2 remote-as 100
```

Inserting prefixes into BGP

- Two main ways to insert prefixes into BGP
  - network command
  - redistribute static
- Both require the prefix to be in the routing table
  - You can't add a prefix to BGP unless the prefix is also in some other routing protocol (e.g. static route, IS-IS, OSPF)

“network” command

- Configuration Example

```
router bgp 1
 network 105.32.4.0 mask 255.255.254.0
ip route 105.32.4.0 255.255.254.0 serial 0
```

- Matching route must exist in the routing table before network is announced!
- Prefix will have Origin code set to “IGP”

“redistribute static”

- Configuration Example:

```
router bgp 1
 redistribute static
ip route 105.32.4.0 255.255.254.0 serial0
```

- Static route must exist before redistribute command will work
- Forces origin to be “incomplete”
- Care required!
  - This will redistribute all static routes into BGP
  - **Redistributing without using a filter is dangerous; you could accidentally get many unwanted routes**

“redistribute static”

- Care required with redistribution
  - redistribute <routing-protocol> means everything in the <routing-protocol> will be transferred into the current routing protocol
  - will not scale if uncontrolled
  - best avoided if at all possible
  - redistribute normally used with “route-maps” and under tight administrative control
    - “route-map” is used to apply policies in BGP, so is a kind of filter

Aggregates and Null0

- Remember: matching route must exist in routing table before it will be announced by BGP

```
router bgp 1
 network 105.32.0.0 mask 255.255.0.0
ip route 105.32.0.0 255.255.0.0 null0 250
```

- Static route to null0 often used for aggregation
  - Packets will be sent here if there is no more specific match in the routing table
  - Distance of 250 ensures last resort
- Often used to nail up routes for stability
  - Can’t flap! ☺

BGP Part 5

Introducing IPv6

Adding IPv6 to BGP…

- RFC4760
  - Defines Multi-protocol Extensions for BGP4
  - Enables BGP to carry routing information of protocols other than IPv4
    - e.g. MPLS, IPv6, Multicast etc
  - Exchange of multiprotocol NLRI must be negotiated at session startup
- RFC2545
  - Use of BGP Multiprotocol Extensions for IPv6 Inter-Domain Routing

RFC4760

- New optional and non-transitive BGP attributes:
  - MP_REACH_ NLRI (Attribute code: 14)
    - Carry the set of reachable destinations together with the next-hop information to be used for forwarding to these destinations (RFC4760)
  - MP_UNREACH_NLRI (Attribute code: 15)
    - Carry the set of unreachable destinations
- Attribute contains one or more Triples:
  - AFI Address Family Information
  - Next-Hop Information (must be of the same address family)
  - NLRI Network Layer Reachability Information

RFC2545

- IPv6 specific extensions
  - Scoped addresses: Next-hop contains a global IPv6 address and/or potentially a link-local address
  - NEXT_HOP and NLRI are expressed as IPv6 addresses and prefix
  - Address Family Information (AFI) = 2 (IPv6)
    - Sub-AFI = 1 (NLRI is used for unicast)
    - Sub-AFI = 2 (NLRI is used for multicast RPF check)
    - Sub-AFI = 3 (NLRI is used for both unicast and multicast RPF check)
    - Sub-AFI = 4 (label)

BGP Considerations

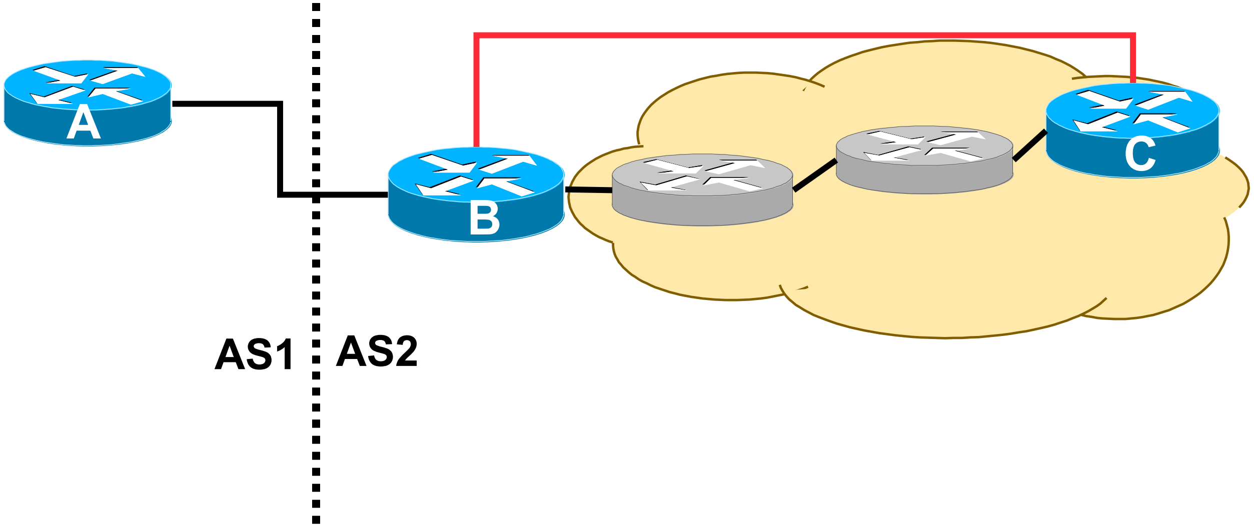
- Rules for constructing the NEXTHOP attribute:
  - When two peers share a common subnet, the NEXTHOP information is formed by a global address and a link local address
  - Redirects in IPv6 are restricted to the usage of link local addresses

Routing Information

- Independent operation
  - One RIB per protocol
    - e.g. IPv6 has its own BGP table
  - Distinct policies per protocol
- Peering sessions **can** be shared when the IPv4 and IPv6 topologies are congruent

BGP next-hop attribute

- Next-hop contains a global IPv6 address
  - (and potentially a link local address)
- Link local address is only set as a next-hop if the BGP peer shares the subnet with both routers (advertising and advertised)



More BGP considerations

- TCP Interaction
  - BGP runs on top of TCP
  - This connection could be set up either over IPv4 or IPv6
- Router ID
  - When no IPv4 is configured, an explicit bgp router-id needs to be configured
    - BGP identifier is a 32 bit integer currently generated from the router identifier – which is generated from an IPv4 address on the router
  - This is needed as a BGP identifier, this is used as a tie breaker, and is sent within the OPEN message

BGP Configuration

- IOS default is to assume that all configured peers are unicast IPv4 neighbours
  - If we want to support IPv6 too, this isn’t useful
  - So we disable the default assumption

