Network Layer - 1

Outline

Network addresses

Forwarding vs Routing

ARP, RARP

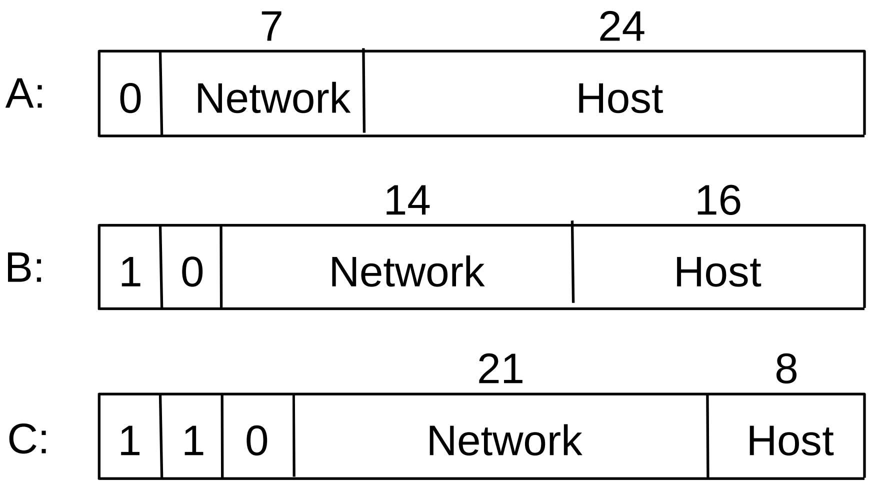
IP Service Model

Addressing

- IP Address: byte-string that identifies a node
  - usually unique (some exceptions)
  - Dotted decimal notation: 128.92.54.32
- Types of addresses
  - unicast: node-specific
  - broadcast: all nodes on the network
  - multicast: some subset of nodes on the network

Global Addresses

- Properties
  - globally unique
  - hierarchical: network + host
- Dotted Decimal Notation
  - 120.3.2.4
  - 128.96.33.81
  - 192.12.69.77
- Address classes
  - A, B, C (shown)



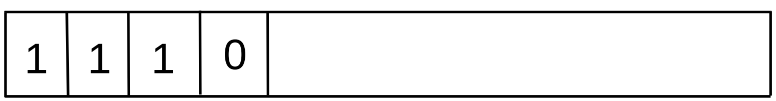
- Network respresented as Network Part / Num. Bits
  - E.g. 120/8 or 128.96/16

Exercise: Find out about private addresses

Other Addresses

- Private address:
  - 10.0.0.0 to 10.255.255.255
  - 172.16.0.0 to 172.16.255.255
  - 192.168.0.0 to 192.168.255.255
  - 169.254.0.0 to 16.254.255.255

- Class D: multicast addresses: 224.0.0.0 to 224.255.255.255



- Host part all 1’s: broadcast in local network
- Host part all 0’s: unspecified (not allowed)

