

Dynamic Routing

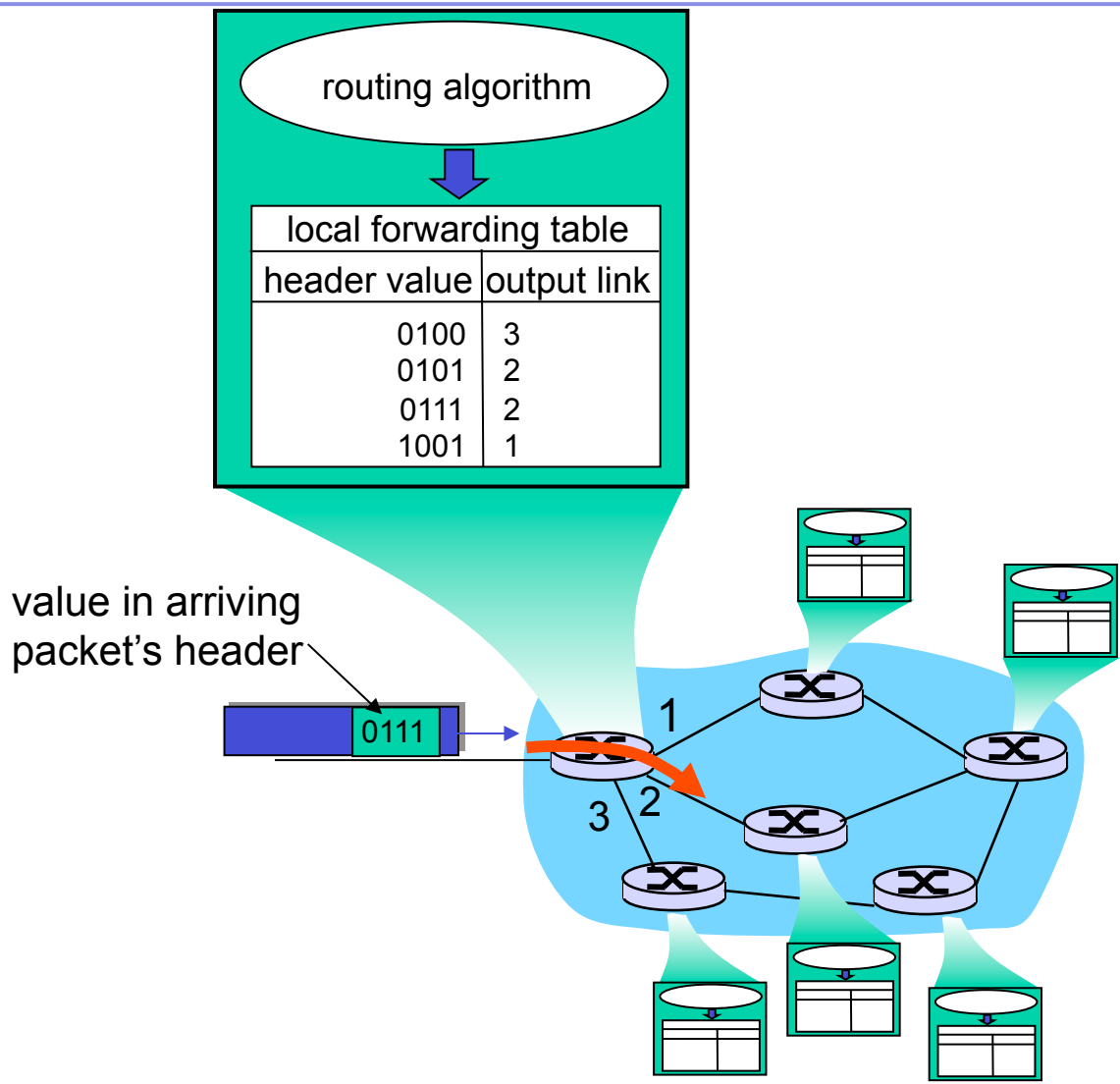


Overview

Desirable Characteristics of Dynamic Routing

- Automatically detect and adapt to topology changes
- Provide optimal routing
- Scalability
- Robustness
- Simplicity
- Rapid convergence
- Some control of routing choices
 - E.g. which links we prefer to use

Interplay between routing & forwarding



IP Routing – finding the path

- Path is derived from information received from the routing protocol
- Several alternative paths may exist
 - best next hop stored in **forwarding** table
- Decisions are updated periodically or as topology changes (event driven)
- Decisions are based on:
 - topology, policies and metrics (hop count, filtering, delay, bandwidth, etc.)

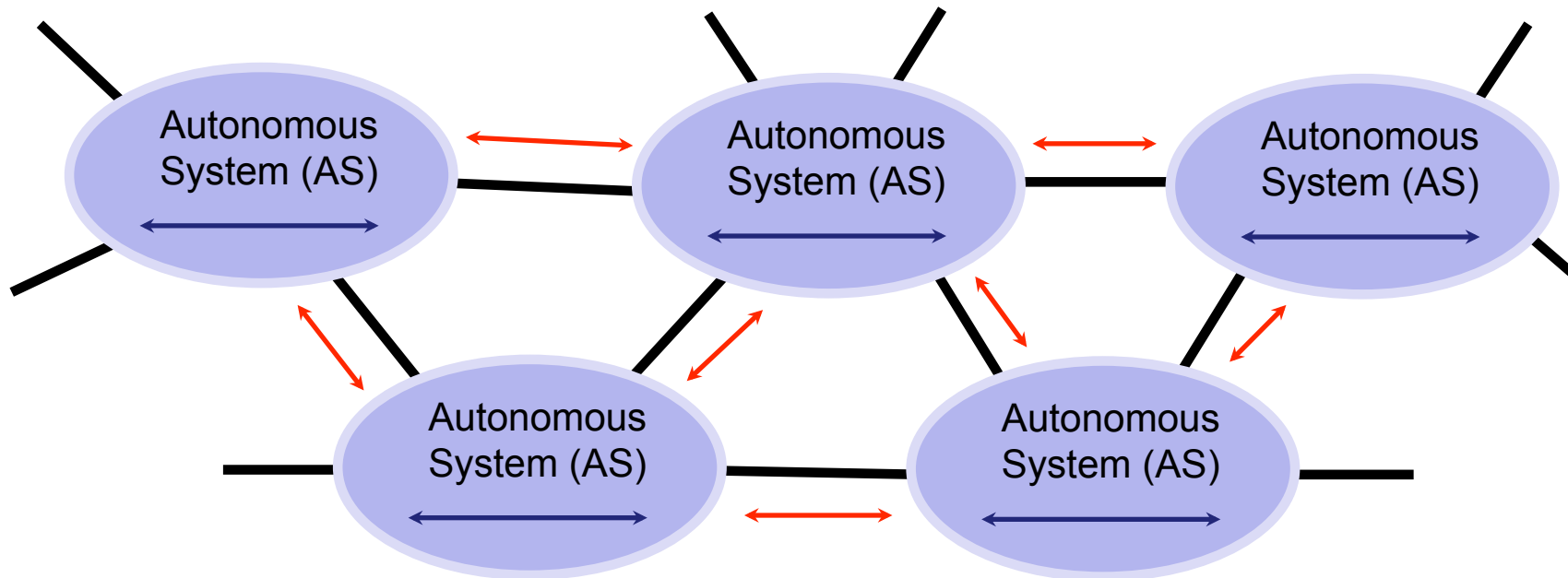
Convergence – why do I care?

- Convergence is when all the routers have a stable view of the network
- When a network is not converged there is network downtime
 - Packets don't get to where they are supposed to go
 - Black holes (packets "disappear")
 - Routing Loops (packets go back and fore between the same devices)
 - Occurs when there is a change in status of router or the links

Internet Routing Hierarchy

- The Internet is composed of Autonomous Systems
- Each Autonomous System is an administrative entity that
 - Uses Interior Gateway Protocols (IGPs) to determine routing within the Autonomous System
 - Uses Exterior Gateway Protocols (EGPs) to interact with other Autonomous Systems

Internet Routing Architecture



Autonomous System: A collection of IP subnets and routers under the same administrative authority.

- Interior Routing Protocol
- Exterior Routing Protocol

Interior Gateway Protocols

- Four well known IGPs today
 - RIP
 - EIGRP
 - OSPF
 - ISIS

Exterior Gateway Protocols

- One single de-facto standard:
 - BGP

Routing's 3 Aspects

- Acquisition of information about the IP subnets that are reachable through an internet
 - static routing configuration information
 - dynamic routing information protocols (e.g., BGP4, OSPF, RIP, ISIS)
 - each mechanism/protocol constructs a Routing Information Base (RIB)

Routing Aspect #2

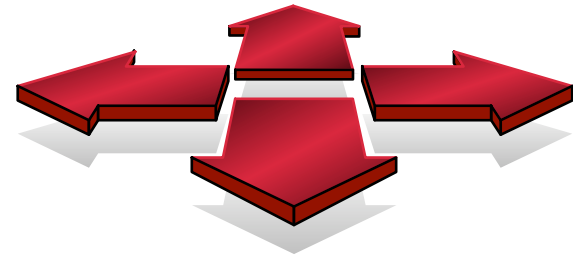
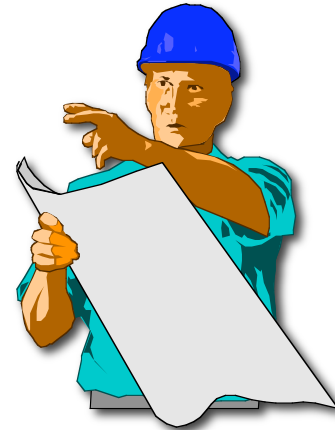
- Construction of a Forwarding Table
 - synthesis of a single table from all the Routing Information Bases (RIBs)
 - information about a destination subnet may be acquired multiple ways
 - a precedence is defined among the RIBs to arbitrate conflicts on the same subnet
 - Also called a Forwarding Information Base (FIB)

Routing #3

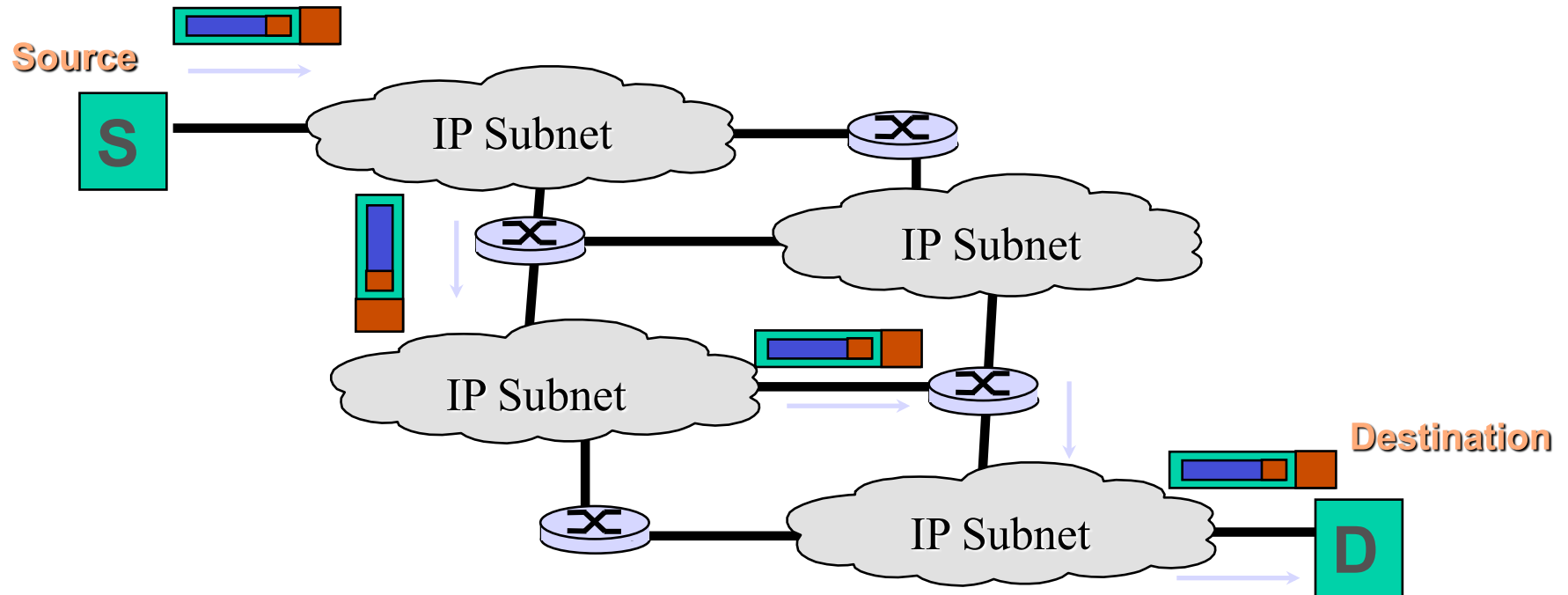
- Use of a Forwarding Table to forward individual packets
 - selection of the next-hop router and interface
 - hop-by-hop, each router makes an independent decision

Routing versus Forwarding

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

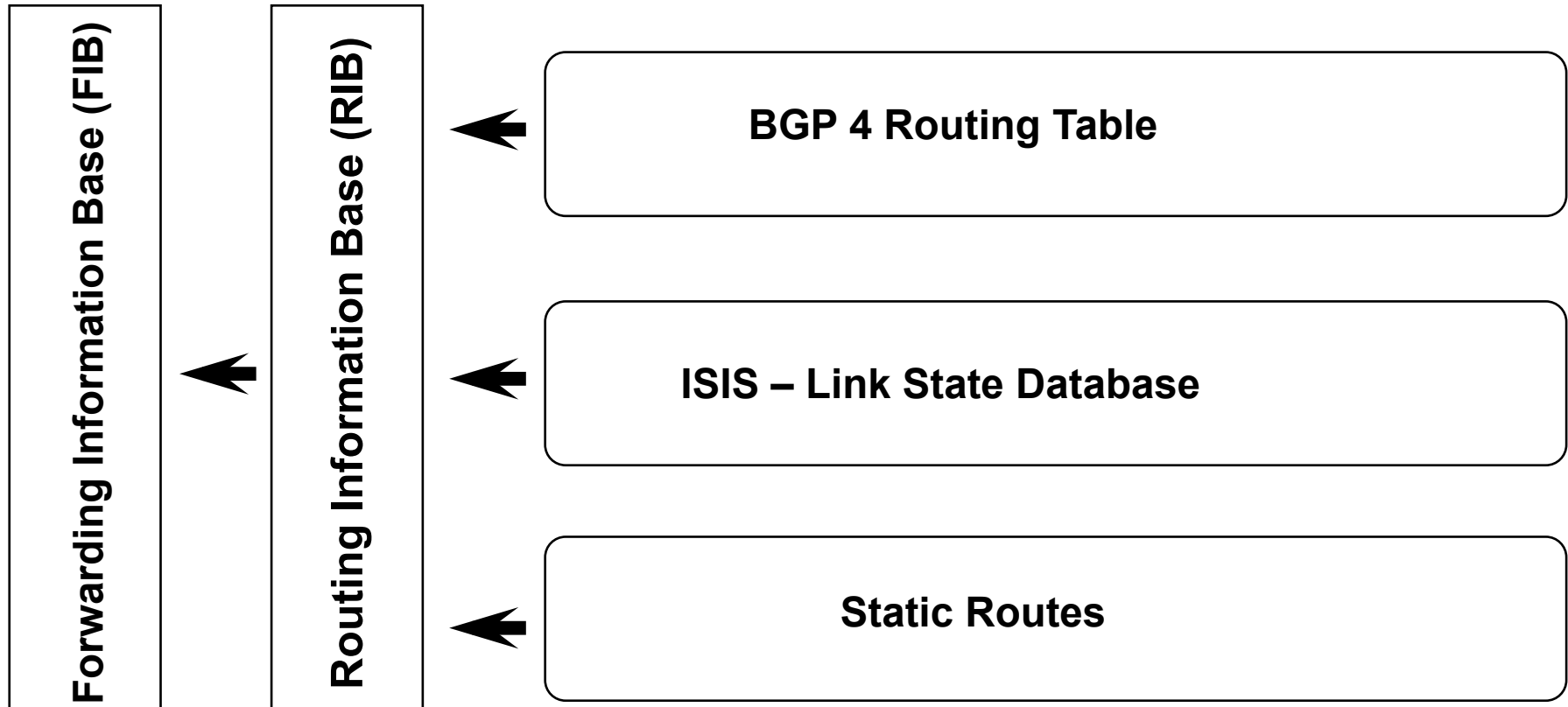


IP Forwarding



- Forwarding decisions:
 - **Destination address**
 - class of service (fair queuing, precedence, others)
 - local requirements (packet filtering)

Routing Tables Feed the Forwarding Table



RIB Construction

- Each routing protocol builds its own Routing Information Base (RIB)
- Each protocol has its own “view” of “costs”
 - e.g., ISIS is administrative weights
 - e.g., BGP4 is Autonomous System path length

FIB Construction

- **There is only ONE forwarding table!**
- An algorithm is used to choose one next-hop toward each IP destination known by any routing protocol
 - the set of IP destinations present in any RIB are collected
 - if a particular IP destination is present in only one RIB, that RIB determines the next hop forwarding path for that destination

FIB Construction

- Choosing FIB entries, cont..
 - if a particular IP destination is present in multiple RIBs, then a precedence is defined to select which RIB entry determines the next hop forwarding path for that destination
 - This process normally chooses exactly one next-hop toward a given destination
- There are no standards for this; it is an implementation (vendor) decision

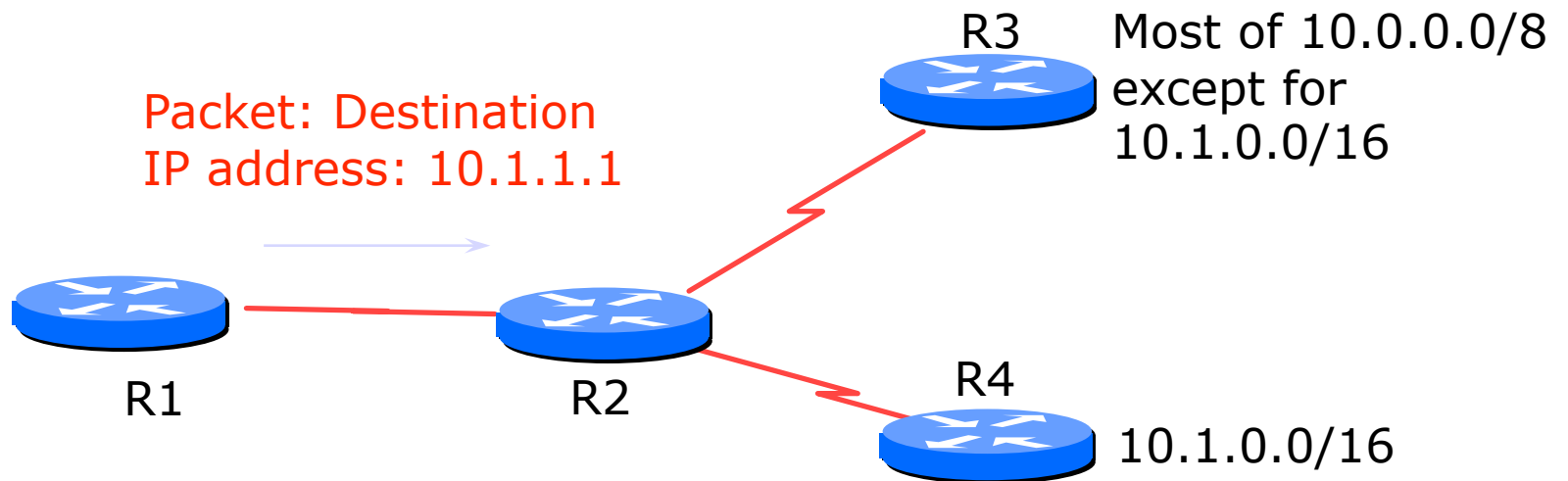
FIB Contents

- IP subnet and mask (or length) of destinations
 - can be the “default” IP subnet
- IP address of the “next hop” toward that IP subnet
- Interface id of the subnet associated with the next hop
- Optional: cost metric associated with this entry in the forwarding table

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

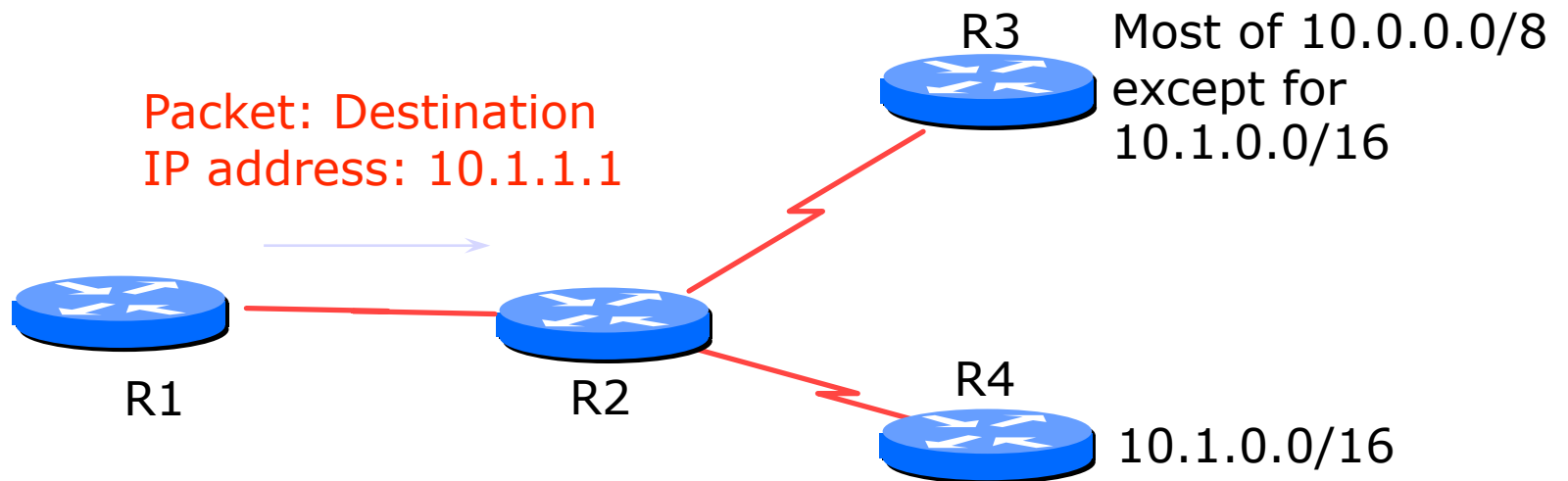


Based on
destination IP
address

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



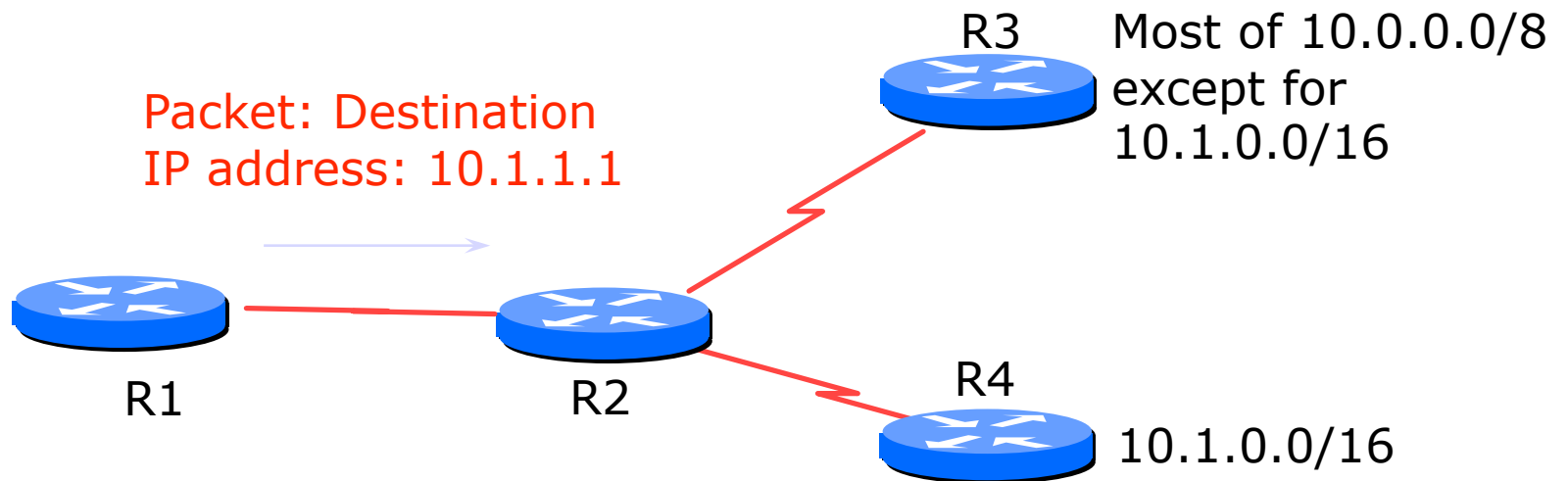
Based on
destination IP
address

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



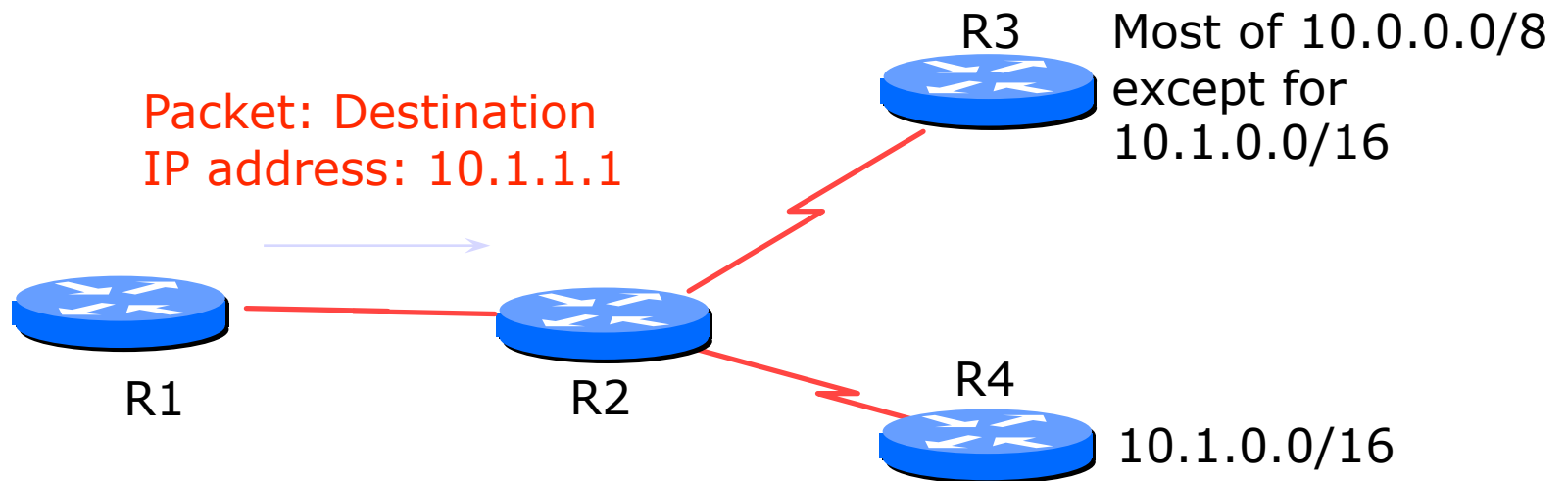
Based on
destination IP
address

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



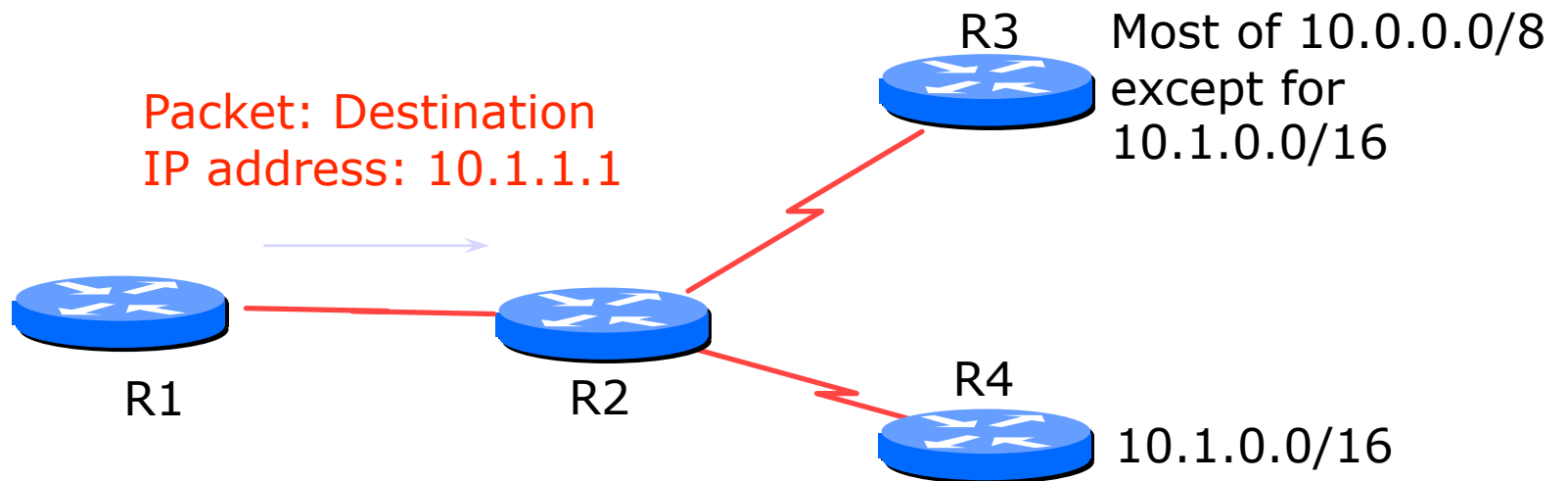
Based on
destination IP
address

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



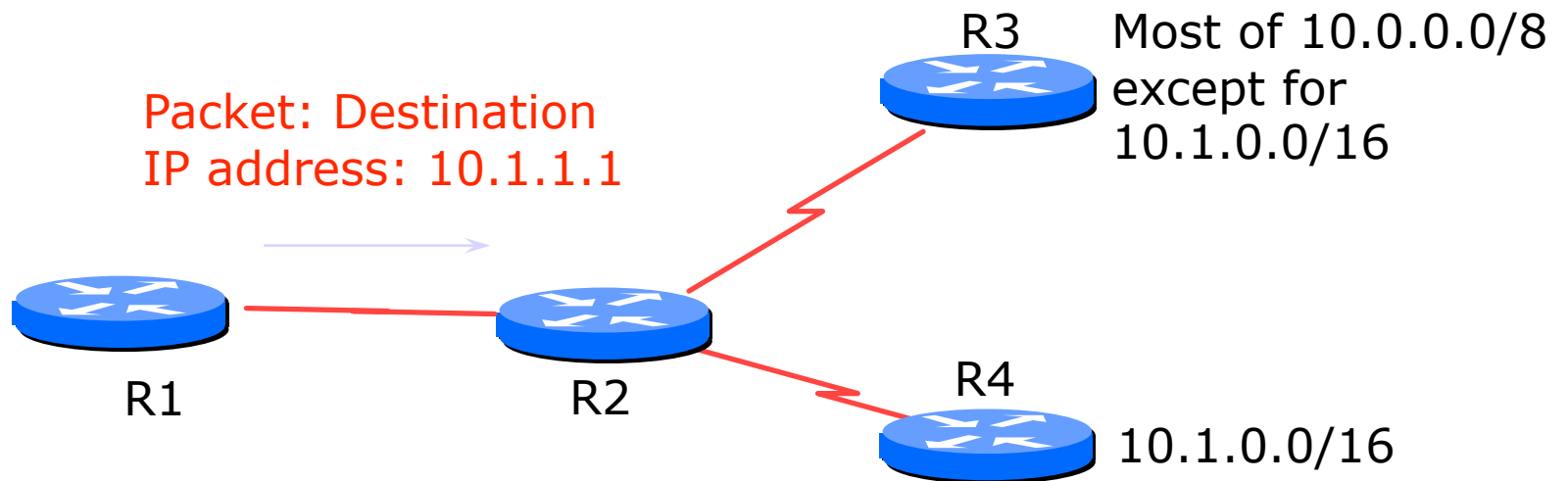
Based on
destination IP
address

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



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

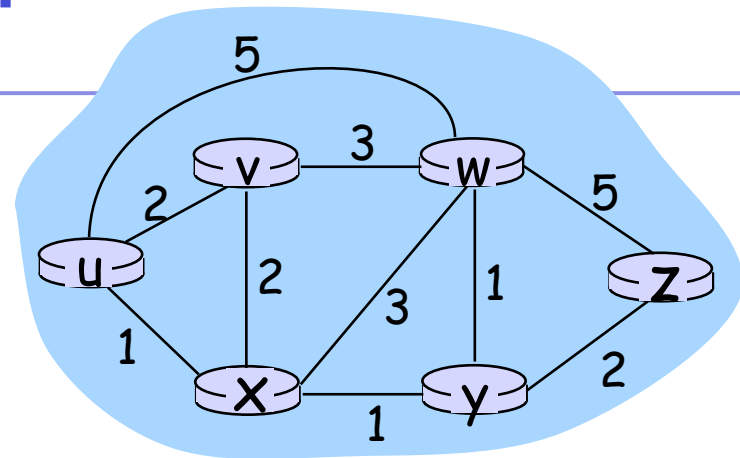
This is the longest
matching prefix (length
16). "R2" will send the
packet to "R4".

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.

Graph abstraction



Graph: $G = (N, E)$

N = set of routers = $\{ u, v, w, x, y, z \}$

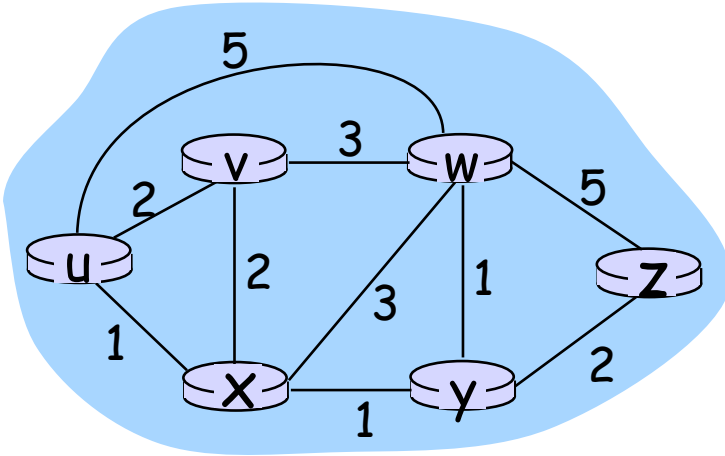
E = set of links = $\{ (u, v), (u, x), (v, x), (v, w), (x, w), (x, y), (w, y), (w, z), (y, z) \}$

Path: Sequence of edges (routers)

Remark: Graph abstr. is useful in other network contexts

Example: P2P, where N is set of peers
and E is set of TCP connections

Graph abstraction: costs



- $c(x,x')$ = cost of link (x,x')
 - e.g., $c(w,z) = 5$
- cost can be always 1, or
- inversely related to bandwidth,
- inversely related to congestion

Cost of path $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

Question: What's the least-cost path between u and z ?

Routing algorithm: alg. that finds "good" path
(typically: least cost path)

Distance Vector and Link State

- Distance Vector
 - Accumulates a metric hop-by-hop as the protocol messages traverse the subnets
- Link State
 - Builds a network topology database
 - Computes best path routes from current node to all destinations based on the topology

Distance Vector Protocols

- Each router only advertises to its neighbors, its “distance” to various IP subnets
- Each router computes its next-hop routing table based on least cost determined from information received from its neighbors and the cost to those neighbors

Distance Vector Algorithm

Bellman-Ford Equation

Define

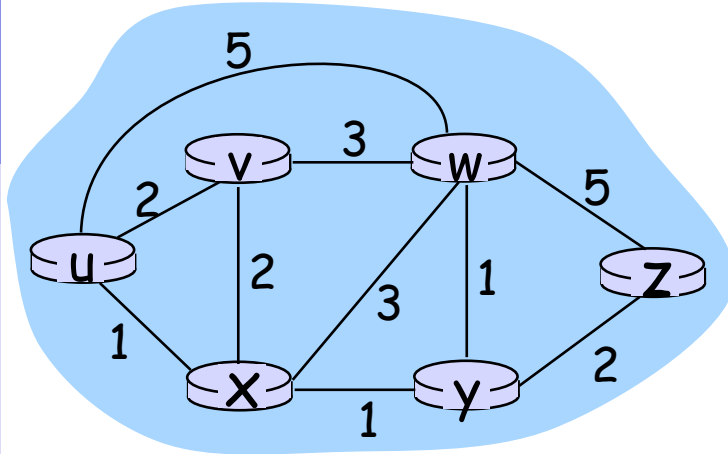
$d_x(y) :=$ cost of least-cost path from x to y

Then

$$d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$$

where min is taken over all neighbors v of x

Bellman-Ford example



Clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

Bellman-Ford equation says:

$$\begin{aligned}d_u(z) &= \min \{ c(u,v) + d_v(z), \\ &\quad c(u,x) + d_x(z), \\ &\quad c(u,w) + d_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4\end{aligned}$$

Node that yields minimum is next hop in shortest path → forwarding table

Distance Vector RIB Parameters

- Accumulated cost
 - cost is a constant administrative assignment for each subnet
 - assignment is typically "1" for each subnet (equivalent to hop-count)
 - included in routing protocol exchange
- Time the update was received (for timeout)

Distance Vector RIB Parameters

- The next-hop the entry was received from
 - sender's id is included in routing protocol exchange
- Accumulated Hop count and Maximum Hop Count
 - used to detect cycles
 - hop count included in routing protocol exchange

Distance Vector: Additions

- When a router learns of new reachable subnets
 - at router startup
 - when an interface is enabled or restored to service
- A routing update is broadcast to all neighbors

Distance Vector: Additions

- Any router receiving the packet compares the cost it received in the new packet with that in its RIB
- If the cost is smaller or the subnet is new
 - the new entry is used in the RIB
 - the new entry is broadcast to all its neighbors (except the one from which it was received)

Distance Vector: Removals

- Each RIB entry is aged
 - a timeout defines when an entry is removed from the RIB
- Periodically, each router re-advertises all the routes it knows to its neighbors
 - this can be done in many ways: from simple neighbor hellos to enumeration of all routes

Distance Vector: Removals

- If a neighbor does not respond within a timeout, all routes learned from that neighbor are removed
- Route removal may be advertised to neighbors

Distance Vector Algorithm (2)

- $D_x(y)$ = estimate of least cost from x to y
- Distance vector: $\mathbf{D}_x = [D_x(y): y \in N]$
- Node x knows cost to each neighbor v : $c(x,v)$
- Node x maintains $\mathbf{D}_x = [D_x(y): y \in N]$
- Node x also maintains its neighbors' distance vectors
 - For each neighbor v , x maintains $\mathbf{D}_v = [D_v(y): y \in N]$

Distance Vector Algorithm (3)

Basic idea:

- Each node periodically sends its own distance vector estimate to neighbors
- When a node x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \quad \text{for each node } y \in N$$

- Under “natural” conditions the estimates of $D_x(y)$ converge to the actual least cost $d_x(y)$

Distance Vector Algorithm (4)

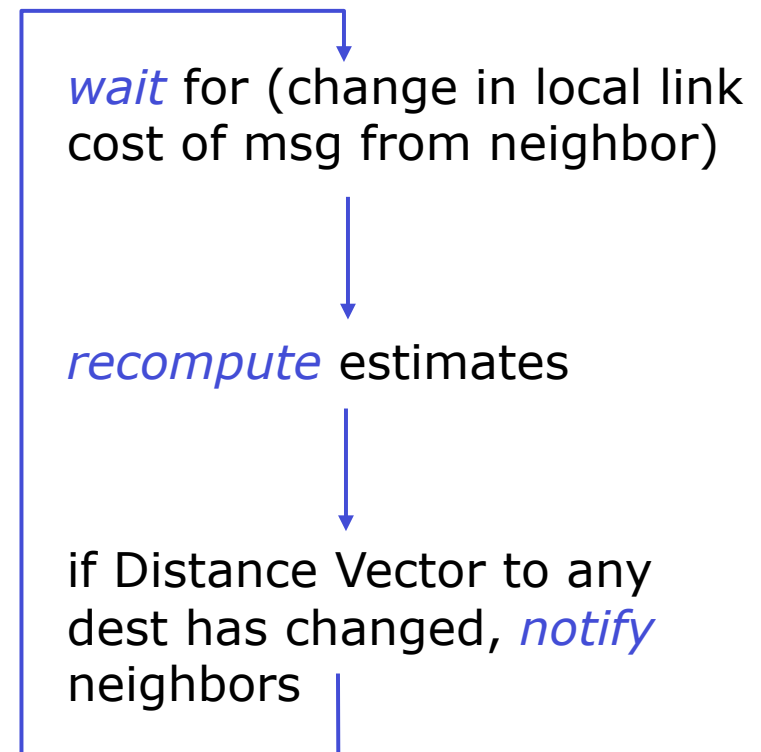
Iterative, asynchronous:

- each local iteration caused by:
 - local link cost change
 - DV update message from neighbor

Distributed:

- each node notifies neighbors *only* when its Distance Vector changes
 - neighbors then notify their neighbors if necessary

Each node:



$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} \\ = \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} \\ = \min\{2+1, 7+0\} = 3$$

node x table

		cost to		
		x	y	z
from	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

node y table

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	7	1	0

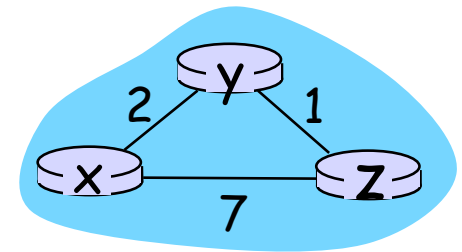
		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

node z table

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	3	1	0

		cost to		
		x	y	z
from	x	0	2	3
	y	2	0	1
	z	3	1	0

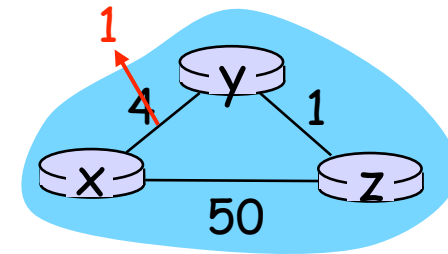


time

Distance Vector (DV): link cost changes

Link cost changes:

- node detects local link cost change
- updates routing info, recalculates distance vector
- if DV changes, notify neighbors

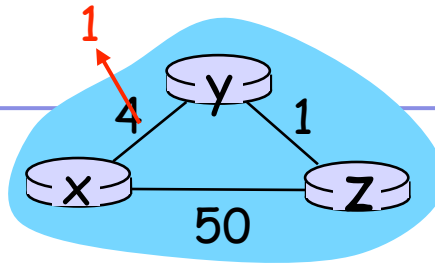


“good
news
travels
fast”

At time t_0 , y detects the link-cost change, updates its DV, and informs its neighbors.