```
no bgp default ipv4-unicast
```

  - This means that we must explicitly state which address family the peer belongs to

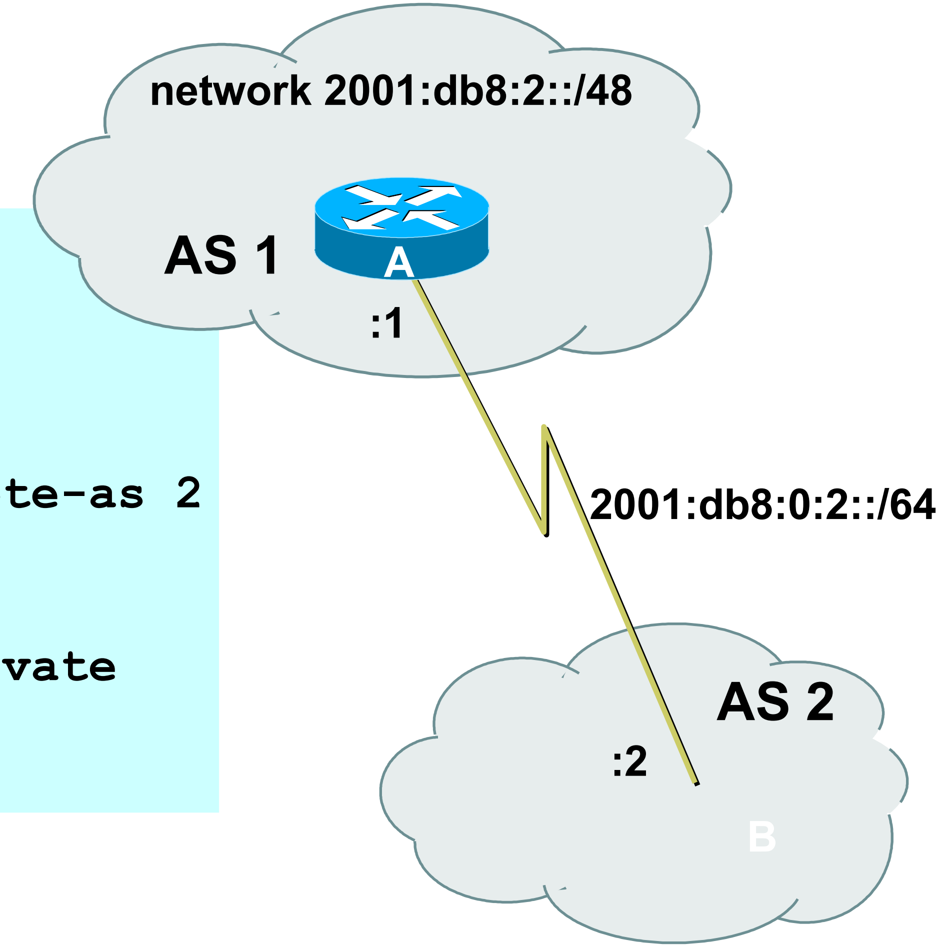
BGP Configuration

- Two options for configuring BGP peering
- Using link local addressing
  - ISP uses FE80:: addressing for BGP neighbours
  - **NOT RECOMMENDED**
    - There are plenty of IPv6 addresses
    - Unnecessary configuration complexity
- Using global unicast addresses
  - As with IPv4
  - **RECOMMENDED**

Regular BGP Peering

Router A