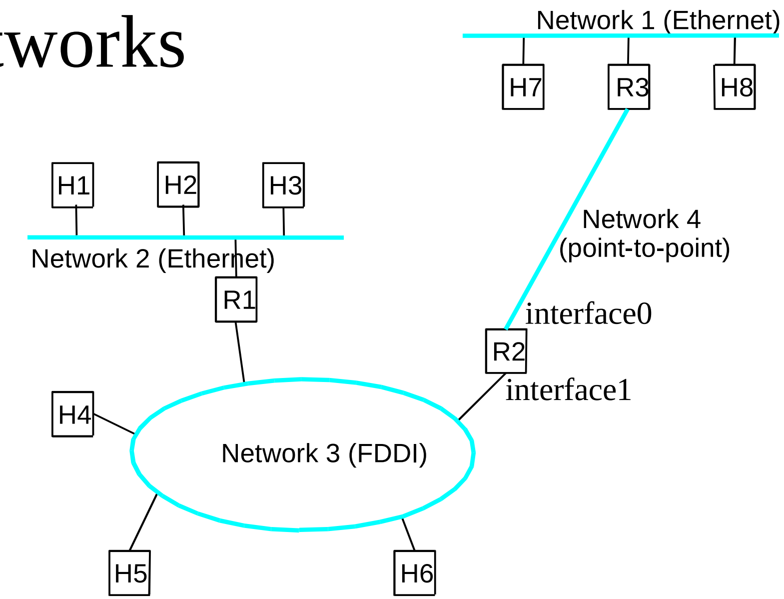
Route Propagation

- Know a smarter router
  - hosts know local router
  - local routers know site routers
  - site routers know core router
  - core routers know everything
- Autonomous System (AS)
  - corresponds to an administrative domain
  - examples: University, company, backbone network
  - assign each AS a 16-bit number
- Two-level route propagation hierarchy
  - interior gateway protocol (each AS selects its own)
  - exterior gateway protocol (Internet-wide standard)