At time t_1 , z receives the update from y and updates its table. It computes a new least cost to x and sends its neighbors its DV.

At time t_2 , y receives z 's update and updates its distance table. y 's least costs do not change and hence y does *not* send any message to z .



node **y** table

		cost to		
		x	y	z
from	x			
	y	4 1	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	1	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	1	0	1
	z	2	1	0

node **z** table

		cost to		
		x	y	z
from	x			
	y	4	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	1	0	1
	z	5 2	1	0

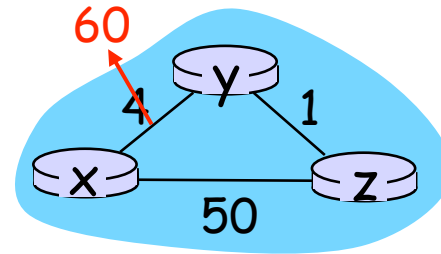
		cost to		
		x	y	z
from	x			
	y	1	0	1
	z	2	1	0

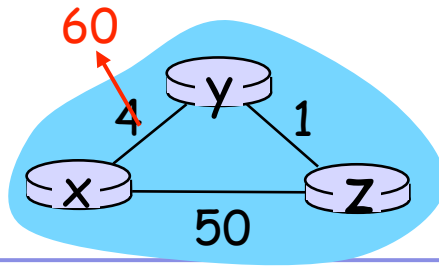
time →

Distance Vector: link cost changes

Link cost changes:

- good news travels fast
- *bad news travels slow*





$$D_y(x) = \min\{c(y,x) + D_x(x), c(y,z) + D_z(x)\}$$

$$= \min\{60 + 0, 1 + 5\} = 6$$

$$D_y(x) = \min\{c(y,x) + D_x(x), c(y,z) + D_z(x)\}$$

$$= \min\{60 + 0, 1 + 7\} = 8$$

node **y** table

		cost to		
		x	y	z
from	x	4 6	0	1
	y	4	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	6	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	6 8	0	1
	z	7	1	0

node **z** table

		cost to		
		x	y	z
from	x			
	y	4	0	1
	z	5	1	0

		cost to		
		x	y	z
from	x			
	y	6	0	1
	z	5 7	1	0

		cost to		
		x	y	z
from	x			
	y	6	0	1
	z	7	1	0

time →

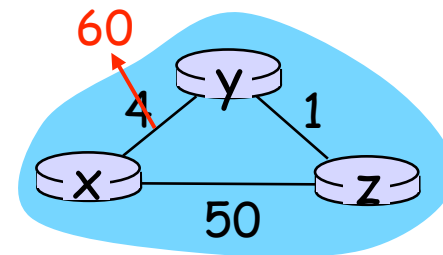
Distance Vector: link cost changes

Link cost changes:

- good news travels fast
- bad news travels slow – “count to infinity” problem!
- 44 iterations before algorithm stabilizes.

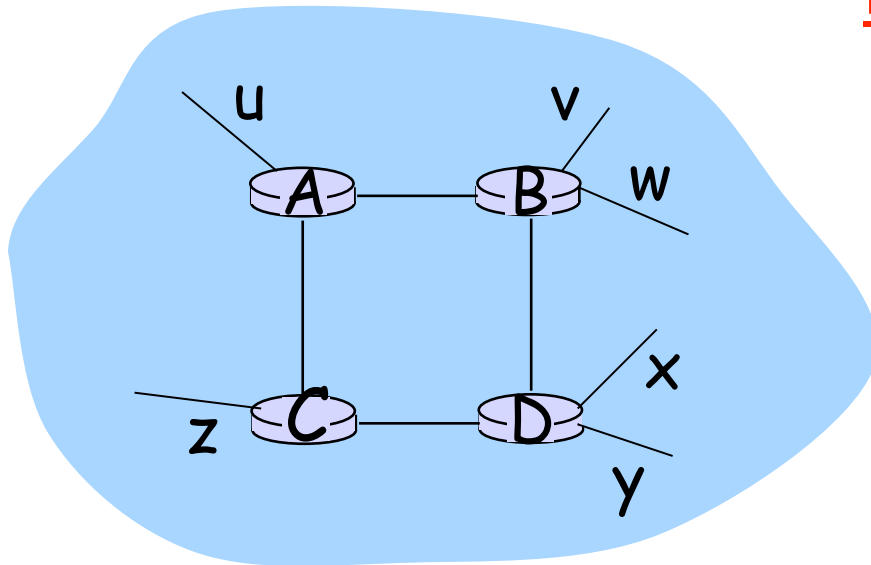
Poisoned reverse:

- If Z routes through Y to get to X :
 - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
- will this completely solve count to infinity problem?



RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)



From router A to subsets:

<u>destination</u>	<u>hops</u>
u	1
v	2
w	2
x	3
y	3
z	2

RIP advertisements

- Distance vectors: exchanged among neighbors every 30 sec via Response Message (also called **advertisement**)
- Each advertisement: list of up to 25 destination nets within AS

RIP: link failure and recovery

- If no advertisement heard after 180 sec, neighbor/link declared dead
 - Routes via the neighbor are invalidated
 - New advertisements sent to neighbors
 - Neighbors in turn send out new advertisements (if their tables changed)
 - Link failure info quickly propagates to entire net
 - Poison reverse used to prevent ping-pong loops (infinite distance = 16 hops)

Why not use RIP?

- RIP is a Distance Vector Algorithm
 - Listen to neighbouring routes
 - Install all routes in routing table
 - Lowest hop count wins
 - Advertise all routes in table
 - Very simple, very stupid
- Only metric is hop count
- Network is max 16 hops (not large enough)
- Slow convergence (routing loops)
- Poor robustness

EIGRP

- “Enhanced Interior Gateway Routing Protocol”
- Predecessor was IGRP which was classfull
 - IGRP developed by Cisco in mid 1980s to overcome scalability problems with RIP
- Cisco proprietary routing protocol
- Distance Vector Routing Protocol
 - Has very good metric control
- Still maybe used in some enterprise networks?
 - Multi-protocol (supports more than IP)
 - Exhibits good scalability and rapid convergence
 - Supports unequal cost load balancing

Link State Protocols

- Each router broadcasts to all the routers in the network the state of its locally attached links and IP subnets
- Each router constructs a complete topology view of the entire network based on these link state updates and computes its next-hop routing table based on this topology view

Link State Protocols

- Attempts to minimize convergence times and eliminate non-transient packet looping at the expense of higher messaging overhead, memory, and processing requirements
- Allows multiple metrics/costs to be used

Link State RIB Parameters

- Topology Database
 - Router IDs
 - Link IDs
 - From Router ID
 - To Router ID
 - Metric(s)
 - Sequence number
- List of Shortest Paths to Destinations

Link State Operation: Removals

- Removals are announcements with the metric set to “infinity”
- Adjacencies must be refreshed
 - neighbors use “hello” protocol
 - if a router loses a neighbor, then routes via that neighbor are recomputed
 - send announcements with link metric to lost neighbor set to infinity

Link State: Shortest Path

- Dijkstra's Shortest Path First graph algorithm
 - Use yourself as starting point
 - Search outward on the graph and add router IDs as you expand the front
- Addresses are associated with routers
 - Hence the SPF algorithm needs to deal only in the number of routers, not the number of routes

Dijkstra's Algorithm

1 **Initialization for A:**

```
2 N' = {A}
3 for all nodes v in Graph
4   if v adjacent to A
5     then D(v) = c(A,v)
6     else D(v) = infinity
7
```

8 **Loop**

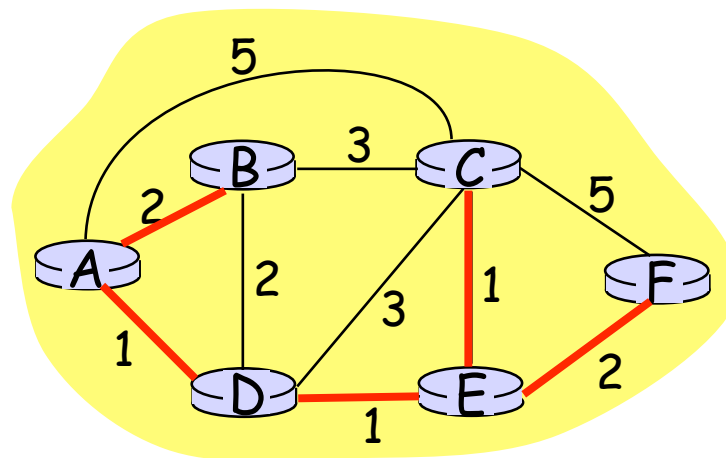
```
9   find w not in N' such that D(w) is a minimum
10  add w to N'
11  update D(v) for all v adjacent to w and not in N':
12    D(v) = min( D(v), D(w) + c(w,v) )
13  /* new cost to v is either old cost to v or known
14     shortest path cost to w plus cost from w to v */
15 until all nodes in N'
```

Notation:

- $c(i,j)$: link cost from node i to j . cost infinite if not direct neighbors
- $D(v)$: current value of cost of path from source to dest. v
- N' : set of nodes whose least cost path definitively known

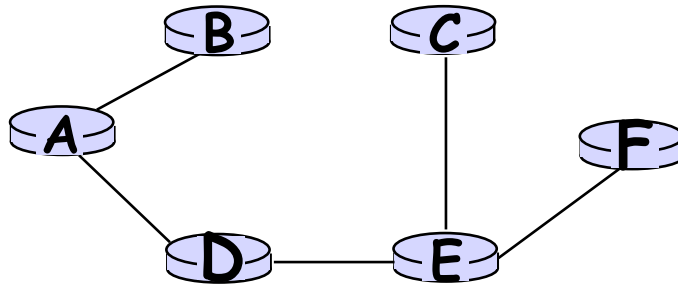
Dijkstra's algorithm: example

Step	start N'	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
→ 0	A	2,A	5,A	1,A	infinity	infinity
→ 1	AD	2,A	4,D		2,D	infinity
→ 2	ADE	2,A	3,E			4,E
→ 3	ADEB		3,E			4,E
→ 4	ADEBC					4,E
5	ADEBCF					



Dijkstra's algorithm: example (2)

Resulting shortest-path tree from A:



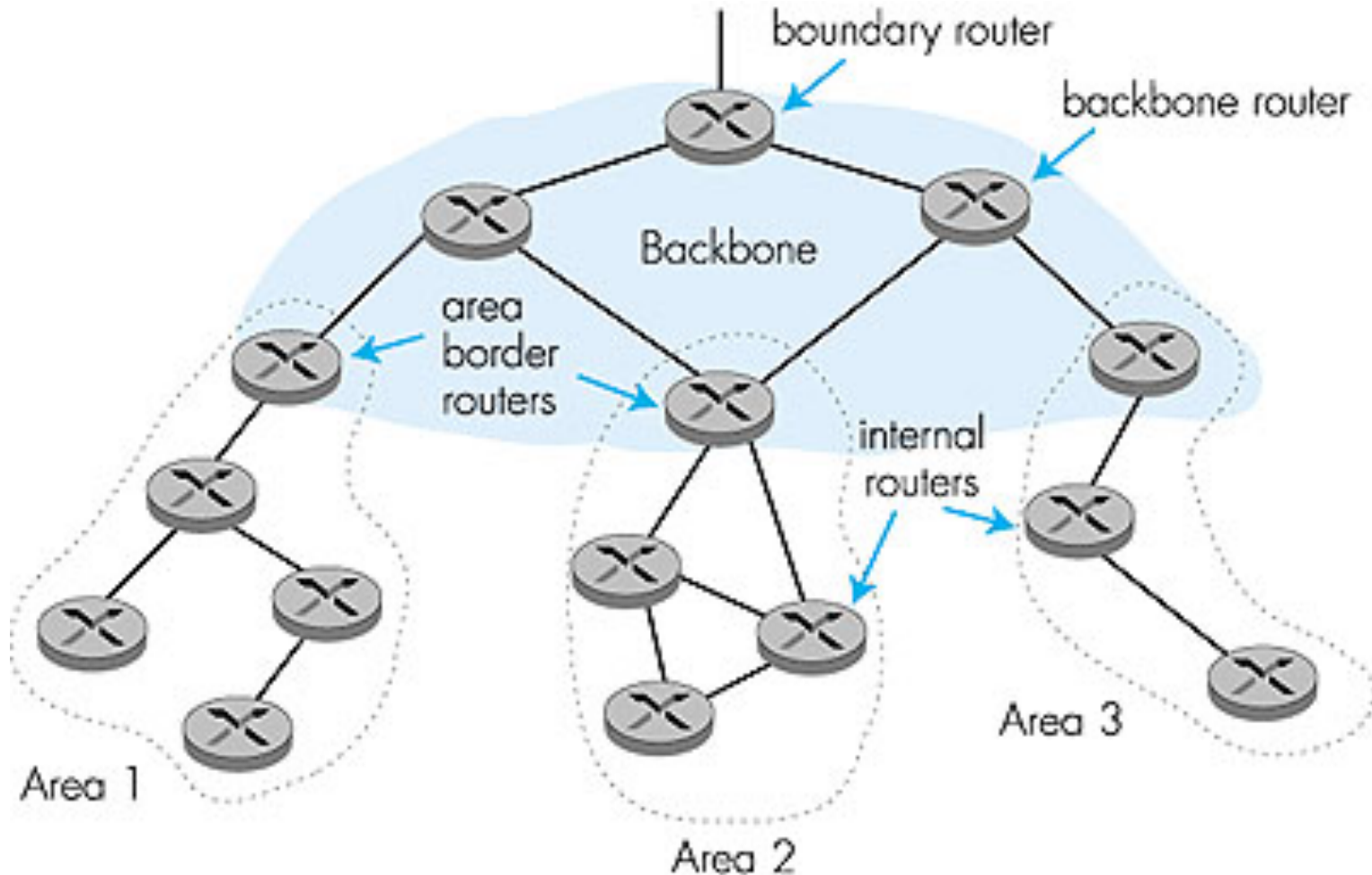
Resulting forwarding table in A:

destination	link
B	(A,B)
D	(A,D)
E	(A,D)
C	(A,D)
F	(A,D)

OSPF

- Open Shortest Path First
 - “Open” means it is public domain
 - Uses “Shortest Path First” algorithm – sometimes called “the Dijkstra algorithm”
- Current generation interior routing protocol based on “link state” concepts (RFC 1131, 10/1/89, obsoleted by OSPF v2, RFC 1723, 11/15/94)
- Supports hierarchies for scalability
- Fast convergence and loop avoidance
 - OSPFv3 based on OSPFv2 designed to support IPv6

Hierarchical OSPF



Hierarchical OSPF

- **Two-level hierarchy**: local area and backbone.
 - Link-state advertisements only in respective areas.
 - Nodes in each area have detailed area topology; only know direction (shortest path) to networks in other areas.
- **Area Border routers** “summarize” distances to networks in the area and advertise them to other Area Border routers.
- **Backbone routers**: run an OSPF routing algorithm limited to the backbone.
- **Boundary routers**: connect to other ASs.

IS-IS



Intermediate-System
to
Intermediate-System

IS-IS Overview

Terminology and Acronyms

Intermediate system (IS)- *Router*

Designated Intermediate System (DIS) - *Designated Router*

Pseudonode - *Broadcast link emulated as virtual node by DIS*

End System (ES) - *Network Host or workstation*

Network Service Access Point (NSAP) - *Network Layer Address*

Subnetwork Point of attachment (SNPA) - *Datalink interface*

Packet data Unit (PDU) - *Analogous to IP Packet*

Link State PDU (LSP) - *Routing information packet*

IS-IS Overview

- The Intermediate Systems to Intermediate System Routing Protocol (IS-IS) was originally designed to route the ISO Connectionless Network Protocol (CLNP) . (ISO10589 or RFC 1142)
- Adapted for routing IP in addition to CLNP (RFC1195) as Integrated or Dual IS-IS
- IS-IS is a Link State Protocol similar to the Open Shortest Path First (OSPF). OSPF supports only IP

IS-IS Overview

- 3 network layer protocols play together to deliver the ISO defined Connectionless Network Service
 - CLNP
 - IS-IS
 - ES-IS – End System to Intermediate System
- All 3 protocols independently go over layer-2

IS-IS Overview

- CLNP is the ISO equivalent of IP for datagram delivery services (ISO 8473, RFC 994)
- ES-IS is designed for routing between network hosts and routers (ISO 9542, RFC 995).
- IS-IS for layer 3 routing between routers. (ISO 10589/RFC 1142). Integrated IS-IS (RFC 1195) works within the ISO CNLS framework even when used for routing only IP.

IS-IS Overview

- End System Hellos (ESH) from Hosts and Intermediate System Hellos (ISH) from Routers used for ES-IS neighbor discovery
- Intermediate System to Intermediate Systems Hellos (IIH) are used for establishing IS-IS layer3 adjacencies
- ES-IS is somehow tied into IS-IS layer 3 adjacency discovery. ES-IS enabled automatically when IS-IS is configured on Cisco

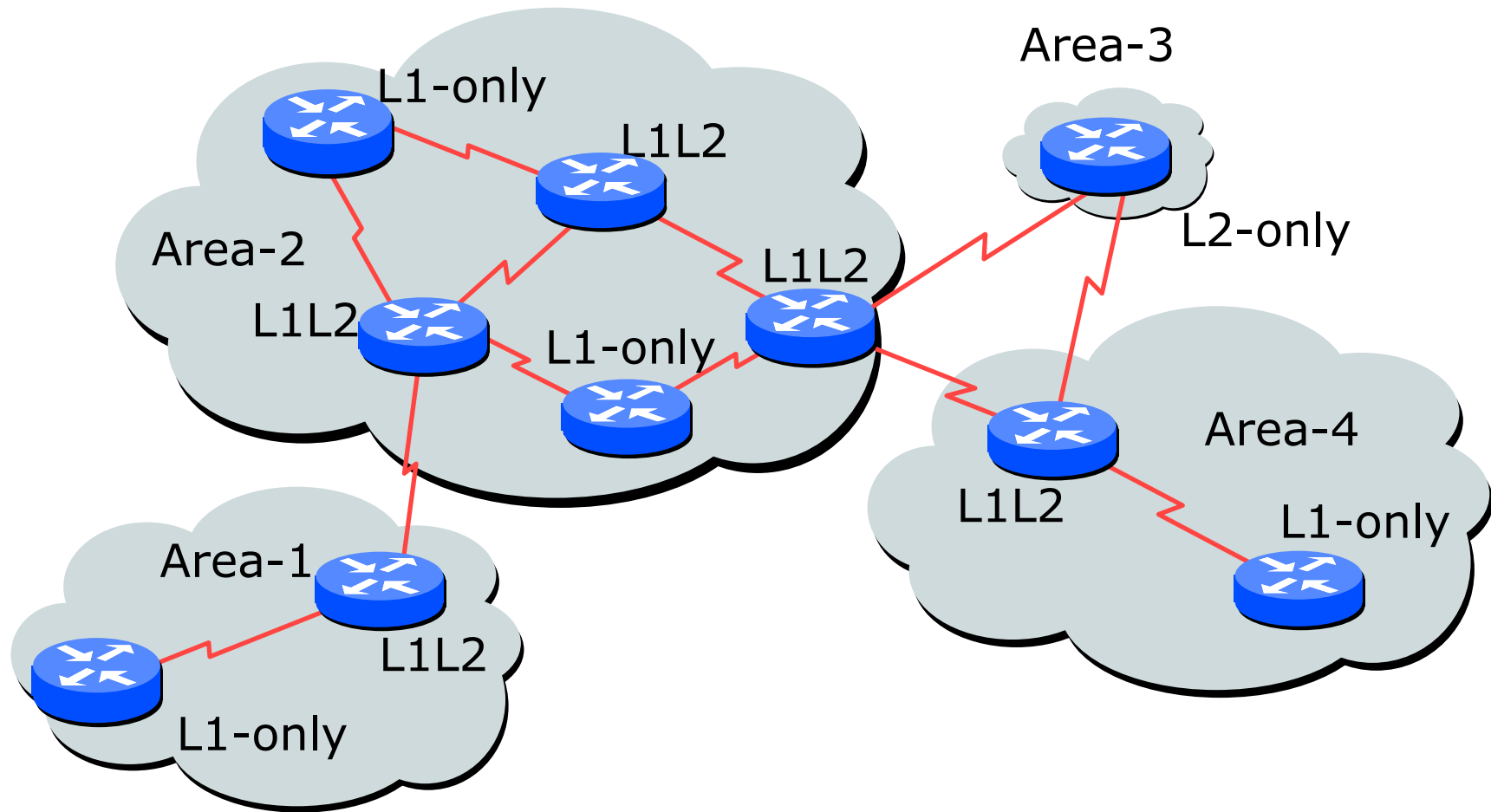
Link State Algorithm

- Each router contains a database containing a map of the whole topology
 - Links
 - Their state (including cost)
- All routers have the same information
- All routers calculate the best path to every destination
- Any link state changes are flooded across the network
 - “Global spread of local knowledge”

ISIS Levels

- ISIS has a 2 layer hierarchy
 - Level-2 (the backbone)
 - Level-1 (the areas)
- A router can be
 - Level-1 (L1) router
 - Level-2 (L2) router
 - Level-1-2 (L1L2) router

L1, L2, and L1L2 Routers



IS-IS Protocol Concepts

IS-IS Packet Types

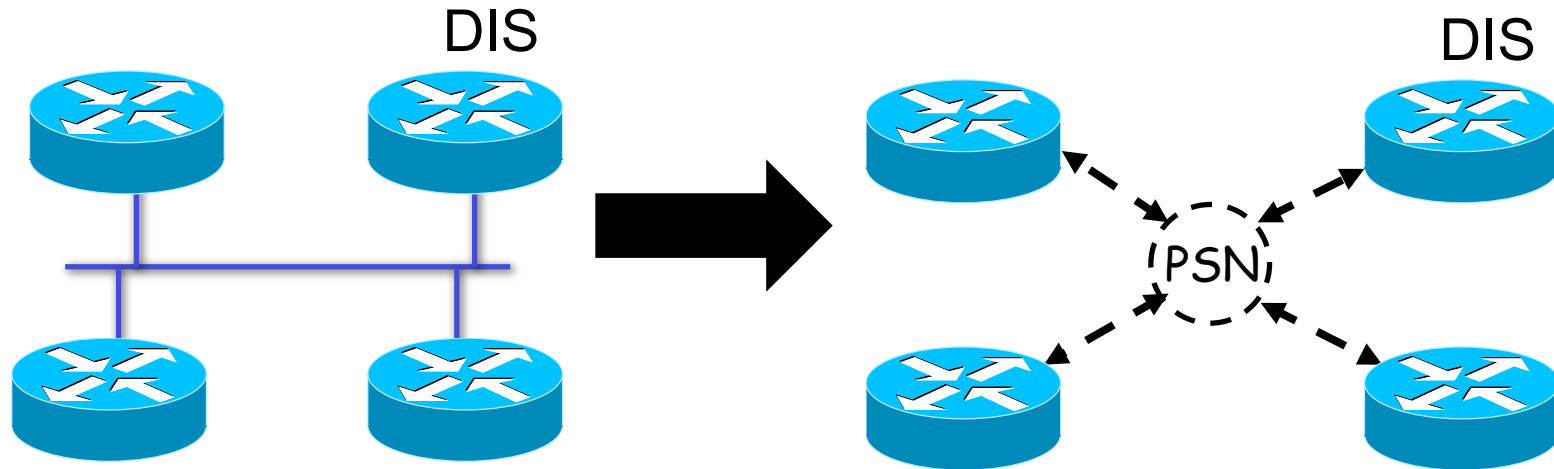
- IS-IS Hello Packets (IIH)
 - Level 1 LAN IS-IS Hello
 - Level 2 LAN IS-IS Hello
 - Point-to-point Hello
- Link State Packets (LSP)
 - Level 1 and Level 2
- Complete Sequence Number packets (CSNP)
 - Level 1 and Level 2
- Partial Sequence Number Packets (PSNP)
 - Level 1 and Level 2

Backbone & Areas

- ISIS does not have a backbone area as such (like OSPF)
- Instead the backbone is the contiguous collection of Level-2 capable routers
- ISIS area borders are on links, not routers
- Each router is identified with Network Entity Title (NET)
 - NET is an NSAP where the n-selector is 0

IS-IS Protocol Concepts

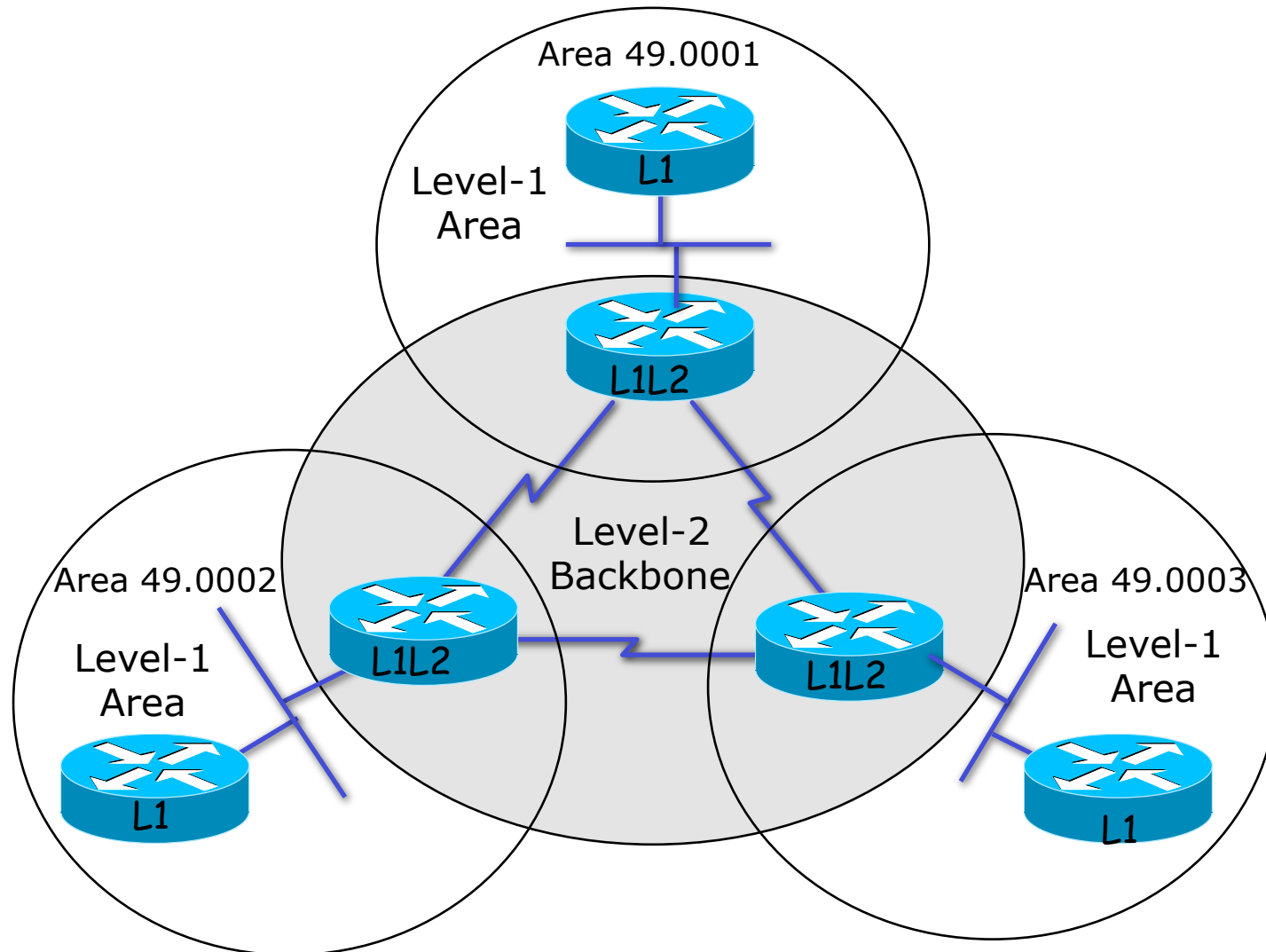
Network Nodes



- Broadcast link represented as virtual node, referred to as Pseudonode (PSN)
- PSN role played by the Designated Router (DIS)
- DIS election is preemptive, based on interface priority with highest MAC address being tie breaker
- IS-IS has only one DIS. DIS/PSN functionality supports database synchronization between routers on a broadcast type link

IS-IS Protocol Concepts

Areas

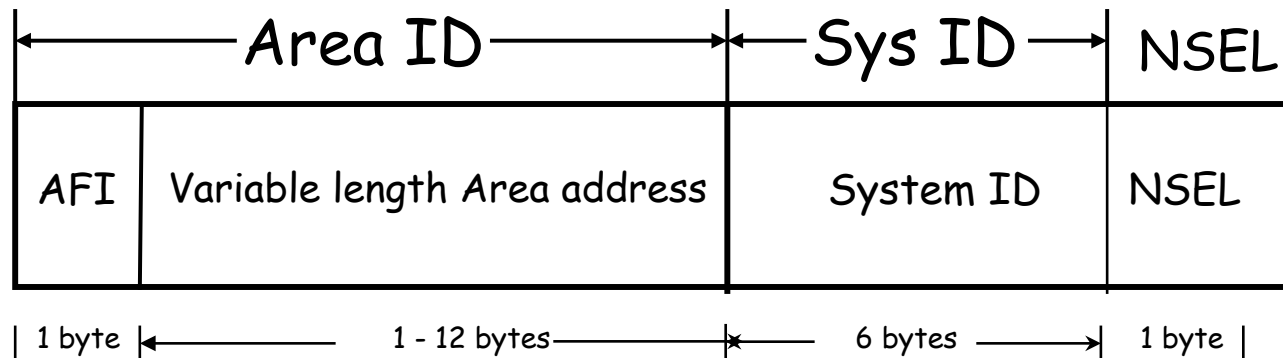


3. CLNS Addressing

- NSAP Format
- AFI Values
- Requirements and Caveats
- Examples
- Globally unique NSAPs

CLNS Addressing

NSAP Format



- NSAP format has 3 main components
 - Area ID
 - System ID
 - N-Selector (NSEL) - value is 0x00 on a router
- NSAP of a router is also called a NET

CLNS Addressing

AFI Values

Address Domain	AFI Value
X.121	37
ISO DCC	39
ISO 6523	47
Local	49

- X.121 - Int'l plan for public data networks
- ISO DCC - Data country code
- ISO 6523 ICD - Telex
- Local - For local use within network domain only

CLNS Addressing

Requirements and Caveats

- At least one NSAP is required per node
- All routers in the same area must have a common Area ID
- Each node in an area must have a unique System ID
- All level 2 routers in a domain must have unique System IDs relative to each other
- All systems belonging to a given domain must have System IDs of the same length in their NSAP addresses

CLNS Addressing

Requirements and Caveats

- Multiple NSAPs allowed on Cisco routers for merging, splitting or renumbering
- All NSAPs on the same router must have the same system ID.
- The maximum size of an NSAP is 20 bytes
- Minimum of 8 bytes allowed on Ciscos.
 - 1 byte for area, 6 bytes for system ID and 1 byte for N-selector.
 - AFI prefix recommended to make minimum of 9 bytes

CLNS Addressing

NSAP Examples

Example 1

47.0001.aaaa.bbbb.cccc.00

Area = 47.0001, SysID = aaaa.bbbb.cccc, NSEL = 00

Example 2

39.0f01.0002.0000.0c00.1111.00

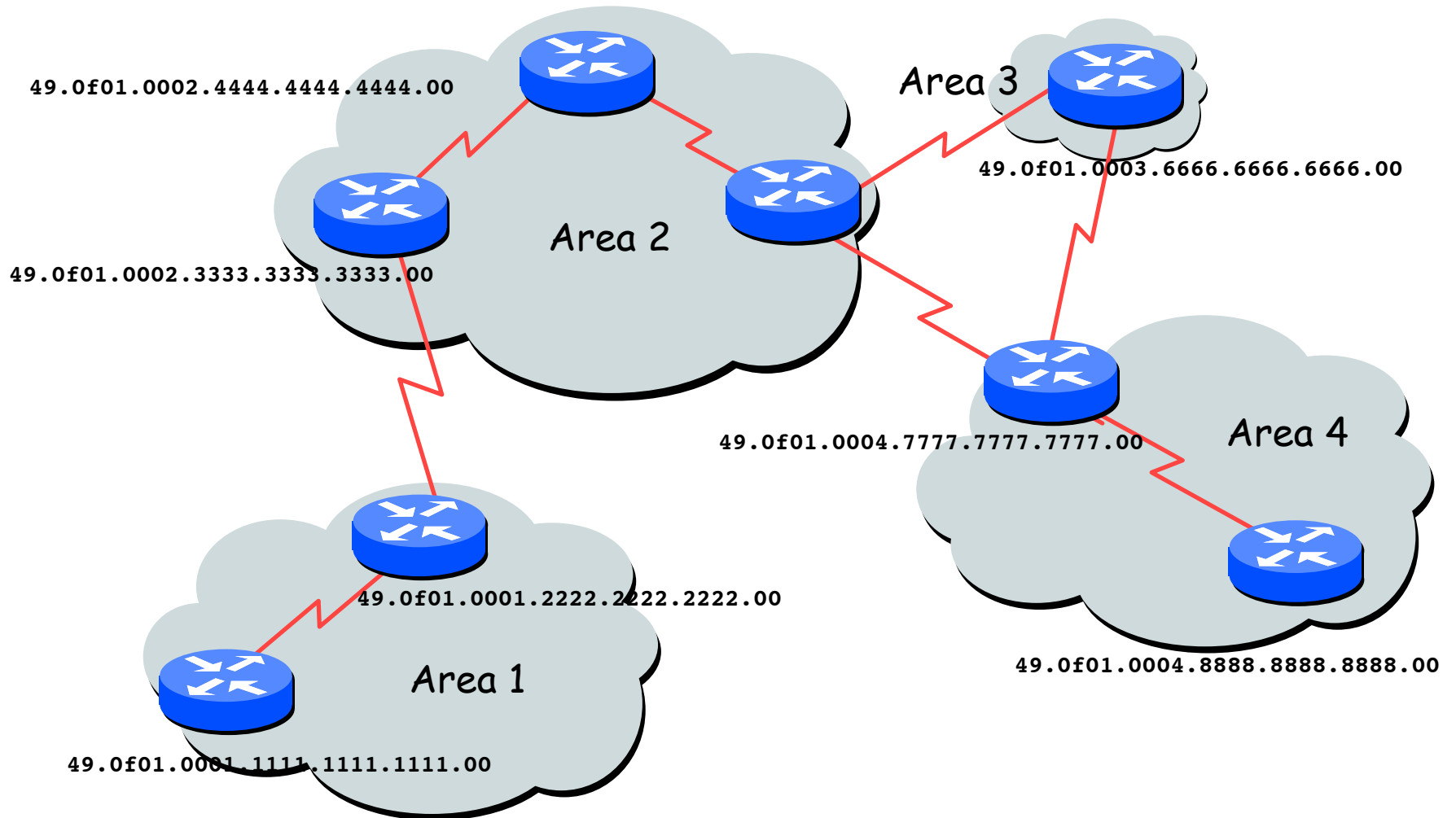
Area = 39.0f01.0002, SysID = 0000.0c00.1111, NSEL = 00

Example 3.

49.0002.0000.0000.0007.00

Area = 49.0002, SysID = 0000.0000.0007, NSEL = 00

An Addressing Example



CLNS Addressing

How do most ISP define System IDs?

```
Interface Loopback 0  
IP address 192.168.3.25
```

```
Router isis  
Net 49.0001.1921.6800.3025.00
```

IP Address conversion process:

192.168.3.25 -> 192.168.003.025



1921.6800.3025



49.0001.1921.6800.3025.00

CLNS Addressing

Globally Unique NSAPs

- AFI 47 (ISO 6523 ICD) is allocated via national sponsoring authority of the International Registration Authority (RA), usually a national standards body
 - NIST - allocated IDI 0005 and 0006
 - BSI subsidiary IOTA allocated 0124 for assignment of ATM End Systems Addresses
- AFI 39 also administered through national institutions
 - IDI 0840 allocated to ANSI

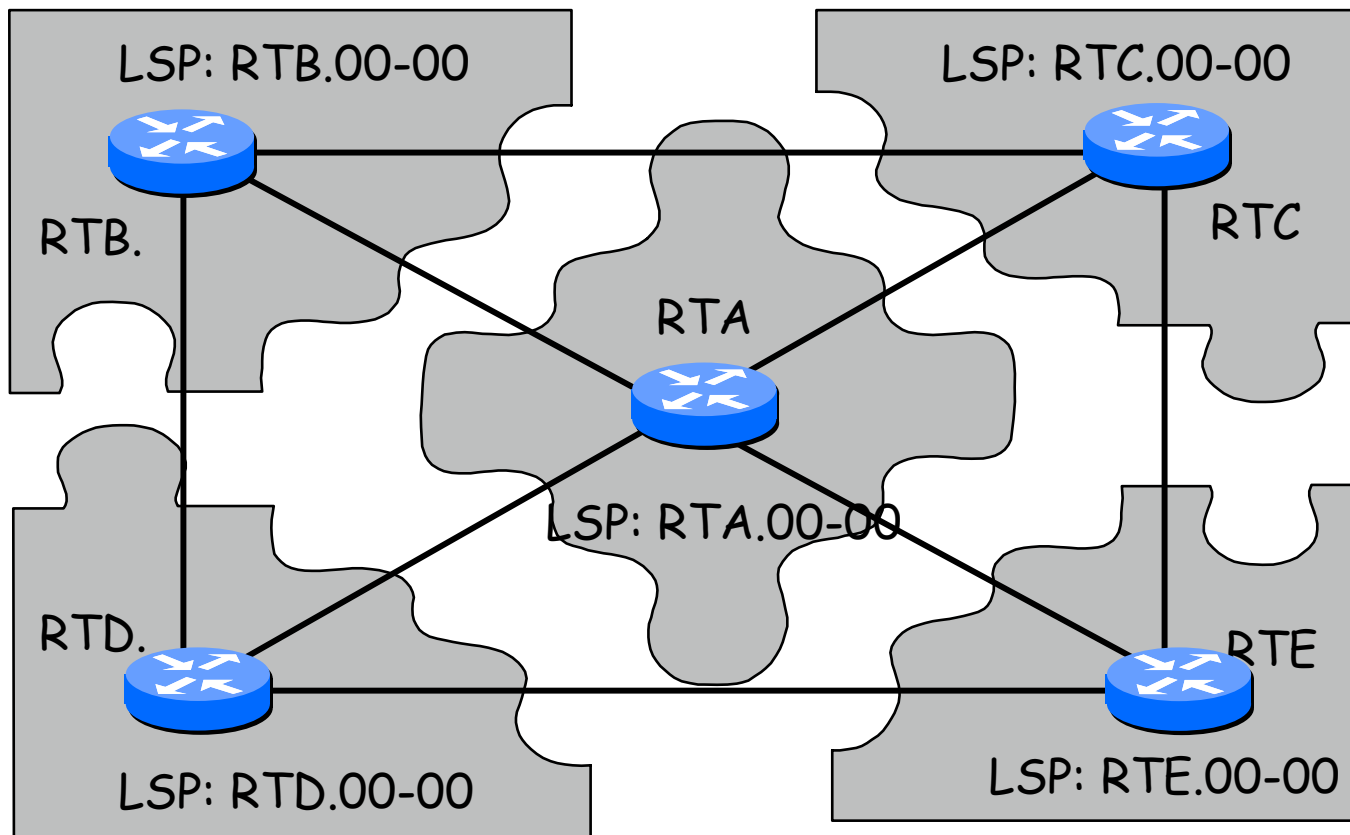
NIST - United States National Institute of Standards

BSI - British Standards Institute

IOTA - Identifiers for Organizations for Telecommunications Addressing

IS-IS LS Database

Link State Packets



IS-IS LS Database

IS-IS Packet Format

IS-IS Packets are made of the following:

- A Fixed Header
 - Contains generic packet information and other specific information about the packet
- Type, Length, Value (TLV) Fields
 - TLVs are blocks of specific routing-related information in IS-IS packets

IS-IS Protocol Concepts

Point-to Adjacencies

Intra-domain Routing Protocol Discriminator			
Length Indicator			
Version/Protocol ID Extension			
ID Length			
R	R	R	PDU Type
Version			
Reserved			
Maximum Area Addresses			
Reserved (6 bits)		Circuit Type	
Source ID			
Holding Time			
PDU Length			
Local Circuit ID			
TLV Fields			

Bytes

1

1

1

1

1

1

1

1

1

ID Length

2

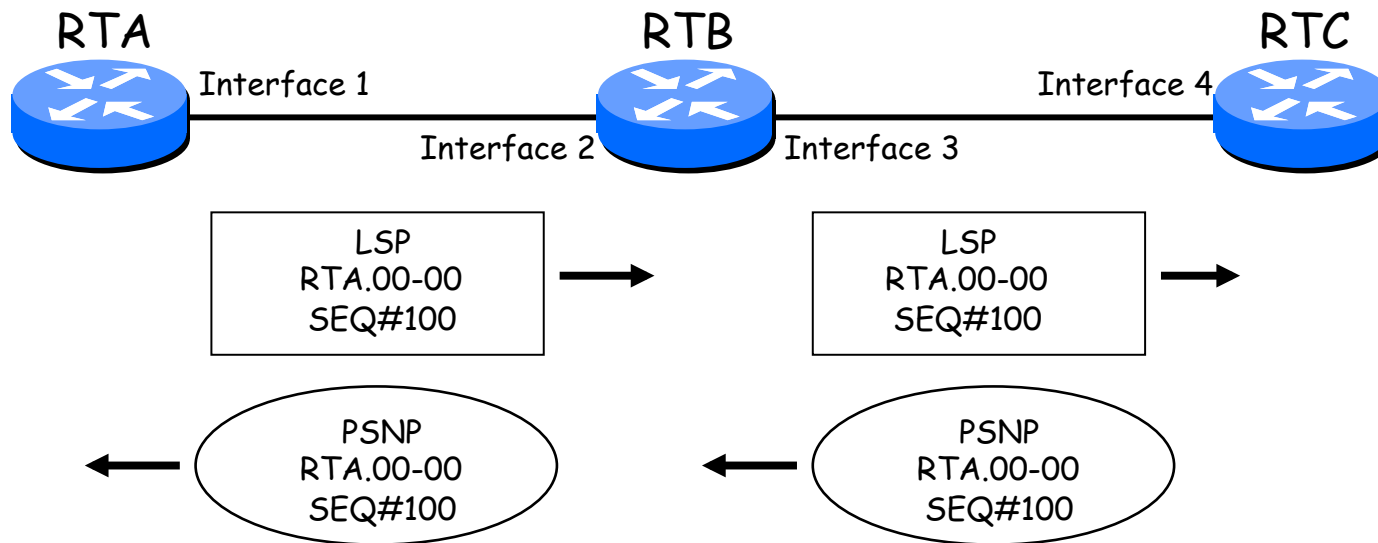
2

1

Variable Length

- Pt-to-pt IIH used to establish level-1 or Level-2 pt-to-point adjacency
- Only two way communication required on pt-to-pt links by ISO 10589
- 3-way reliable process recently proposed for standardization in the IETF. Introduces pt-to-pt adjacency state TLV (Type 240)

Flooding on Pt-to-pt links



IS-IS Database Timers

Timer	Default Value	Cisco IOS Command
Maxage	1200s	isis max-lsp-interval
LSP Refresh Interval	900s	isis refresh-interval
LSP Transmission Interval	33ms	isis lsp-interval
LSP Retransmit Interval	5s	isis retransmit-interval
CSNP Interval	10s	isis csnp-interval

SPF Algorithm

- In default mode, SPF process runs no frequent than every 5s
- Full SPF is run when topology changes
- When leaf elements such as IP prefixes change, routing table is adjusted with Partial Route Calculation (PRC)
- PRC evaluates only routes that changed hence less CPU intensive and relatively fast

SPF Algorithm

- Duration of SPF depends on many factors such as:
 - Number of nodes
 - Number of links
 - Number of IP prefixes
 - Degree of mesh (especially for NBMA)
 - Speed of Route Processor

Synchronous Optical Networks (SONET)

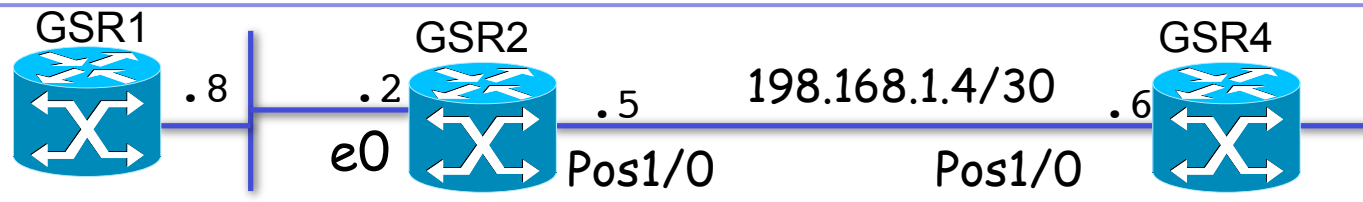
- Packets over SONET or SDH (synchronous digital hierarchy).
Interfaces often called POS.