```
router bgp 1
 no bgp default ipv4 unicast
 neighbor 2001:db8:0:2::2 remote-as 2
!
 address-family ipv6
 neighbor 2001:db8:0:2::2 activate
 network 2001:db8:2::/48
!
```

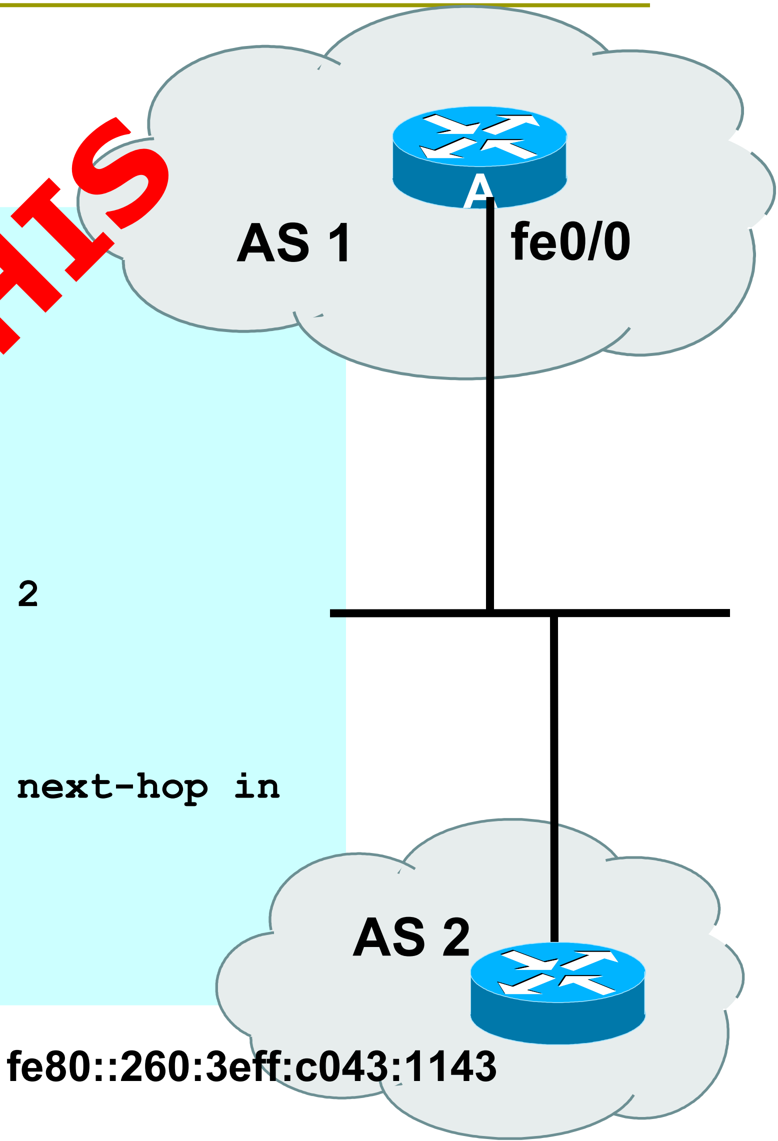


Link Local Peering

Router A