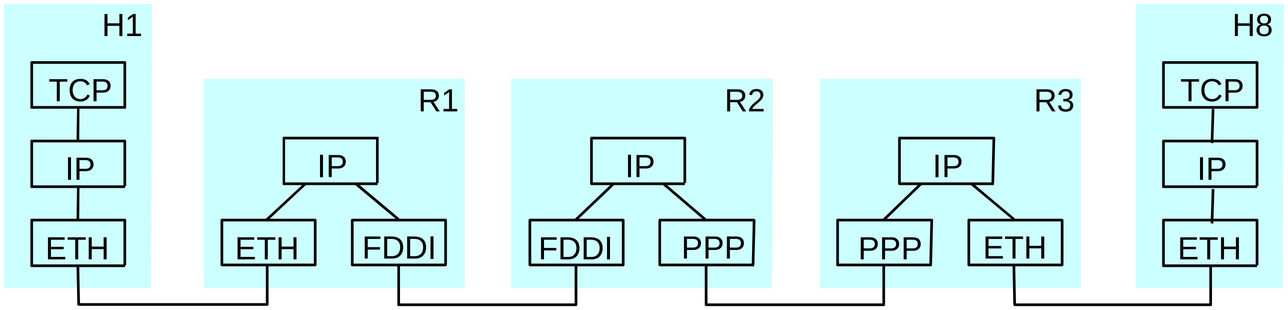
Data forwarding

IP Internet

- Concatenation of Networks



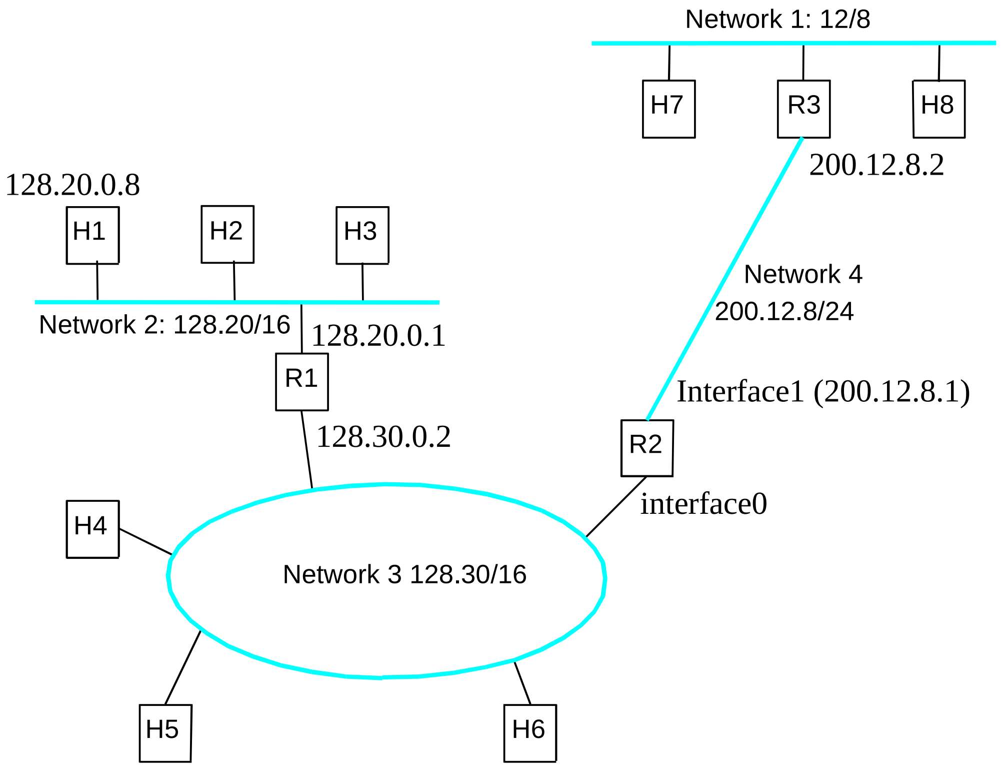
- Protocol Stack



Forwarding and Routing

- Routing: involves computation of routes
  - Which path to take
- Forwarding: select an output interface at each hop
  - Assumes routes have been computed
  - Depends only on destination IP address
- They are independent of each other

Forwarding


Network 1: 12/8
H7
R3
H8
200.12.8.2
128.20.0.8
H1
H2
H3
Network 4
200.12.8/24
Network 2: 128.20/16
128.20.0.1
R1
Interface1 (200.12.8.1)
128.30.0.2
R2
interface0
H4
Network 3 128.30/16
H5
H6

Forwarding Tables

- Suppose there are *n* possible destinations, how many bits are needed to represent addresses in a routing table?
  - $\log_2 n$
- So, we need to store and search $n * \log_2 n$ bits in routing tables?
  - We're smarter than that!

Datagram Forwarding

- Strategy
  - every datagram contains destination's address
  - if directly connected to destination network, then forward to host
  - if not directly connected to destination network, then forward to some router
  - forwarding table maps network number into next hop
  - each host has a default router
  - each router maintains a forwarding table
- Example

for router R2
in previous figure

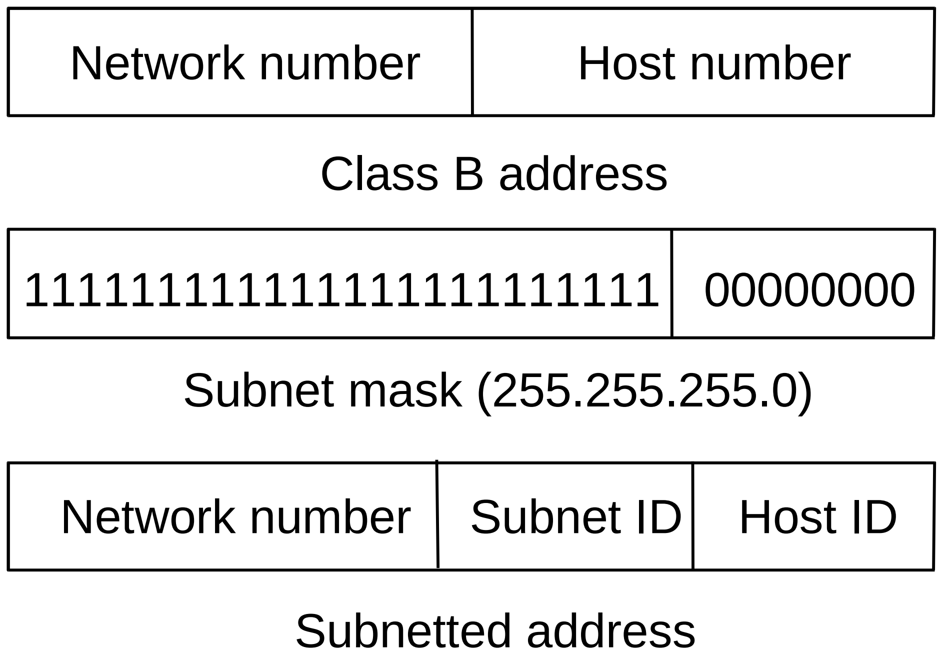
Network	Next Hop
1	R3
2	R1
3	interface 1
4	interface 0
default	R3