Wide-Area-Network speeds

- OC-1 is a SONET line with transmission speeds of up to **51.84** Mbit/s.
- OC-3 / STM-1x : **155.52** Mbit/s
- OC-12 / STM-4x : **622.08** Mbit/s
- OC-48 / STM-16x / **2.5G** SONET
- OC-192 / STM-64x / **10G** SONET
- OC-768 / STM-256x / **40G**

Basic Configuration

12.1.1.0/24



```
hostname GSR2
clns routing
!
interface Loopback0
 ip address 13.1.1.2 255.255.255.0
 ip router isis

interface Ethernet0
 ip address 12.1.1.2 255.255.255.0
 ip router isis
!
interface POS2/0
 ip address 10.1.1.1 255.255.255.252
 ip router isis
!
router isis
 net 49.0001.0000.0000.0002.00
!
clns host GSR1 49.0001.0000.0000.0008.00
```

```
hostname GSR4
clns routing
!
interface Loopback0
 ip address 13.1.1.2 255.255.255.0
 ip router isis
!
interface POS2/0
 ip address 10.1.1.2 255.255.255.0
 ip router isis
!
router isis
 net 49.0002.0000.0000.0004.00
```

Verifying Operation

show clns neighbors

```
GSR2#show clns neighbors
```

System Id	Interface	SNPA	State	Holdtime	Type	Protocol
GSR1	Et0	00d0.58eb.d601	Up	8	L1L2	IS-IS
GSR4	PO2/0	*HDLC*	Up	25	L2	IS-IS

```
GSR2#show clns neighbors detail
```

System Id	Interface	SNPA	State	Holdtime	Type	Protocol
GSR1	Et0	00d0.58eb.d601	Up	9	L1L2	IS-IS
Area Address(es): 49.0001						
IP Address(es): 12.1.1.8*						
Uptime: 00:08:57						
GSR4	PO2/0	*HDLC*	Up	24	L2	IS-IS
Area Address(es): 49.0002						
IP Address(es): 10.1.1.2*						
Uptime: 00:24:08						

Verifying Operation

show isis topology

```
GSR2#sh isis topology
```

```
IS-IS paths to level-1 routers
```

System Id	Metric	Next-Hop	Interface	SNPA
GSR2	--			
GSR1	10	GSR1	Et0	00d0.58eb.d601

```
IS-IS paths to level-2 routers
```

System Id	Metric	Next-Hop	Interface	SNPA
GSR2	--			
GSR4	10	GSR4	PO2/0	*HDLC*
GSR1	10	GSR1	Et0	00d0.58eb.d601

Verifying Operation

show isis database level-n detail <lspid>

```
GSR2#show isis database level-1 detail GSR2.00-00
```

```
IS-IS Level-1 LSP GSR2.00-00
```

```
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
```

```
GSR2.00-00          * 0x0000000E    0xDAE4        1197          1/0/0
```

```
Area Address: 49.0001
```

```
NLPID:             0xCC
```

```
Hostname: GSR2
```

```
IP Address:       13.1.1.2
```

```
Metric: 10        IP 12.1.1.0 255.255.255.0
```

```
Metric: 10        IP 10.1.1.0 255.255.255.252
```

```
Metric: 10        IP 13.1.1.2 255.255.255.255
```

```
Metric: 10        IS GSR2.02
```

```
Metric: 10        IS GSR1.01
```

```
Metric: 0         ES GSR2
```

Verifying Operation

show isis database level-n detail <lspid>

```
GSR2#sh isis dat level-1 detail GSR1.01-00
```

```
IS-IS Level-1 LSP GSR1.01-00
```

LSPID	LSP Seq Num	LSP Checksum	LSP Holdtime	ATT/P/OL
GSR1.01-00	0x00000007	0xAF8E	616	0/0/0
Metric: 0	IS GSR1.00			
Metric: 0	IS GSR2.00			

- Pseudonode LSP (GSR1.01-00) is generated by GSR1 which is DIS on ethernet0 of GSR2
- PSN LSP Lists all known routers connected to LAN

Verifying Operation

show ip route [isis]

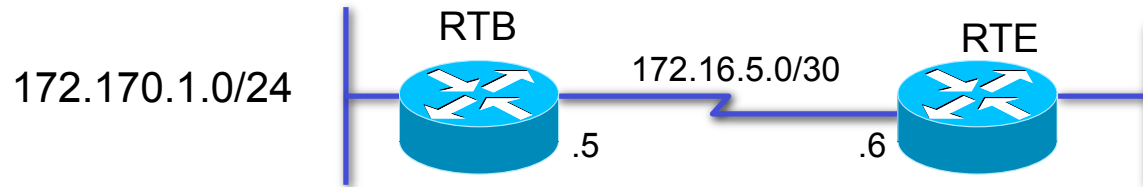
```
GSR2#sh ip route
Codes: C - connected, S - static,
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area

       10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, POS2/0
       12.0.0.0/24 is subnetted, 1 subnets
C       12.1.1.0 is directly connected, Ethernet0
       13.0.0.0/32 is subnetted, 3 subnets
i L1    13.1.1.8 [115/20] via 12.1.1.8, Ethernet0
i L2    13.1.1.4 [115/20] via 10.1.1.2, POS2/0
C       13.1.1.2 is directly connected, Loopback0
```

Typical ISP Router Configuration

```
GSR1#
interface Loopback0
ip address 172.160.250.1 255.255.255.255
!
interface POS1/0
ip address 192.168.1.1 255.255.255.0
isis metric 100 level-2
isis hello-interval 12 level-2
isis hello-multiplier 5 level-2
isis retransmit-interval 100
!
router isis SJ
summary-address 172.160.0.0 255.255.0.0
passive-interface Loopback0
distance 15 ip
net 49.0001.0001.0000.0001.0002.0001.1721.6025.0001.00
is-type level-2-only
metric-style wide
spf-interval 30
log-adjacency-changes
```

Summarization



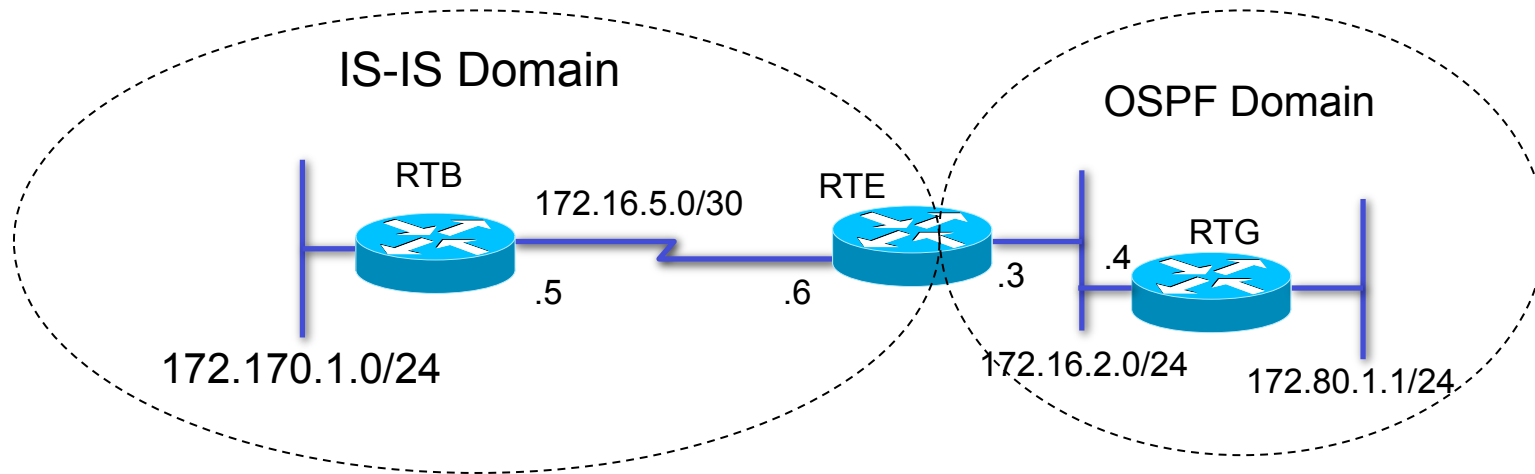
```
hostname RTB
!  
interface Ethernet0  
 ip address 172.170.1.1 255.255.255.0  
 ip router isis  
!  
router isis  
 summary-address 172.170.0.0 255.255.0.0  
 net 49.0001.0000.0000.0001.00
```

Summarization

```
RTE#sh ip route
Gateway of last resort is not set
  i L2 172.170.0.0/16 [115/20] via 172.16.5.5, Serial 0
172.16.0.0/16 is subnetted, 1 subnets
C      172.16.5.4/30 is directly connected, Serial0
```

```
RTB#sh isis dat RTB.00-00 l2 detail
IS-IS Level-2 LSP RTB.00-00
LSPID                LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
RTB.00-00            * 0x00000096  0x86F6                877           0/0/0
  Area Address: 49.0001
  NLPID:          0x81 0xCC
  IP Address:    172.170.1.1
  Metric: 10 IS RTB.02
  Metric: 10 IS RTE.00
  Metric: 10 IS RTF.00
  Metric: 10 IP 172.16.5.4 255.255.255.252
  Metric: 10 IP 172.170.0.0 255.255.0.0
```

Redistribution



```
RTE
router ospf 1
 network 172.16.2.0 0.0.0.255 area 0
!
router isis
 redistribute ospf 1 metric 20 metric-type internal level-2
 net 49.0002.0000.0000.0002.00
```


Troubleshooting CLNS Commands

- show clns int
- show clns protocol
- show clns neighbors detail
- show clns is-neighbors
- show clns es-neighbors
- show clns route
- show clns cache
- show clns traffic
- show isis spf-log
- show isis database detail
- show isis database<lspid>
- show isis route
- show isis database L1|L2

Troubleshooting SPF Logs

```
RTB#sh isis spf-log
```

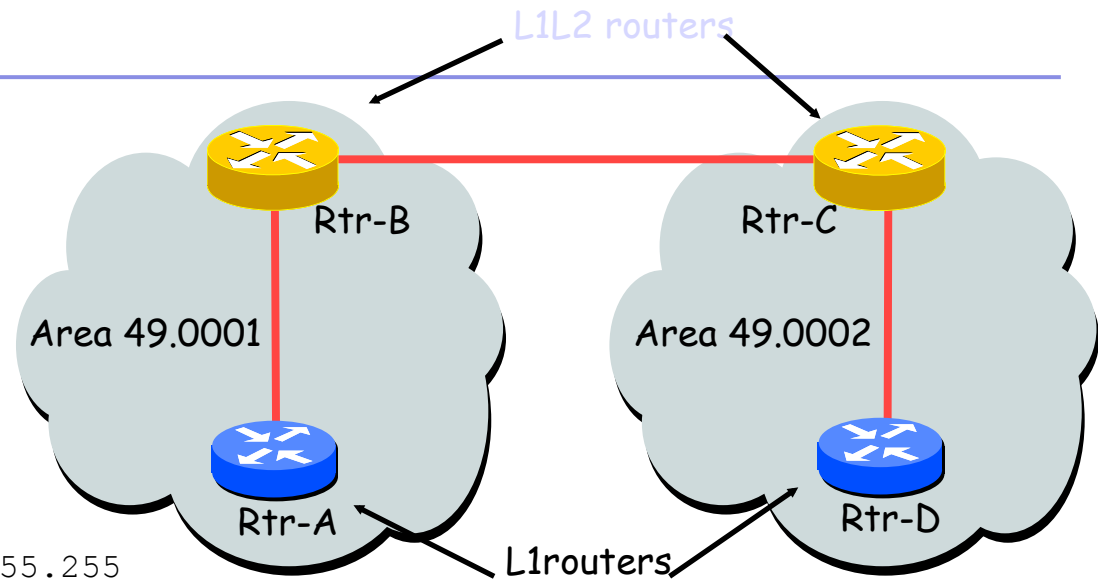
```
Level 1 SPF log
```

When	Duration	Nodes	Count	Triggers	
00:25:27	8	4	1		PERIODIC
00:18:09	12	5	2		NEWLSP TLVCONTENT
00:10:27	8	5	1		PERIODIC

```
Level 2 SPF log
```

When	Duration	Nodes	Count	Triggers	
00:40:35	8	3	1		PERIODIC
00:25:35	8	3	1		PERIODIC
00:18:17	8	3	1		TLVCONTENT
00:10:34	8	3	1		PERIODIC

Configuration for A&B



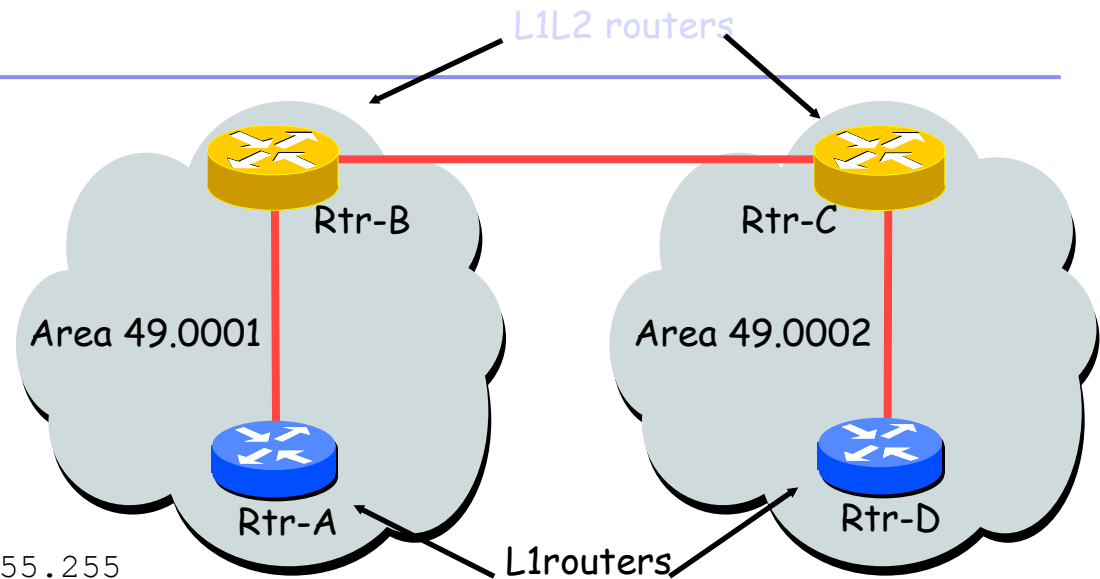
Router-B

```
Interface Loopback0
 ip address 192.168.1.1 255.255.255.255
!
Interface Pos2/0/0
 ip address 192.168.222.1 255.255.255.0
 ip router isis
 isis circuit-type level-2
!
FastEthernet4/0/0
 ip address 192.168.120.10 255.255.255.0
 ip router isis
 isis circuit-type level-1
!
router isis
 passive-interface Loopback0
 net 49.0001.1921.6800.1001.00
```

Router-A

```
Interface Loopback0
 ip address 192.168.1.5 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.120.5 255.255.255.0
 ip router isis
!
router isis
 is-type level-1
 passive-interface Loopback0
 net 49.0001.1921.6800.1005.00
```

Configuration for C&D



Router-C

```
Interface Loopback0
 ip address 192.168.2.2 255.255.255.255
!
Interface Pos1/0/0
 ip address 192.168.222.2 255.255.255.0
 ip router isis
 isis circuit-type level-2
!
interface Fddi3/0
 ip address 192.168.111.2 255.255.255.0
 ip router isis
 isis circuit-type level-1
!
router isis
 passive-interface Loopback0
 net 49.0002.1921.6800.2002.00
```

Router-D

```
Interface Loopback0
 ip address 192.168.2.4 255.255.255.255
!
interface Fddi6/0
 ip address 192.168.111.4 255.255.255.0
 ip router isis
!
router isis
 is-type level-1
 passive-interface Loopback0
 net 49.0002.1921.6800.2004.00
```

Adding interfaces to ISIS

- To activate ISIS on an interface:
 - `interface HSSI 4/0`
 - `ip route isis`
 - `isis circuit-type level-2`
- To disable ISIS on an interface:
 - `router isis`
 - `passive-interface GigabitEthernet 0/0`
 - Disables CLNS on that interface
 - Puts the interface subnet address into the LSDB
- No ISIS configuration on an interface
 - No CLNS run on interface, no interface subnet in the LSDB

Adding interfaces to ISIS

- **Scaling ISIS: passive-interface default**
 - Disables ISIS processing on all interfaces apart from those marked as no-passive
 - Places all IP addresses of all connected interfaces into ISIS
 - Must be at least one non-passive interface:
 - `router isis`
 - `passive-interface default`
 - `no passive-interface GigabitEthernet 0/0`
 - `interface GigabitEthernet 0/0`
 - `ip router isis`
 - `isis metric 1 level-2`

Network Design Issues

- As in all IP network designs, the key issue is the addressing lay-out
- ISIS supports a large number of routers in a single area
- When using areas, use summary-addresses
- >400 routers in the backbone is quite doable

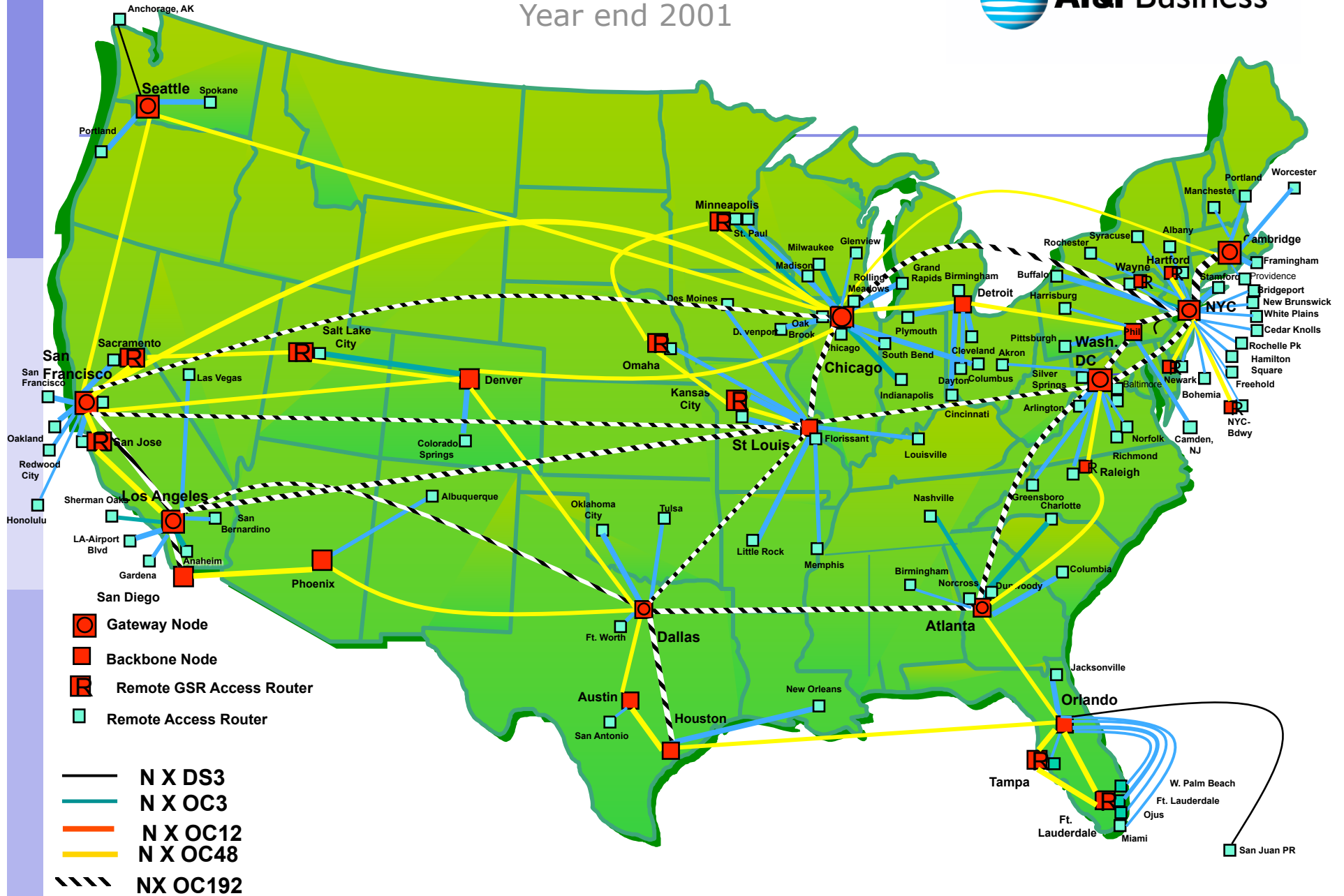
Border Gateway Protocol



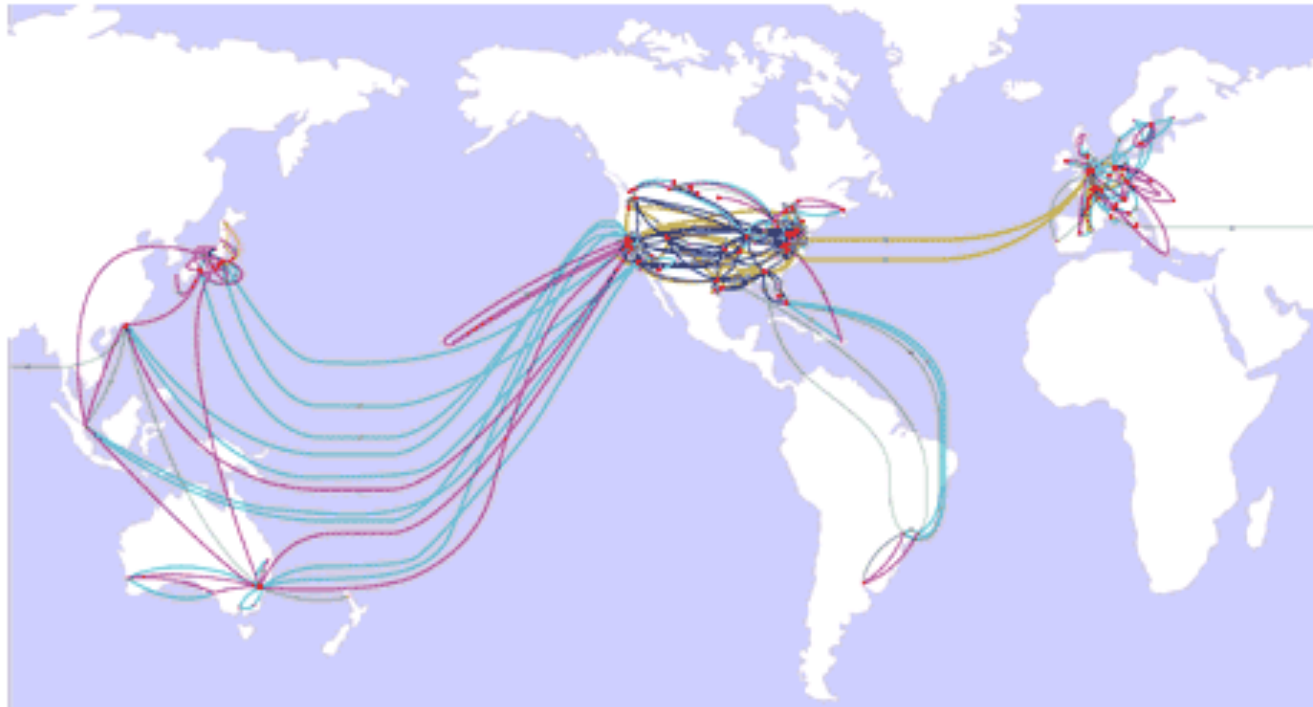
Introduction

AT&T IP Backbone

Year end 2001

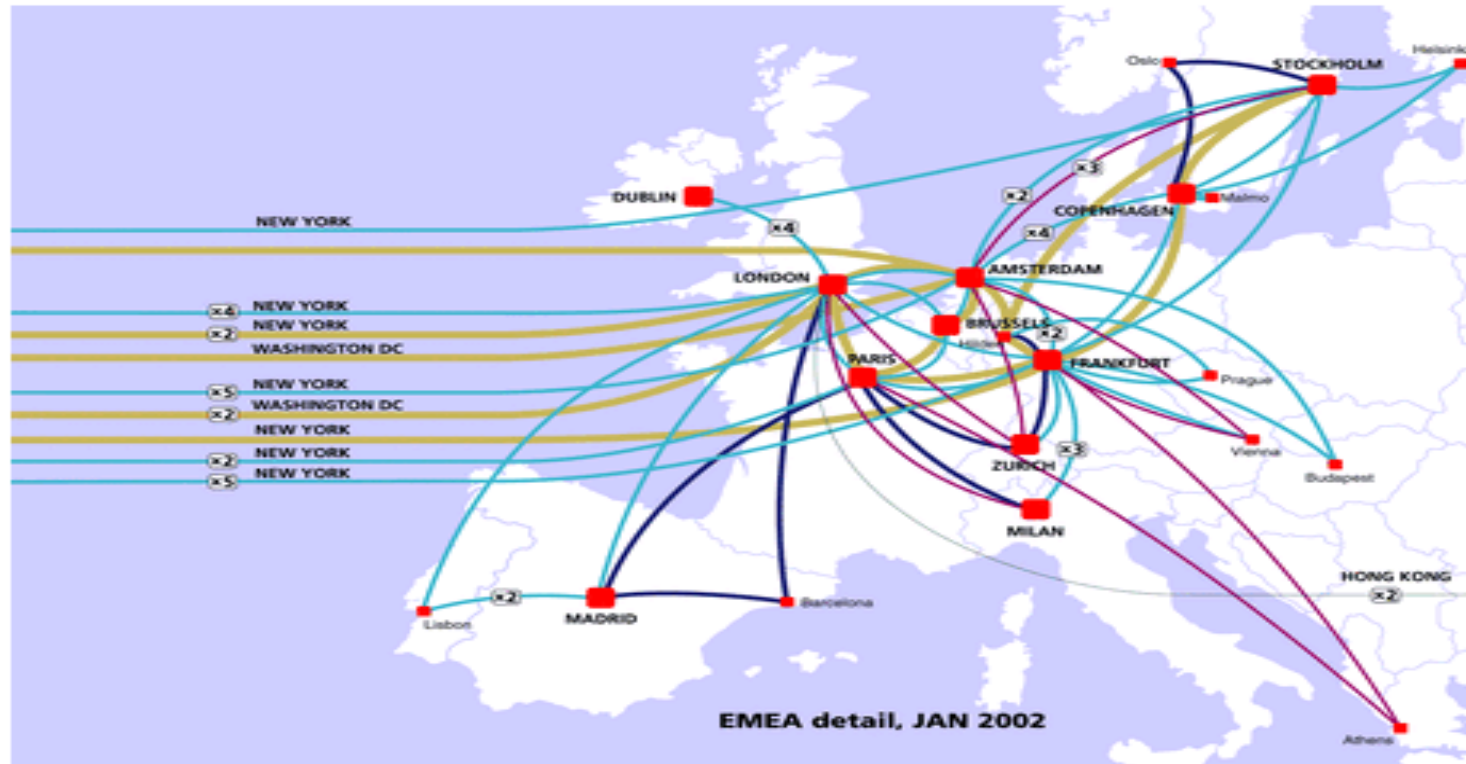


Verizon (UUNet)



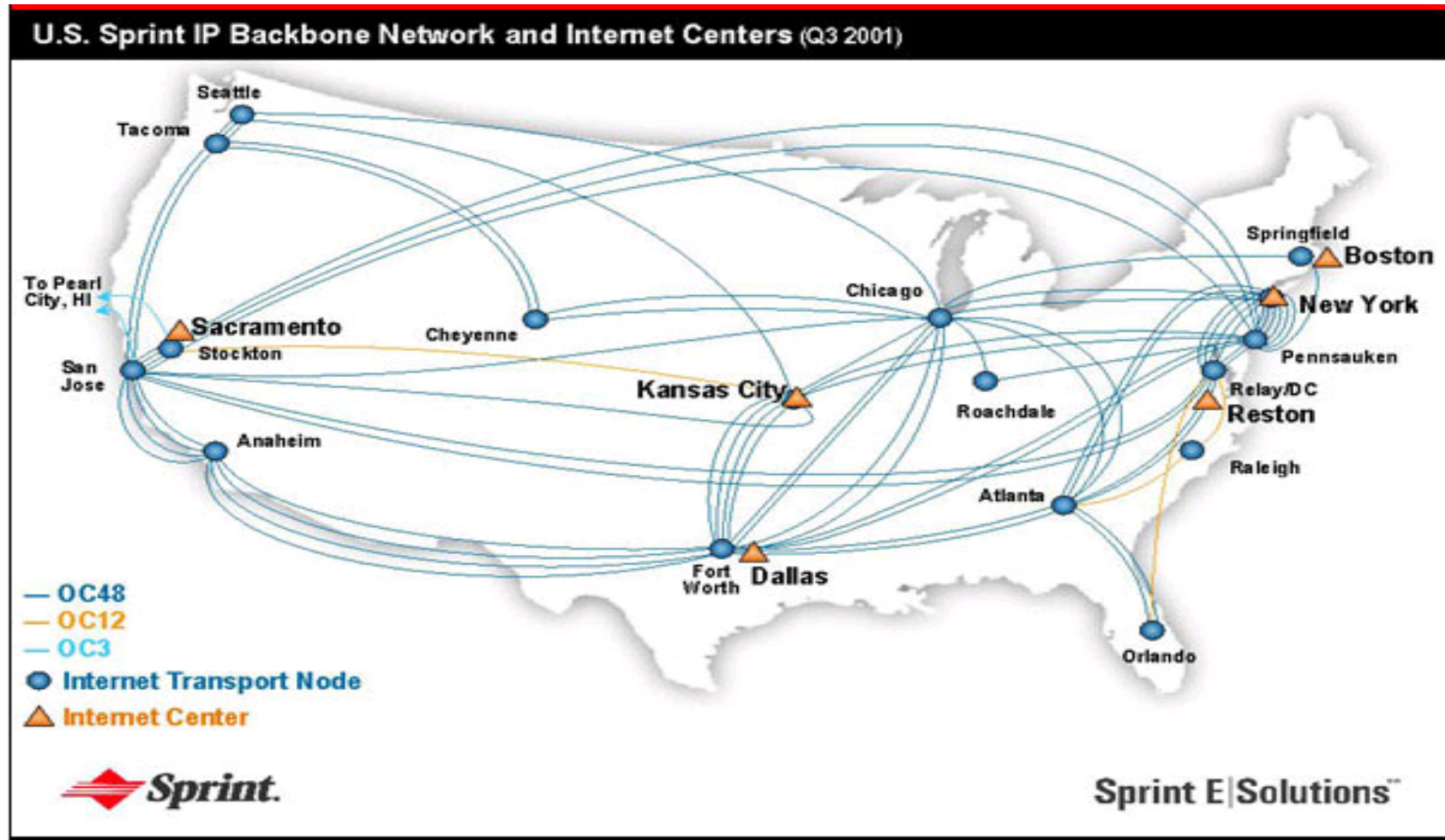
- | | |
|-------------------------------|--------------------------|
| — 64 Kbps | — OC12c/STM4 (622 Mbps) |
| — T1/E1 (1.5 Mbps/2 Mbps) | — OC48c/STM16 (2.5 Gbps) |
| — E3/T3/DS3 (35 Mbps/45 Mbps) | — OC192c/STM64 (10 Gbps) |
| — T2 (6 Mbps) | ● Single Hub City |
| — OC3c/STM1 (155 Mbps) | ■ Multiple Hubs City |
| | ■ Data Center Hub |

Verizon, Europe

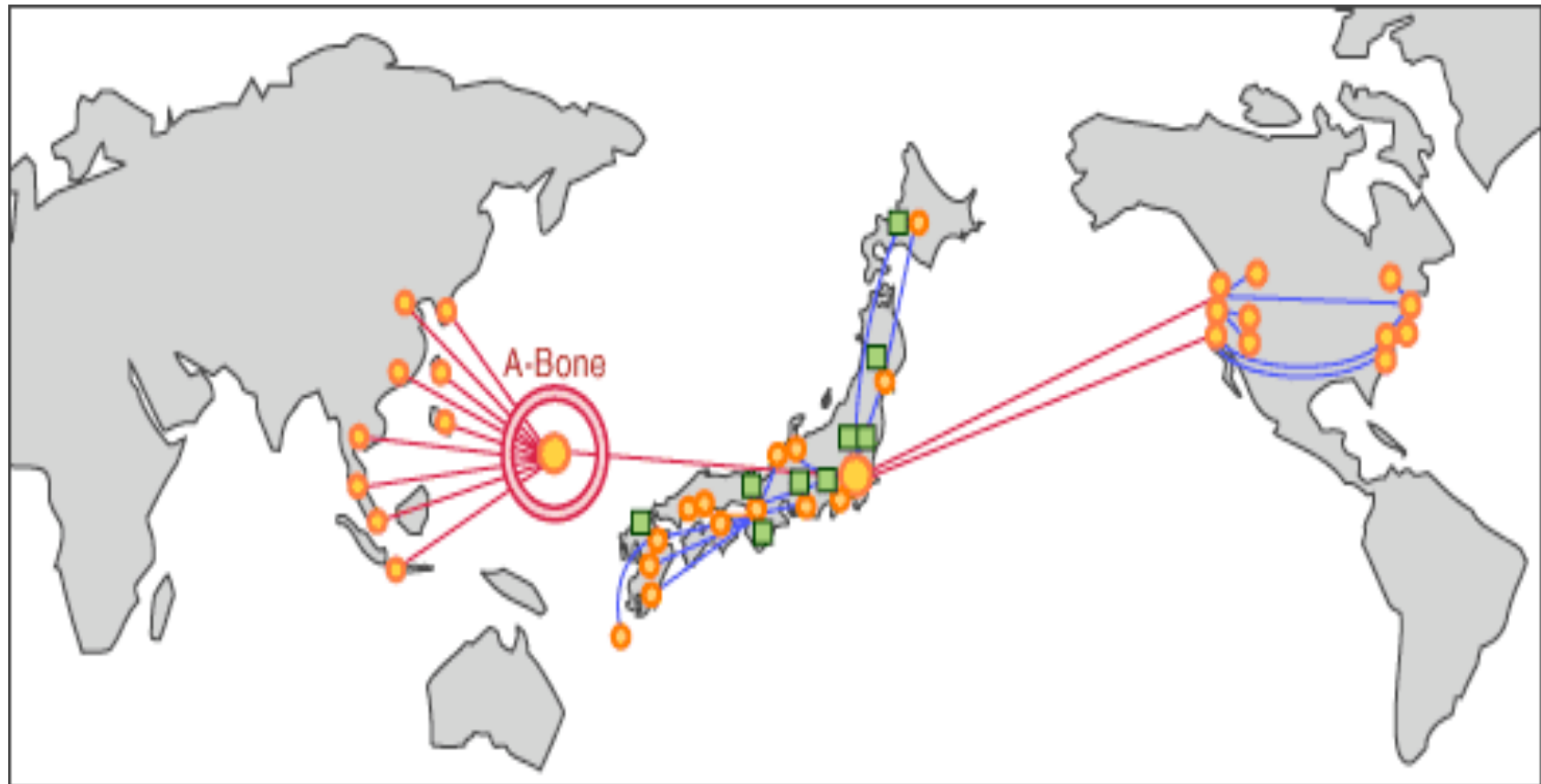


- | | |
|-------------------------------|--------------------------|
| — 64 Kbps | — OC12c/STM4 (622 Mbps) |
| — T1/E1 (1.5 Mbps/2 Mbps) | — OC48c/STM16 (2.5 Gbps) |
| — E3/T3/DS3 (35 Mbps/45 Mbps) | — OC192c/STM64 (10 Gbps) |
| — T2 (6 Mbps) | • Single Hub City |
| — OC3c/STM1 (155 Mbps) | ■ Multiple Hubs City |
| | ■ Data Center Hub |

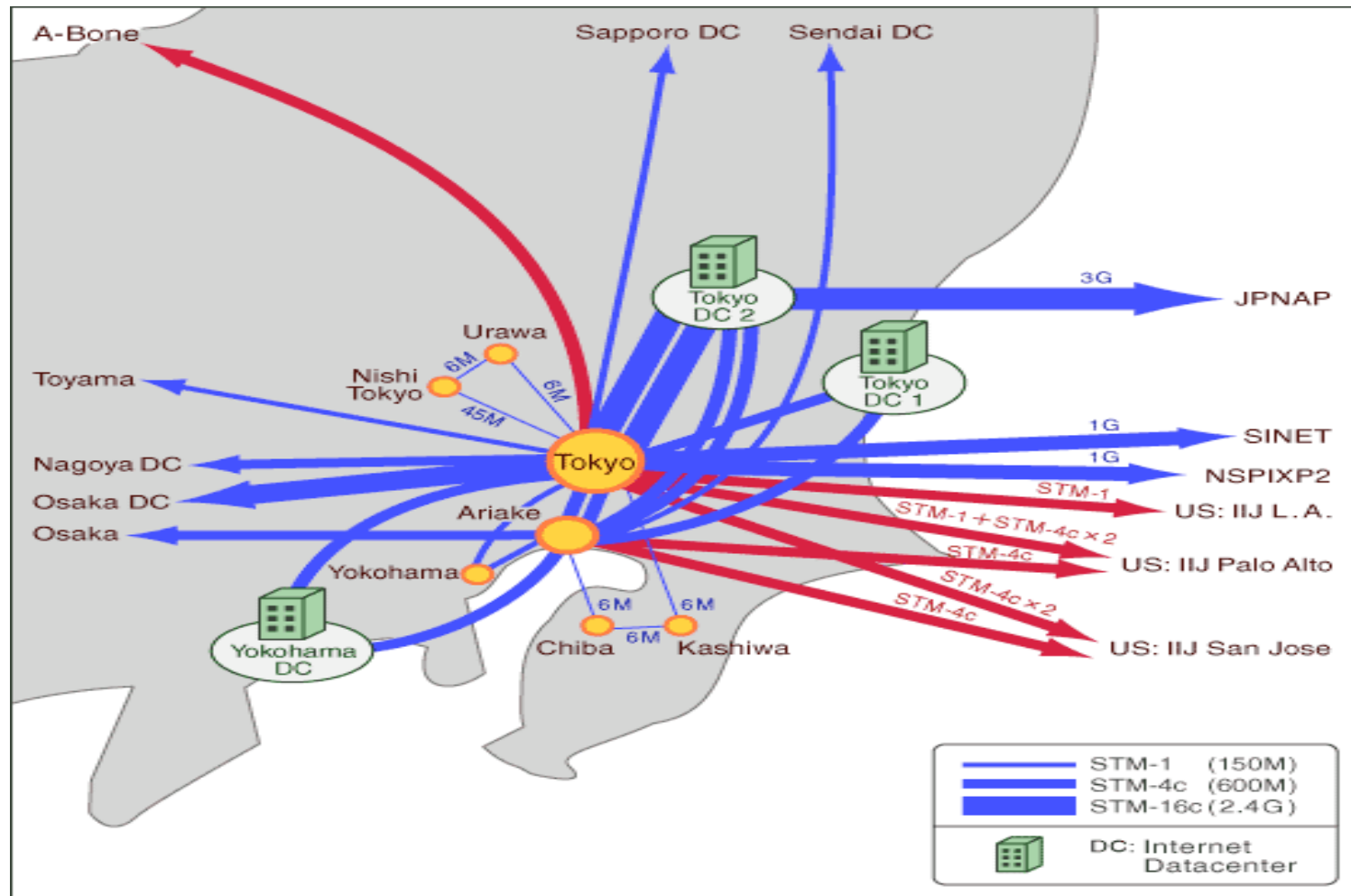
Sprint, USA



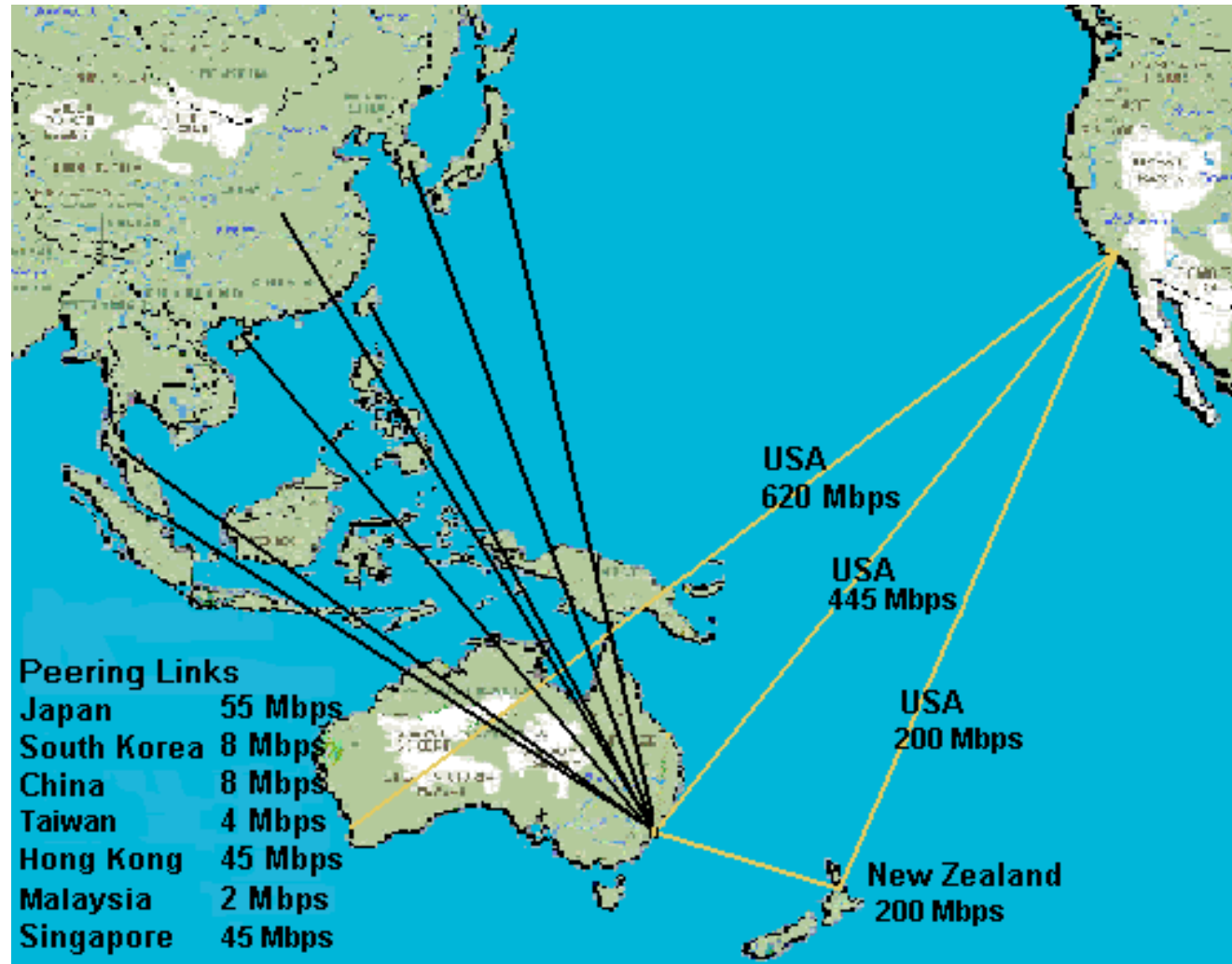
Internet Initiative Japan (IIJ)