```
interface fastethernet 0/0
 ipv6 address 2001:db8:0:1::1/64
!
router bgp 1
 no bgp default ipv4 unicast
 neighbor fe80::260:3eff:c043:1143 remote-as 2
!
address-family ipv6
 neighbor fe80::260:3eff:c043:1143 activate
 neighbor fe80::260:3eff:c043:1143 route-map next-hop in
!
route-map next-hop permit 5
 set ipv6 next-hop 2001:db8:0:1::1
!
```

DO NOT DO THIS



IPv4 and IPv6

```
router bgp 10
 no bgp default ipv4-unicast
 neighbor 2001:db8:1:1019::1 remote-as 20
 neighbor 172.16.1.2 remote-as 30
!
 address-family ipv4
 neighbor 172.16.1.2 activate
 neighbor 172.16.1.2 prefix-list ipv4-ebgp in
 neighbor 172.16.1.2 prefix-list v4out out
 network 172.16.0.0
 exit-address-family
!
 address-family ipv6
 neighbor 2001:db8:1:1019::1 activate
 neighbor 2001:db8:1:1019::1 prefix-list ipv6-ebgp in
 neighbor 2001:db8:1:1019::1 prefix-list v6out out
 network 2001:db8::/32
 exit-address-family
!
```

BGP Configuration IPv4 and IPv6

- When configuring the router, recommendation is:
  - Put **all** IPv6 configuration directly into IPv6 address family
  - Put **all** IPv4 configuration directly into IPv4 address family
- Router will sort generic from specific address family configuration when the configuration is saved to NVRAM or displayed on the console
- Example follows…
  - Notice how **`activate`** is required to indicate that the peering is activated for the particular address family

BGP Address Families Applied Configuration

```
router bgp 10
 no bgp default ipv4-unicast
!
address family ipv4
 neighbor 172.16.1.2 remote-as 30
 neighbor 172.16.1.2 prefix-list ipv4-ebgp in
 neighbor 172.16.1.2 prefix-list v4out out
 neighbor 172.16.1.2 activate
 network 172.16.0.0
!
 address-family ipv6
 neighbor 2001:db8:1:1019::1 remote-as 20
 neighbor 2001:db8:1:1019::1 prefix-list ipv6-ebgp in
 neighbor 2001:db8:1:1019::1 prefix-list v6out out
 neighbor 2001:db8:1:1019::1 activate
 network 2001:db8::/32
!
ip prefix-list ipv4-ebgp permit 0.0.0.0/0 le 32
ip prefix-list v4out permit 172.16.0.0/16
ipv6 prefix-list ipv6-ebgp permit ::/0 le 128
ipv6 prefix-list v6out permit 2001:db8::/32
```

Generic Configuration

Specific Configuration

BGP Address Families
End result – line order has changed

```
router bgp 10
 no bgp default ipv4-unicast
 neighbor 2001:db8:1:1019::1 remote-as 20
 neighbor 172.16.1.2 remote-as 30
!
 address-family ipv4
 neighbor 172.16.1.2 activate
 neighbor 172.16.1.2 prefix-list ipv4-ebgp in
 neighbor 172.16.1.2 prefix-list v4out out
 network 172.16.0.0
 exit-address-family
!
 address-family ipv6
 neighbor 2001:db8:1:1019::1 activate
 neighbor 2001:db8:1:1019::1 prefix-list ipv6-ebgp in
 neighbor 2001:db8:1:1019::1 prefix-list v6out out
 network 2001:db8::/32
 exit-address-family
!
ip prefix-list ipv4-ebgp permit 0.0.0.0/0 le 32
ip prefix-list v4out permit 172.16.0.0/16
ipv6 prefix-list ipv6-ebgp permit ::/0 le 128
ipv6 prefix-list v6out permit 2001:db8::/32
```

Generic Configuration

Specific Configuration

Summary

- We have learned:
  - Why we use BGP
  - About the difference between Forwarding and Routing
  - About Interior and Exterior Routing
  - What the BGP Building Blocks are
  - How to configure BGP
  - How BGP has been enhanced to support IPv6