Subnetting and Supernetting

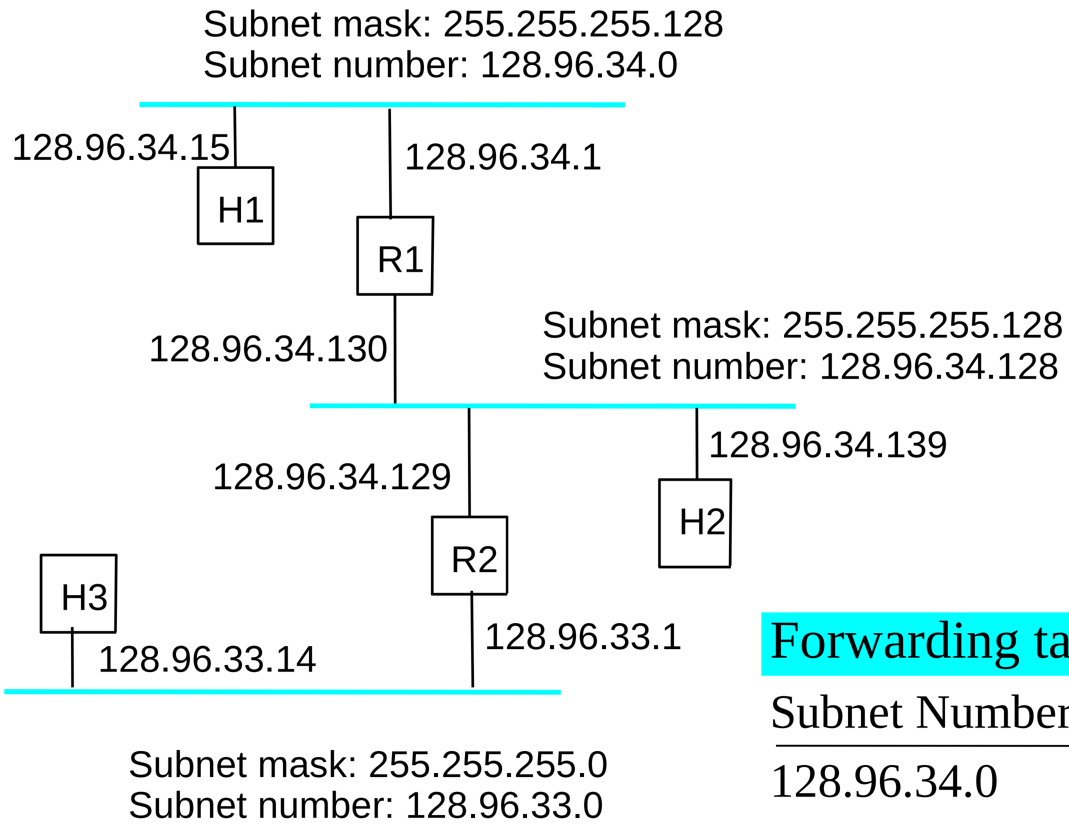
- Fixed network sizes are wasteful
  - What happens if a site asks for 256 IP addresses?
  - Subnetting
- Too many entries at a router can be combined
  - Keep routing tables small
  - Supernetting
- Classless Inter-Domain Routing (CIDR)

Subnetting

- Add another level to address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part
- Subnets visible only within site