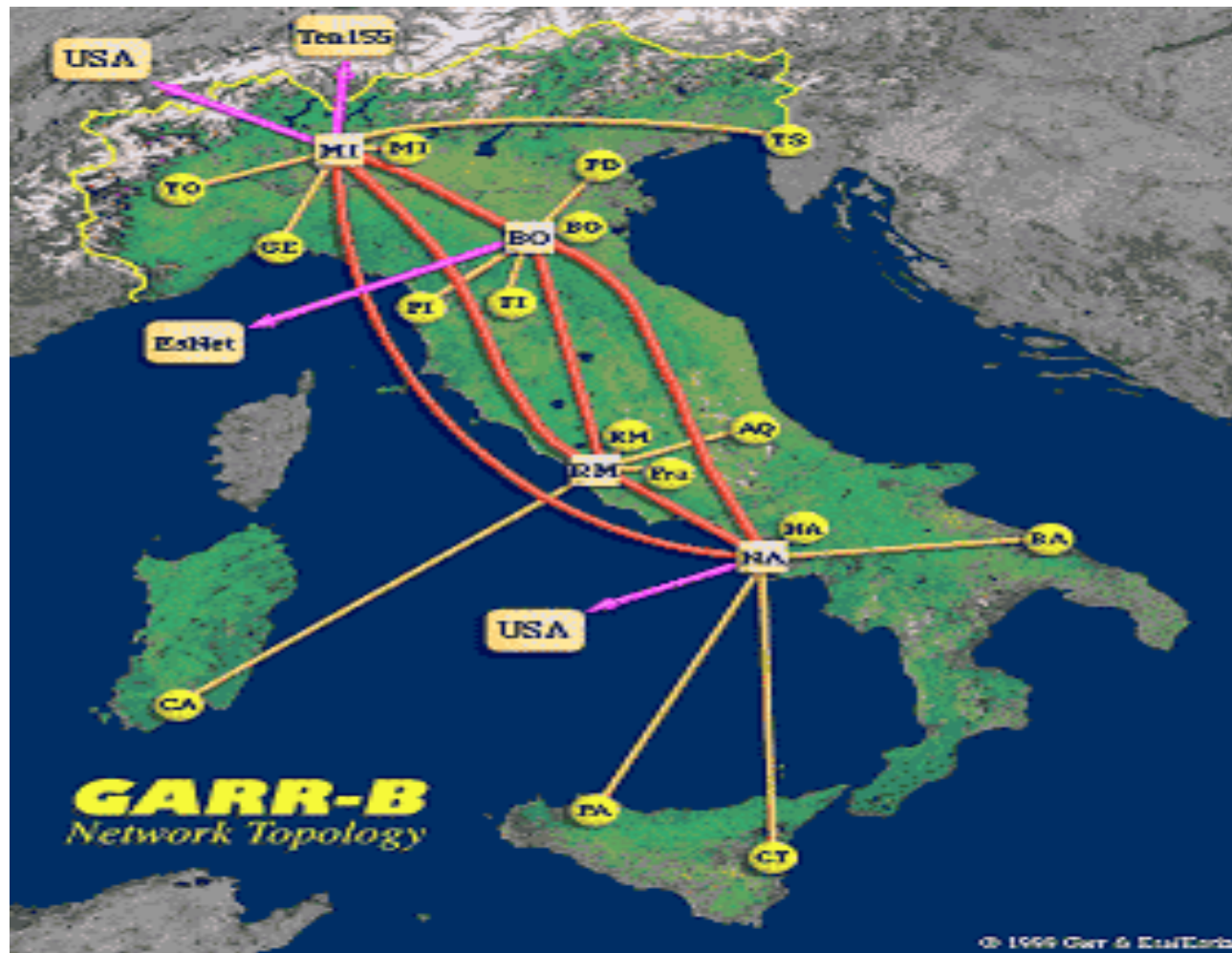
IIJ, Tokyo



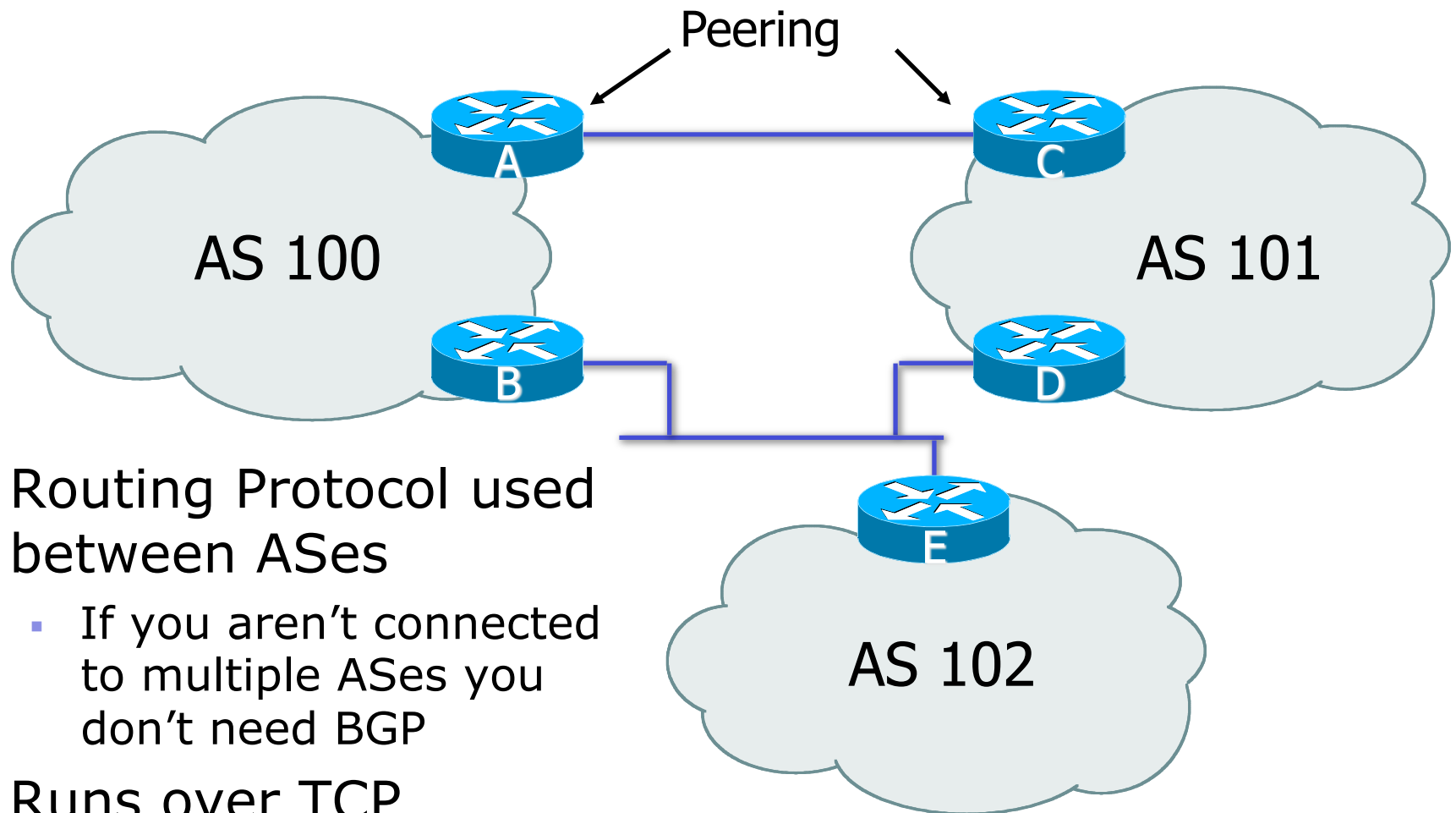
Telstra international



GARR-B



BGP Protocol Basics

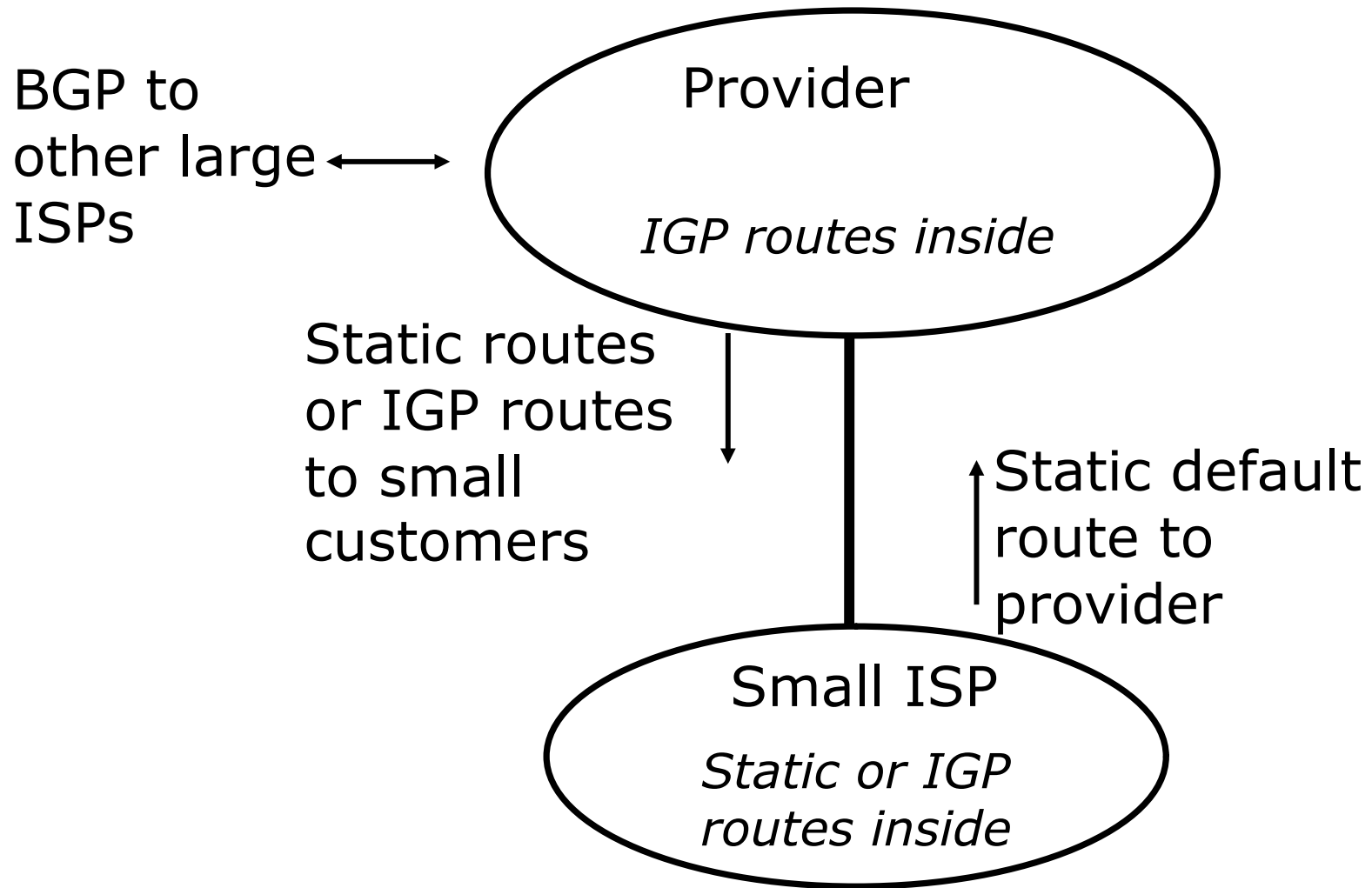


- Routing Protocol used between ASes
 - If you aren't connected to multiple ASes you don't need BGP
- Runs over TCP

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

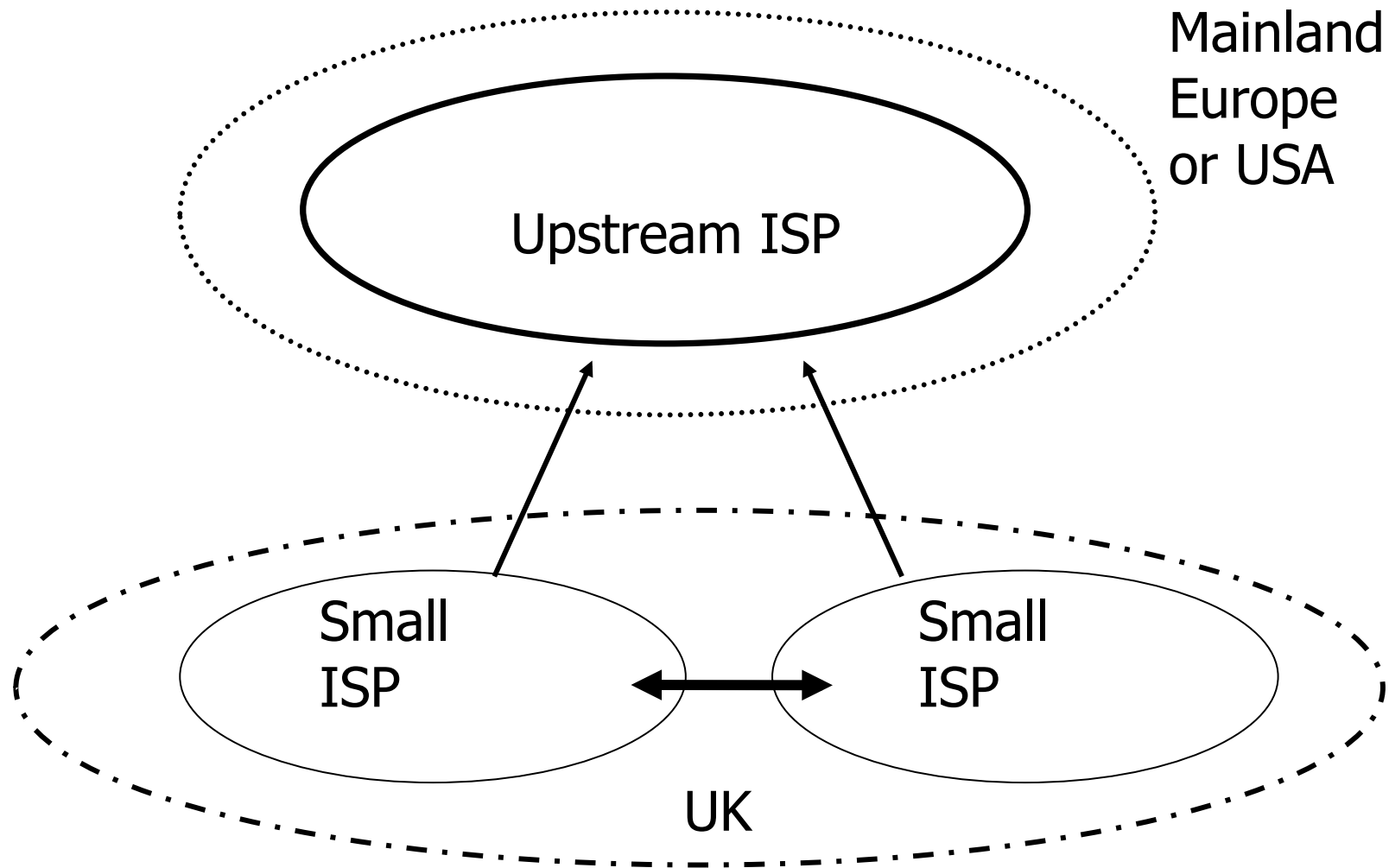
Small ISP with one upstream provider



What happens with other ISPs in the same region/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

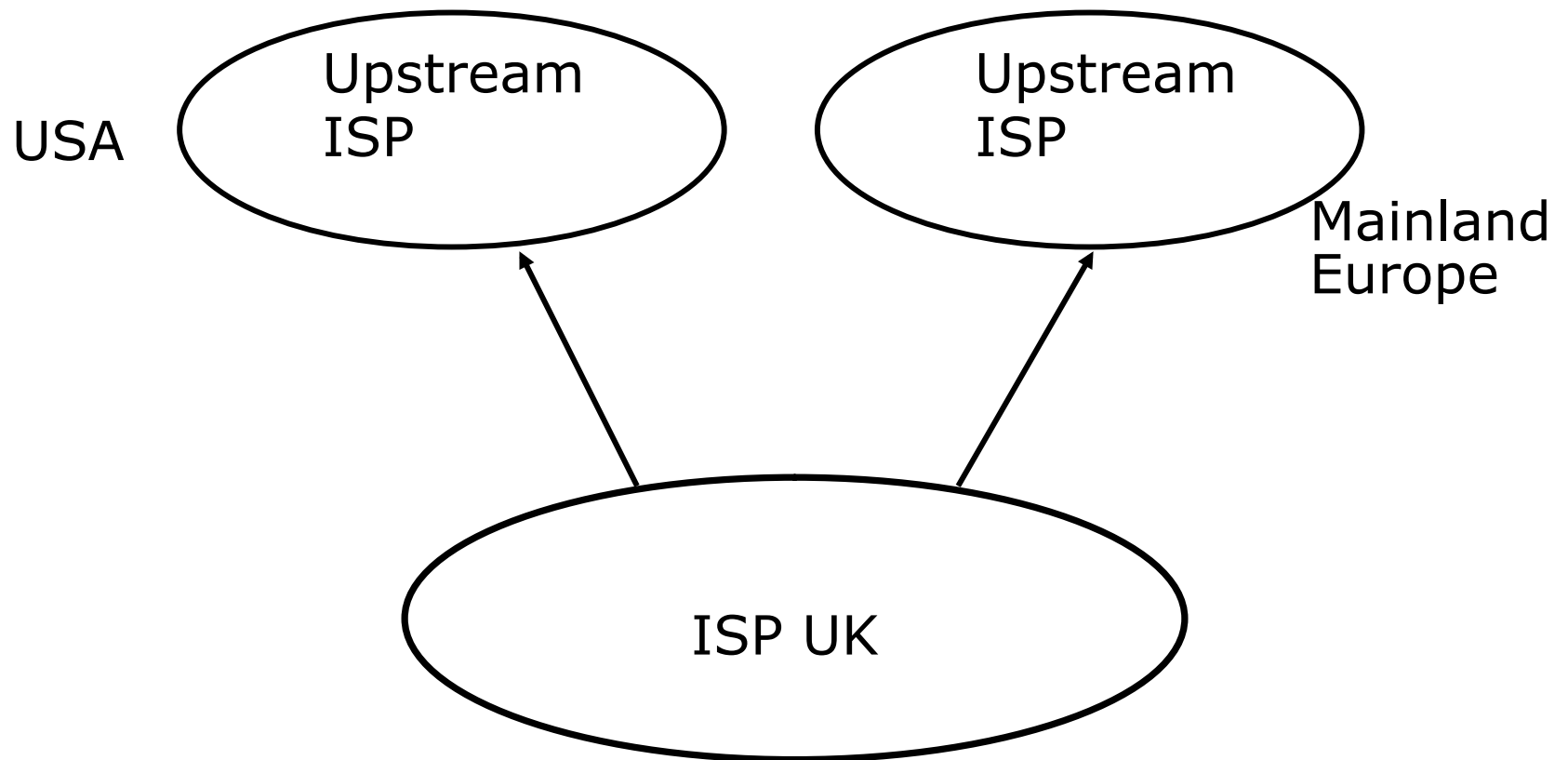
Keeping Local Traffic Local



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:

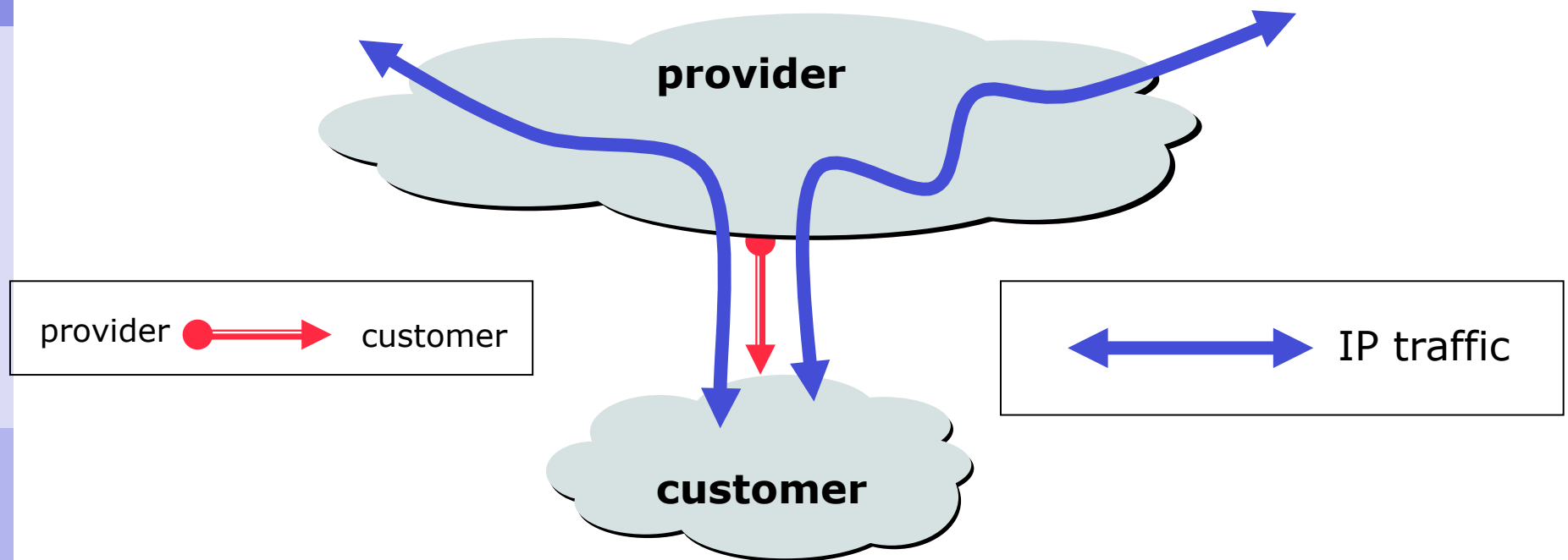
“Peering” and “Transit”

- **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 through the other ISP to other ISP networks
 - ... getting connectivity to rest of world (or part thereof)
 - Usually at cost (customer-provider relationship)

Terminology: “Aggregation”

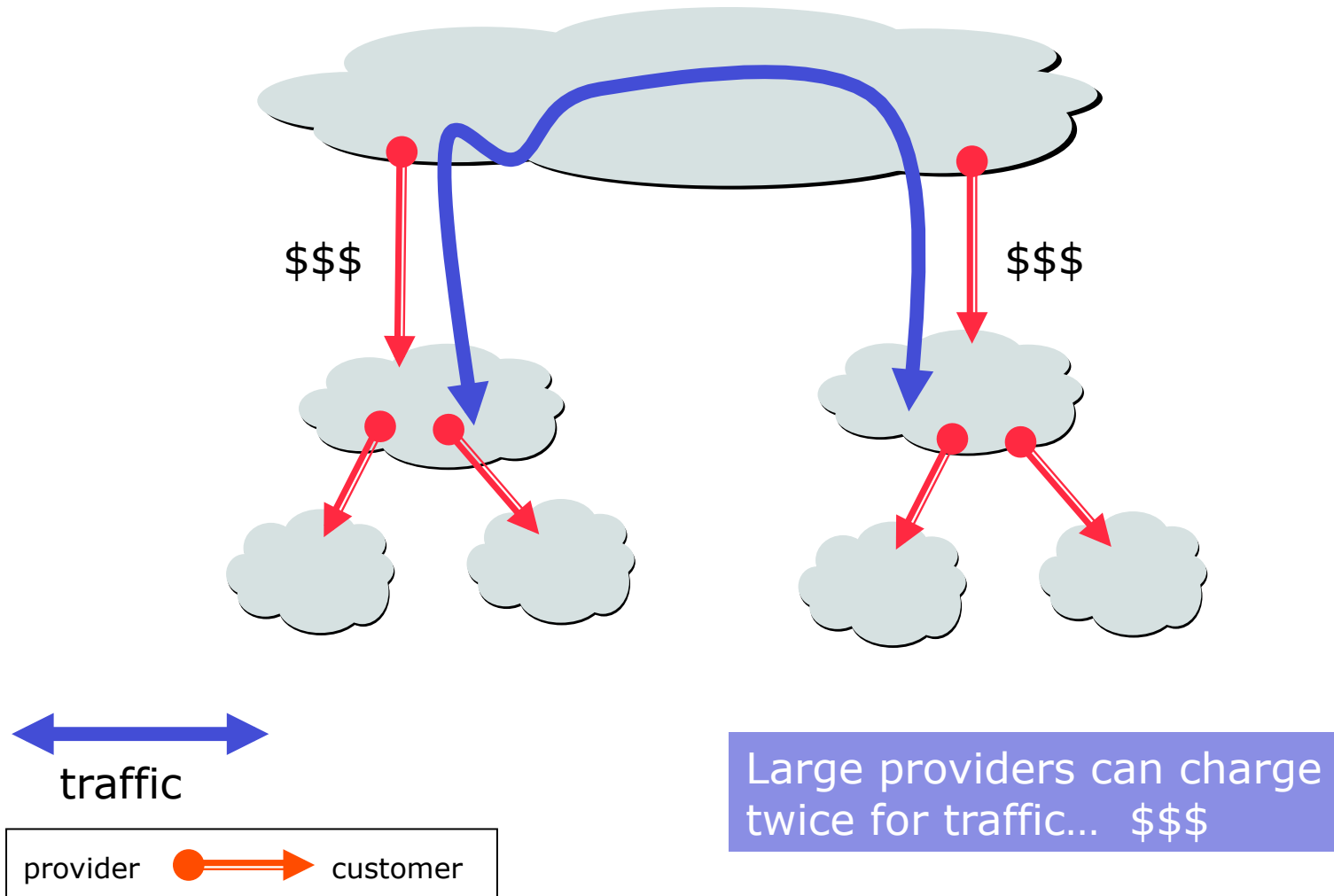
- Combining of several smaller blocks of address space into a larger block
- For example:
 - 192.168.4.0/24 and 192.168.5.0/24 are contiguous address blocks
 - They can be combined and represented as 192.168.4.0/23...
 - ...with no loss of information!

Customers and Providers

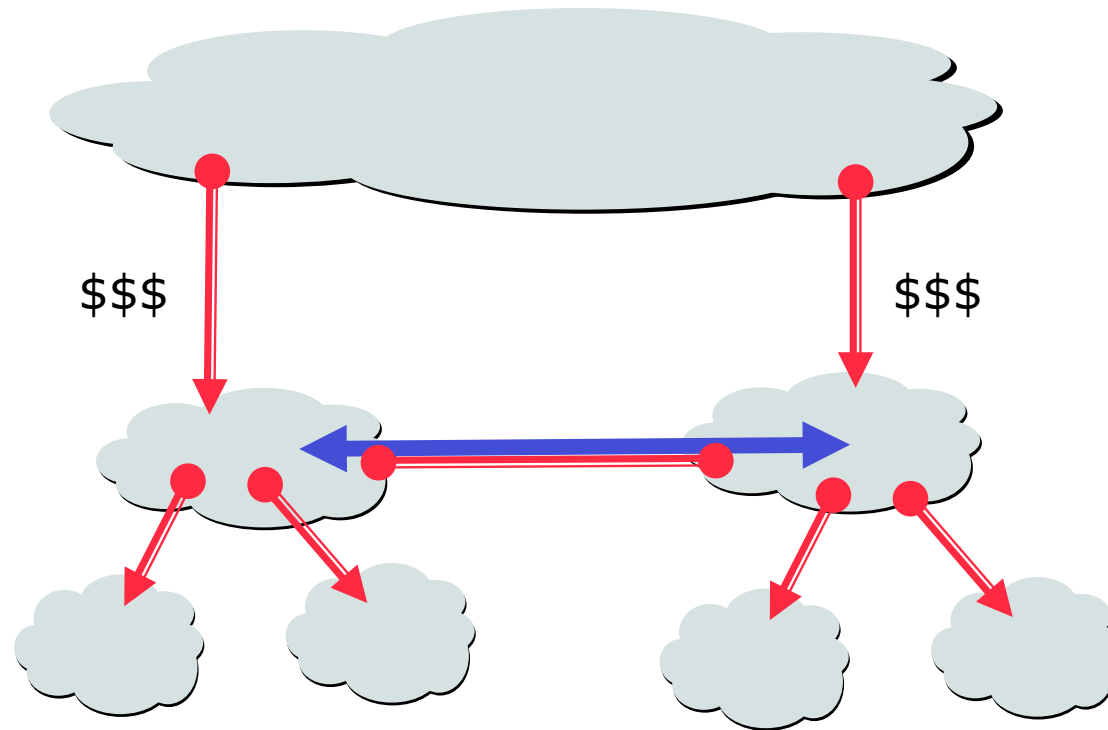


Customer pays provider for access to the Internet

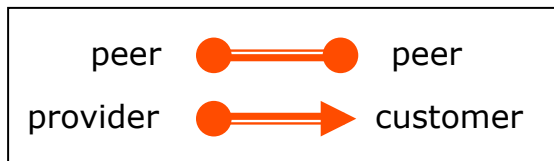
Big tier-1 providers



The “Peering” Relationship

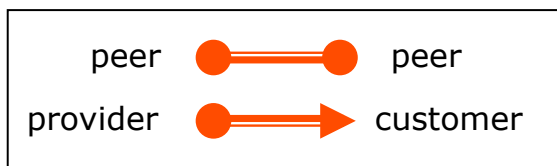
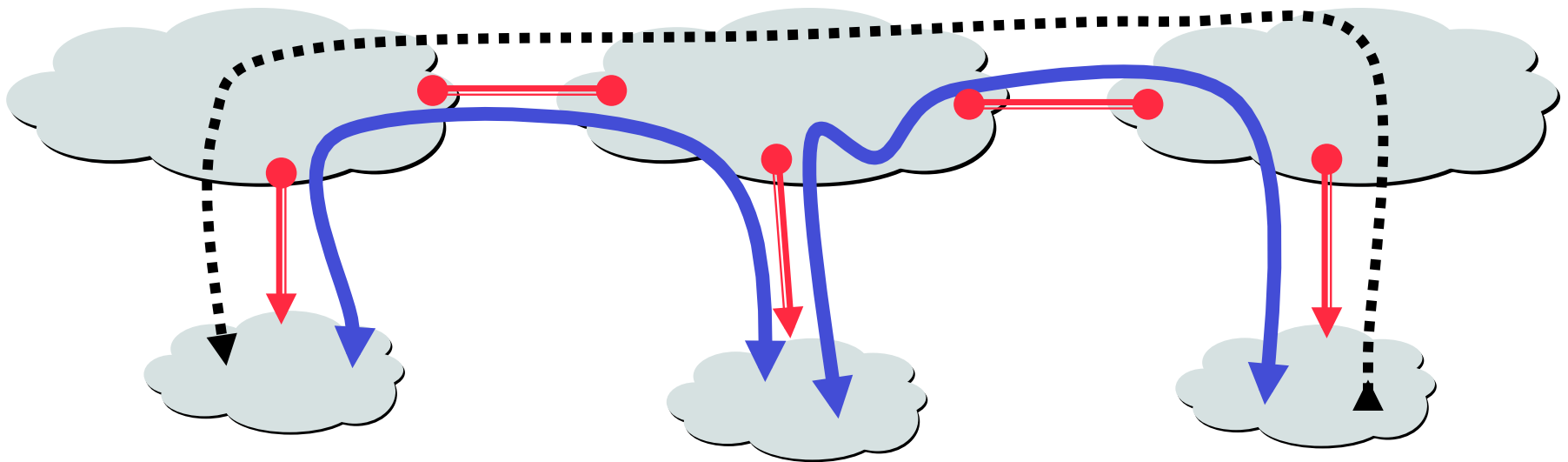


traffic



Peerings are mutual agreements.
Both partners benefit...

The "Peering" Relationship



traffic
allowed



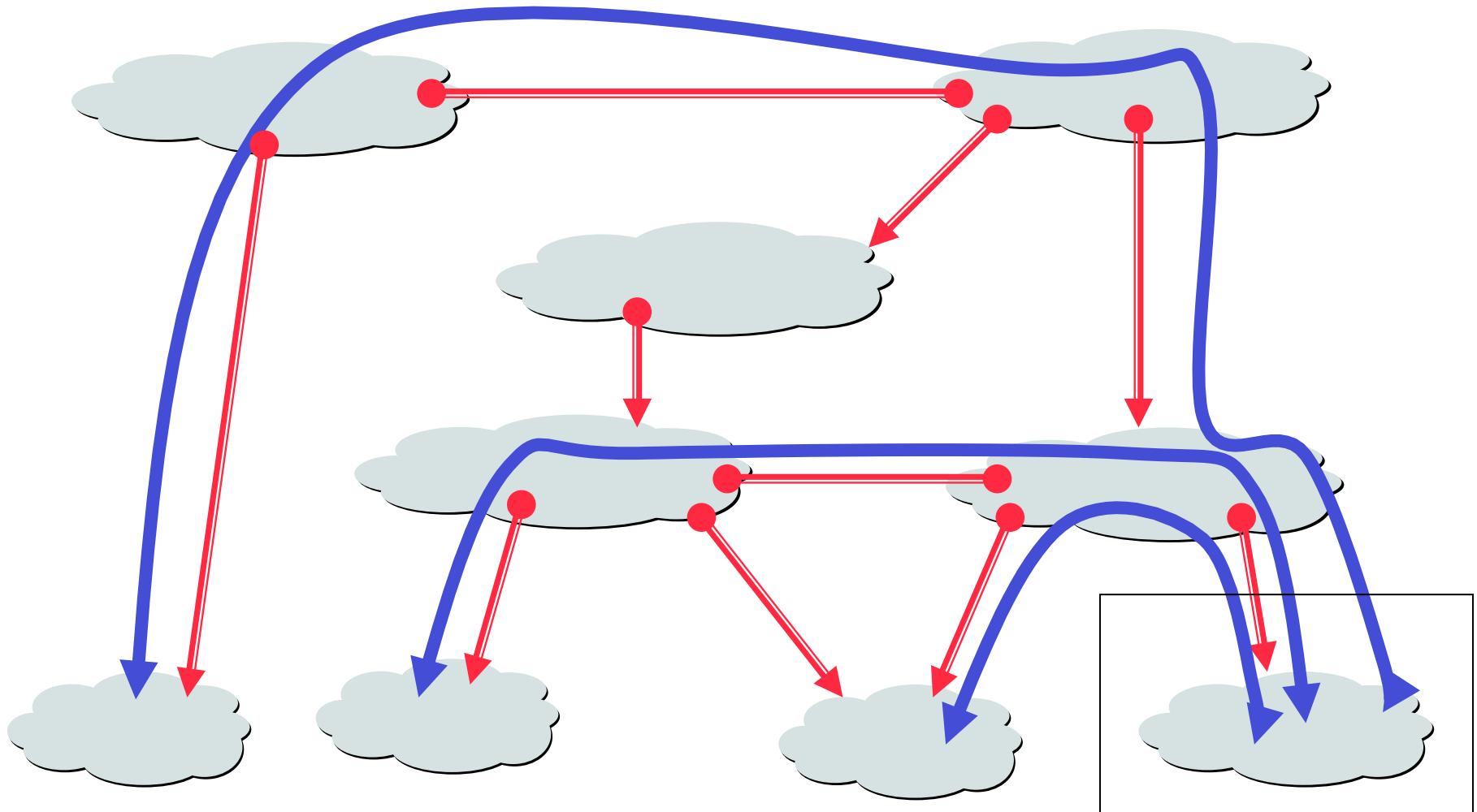
traffic NOT
allowed

Peers provide transit between their respective customers

Peers do not provide transit between peers

Peers (often) do not exchange \$\$\$

Economic Relationships can get complex



Peering Wars

Peer

- Reduces upstream transit costs
- Can increase end-to-end performance
- May be the only way to connect your customers to some part of the Internet ("Tier 1")

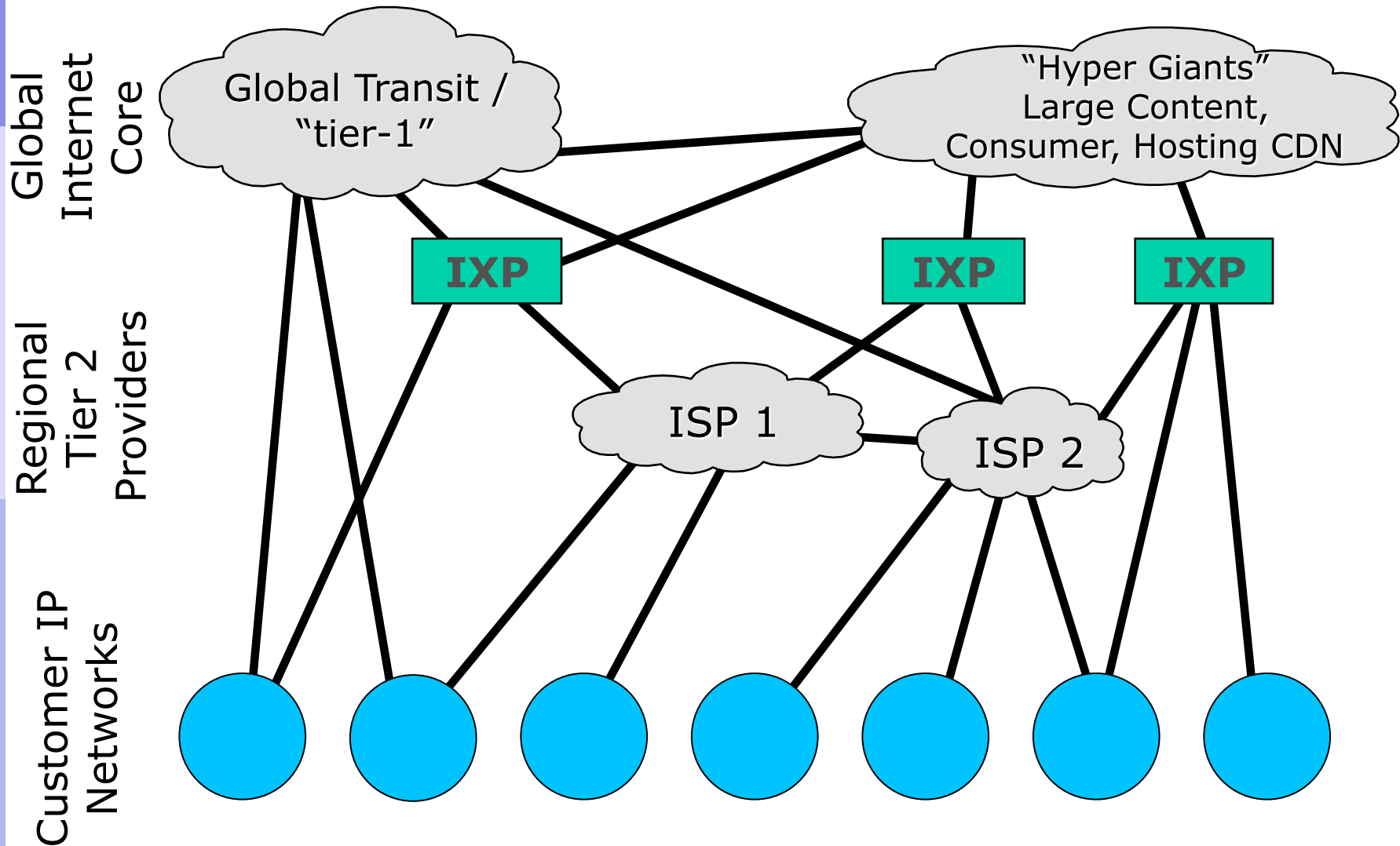
Don't Peer

- You would rather have customers
- Peers are usually your competition
- Peering relationships may require periodic renegotiation

Peering struggles are by far the most contentious issues in the ISP world!

Peering agreements are often confidential.

Structure of the Internet



Summary:

Why do I need BGP?

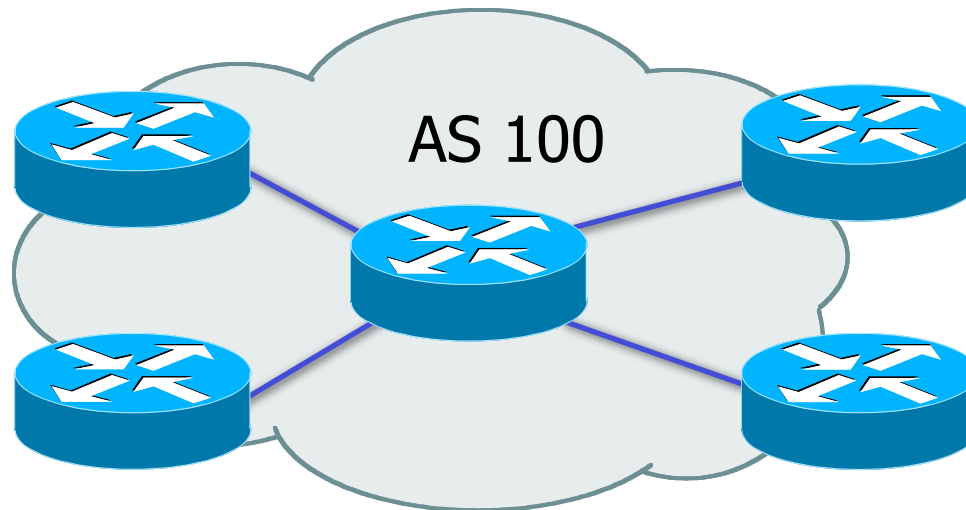
- Multi-homing – connecting to multiple providers
 - upstream providers
 - local networks – regional peering to get local traffic
- Policy discrimination
 - controlling how traffic flows
 - do not accidentally provide transit to non-customers

BGP Part II



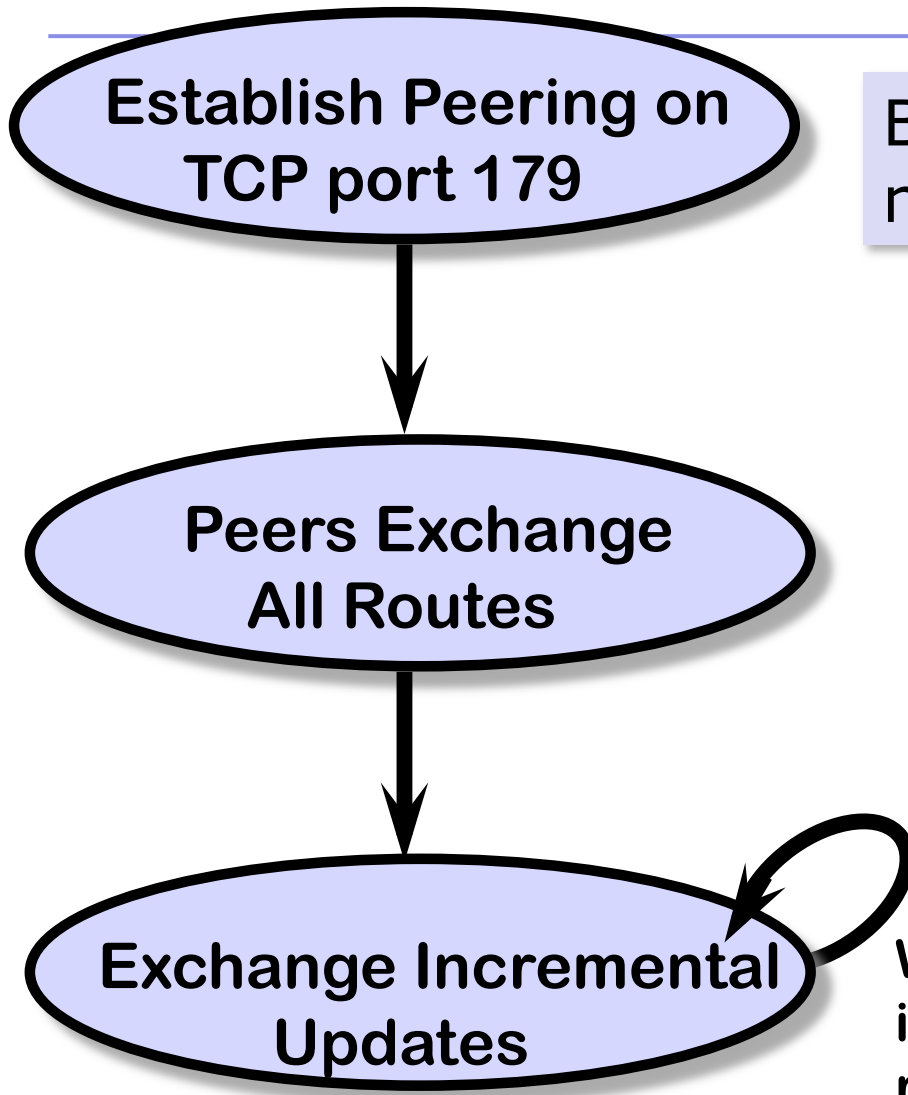
BGP Building Blocks

Autonomous System (AS)

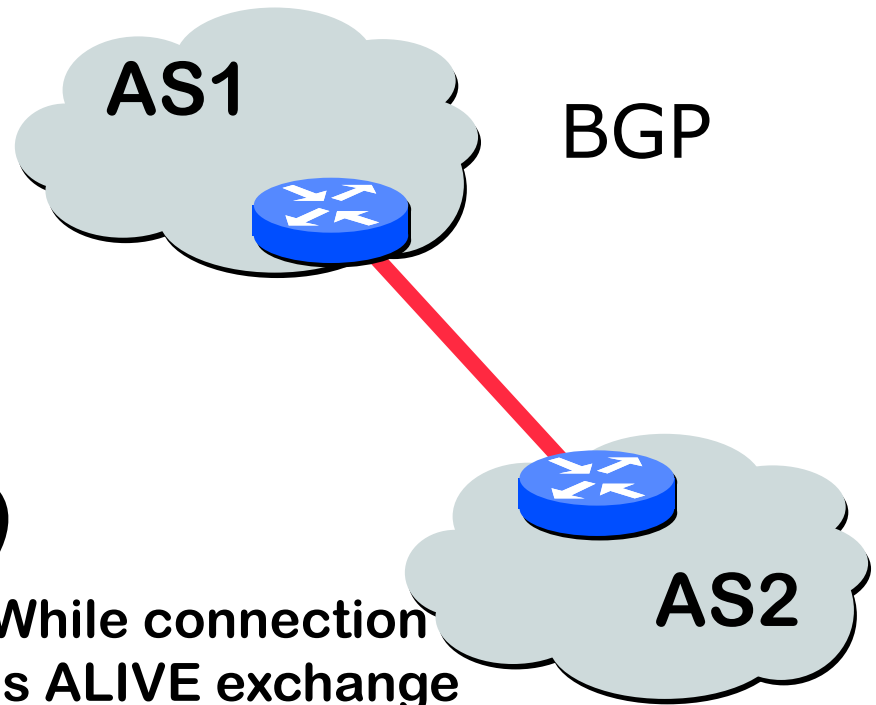


- Collection of networks with same policy
- Single routing protocol
- Usually under single administrative control
- IGP to provide internal connectivity

BGP Operations Simplified



BGP Route = network prefix + attributes



While connection is ALIVE exchange route UPDATE messages

BGP Messages

- **OPEN:**
 - opens TCP conn. to peer
 - authenticates sender
- **UPDATE:**
 - "Announcement": prefix is reachable
 - "Withdraw": prefix is not reachable
- **KEEPALIVE:**
 - keeps connection alive in absence of UPDATES
 - serves as ACK to an OPEN request
- **NOTIFICATION:**
 - reports errors in previous msg;
 - closes a connection

BGP Attributes

...	Next Hop	AS Path	Local-Pref.	MED	Community	...
-----	----------	---------	-------------	-----	-----------	-----

- Attributes are “knobs” for
 - traffic engineering
 - capacity planning

BGP Protocol Basics

- Uses Incremental updates
 - sends one copy of the RIB at the beginning, then sends changes as they happen
- Path Vector protocol
 - keeps track of the AS path of routing information
- Many options for policy enforcement

Terminology

- Neighbour
 - Configured BGP peer
- NLRI/Prefix
 - NLRI – network layer reachability information
 - Reachability information for an IP address & mask
- Router-ID
 - 32 bit integer to uniquely identify router
 - Comes from Loopback or Highest IP address configured on the router
- Route/Path
 - NLRI advertised by a neighbour

Terminology

- Transit – carrying network traffic across a network, usually for a fee
- Peering – exchanging routing information and traffic
 - your customers and your peers' customers network information only.
 - not your peers' peers; not your peers' providers.
- Peering also has another meaning:
 - BGP neighbour, whether or not transit is provided
- Default – where to send traffic when there is no explicit route in the routing table

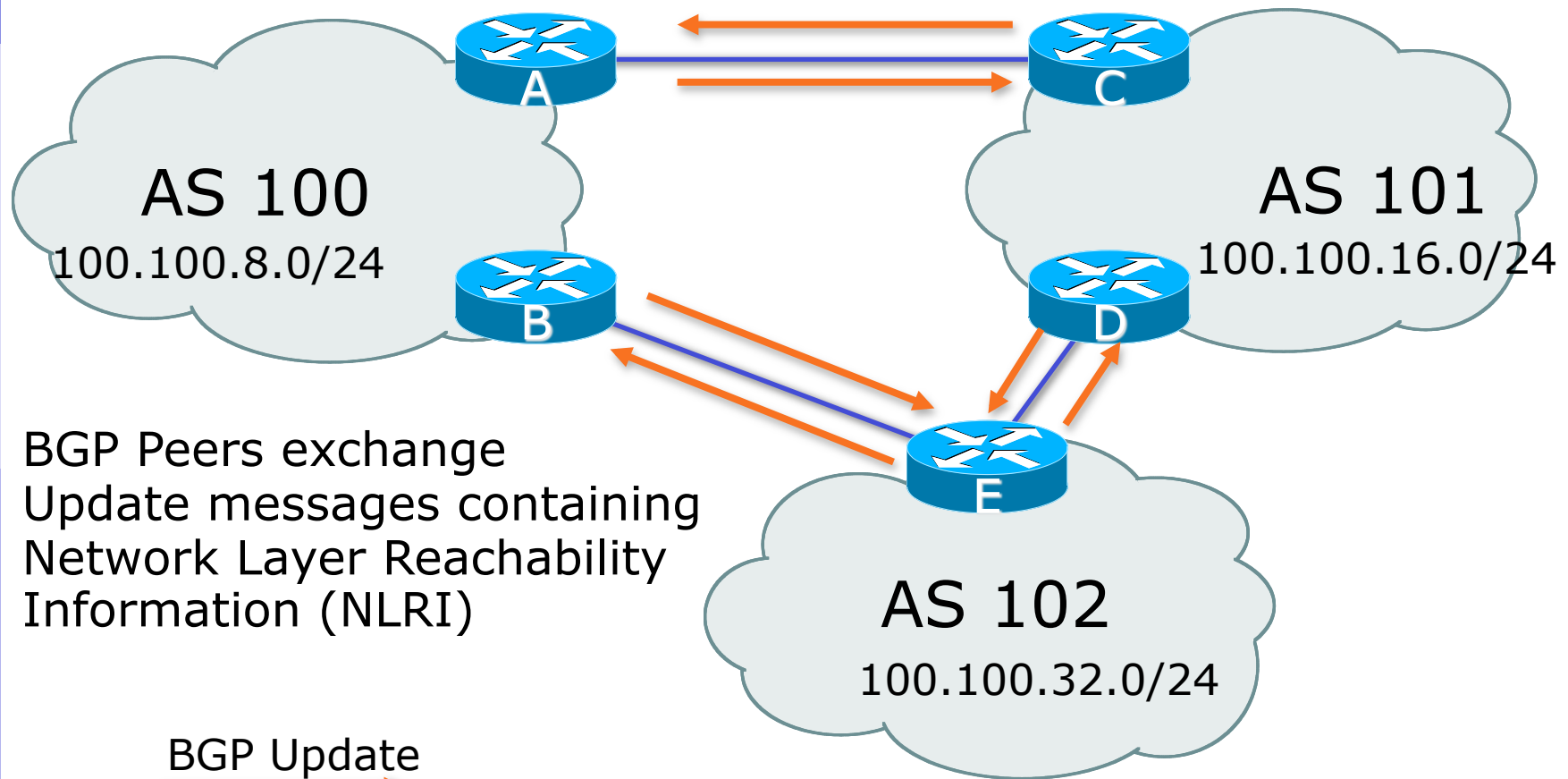
BGP Basics ...

- Each AS originates a set of NLRI (routing announcements)
- NLRI is exchanged between BGP peers
- Can have multiple paths for a given prefix
- BGP picks the best path and installs in the IP forwarding table
- Policies applied (through attributes) influences BGP path selection

Interior BGP vs. Exterior BGP

- Interior BGP (iBGP)
 - Between routers in the same AS
 - Often between routers that are far apart
 - Should be a full mesh: every iBGP router talks to all other iBGP routers in the same AS
- Exterior BGP (eBGP)
 - Between routers in different ASes
 - Almost always between directly-connected routers (ethernet, serial line, etc.)

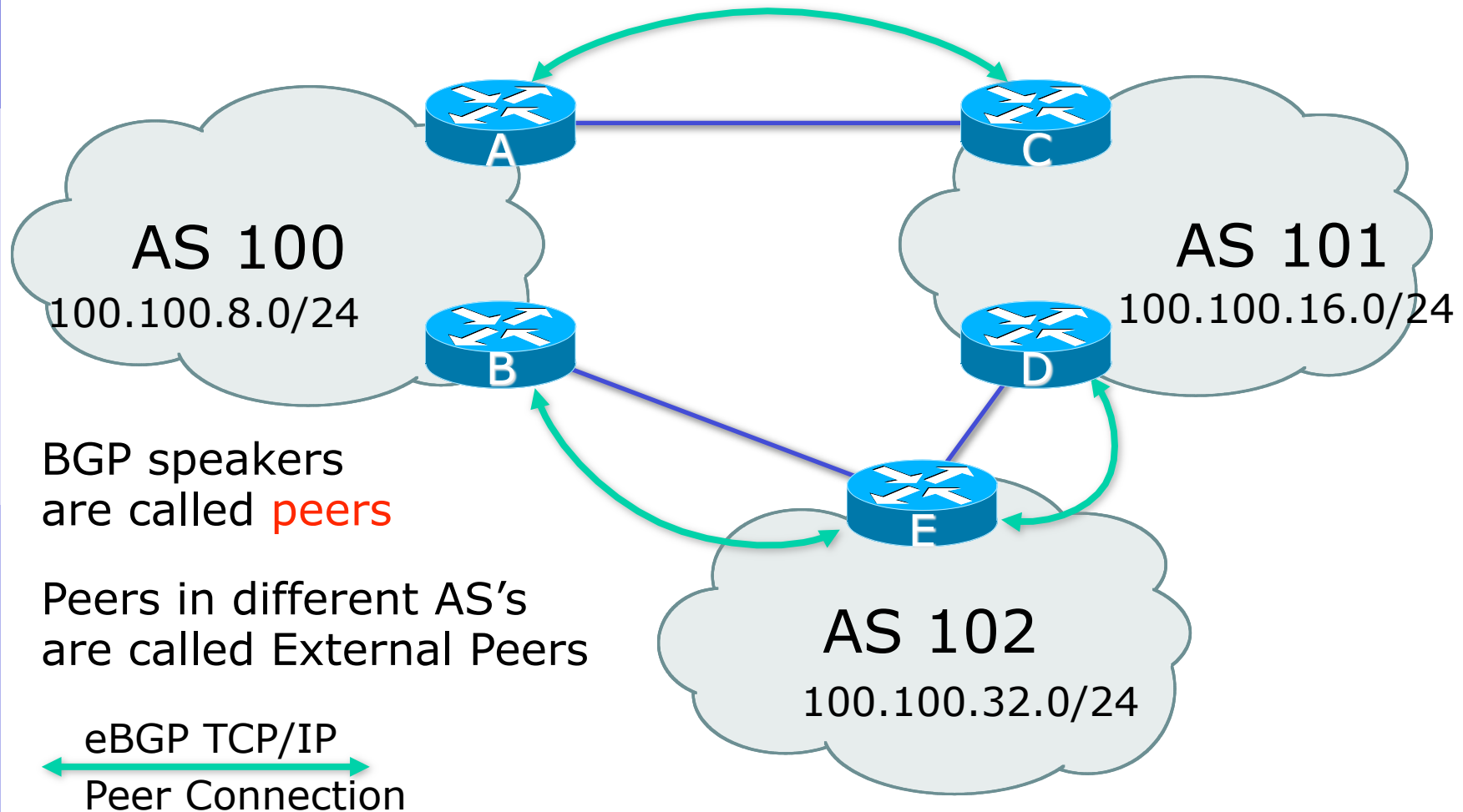
BGP Peers



BGP Peers exchange Update messages containing Network Layer Reachability Information (NLRI)

BGP Update Messages →

BGP Peers – External (eBGP)

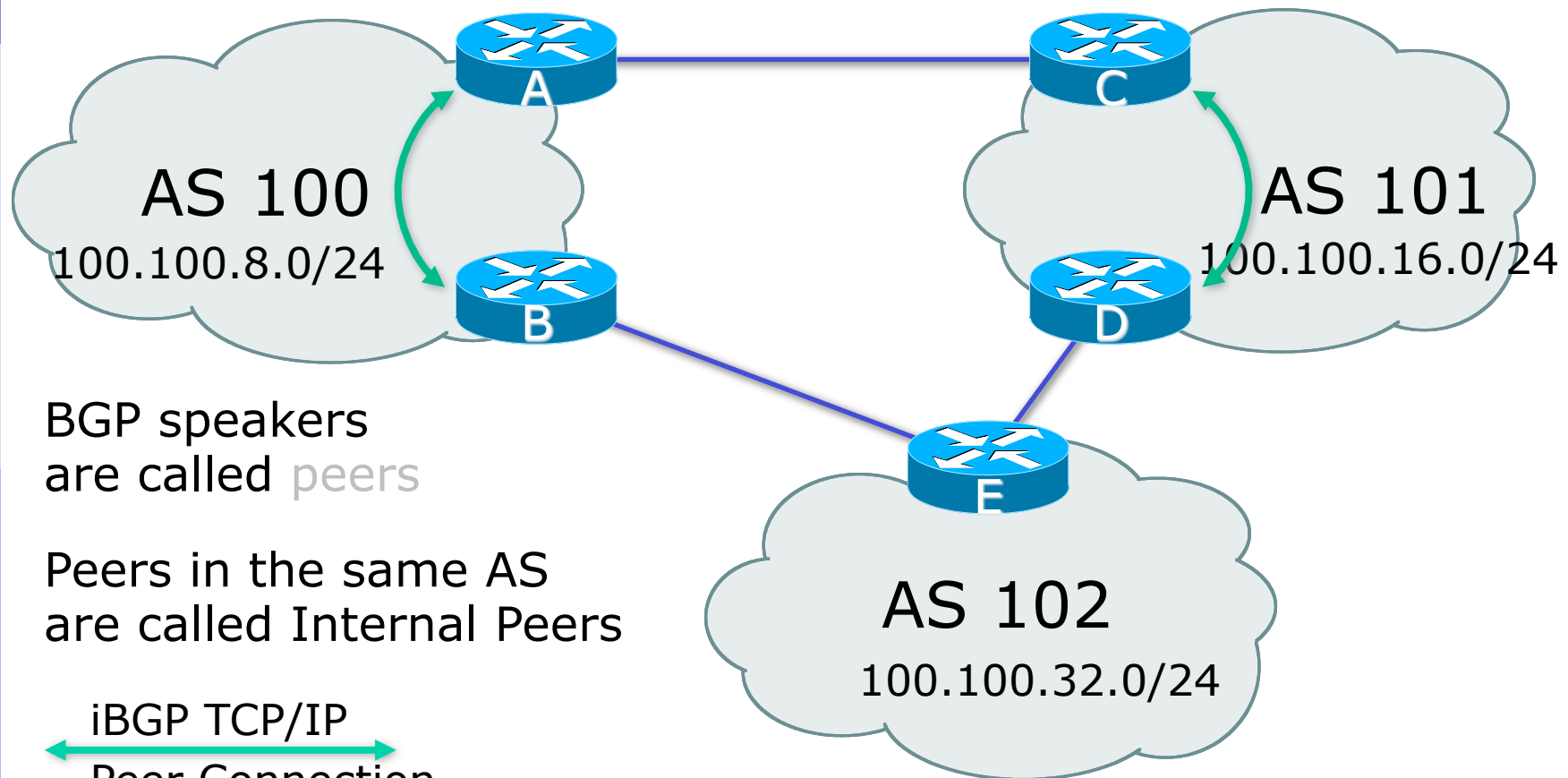


BGP speakers are called **peers**

Peers in different AS's are called External Peers

Note: eBGP Peers normally should be directly connected.

BGP Peers – Internal (iBGP)



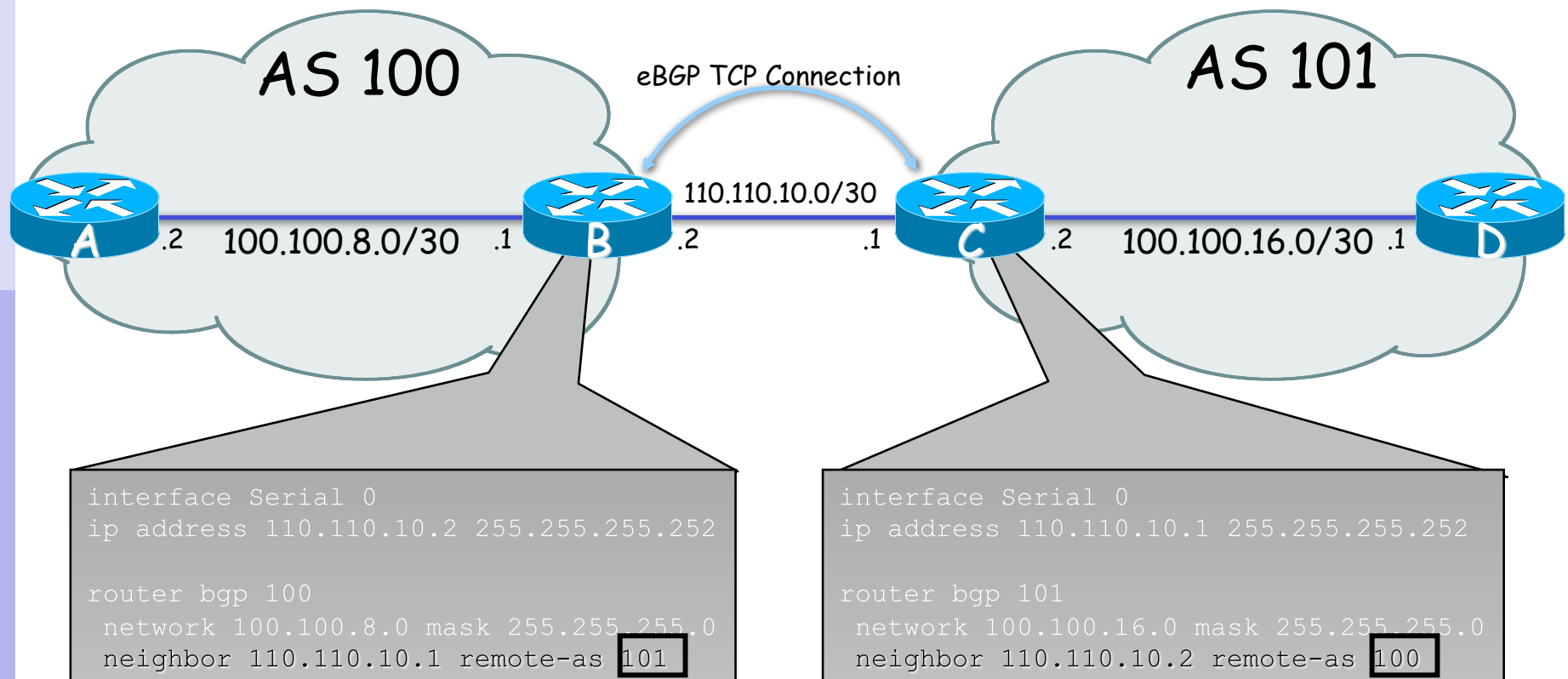
BGP speakers
are called **peers**

Peers in the same AS
are called **Internal Peers**

Note: iBGP Peers don't have to be directly connected.

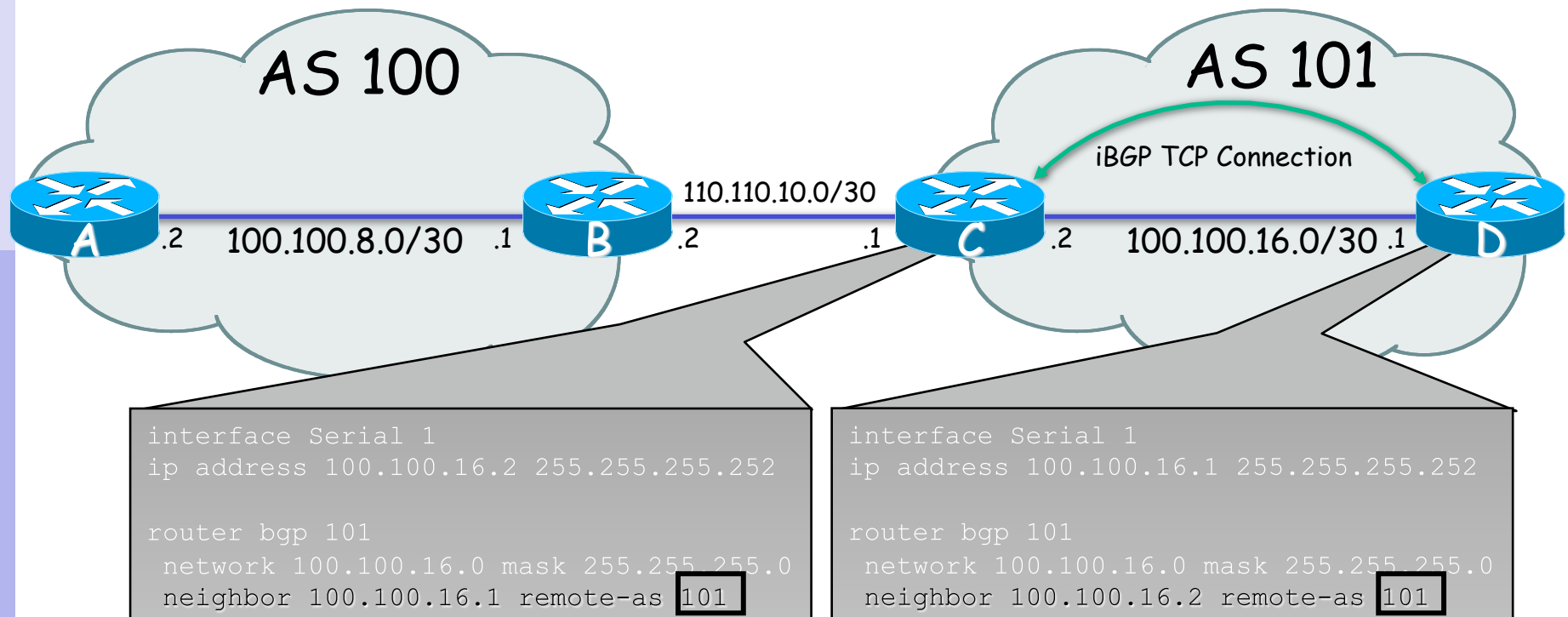
Configuring eBGP peers

- BGP peering sessions are established using the BGP "neighbor" command
 - eBGP is configured when AS numbers are different



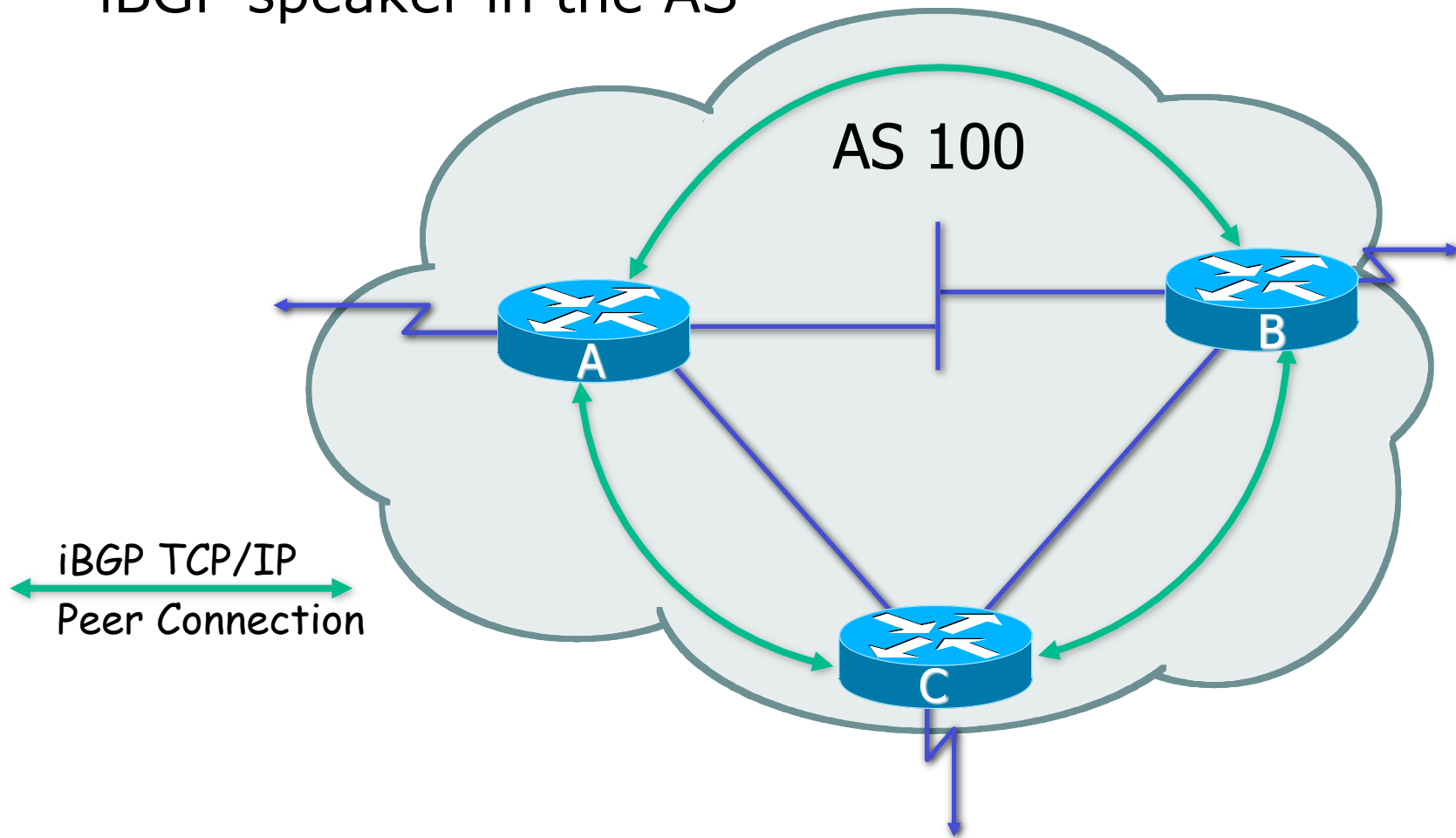
Configuring iBGP peers

- BGP peering sessions are established using the BGP "neighbor" command
 - iBGP is configured when AS numbers are the same



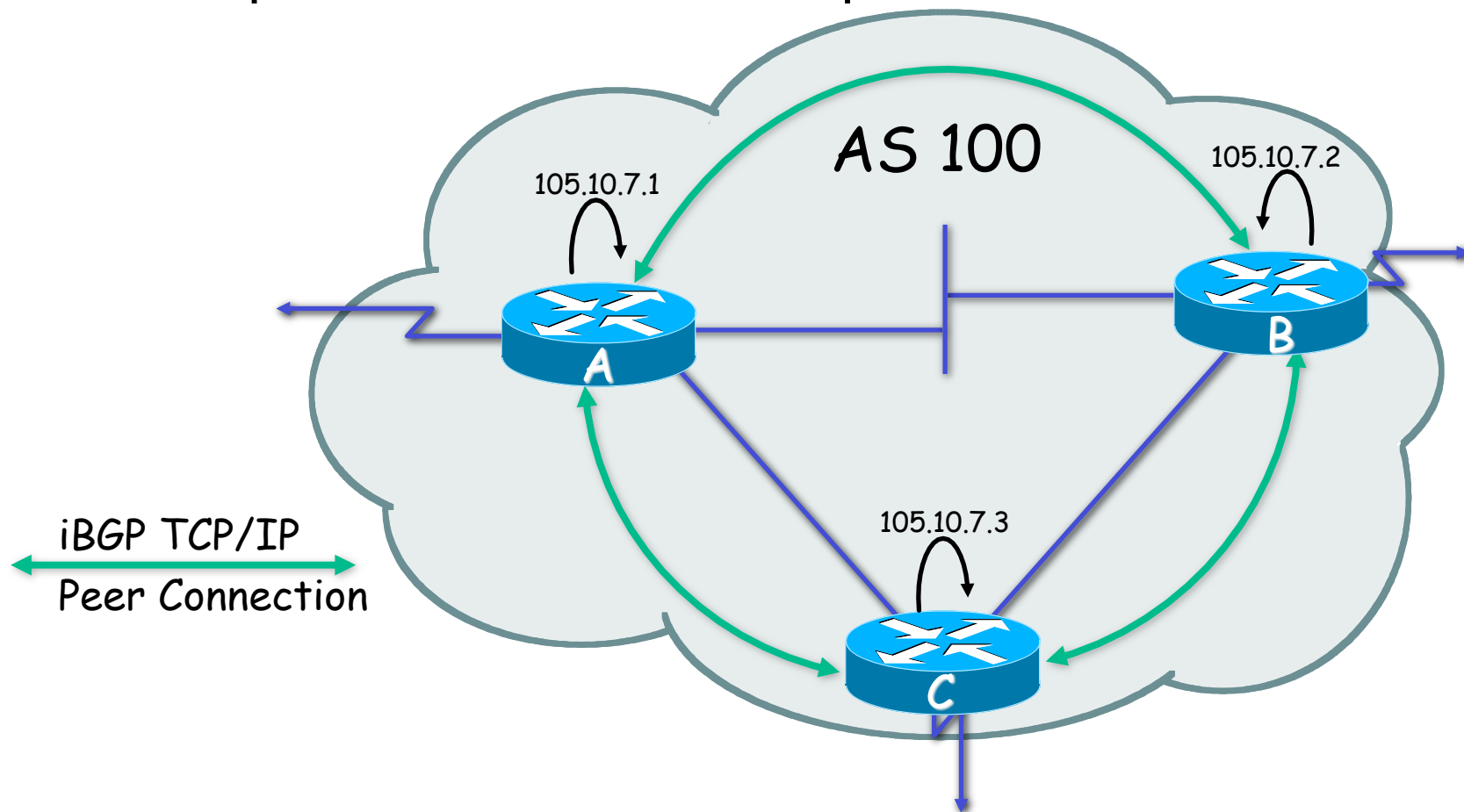
Configuring iBGP peers: Full mesh

- Each iBGP speaker must peer with every other iBGP speaker in the AS

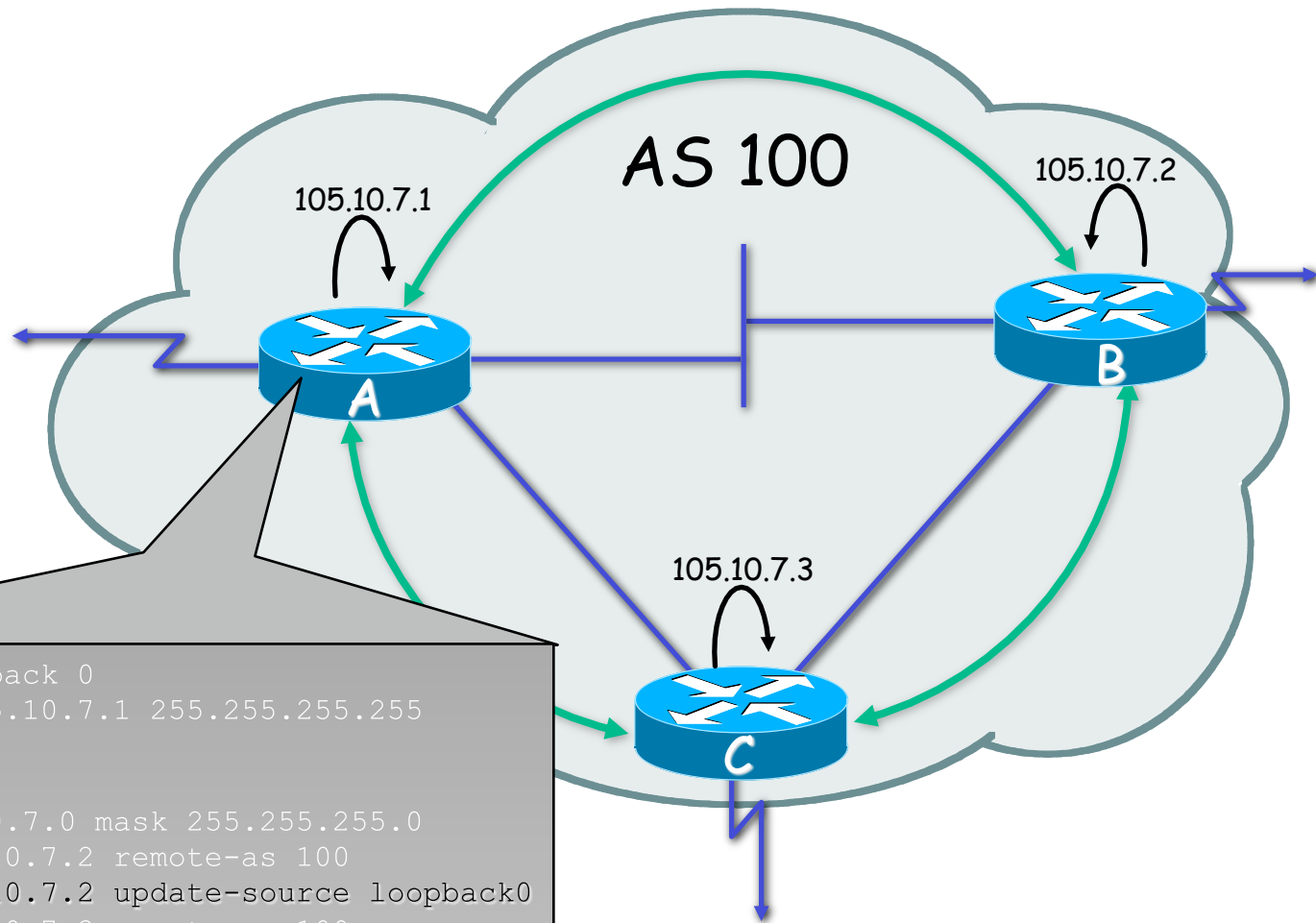


Configuring iBGP peers: Loopback interface

- Loopback interfaces are normally used as the iBGP peer connection end-points



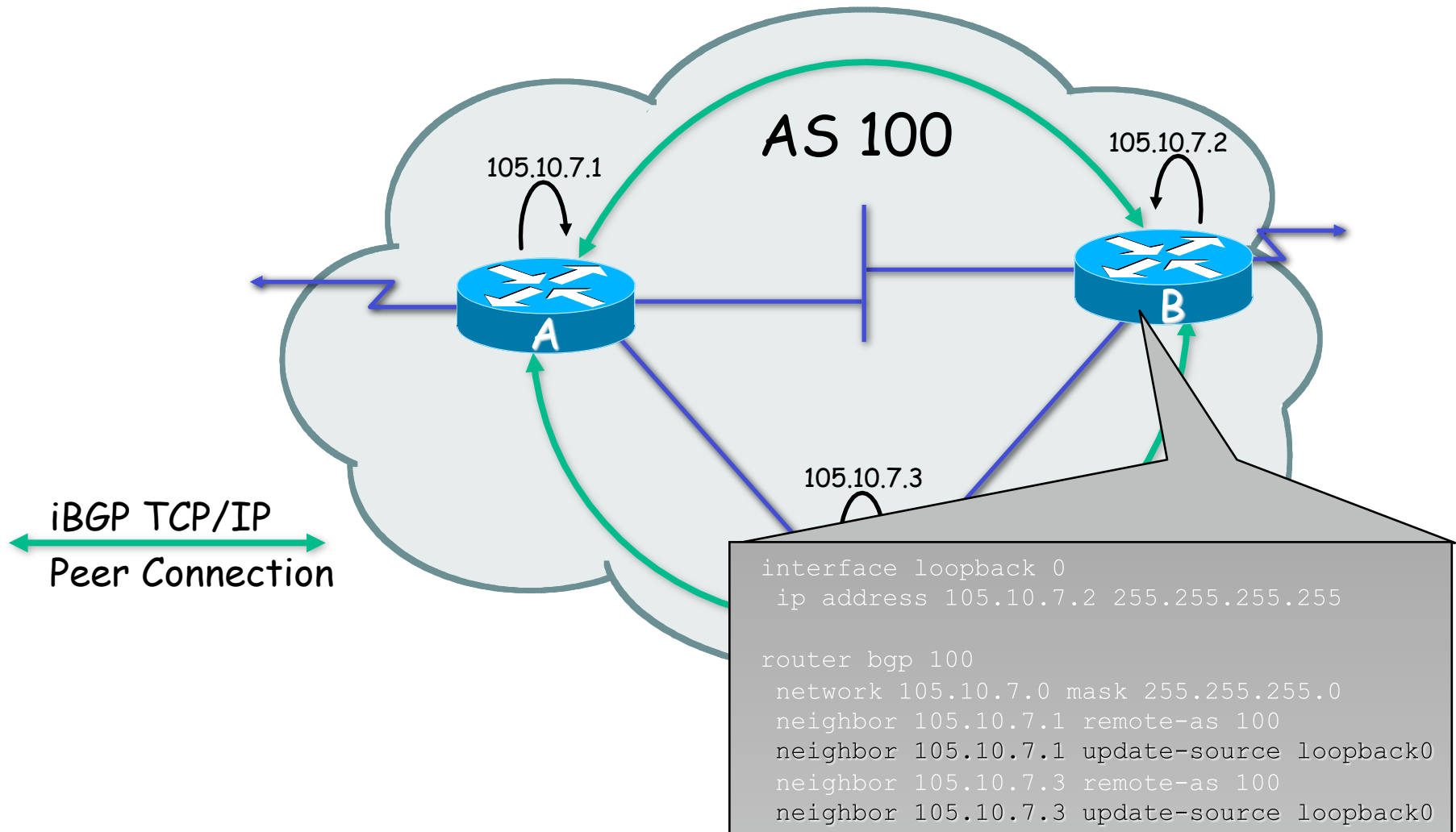
Configuring iBGP peers



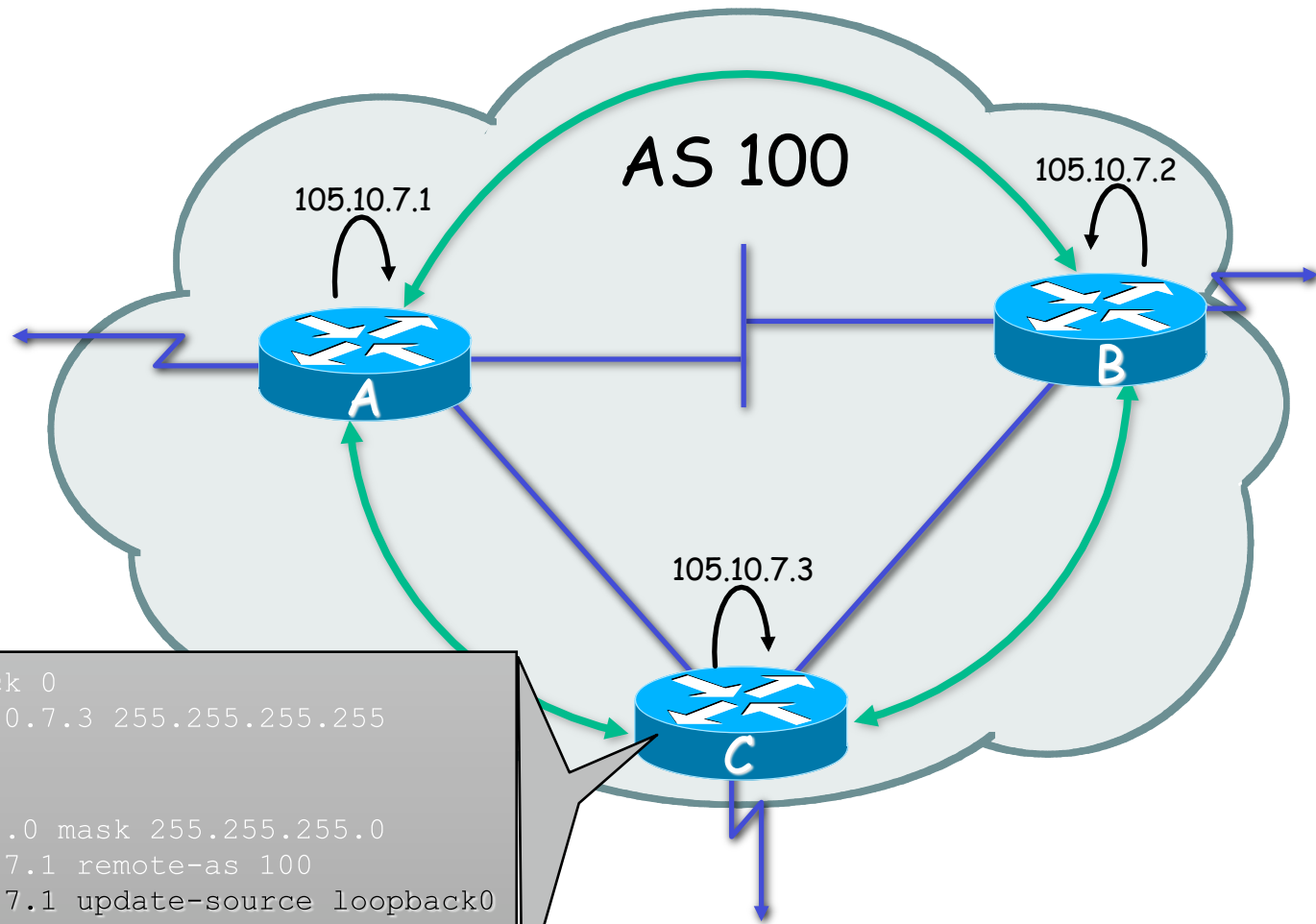
```
interface loopback 0
 ip address 105.10.7.1 255.255.255.255

router bgp 100
 network 105.10.7.0 mask 255.255.255.0
 neighbor 105.10.7.2 remote-as 100
 neighbor 105.10.7.2 update-source loopback0
 neighbor 105.10.7.3 remote-as 100
 neighbor 105.10.7.3 update-source loopback0
```