Subnet Example



Forwarding table at router R1

Subnet Number	Subnet Mask	Next Hop
128.96.34.0	255.255.255.128	interface 0
128.96.34.128	255.255.255.128	interface 1
128.96.33.0	255.255.255.0	R2

Forwarding Algorithm

```
D = destination IP address
for each entry (SubnetNum, SubnetMask, NextHop)
   D1 = SubnetMask & D
   if D1 = SubnetNum
      if NextHop is an interface
         deliver datagram directly to D
      else
         deliver datagram to NextHop
```

- Use a default router if nothing matches
- Not necessary for all 1s in subnet mask to be contiguous
- Can put multiple subnets on one physical network
- Subnets not visible from the rest of the Internet

Supernetting

- Assign block of contiguous network numbers to nearby networks
- Called CIDR: Classless Inter-Domain Routing
- Represent blocks with a single pair

  `(first_network_address, count)`
- Restrict block sizes to powers of 2
- Use a bit mask (CIDR mask) to identify block size
- All routers must understand CIDR addressing

Forwarding Table Lookup

- Longest prefix match
  - Each entry in the forwarding table is:

    < Network Number / Num. Bits> | interface-id

Suppose we have:

192.20./16	|	i0
192.20.12/24	|	i1

And destination address is: 192.20.12.7, choose i1

Address Translation

- Map IP addresses into physical addresses
  - destination host
  - next hop router
- Techniques
  - encode physical address in host part of IP address
  - table-based
- ARP
  - table of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed

ARP Details

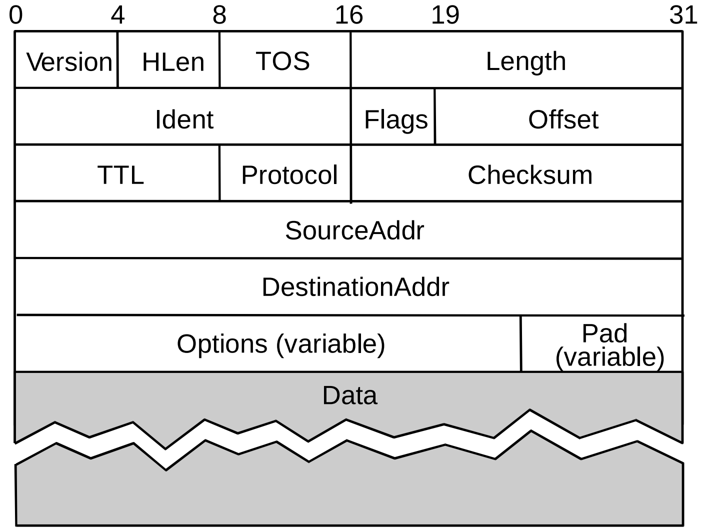
- Request Format
  - HardwareType: type of physical network (e.g., Ethernet)
  - ProtocolType: type of higher layer protocol (e.g., IP)
  - HLEN & PLEN: length of physical and protocol addresses
  - Operation: request or response
  - Source/Target-Physical/Protocol addresses
- Notes
  - table entries timeout in about 10 minutes
  - update table with source when you are the target
  - update table if already have an entry
  - do not refresh table entries upon reference

ARP Packet Format

0	8	16 31
Hardware type = 1		ProtocolType = 0x0800
HLen = 48	PLen = 32	Operation
SourceHardwareAddr (bytes 0 – 3)		
SourceHardwareAddr (bytes 4 – 5)		SourceProtocolAddr (bytes 0 – 1)
SourceProtocolAddr (bytes 2 – 3)		TargetHardwareAddr (bytes 0 – 1)
TargetHardwareAddr (bytes 2 – 5)		
TargetProtocolAddr (bytes 0 – 3)		