Configuring iBGP peers



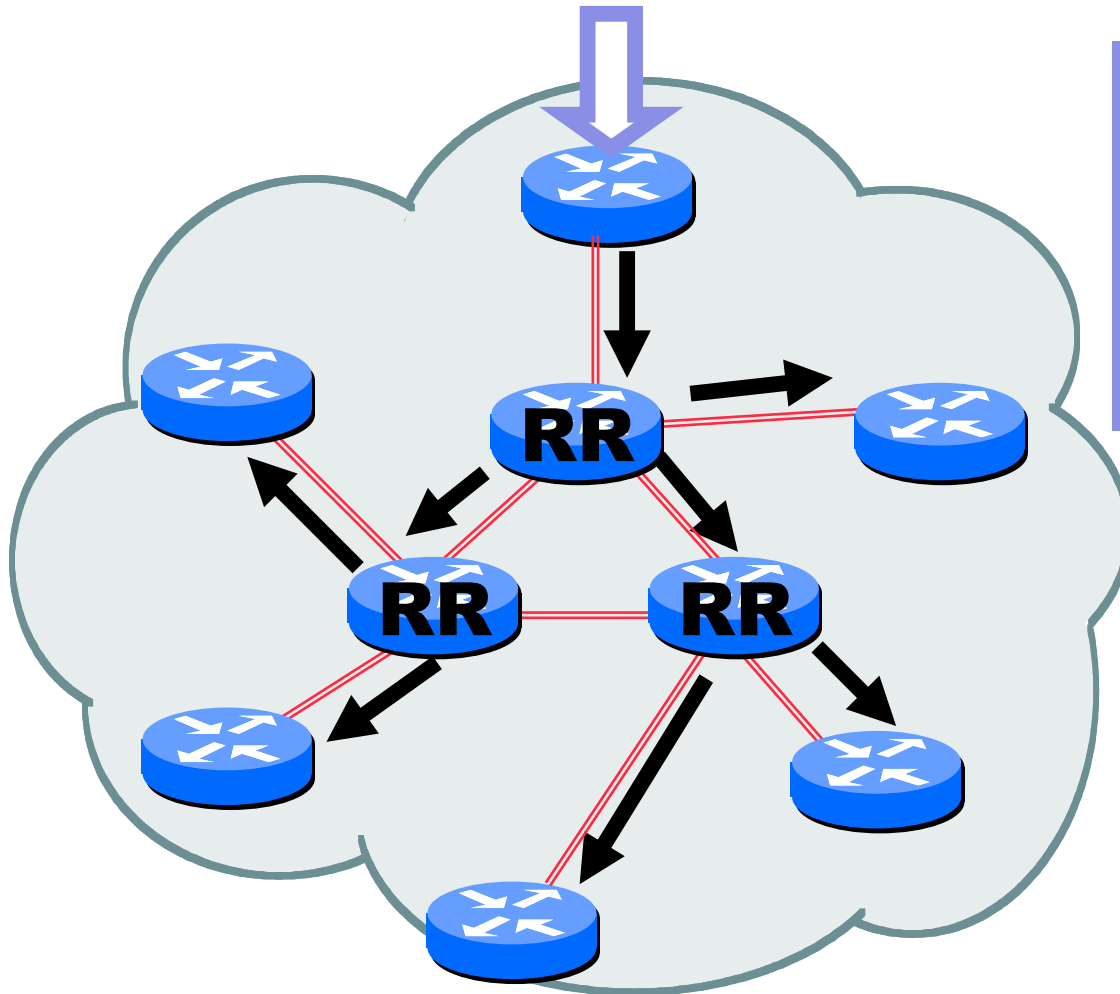
Configuring iBGP peers



```
interface loopback 0
ip address 105.10.7.3 255.255.255.255

router bgp 100
network 105.10.7.0 mask 255.255.255.0
neighbor 105.10.7.1 remote-as 100
neighbor 105.10.7.1 update-source loopback0
neighbor 105.10.7.2 remote-as 100
neighbor 105.10.7.2 update-source loopback0
```


Route Reflectors



- Route reflectors can pass on iBGP updates to clients
- Each RR passes along ONLY best routes
- ORIGINATOR_ID and CLUSTER_LIST attributes are needed to avoid loops

BGP Part III



BGP Protocol – A little more detail

BGP Updates — NLRI

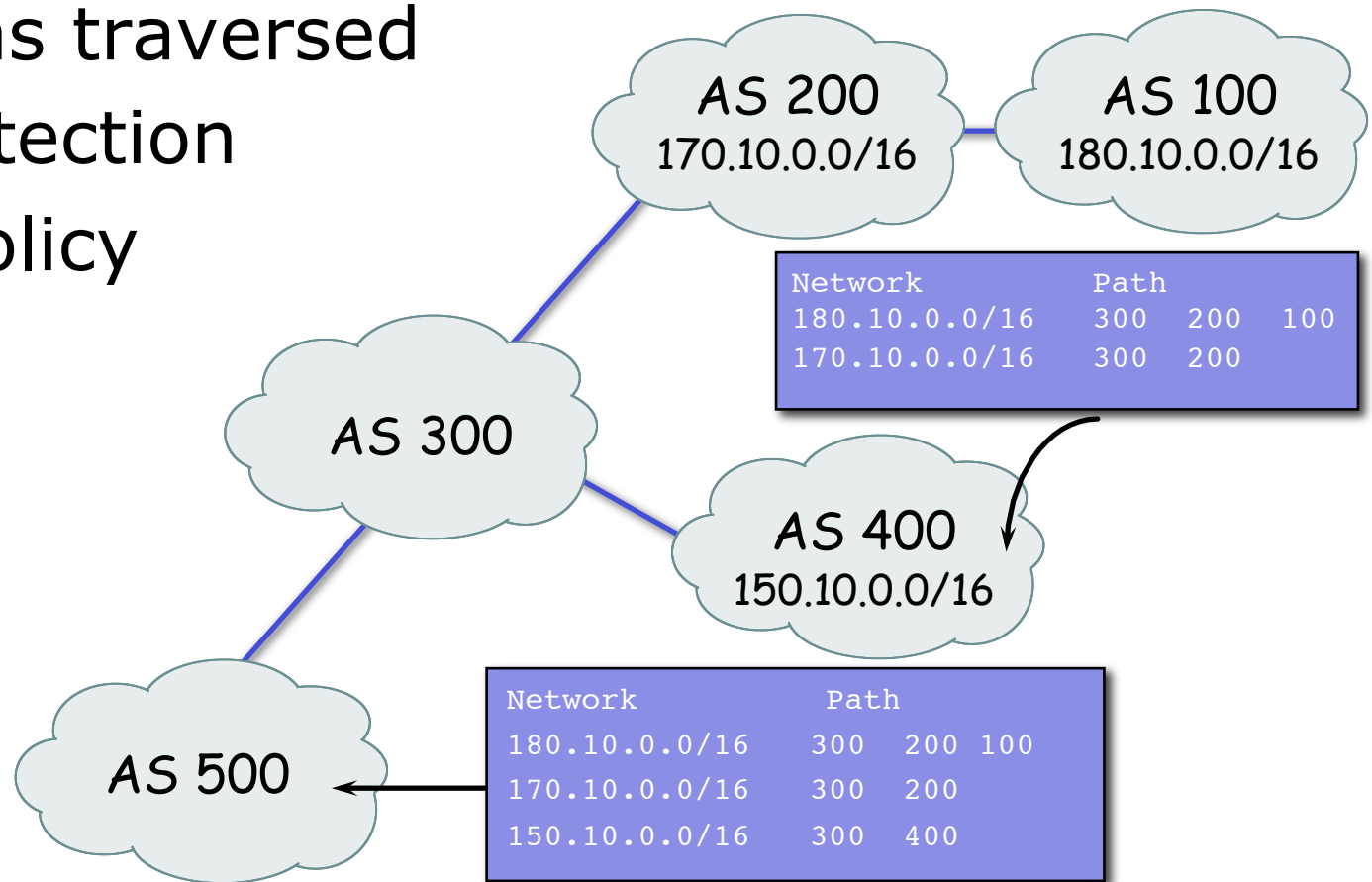
- Network Layer Reachability Information
- Used to advertise feasible routes
- Composed of:
 - Network Prefix
 - Mask Length

BGP Updates — Attributes

- Used to convey information associated with NLRI
 - AS path
 - Next hop
 - Local preference
 - Multi-Exit Discriminator (MED)
 - Community
 - Origin
 - Aggregator

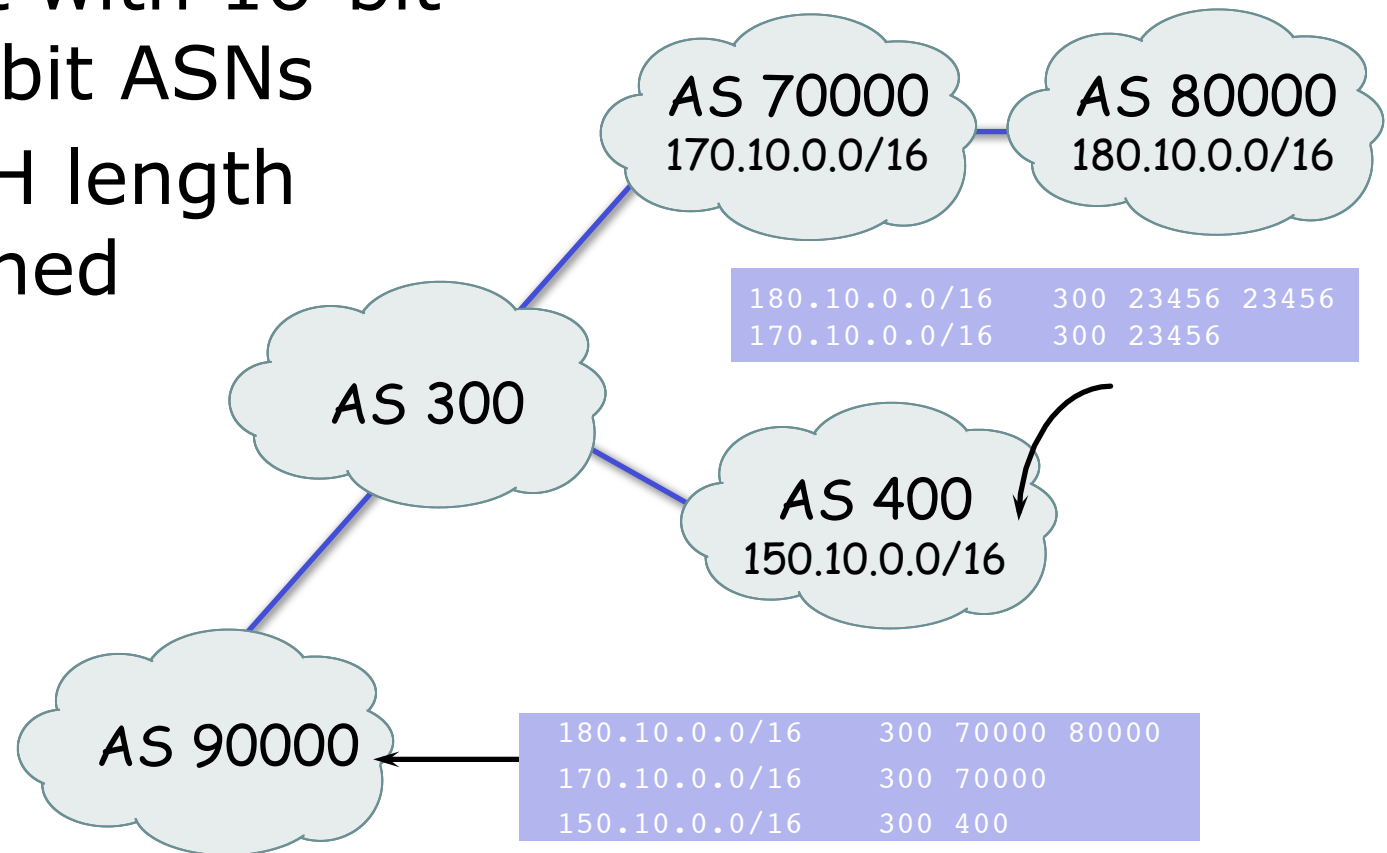
AS-Path Attribute

- Sequence of ASes a route has traversed
- Loop detection
- Apply policy



AS-Path (with 16 and 32-bit ASNs)

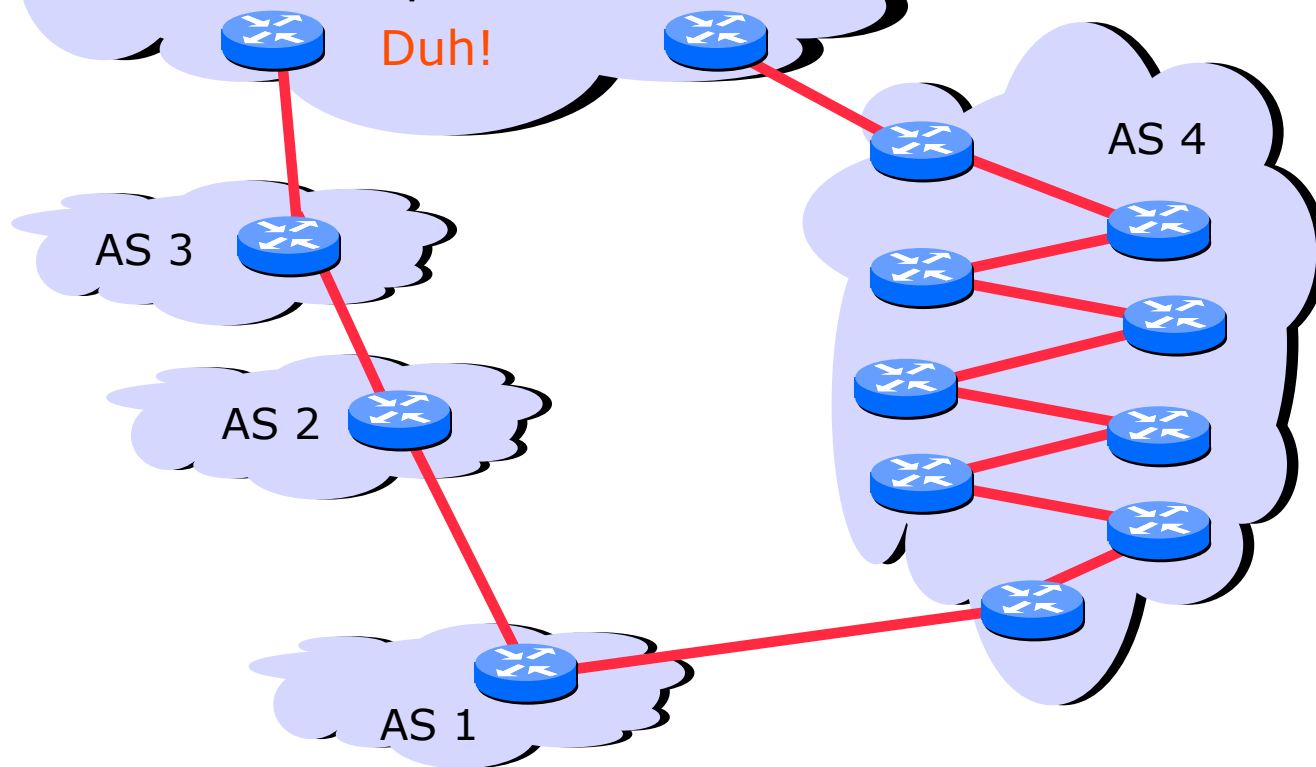
- Internet with 16-bit and 32-bit ASNs
- AS-PATH length maintained



Shorter Doesn't Always Mean Shorter

Mr. BGP says that path 4 1 is better than path 3 2 1

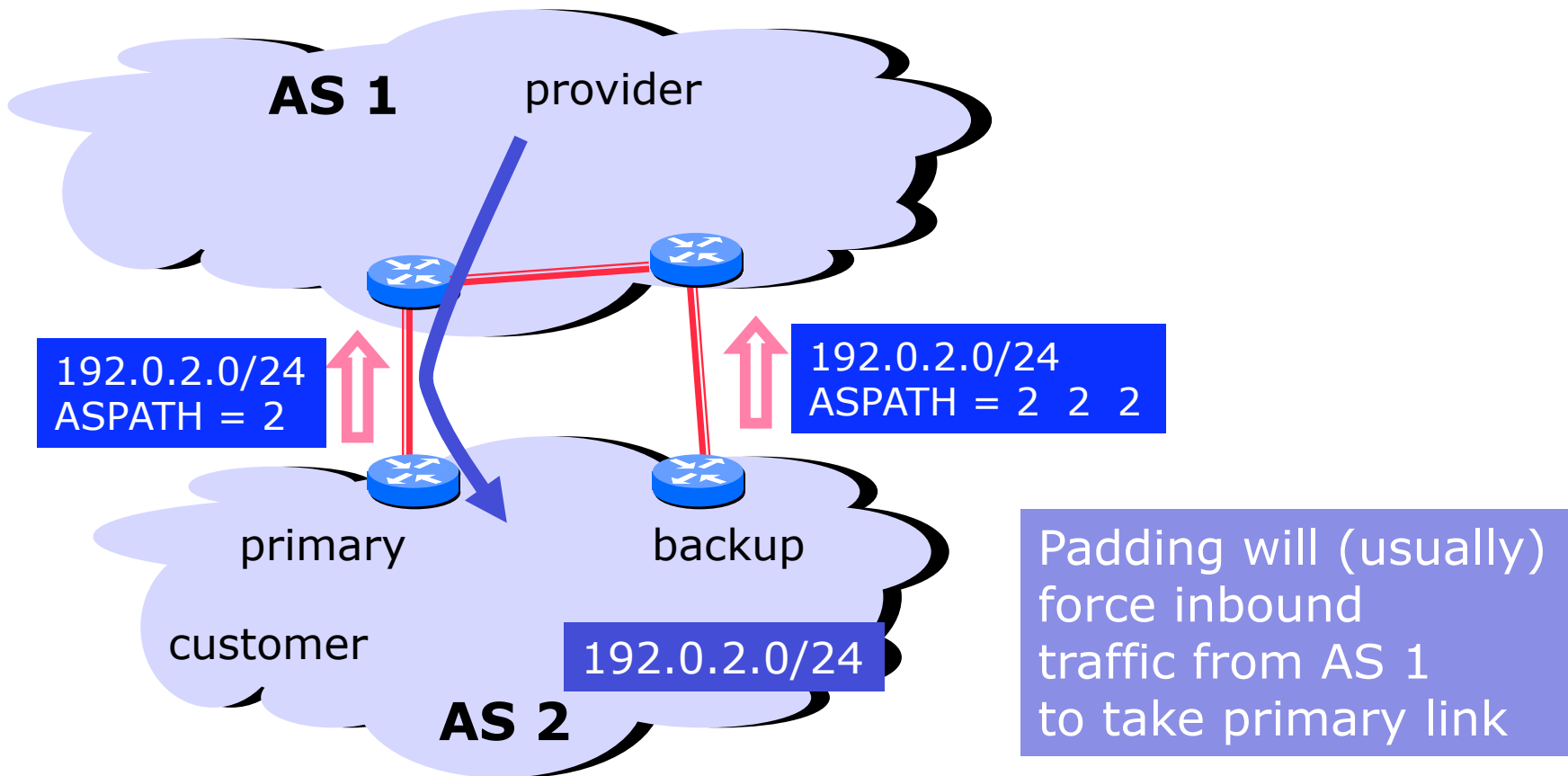
Duh!



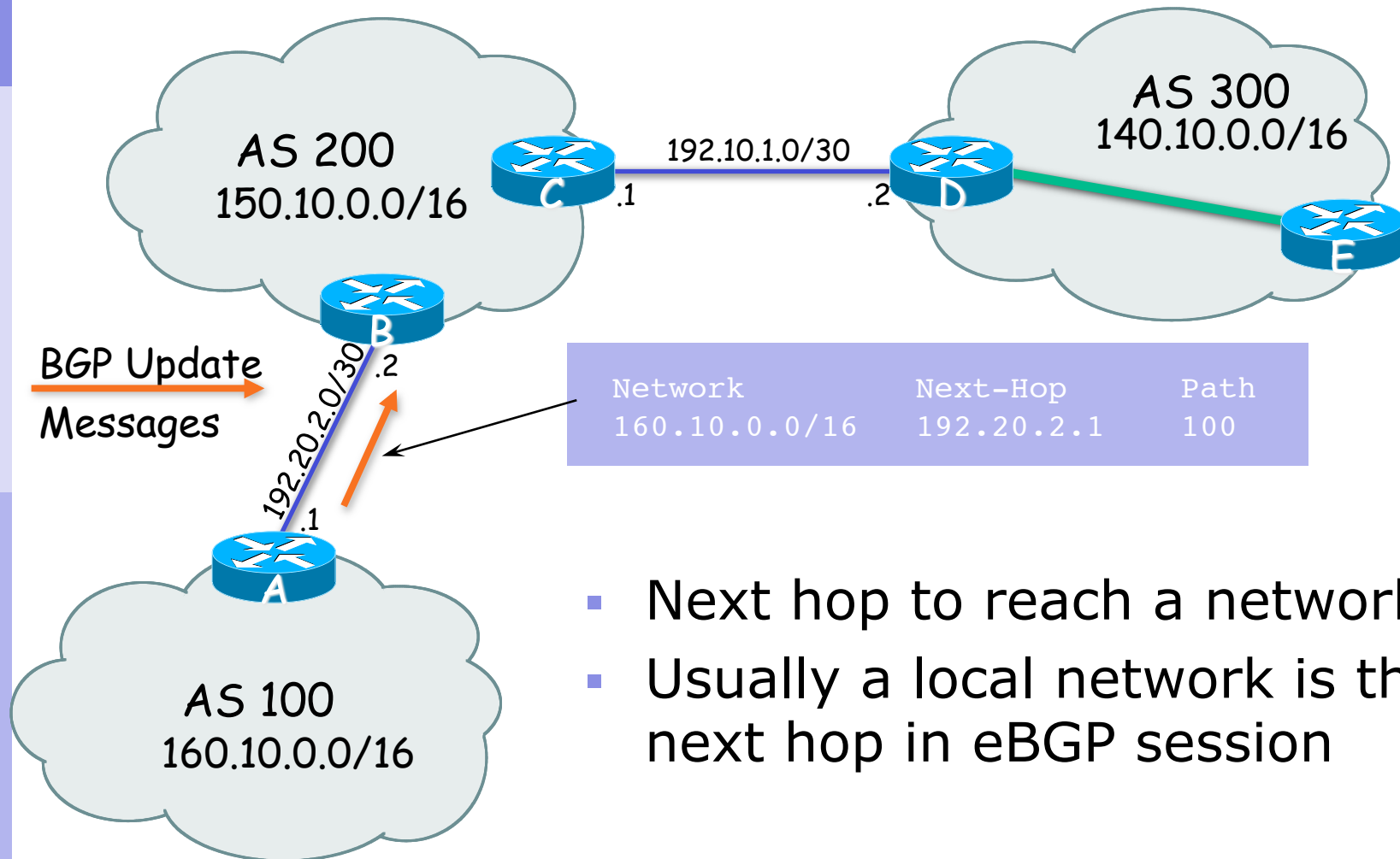
In fairness:
could you do
this "right" and
still scale?

Exporting
internal
state would
dramatically
increase global
instability and
amount of
routing
state

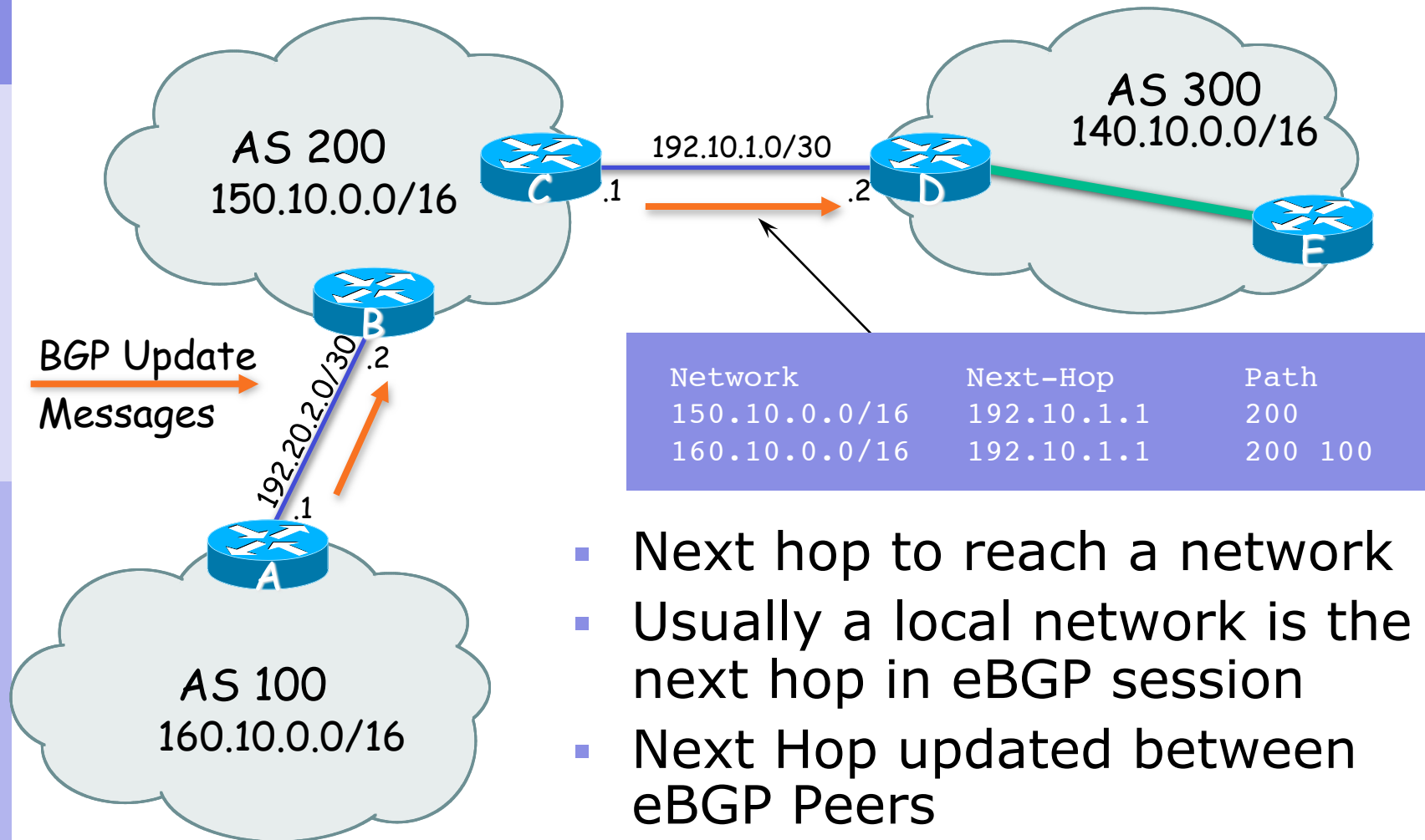
ASPATH Padding



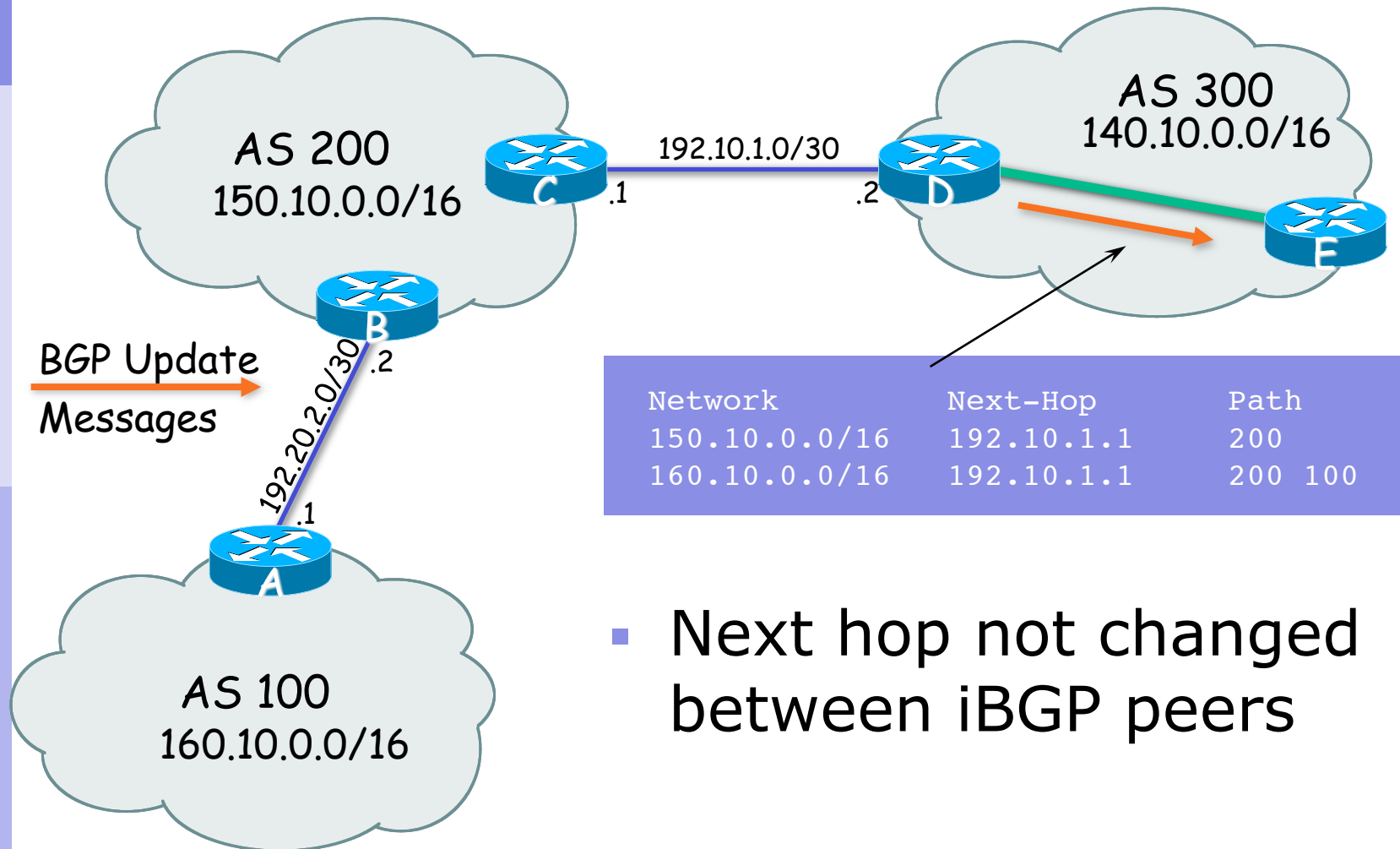
Next Hop Attribute



Next Hop Attribute



Next Hop Attribute



Next Hop Attribute (more)

- IGP is used to carry route to next hops
- Recursive route look-up
 - BGP looks into IGP to find out next hop information
 - BGP is not permitted to use a BGP route as the next hop
- Unlinks BGP from actual physical topology
- Allows IGP to make intelligent forwarding decision

Next Hop Best Practice

- Cisco IOS default is for external next-hop to be propagated unchanged to iBGP peers
 - This means that IGP has to carry external next-hops
 - Forgetting means external network is invisible
 - With many eBGP peers, it is extra load on IGP
- **ISP best practice is to change external next-hop to be that of the local router**
`neighbor x.x.x.x next-hop-self`

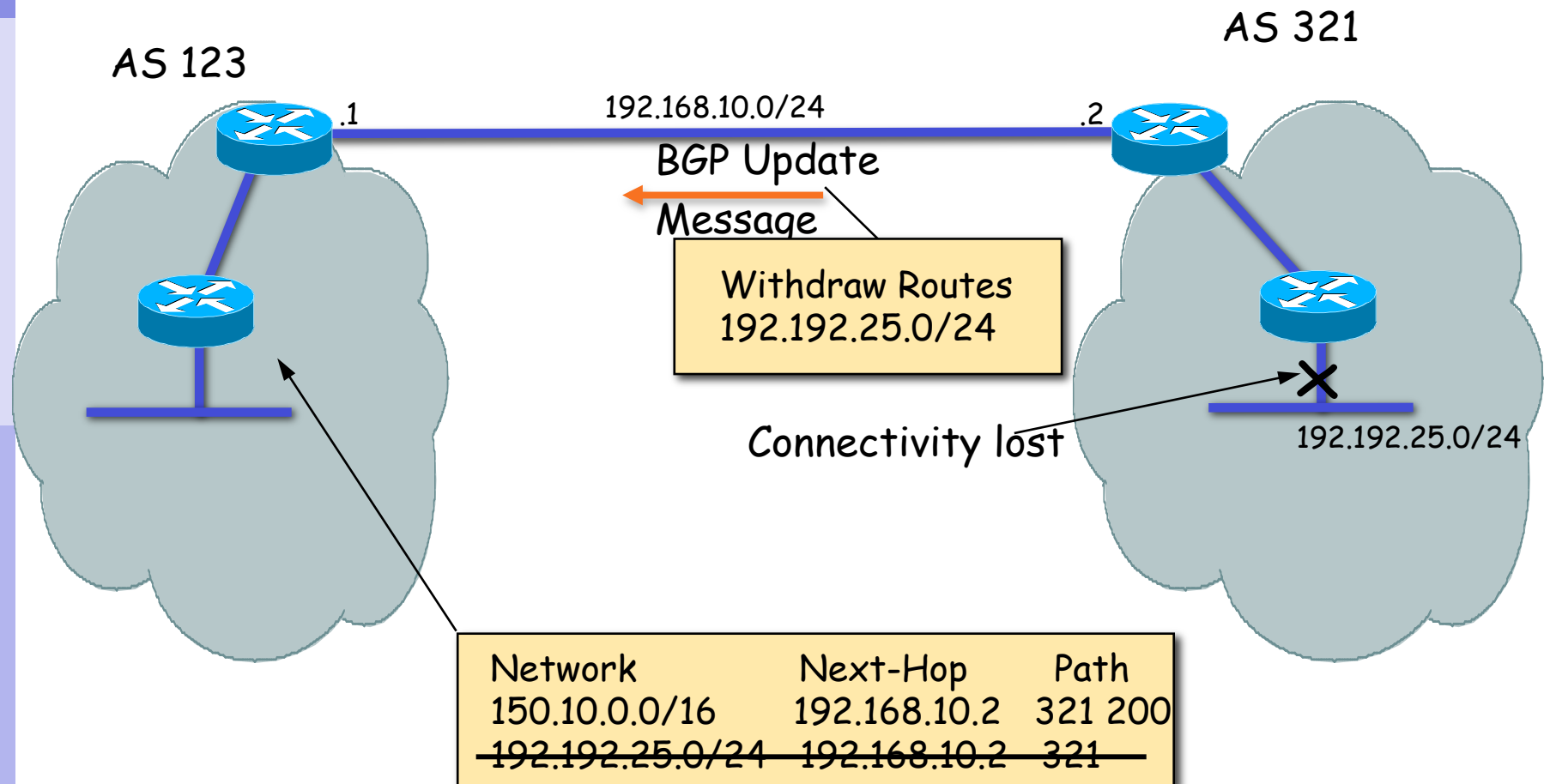
Community Attribute

- 32-bit number
- Conventionally written as two 16-bit numbers separated by colon
 - First half is usually an AS number
 - ISP determines the meaning (if any) of the second half
- Carried in BGP protocol messages
 - Used by administratively-defined filters
 - Not directly used by BGP protocol (except for a few “well known” communities)

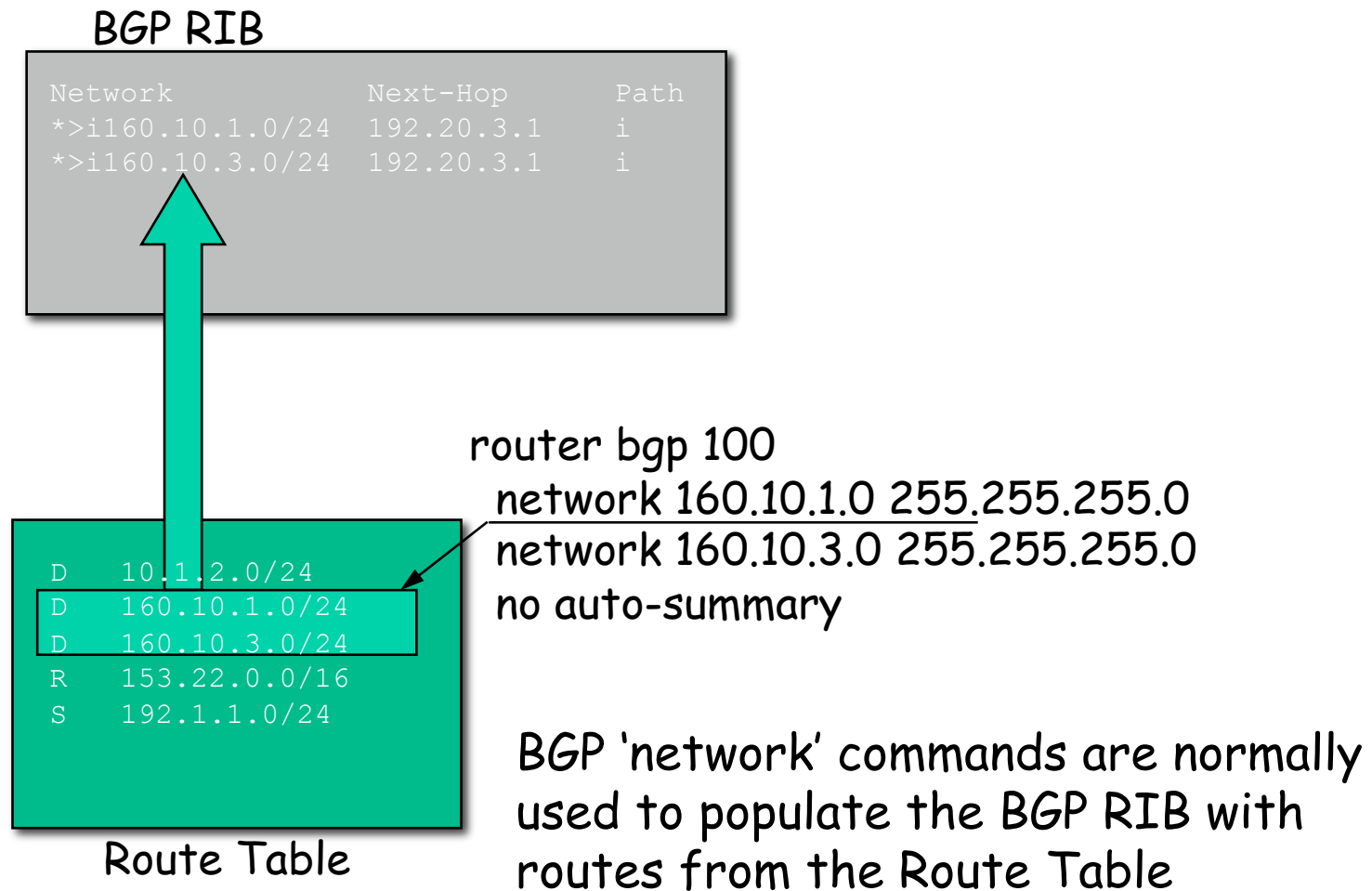
BGP Updates: Withdrawn Routes

- Used to “withdraw” network reachability
- Each withdrawn route is composed of:
 - Network Prefix
 - Mask Length