IP Service Model

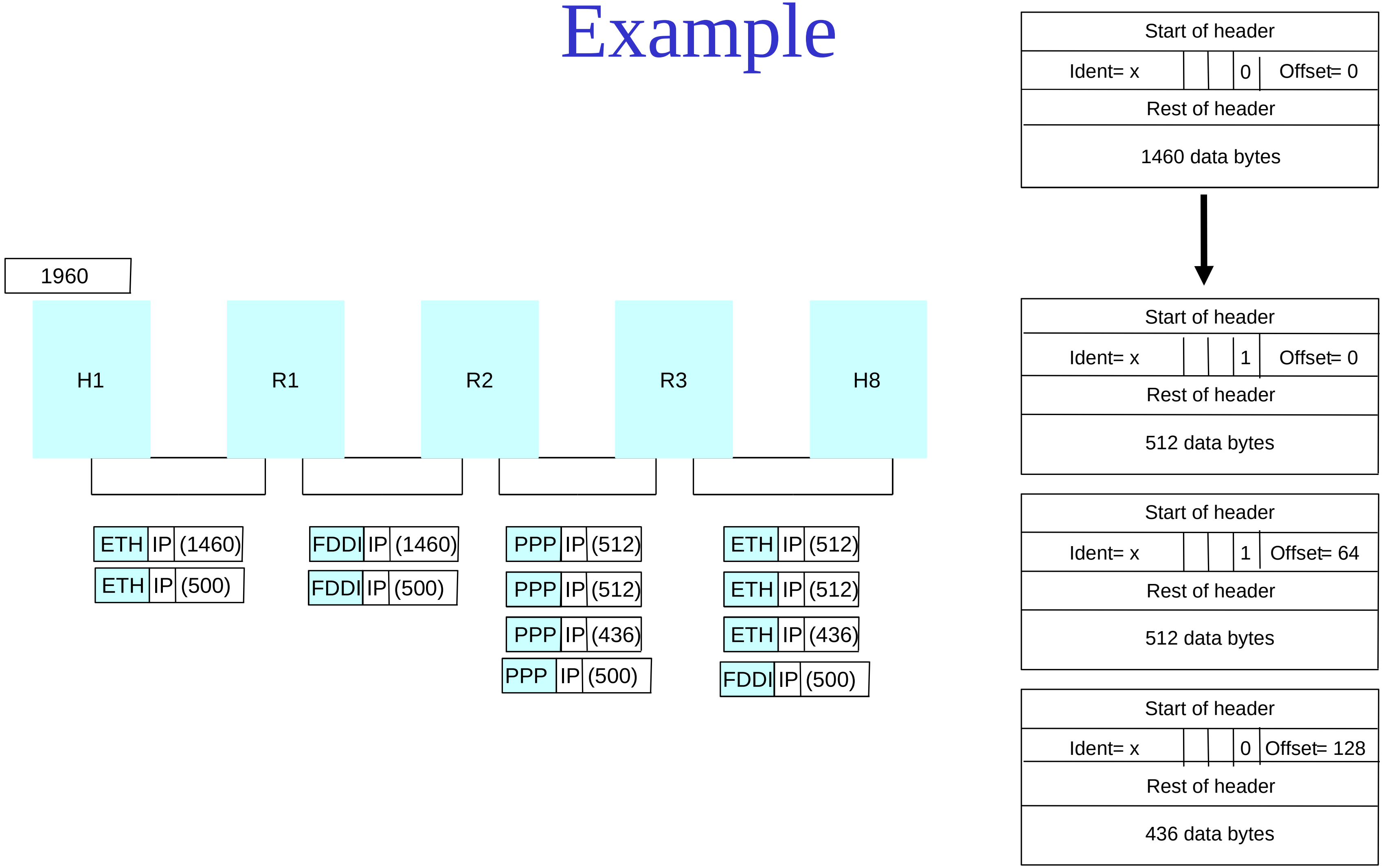
- Connectionless (datagram/packet-based)
- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time
- Datagram format



Fragmentation and Reassembly

- Each network has some MTU
- Design decisions
  - fragment when necessary (MTU < Datagram)
  - try to avoid fragmentation at source host
  - re-fragmentation is possible
  - fragments are self-contained datagrams
  - delay reassembly until destination host
  - do not recover from lost fragments

Example

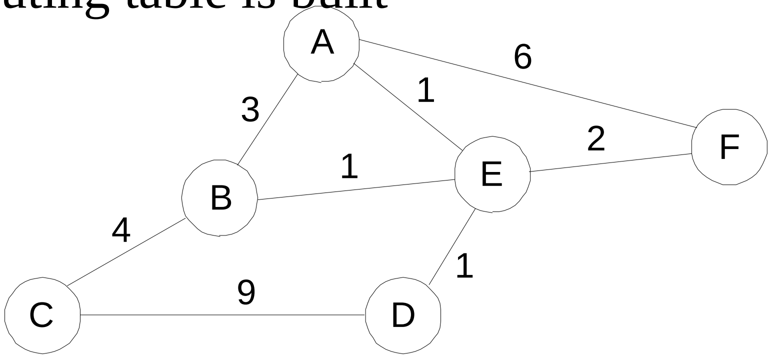

1960
H1
R1
R2
R3
H8
ETH IP (1460)
ETH IP (500)
FDDI IP (1460)
FDDI IP (500)
PPP IP (512)
PPP IP (512)
PPP IP (436)
PPP IP (500)
ETH IP (512)
ETH IP (512)
ETH IP (436)
FDDI IP (500)
Start of header
Ident= x 0 Offset= 0
Rest of header
1460 data bytes
Start of header
Ident= x 1 Offset= 0
Rest of header
512 data bytes
Start of header
Ident= x 1 Offset= 64
Rest of header
512 data bytes
Start of header
Ident= x 0 Offset= 128
Rest of header
436 data bytes

Network Layer - 2

Intra-domain Routing
Inter-domain Routing

Overview

- Forwarding vs Routing
  - forwarding: to select an output port based on destination address and routing table
  - routing: process by which routing table is built
- Network as a Graph

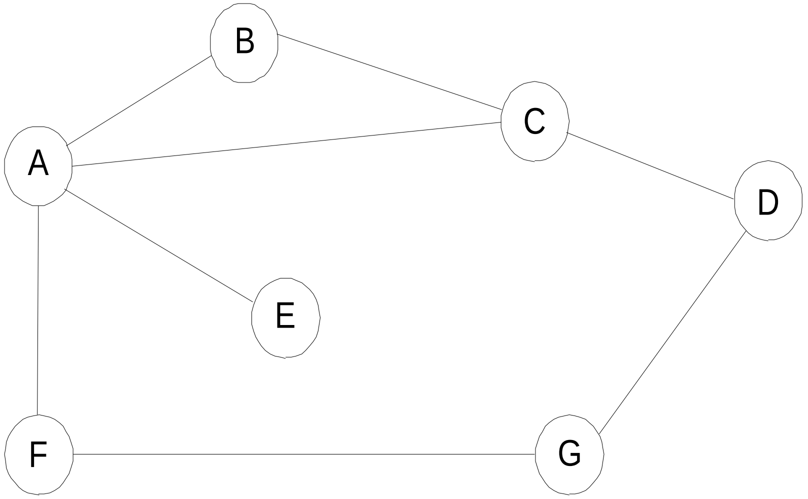


- Problem: Find lowest cost path between two nodes
- Factors
  - static: topology
  - dynamic: load

Distance Vector

- Each node maintains a set of triples
  - **`(Destination, Cost, NextHop)`**
- Directly connected neighbors exchange updates
  - periodically (on the order of several seconds)
  - whenever table changes (called *triggered* update)
- Each update is a list of pairs:
  - (**`Destination, Cost)`**
- Update local table if receive a “better” route
  - smaller cost
  - came from next-hop
- Refresh existing routes; delete if they time out