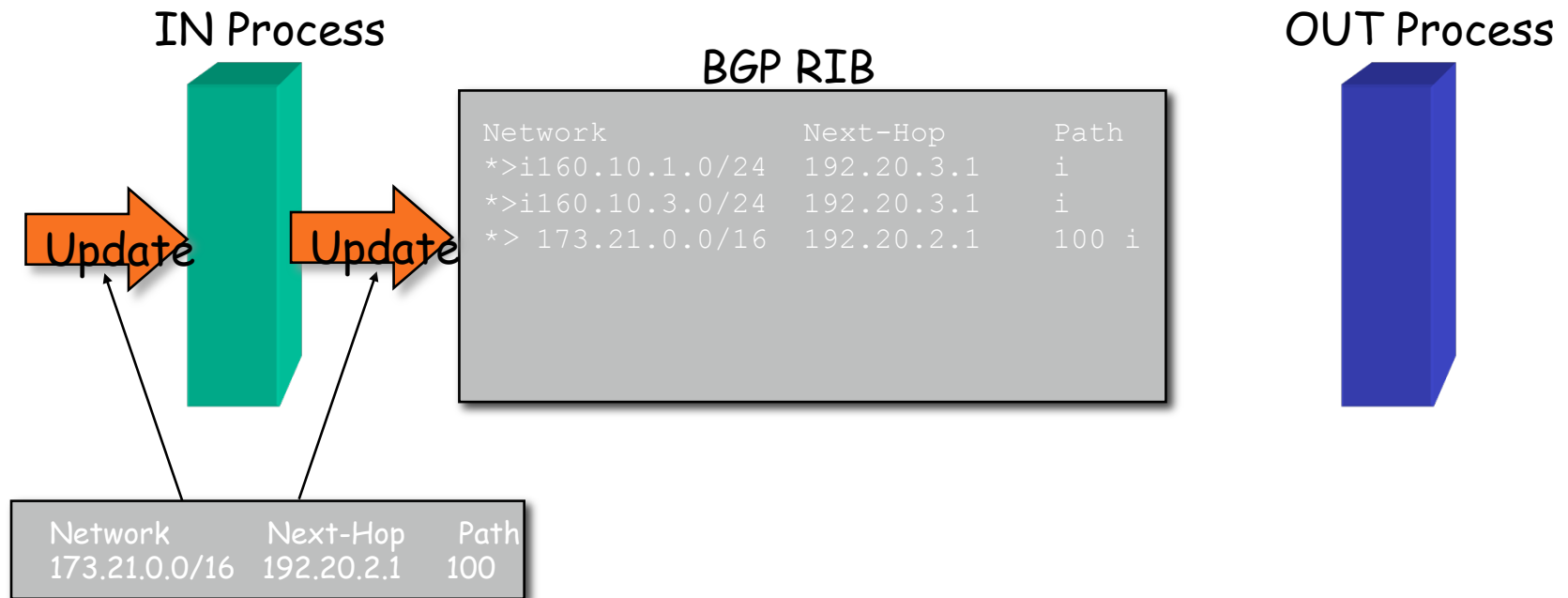
BGP Updates: Withdrawn Routes



BGP Routing Information Base



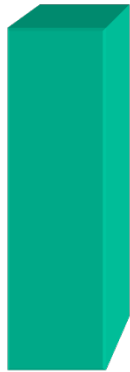
BGP Routing Information Base



- BGP "in" process
 - receives path information from peers
 - results of BGP path selection placed in the BGP table
 - "best path" flagged (denoted by ">")

BGP Routing Information Base

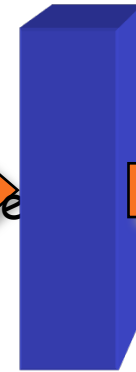
IN Process



BGP RIB

Network	Next-Hop	Path
*>i160.10.1.0/24	192.20.3.1	i
*>i160.10.3.0/24	192.20.3.1	i
*> 173.21.0.0/16	192.20.2.1	100

OUT Process



Network	Next-Hop	Path
160.10.1.0/24	192.20.3.1	200
160.10.3.0/24	192.20.3.1	200
173.21.0.0/16	192.20.2.1	200 100

- BGP "out" process
 - builds update using info from RIB
 - may modify update based on config
 - Sends update to peers

BGP Routing Information Base

BGP RIB

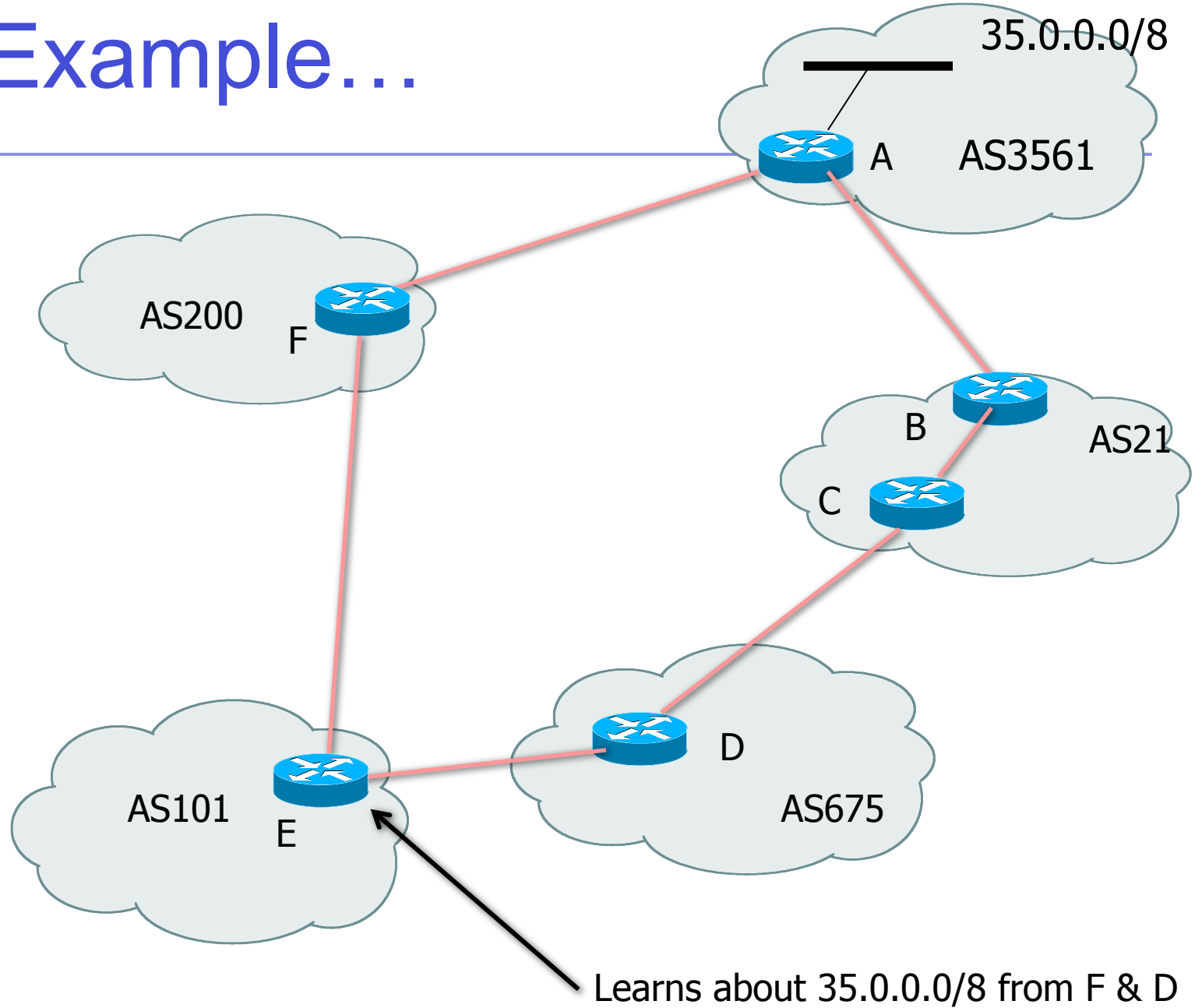
Network	Next-Hop	Path
*>i160.10.1.0/24	192.20.3.1	i
*>i160.10.3.0/24	192.20.3.1	i
*> 173.21.0.0/16	192.20.2.1	100

D	10.1.2.0/24
D	160.10.1.0/24
D	160.10.3.0/24
R	153.22.0.0/16
S	192.1.1.0/24
B	173.21.0.0/16

Route Table

- Best paths installed in routing table if:
 - prefix and prefix length are unique
 - lowest "protocol distance"

An Example...



BGP Part IV



Routing Policy
Filtering

Terminology: “Policy”

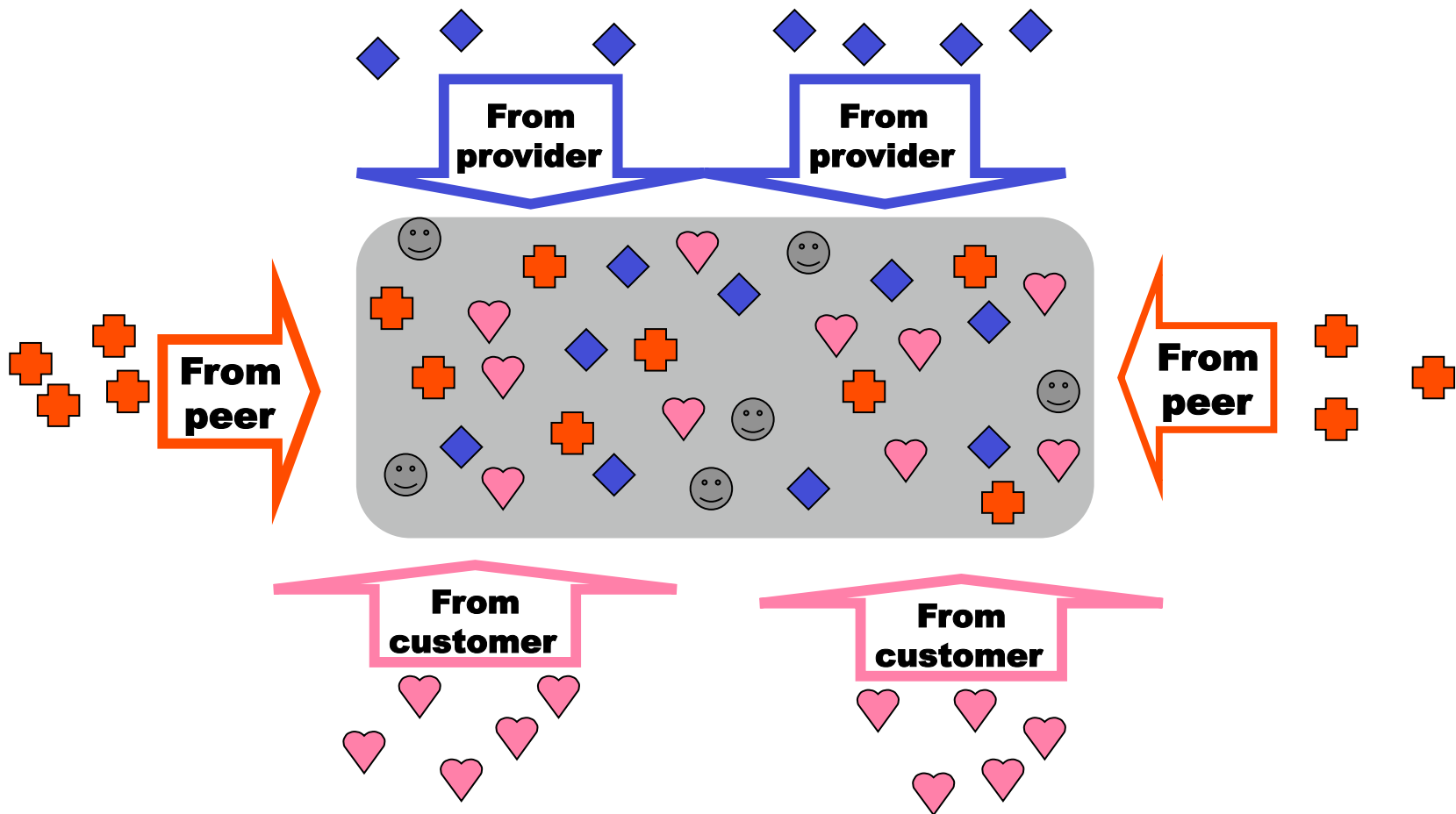
- Where do you want your traffic to go?
 - It is difficult to get what you want, but you can try
- Control of how you accept and send routing updates to neighbors
 - prefer cheaper connections, load-sharing, etc.
- Accepting routes from some ISPs and not others
- Sending some routes to some ISPs and not others
- Preferring routes from some ISPs over others

Routing Policy

- Why?
 - To steer traffic through preferred paths
 - Inbound/Outbound prefix filtering
 - To enforce Customer-ISP agreements
- How?
 - AS based route filtering – filter list
 - Prefix based route filtering – prefix list
 - BGP attribute modification – route maps
 - Complex route filtering – route maps

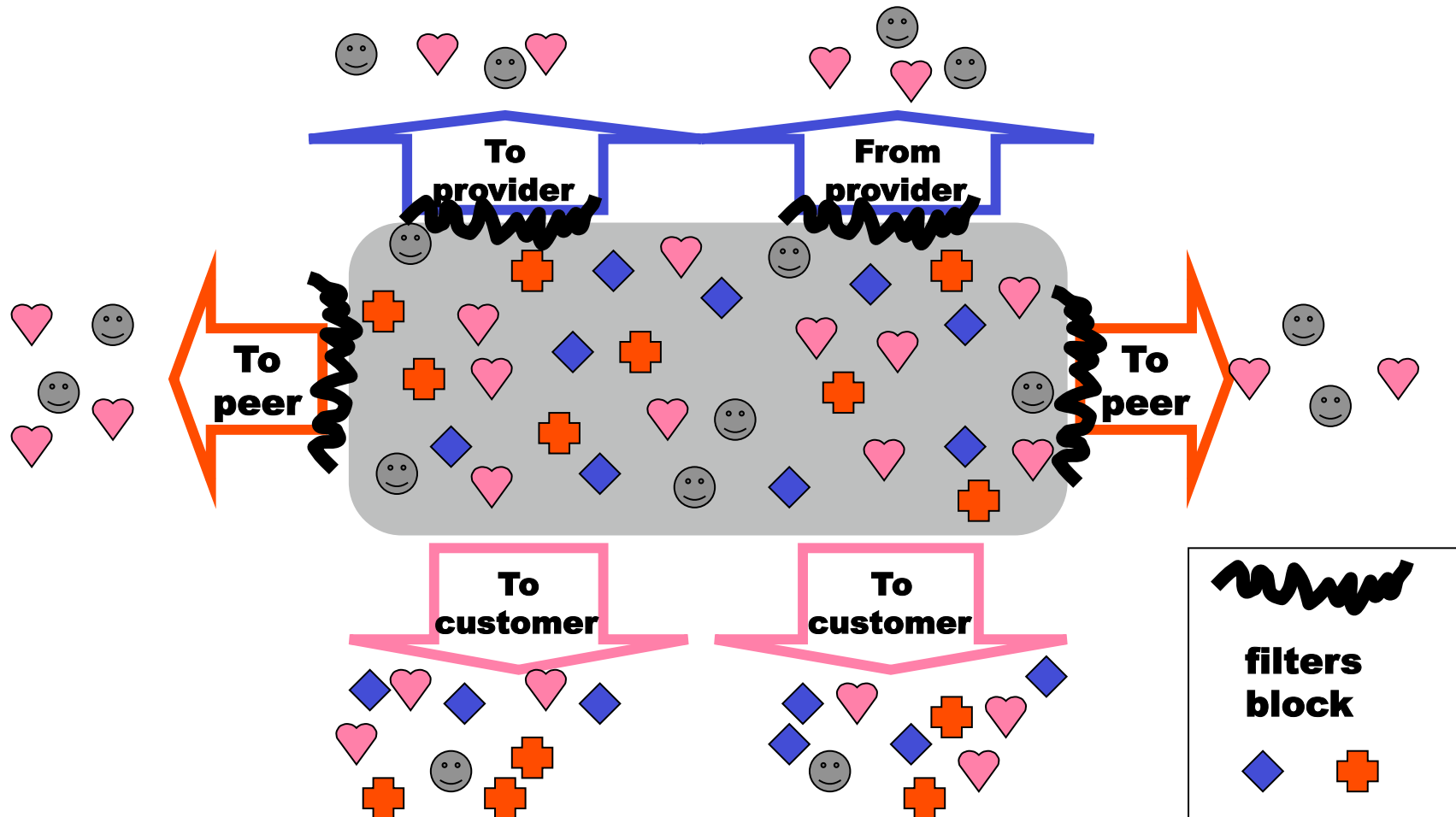
Import Routes

◆ provider route + peer route ♥ customer route ☺ ISP route



Export Routes

◆ provider route + peer route ♥ customer route ☺ ISP route



Filter list rules:

Regular Expressions

- Regular Expression is a pattern to match against an input string
- Used to match against AS-path attribute
- ex: `^3561_.*_100_.*_1$`
- Flexible enough to generate complex filter list rules

Regular expressions (cisco specific)

- `^` matches start
- `$` matches end
- `_` matches start, or end, or space (boundary between words or numbers)
- `.*` matches anything (0 or more characters)
- `.+` matches anything (1 or more characters)
- `[0-9]` matches any number between 0 and 9
- `^$` matches the local AS
- There are many more possibilities

Filter list – using as-path access list

- Listen to routes originated by AS 3561. Implicit deny everything else inbound.
- Don't announce routes originated by AS 35, but announce everything else (outbound).

```
ip as-path access-list 1 permit _3561$  
ip as-path access-list 2 deny _35$  
ip as-path access-list 2 permit .*
```

```
router bgp 100  
  neighbor 171.69.233.33 remote-as 33  
  neighbor 171.69.233.33 filter-list 1 in  
  neighbor 171.69.233.33 filter-list 2 out
```

Policy Control – Prefix Lists

- Per neighbor prefix filter
 - incremental configuration
- High performance access list
- Inbound or Outbound
- Based upon network numbers (using CIDR address/mask format)
- First relevant “allow” or “deny” rule wins
- Implicit Deny All as last entry in list

Prefix Lists – Examples

- Deny default route
 - `ip prefix-list Example deny 0.0.0.0/0`
- Permit the prefix 35.0.0.0/8
 - `ip prefix-list Example permit 35.0.0.0/8`
- Deny the prefix 172.16.0.0/12, and all more-specific routes
 - `ip prefix-list Example deny 172.16.0.0/12 ge 12`
 - “ge 12” means “prefix length /12 or longer”. For example, 172.17.0.0/16 will also be denied.
- In 192.0.0.0/8, allow any /24 or shorter prefixes
 - `ip prefix-list Example permit 192.0.0.0/8 le 24`
 - This will not allow any /25, /26, /27, /28, /29, /30, /31 or /32

Prefix Lists – More Examples

- In 192/8 deny /25 and above

```
ip prefix-list Example deny 192.0.0.0/8 ge 25
```

 - This denies all prefix sizes /25, /26, /27, /28, /29, /30, /31 and /32 in the address block 192.0.0.0/8
 - It has the same effect as the previous example
- In 192/8 permit prefixes between /12 and /20

```
ip prefix-list Example permit 192.0.0.0/8 ge 12 le 20
```

 - This denies all prefix sizes /8, /9, /10, /11, /21, /22 and higher in the address block 193.0.0.0/8
- Permit all prefixes
 - `ip prefix-list Example 0.0.0.0/0 le 32`

Policy Control Using Prefix Lists

- Example Configuration

```
router bgp 200
  network 215.7.0.0
  neighbor 220.200.1.1 remote-as 210
  neighbor 220.200.1.1 prefix-list PEER-IN in
  neighbor 220.200.1.1 prefix-list PEER-OUT out
!
ip prefix-list PEER-IN deny 218.10.0.0/16
ip prefix-list PEER-IN permit 0.0.0.0/0 le 32
ip prefix-list PEER-OUT permit 215.7.0.0/16
ip prefix-list PEER-OUT deny 0.0.0.0/0 le 32
```

- Accept everything except our network from our peer
- Send only our network to our peer

Prefix-lists in IPv6

- Prefix-lists in IPv6 work the same way as they do in IPv4
 - Caveat: ipv6 prefix-lists cannot be used for ipv4 neighbours - and vice-versa
 - Syntax is very similar, for example:

```
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
```

Policy Control – Route Maps

- A route-map is like a “program” for Cisco IOS
- Has “line” numbers, like programs
- Each line is a separate condition/action
- Concept is basically:
 - if match then do expression and exit*
 - else*
 - if match then do expression and exit*
 - else etc*

Route-map match & set clauses

- Match Clauses
 - AS-path
 - Community
 - IP address
- Set Clauses
 - AS-path prepend
 - Community
 - Local-Preference
 - MED
 - Origin
 - Weight
 - Others...

Route Map: Example One

```
router bgp 300
  neighbor 2.2.2.2 remote-as 100
  neighbor 2.2.2.2 route-map SETCOMMUNITY out
!
route-map SETCOMMUNITY permit 10
  match ip address 1
  match community 1
  set community 300:100
!
access-list 1 permit 35.0.0.0
ip community-list 1 permit 100:200
```

Route Map: Example Two

- Example Configuration as AS PATH prepend

```
router bgp 300
  network 215.7.0.0
  neighbor 2.2.2.2 remote-as 100
  neighbor 2.2.2.2 route-map SETPATH out
!
route-map SETPATH permit 10
  set as-path prepend 300 300
```

- Use your own AS number for prepending
 - Otherwise BGP loop detection will cause disconnects

BGP Part V



More detail than you want

BGP Attributes
Synchronization
Path Selection

BGP Path Attributes: Why ?

- Encoded as Type, Length & Value (TLV)
- Transitive/Non-Transitive attributes
- Some are mandatory
- Used in path selection
- To apply policy for steering traffic

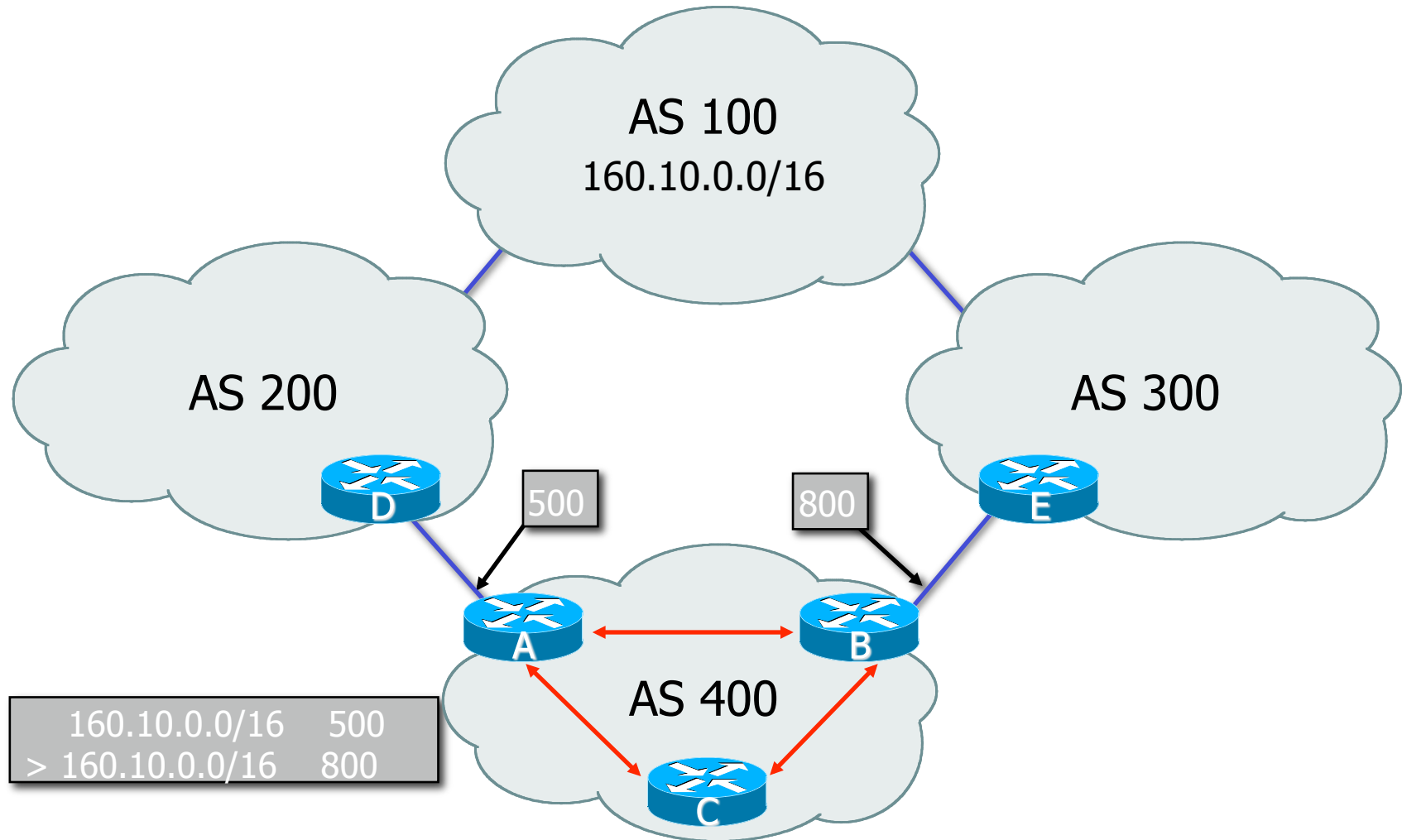
BGP Attributes

- Used to convey information associated with NLRI
 - AS path
 - Next hop
 - Local preference
 - Multi-Exit Discriminator (MED)
 - Community
 - Origin
 - Aggregator

Local Preference

- Not used by eBGP, mandatory for iBGP
- Default value of 100 on Cisco IOS
- Local to an AS
- Used to prefer one exit over another
- Path with highest local preference wins

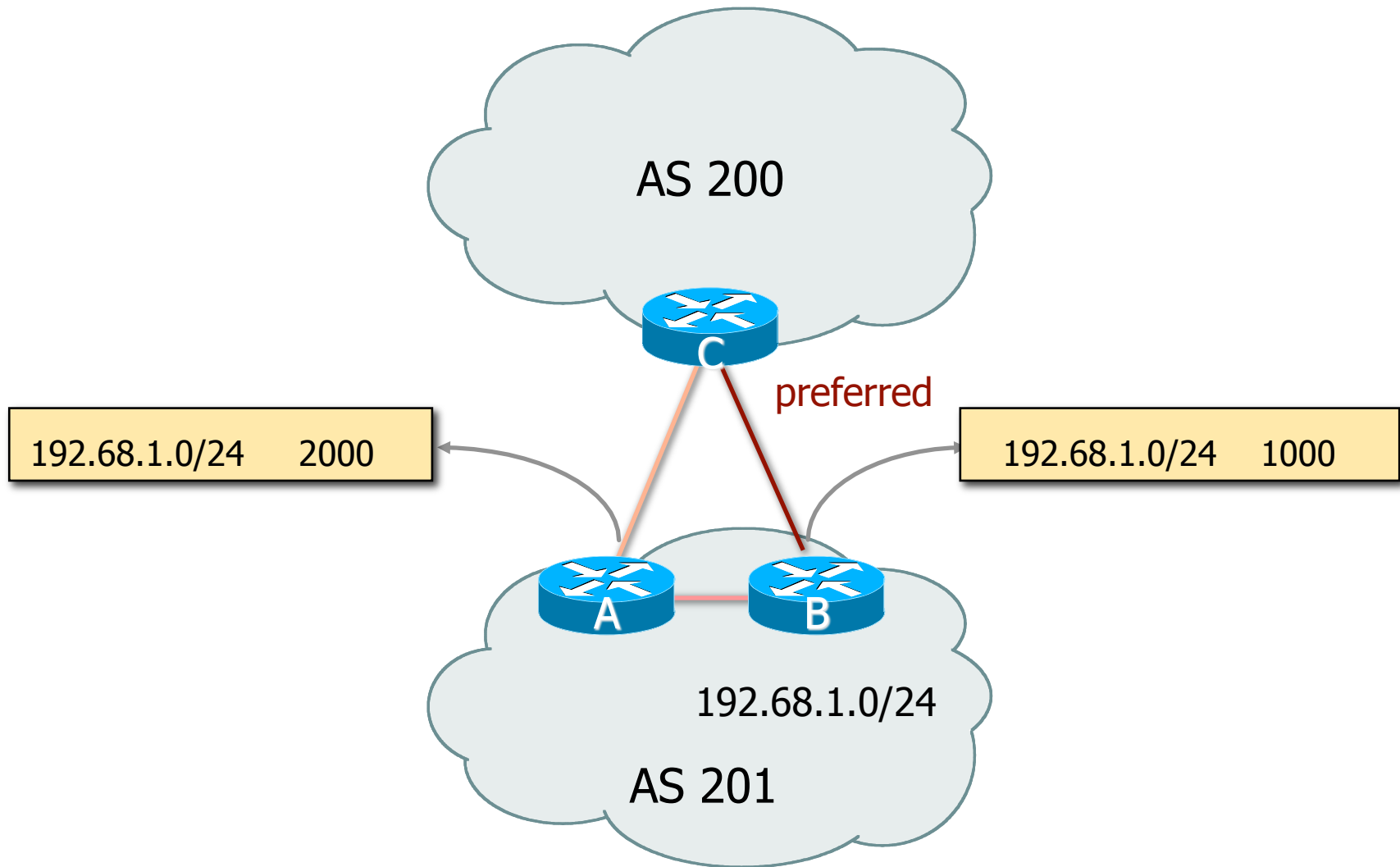
Local Preference



Multi-Exit Discriminator

- Non-transitive
- Represented as a numerical value
 - Range 0x0 – 0xffffffff
- Used to convey relative preference of entry points to an AS
- Comparable if the paths are from the same AS
- Path with the lowest MED wins
- IGP metric can be conveyed as MED

Multi-Exit Discriminator (MED)



Origin

- Conveys the origin of the prefix
 - **Historical** attribute
- Three values:
 - IGP – from BGP network statement
 - E.g. – *network 35.0.0.0*
 - EGP – redistributed from EGP (not used today)
 - Incomplete – redistributed from another routing protocol
 - E.g. – *redistribute static*
- IGP < EGP < incomplete
 - Lowest origin code wins

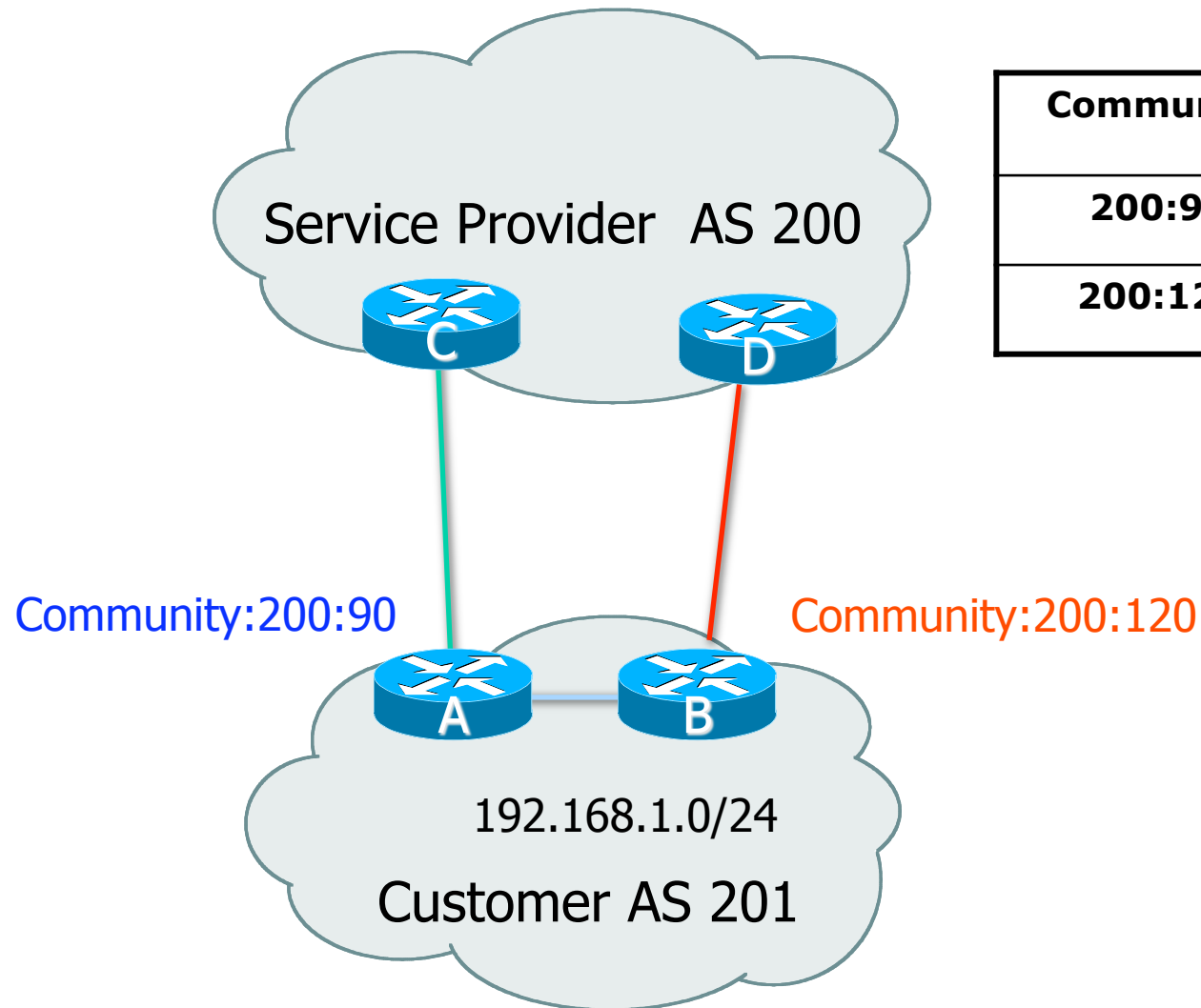
Weight

- Not really an attribute
- Used when there is more than one route to same destination
- Local to the router on which it is assigned, and not propagated in routing updates
- Default is 32768 for paths that the router originates and zero for other paths
- Routes with a higher weight are preferred when there are multiple routes to the same destination

Communities

- Transitive, Non-mandatory
- Represented as a numeric value
 - 0x0 – 0xffffffff
 - Internet convention is ASn:<0-65535>
- Used to group destinations
- Each destination could be member of multiple communities
- Flexibility to scope a set of prefixes within or across AS for applying policy

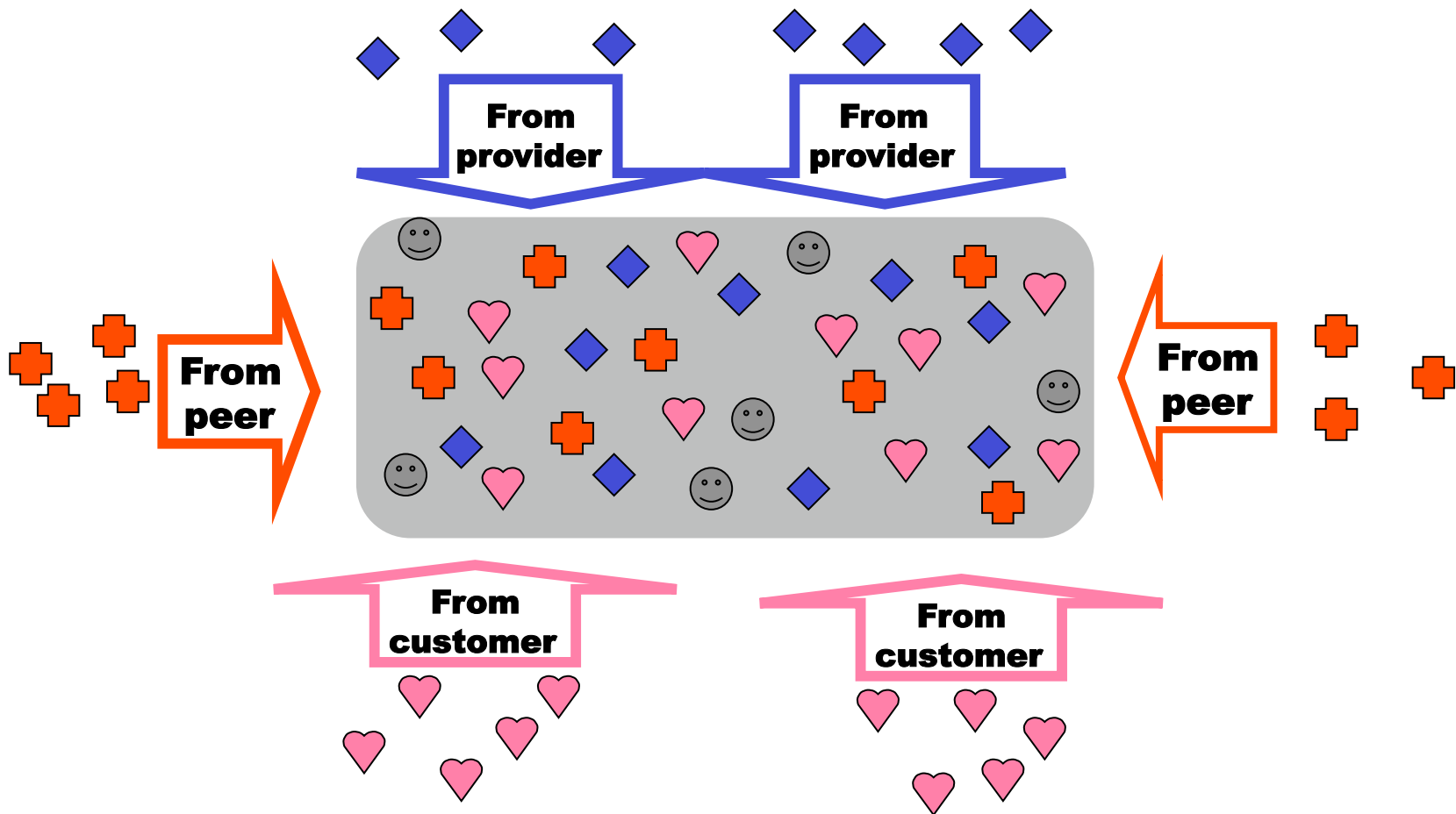
Communities



Community	Local Preference
200:90	90
200:120	120

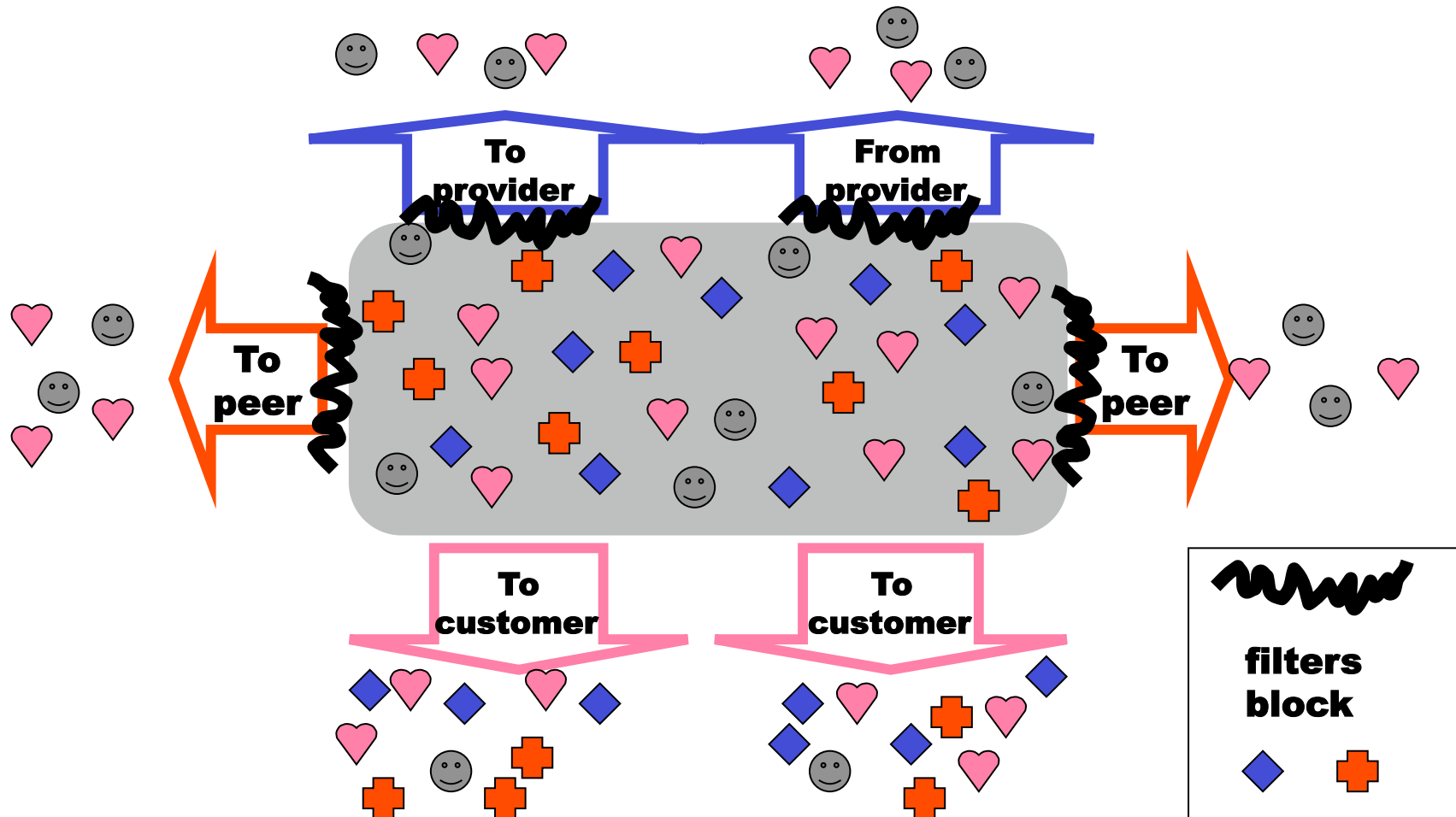
Import Routes

◆ provider route + peer route ♥ customer route ☺ ISP route



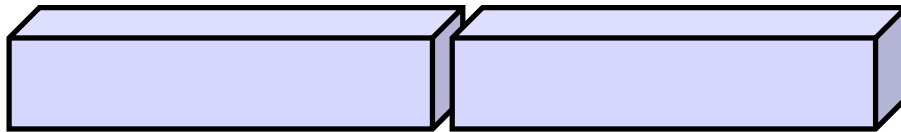
Export Routes

◆ provider route + peer route ♥ customer route ☺ ISP route



How Can Routes be Colored? BGP Communities!

A community value is 32 bits



By convention,
first 16 bits is
ASN indicating
who is giving it
an interpretation




community
number

Used for signaling
within and between
ASes

Very powerful
BECAUSE it
has no (predefined)
meaning

Community Attribute = a list of community values.
(So one route can belong to multiple communities)

Communities Example

- 1:100 
 - Customer routes
- 1:200 
 - Peer routes
- 1:300 
 - Provider Routes

Import

- To Customers
 - 1:100, 1:200, 1:300
- To Peers
 - 1:100
- To Providers
 - 1:100

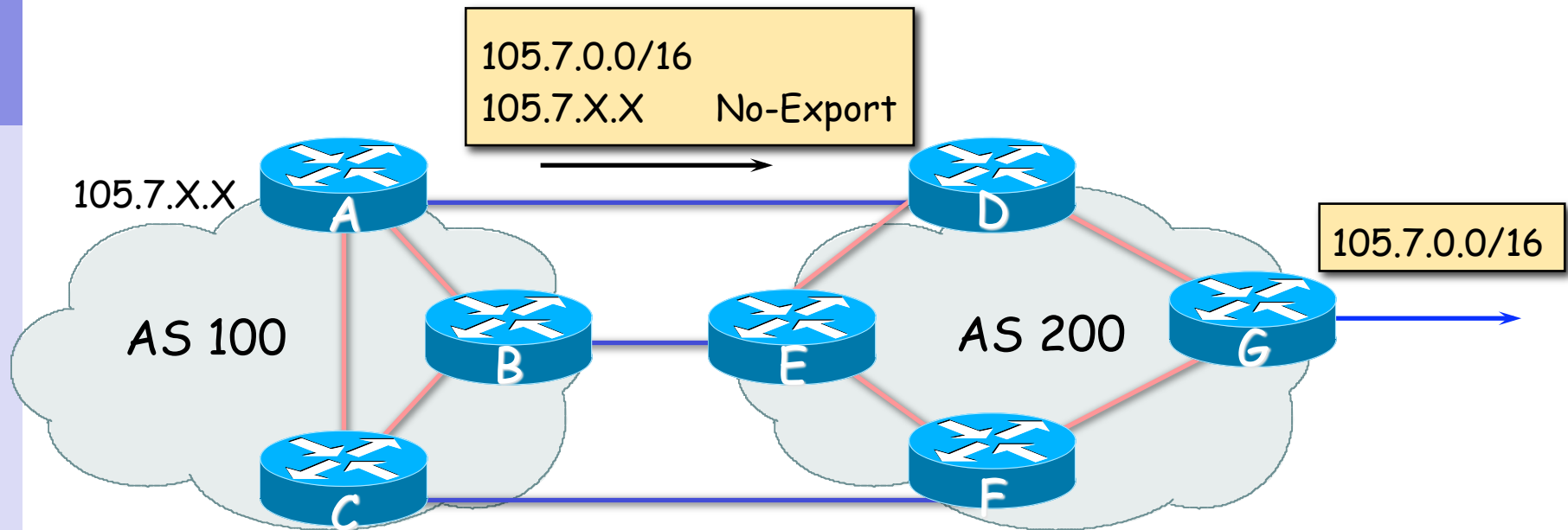
Export

AS 1

Well-Known Communities

- Several well known communities
www.iana.org/assignments/bgp-well-known-communities
- no-export 65535:65281
 - do not advertise to any eBGP peers
- no-advertise 65535:65282
 - do not advertise to any BGP peer
- no-export-subconfed 65535:65283
 - do not advertise outside local AS (only used with confederations)
- no-peer 65535:65284
 - do not advertise to bi-lateral peers (RFC3765)

No-Export Community

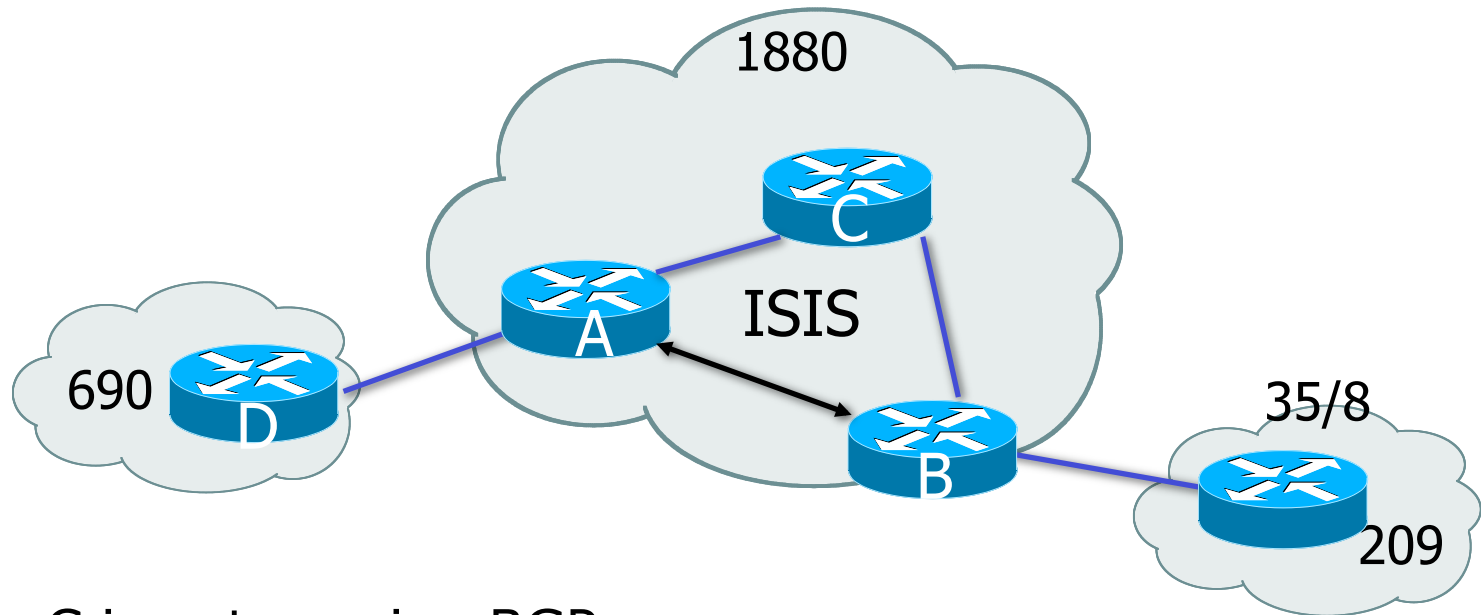


- AS100 announces aggregate and subprefixes
 - Intention is to improve loadsharing by leaking subprefixes
- Subprefixes marked with no-export community
- Router G in AS200 does not announce prefixes with no-export community set

Administrative Distance

- Routes can be learned via more than one protocol
 - Used to discriminate between them
- Route with lowest distance installed in forwarding table
- BGP defaults
 - Local routes originated on router: 200
 - iBGP routes: 200
 - eBGP routes: 20
- Does not influence the BGP path selection algorithm but influences whether BGP learned routes enter the forwarding table

Synchronization



- C is not running BGP
- A won't advertised 35/8 to D until the IGP is in sync
- Turn synchronization off!
`router bgp 1880`
`no synchronization`

Synchronization

- In Cisco IOS, BGP does not advertise a route before all routers in the AS have learned it via an IGP
 - Default in IOS prior to 12.4; very unhelpful to most ISPs
- Disable synchronization if:
 - AS doesn't pass traffic from one AS to another, or
 - All transit routers in AS run BGP, or
 - iBGP is used across backbone
- You should always use iBGP
 - so, always use "no synchronization"

BGP Policies

- Implements policies to enable politics and traffic engineering
- Decision process (in each router):



BGP route selection (bestpath)

- Route has to be synchronized
 - Only if synchronization is enabled
 - e.g., always use "*no synchronization*"
- Next-hop has to be accessible
 - Next-hop must be in forwarding table
- Largest weight
- Largest local preference

BGP route selection (bestpath)

- Locally sourced
 - Via redistribute or network statement
- Shortest AS path length
 - Number of ASes in the AS-PATH attribute
- Lowest origin
 - IGP < EGP < incomplete
- Lowest MED
 - Compared from paths from the same AS

BGP route selection (bestpath)

- External before internal
 - Choose external path before internal
- Closest next-hop
 - Lower IGP metric, nearest exit to router
- Lowest router ID
- Lowest IP address of neighbour

BGP Part VI



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>
```


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

Configure iBGP

- The two routers in your AS should talk iBGP to each other
 - no filtering here
 - use “update-source loopback 0”

“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

“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

BGP Part VII

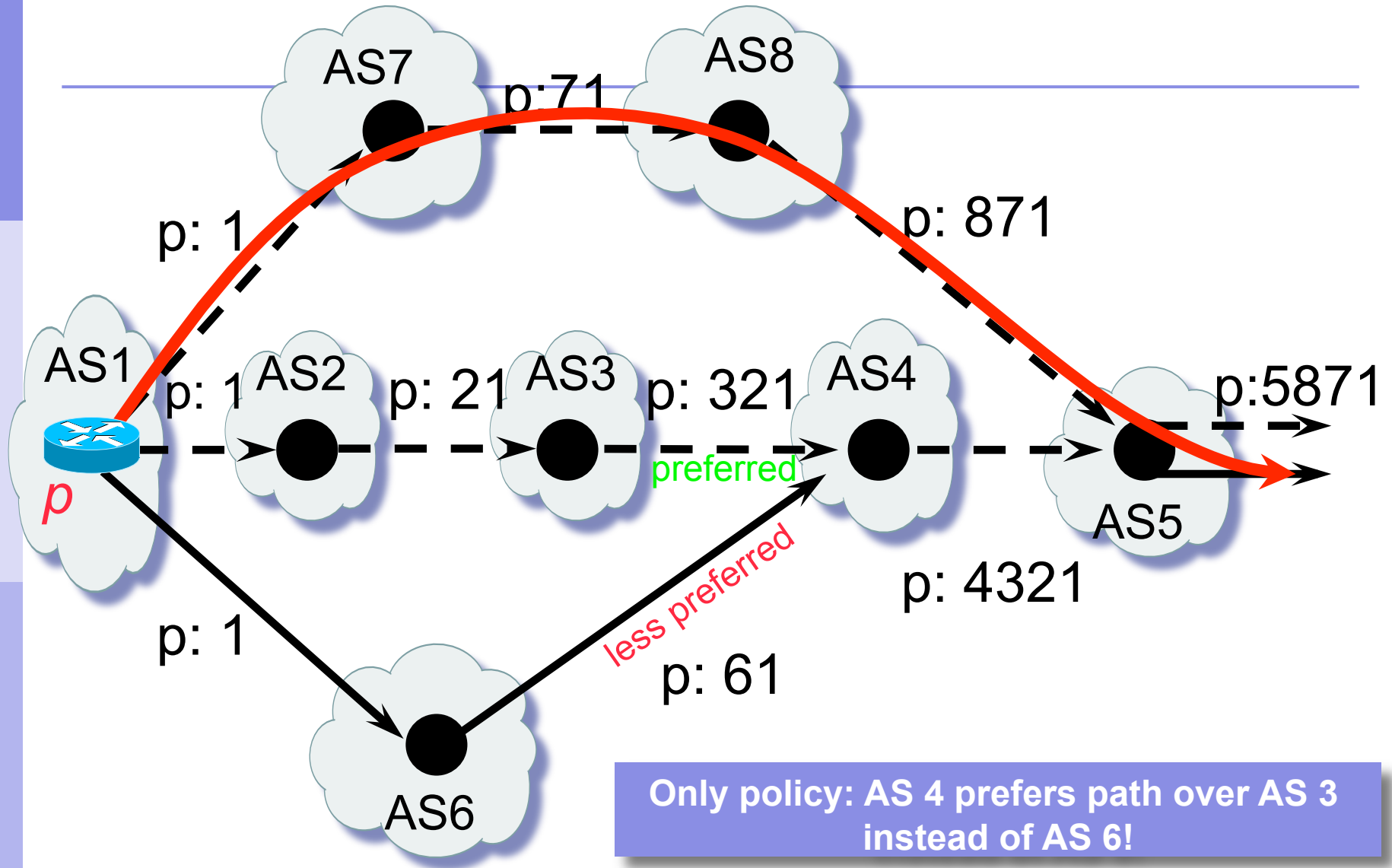


Complexity of large networks

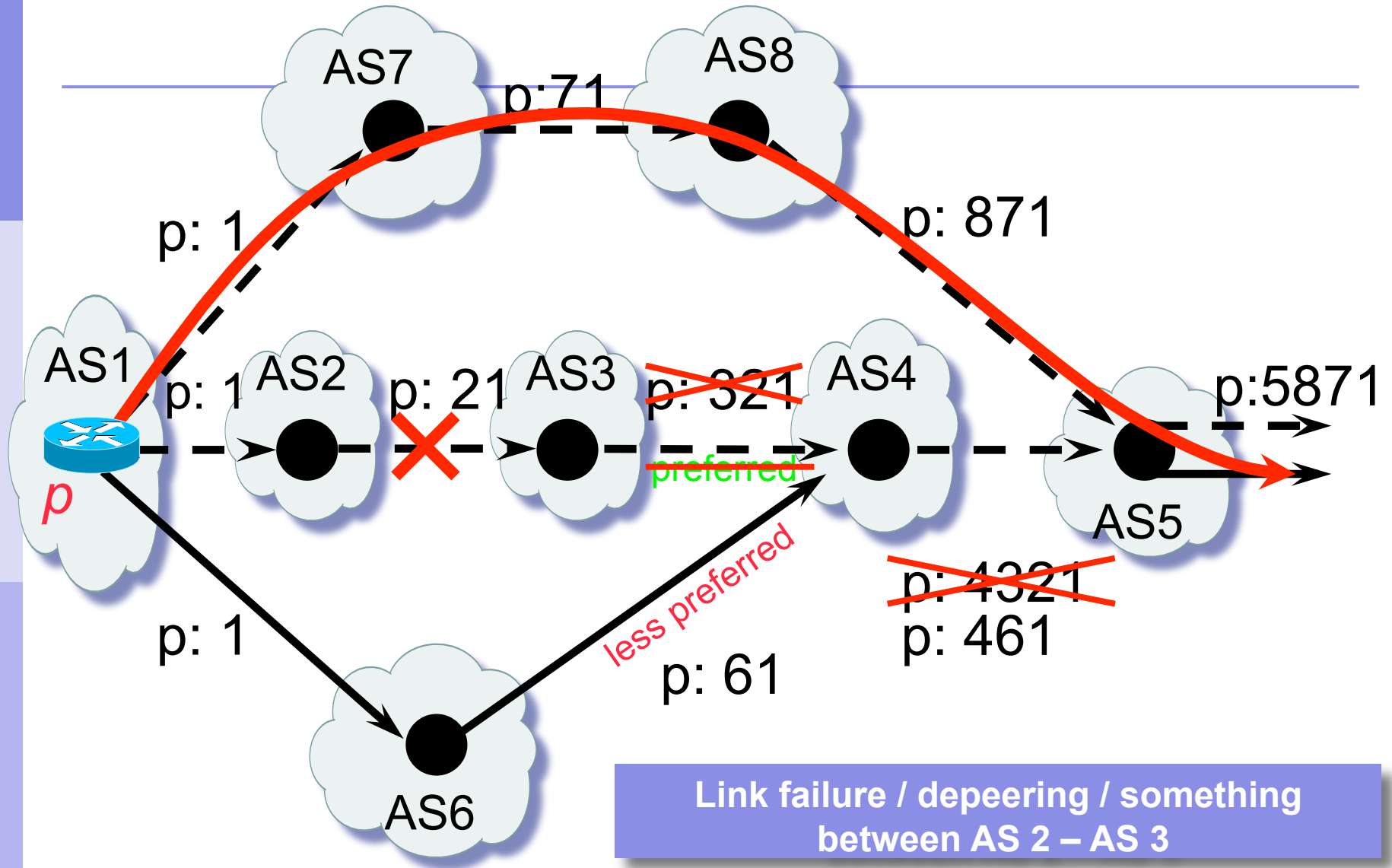
BGP Wedgies

(Tim Griffin)

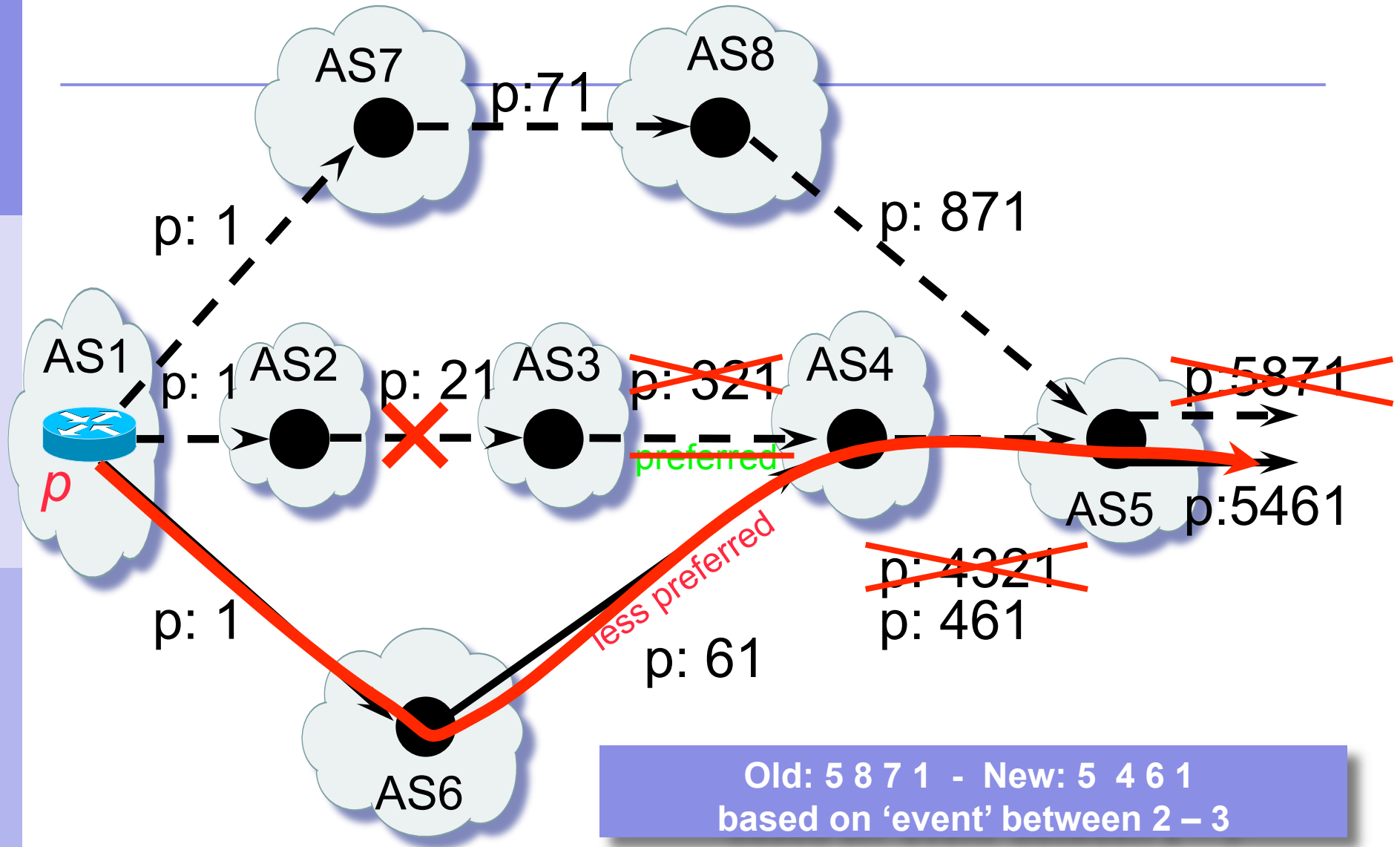
Policy Interactions



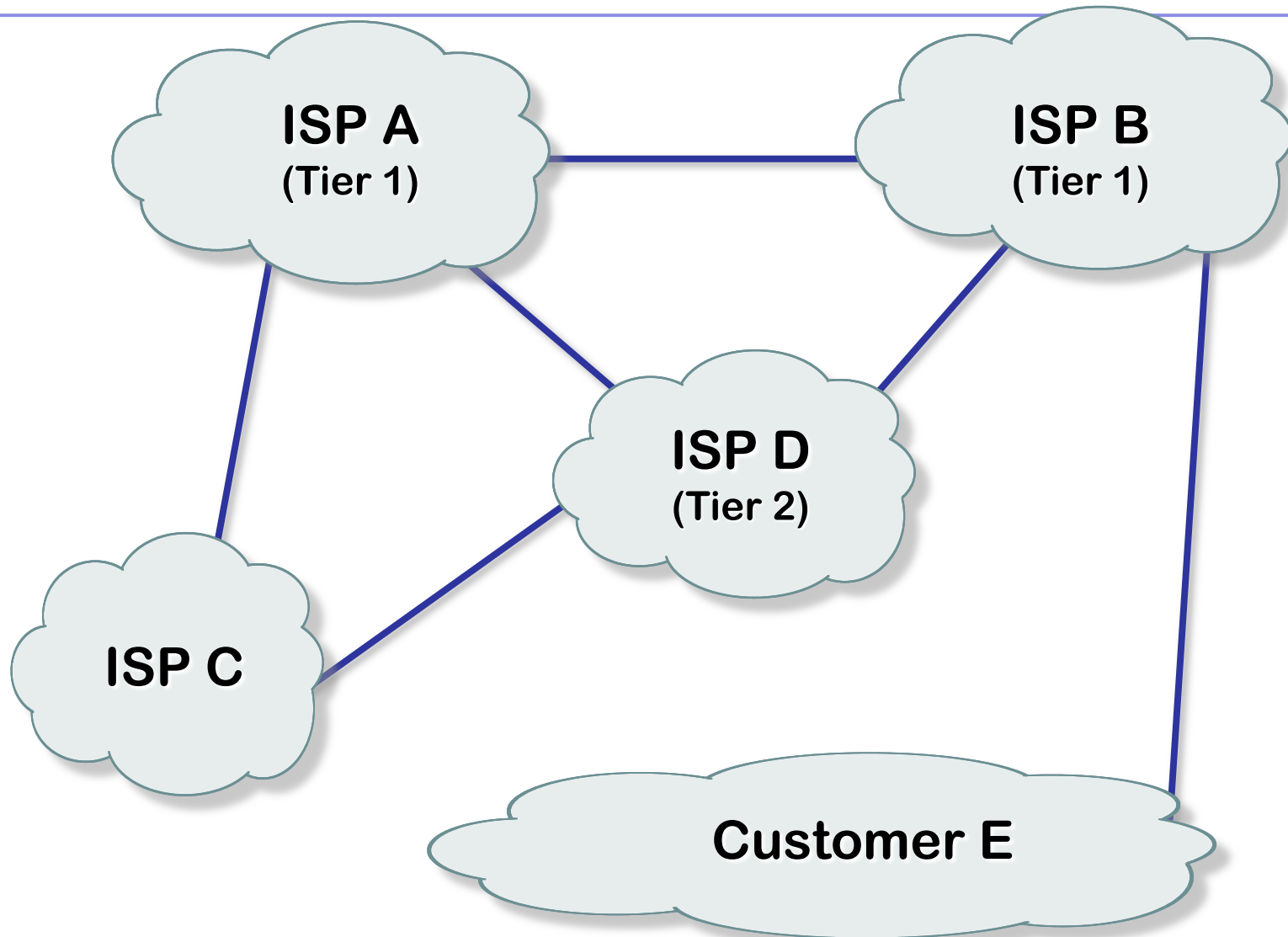
Policy Interactions



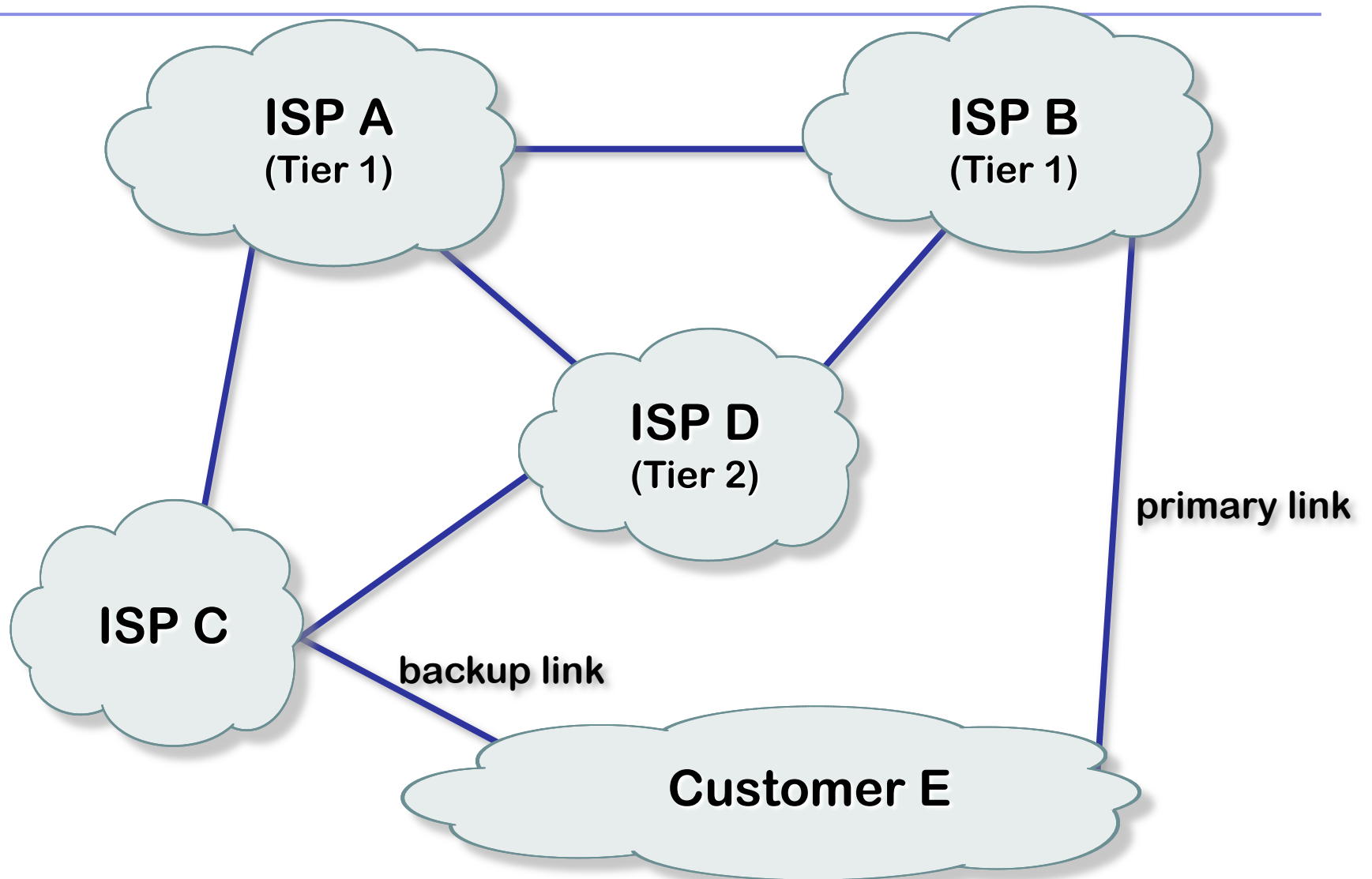
Policy Interactions



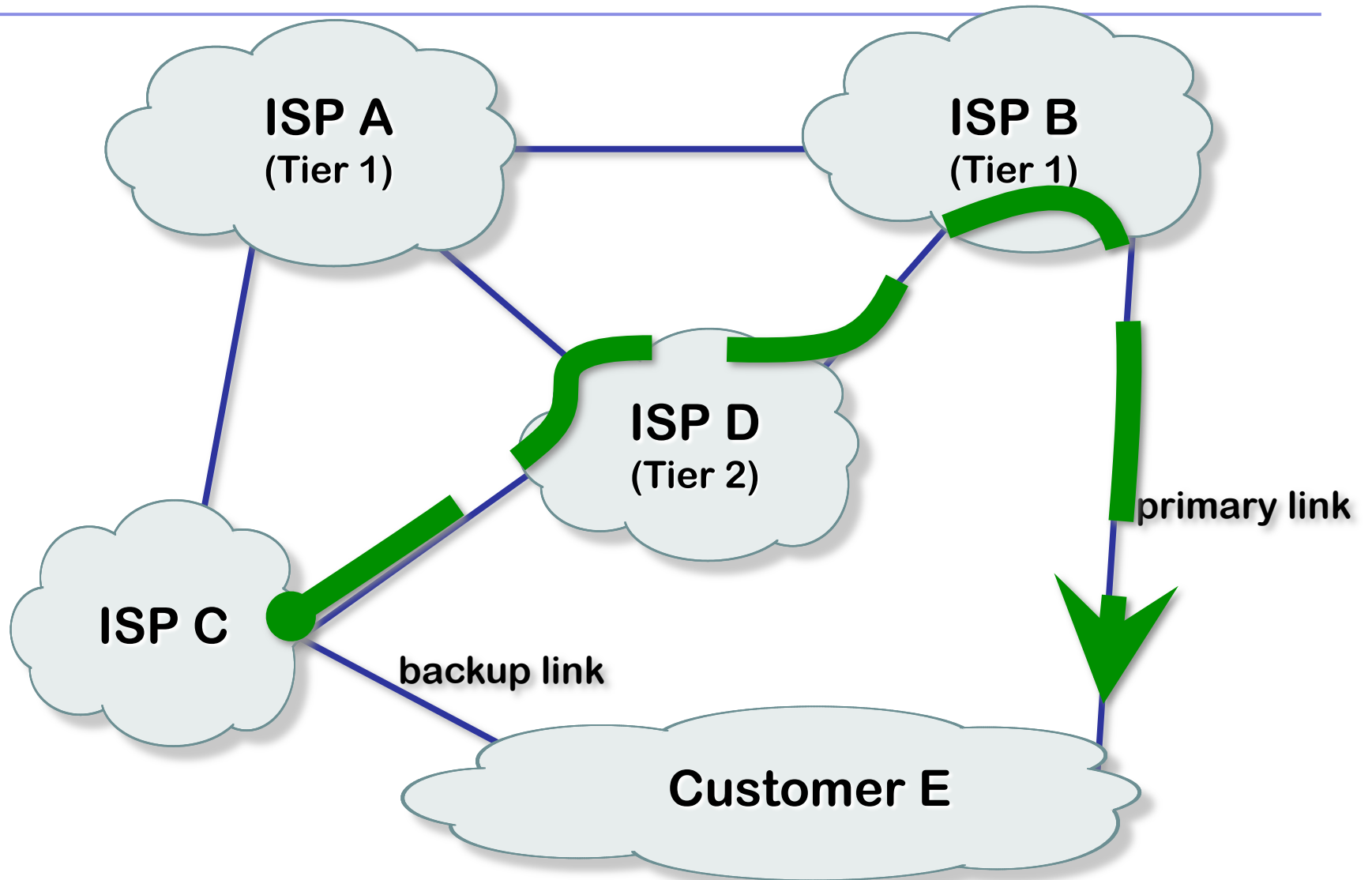
Tim Griffin: “BGP Wedgies”



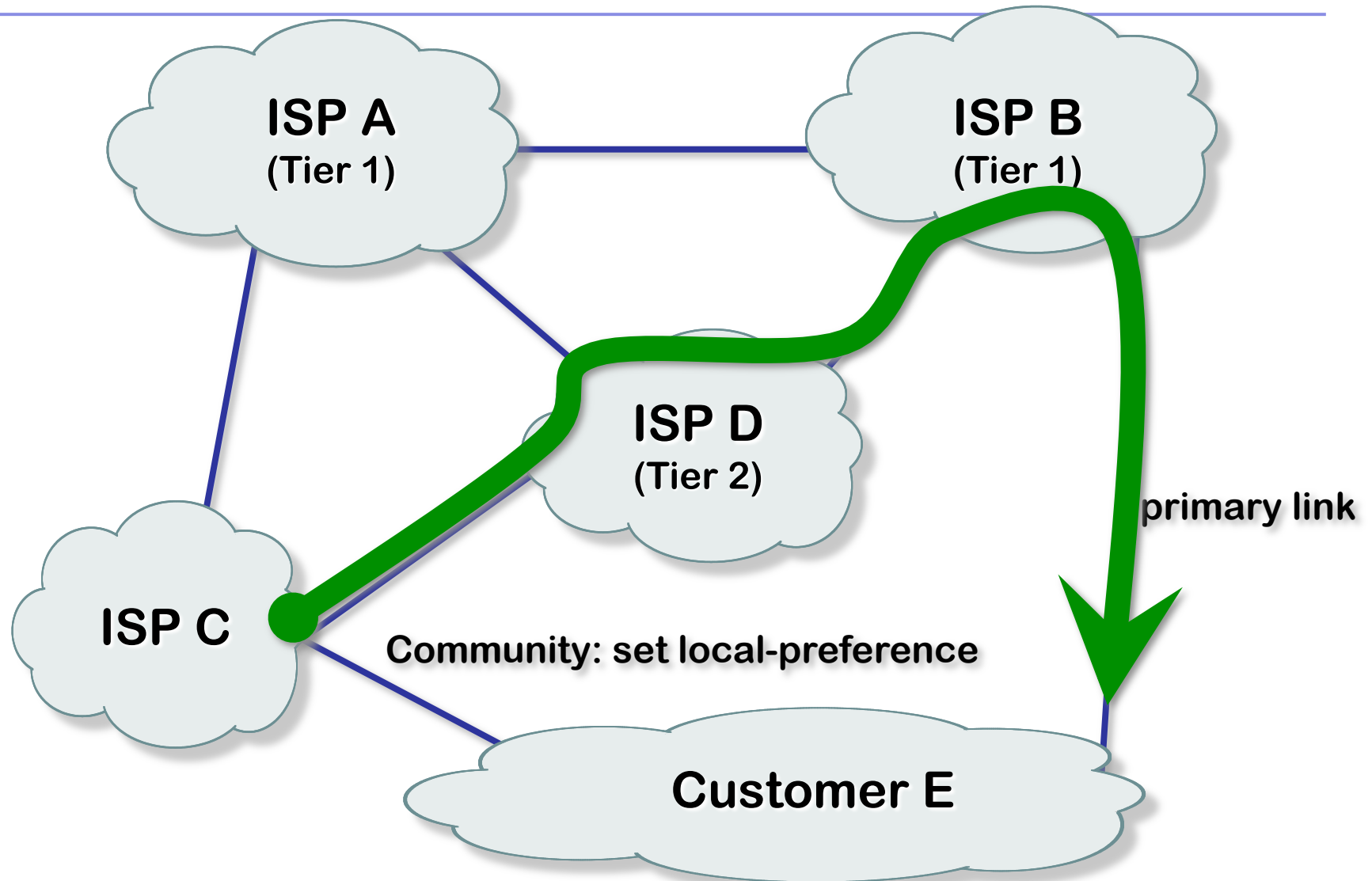
Tim Griffin: "BGP Wedgies"



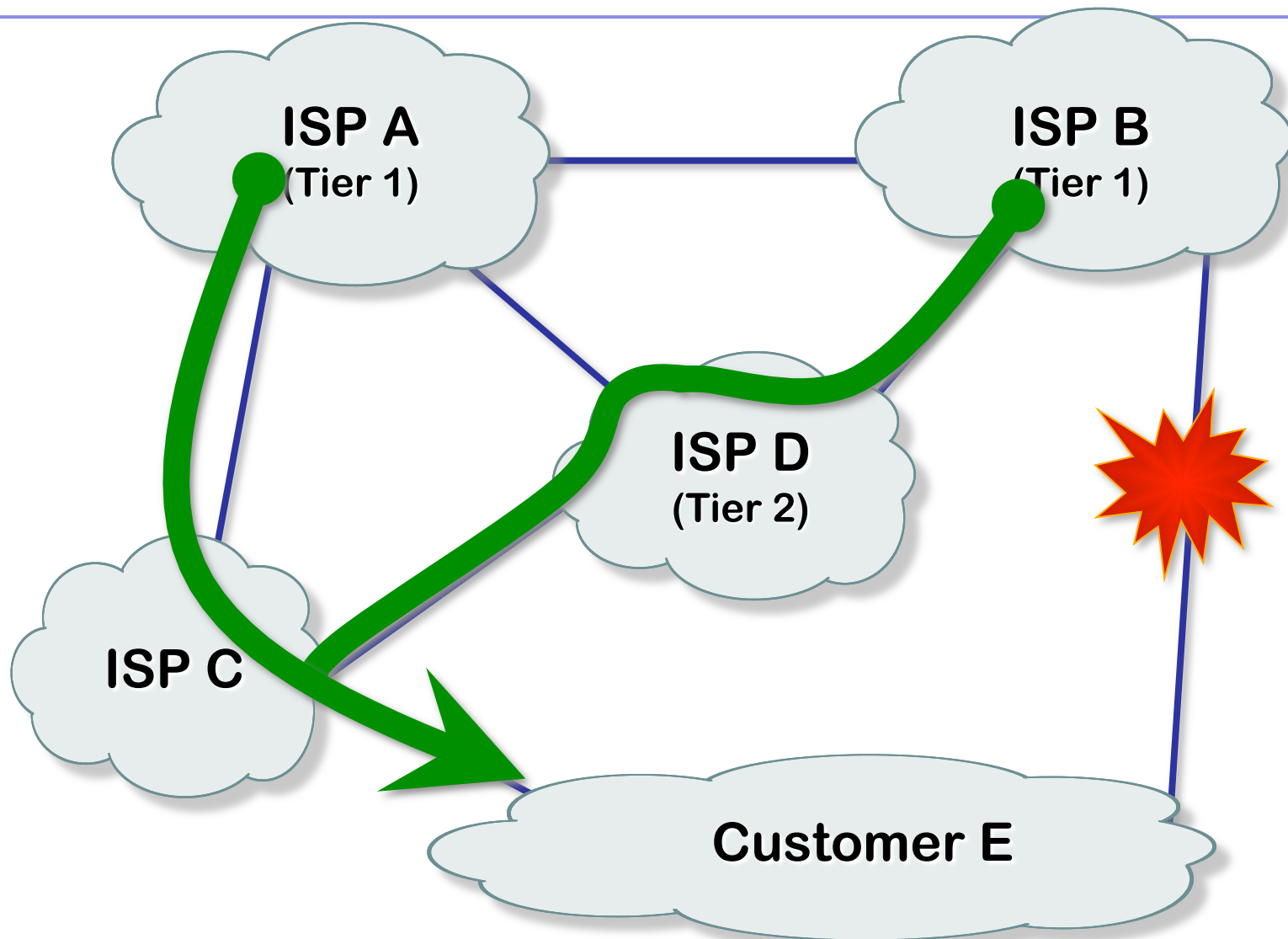
Desired Situation...



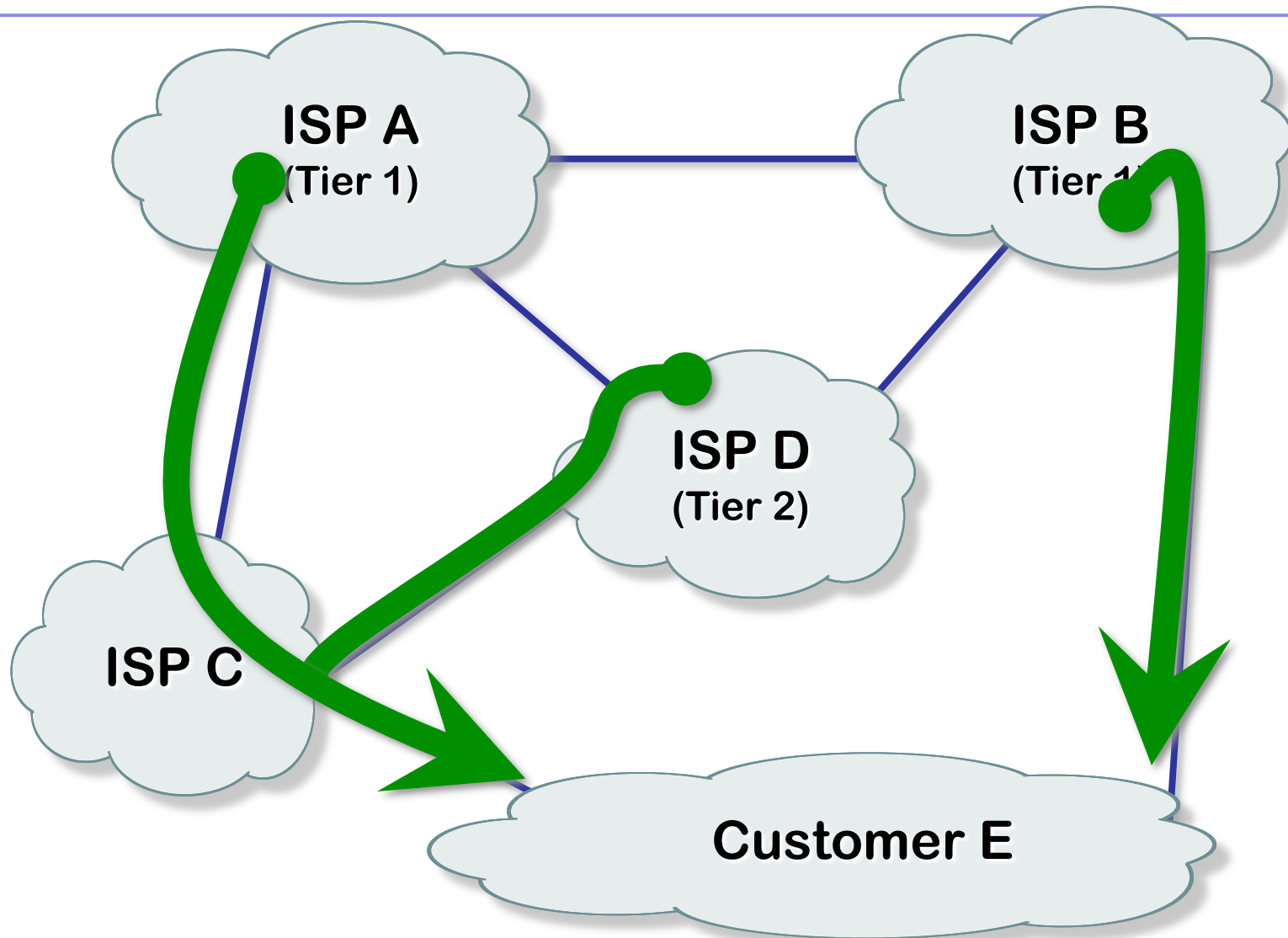
Desired Situation via communities



Primary link fails...



Primary link recovers...



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
 - Where complexity comes from...
 - Limitations of the “Internet”