Example



Routing table for B

Destination	Cost	NextHop
A	1	A
C	1	C
D	2	C
E	2	A
F	2	A
G	3	A

Routing Loops

- Example 1
  - F detects that link to G has failed
  - F sets distance to G to infinity and sends update t o A
  - A sets distance to G to infinity since it uses F to reach G
  - A receives periodic update from C with 2-hop path to G
  - A sets distance to G to 3 and sends update to F
  - F decides it can reach G in 4 hops via A
- Example 2
  - link from A to E fails
  - A advertises distance of infinity to E
  - B and C advertise a distance of 2 to E
  - B decides it can reach E in 3 hops; advertises this to A
  - A decides it can read E in 4 hops; advertises this to C
  - C decides that it can reach E in 5 hops…

Loop-Breaking Heuristics

- Set infinity to 16
- Split horizon
- Split horizon with poison reverse

Link State

- Strategy
  - send to all nodes (not just neighbors) information about directly connected links (not entire routing table)
- Link State Packet (LSP)
  - id of the node that created the LSP
  - cost of link to each directly connected neighbor
  - sequence number (SEQNO)
  - time-to-live (TTL) for this packet

Link State (cont)

- Reliable flooding
  - store most recent LSP from each node
  - forward LSP to all nodes but one that sent it
  - generate new LSP periodically
    - increment SEQNO
  - start SEQNO at 0 when reboot
  - decrement TTL of each stored LSP
    - discard when TTL=0

Route Calculation

- Dijkstra's shortest path algorithm
- Let
  - *N* denotes set of nodes in the graph
  - *l* (*i*, *j*) denotes non-negative cost (weight) for edge (*i*, *j*)
  - *s* denotes this node
  - *M* denotes the set of nodes incorporated so far
  - *C*(*n*) denotes cost of the path from *s* to node *n*

```
M = {s}
for each n in N - {s}
   C(n) = l(s, n)
while (N != M)
   M = M union {w} such that C(w) is the minimum for
       all w in (N - M)
   for each n in (N - M)
      C(n) = MIN(C(n), C (w) + l(w, n ))
```

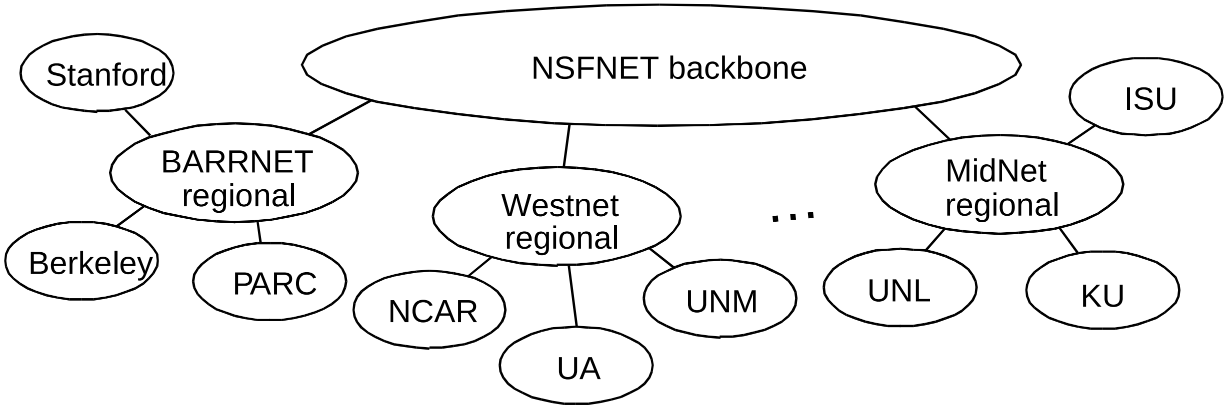
Metrics

- Original ARPANET metric
  - measures number of packets queued on each link
  - took neither latency or bandwidth into consideration
- New ARPANET metric
  - stamp each incoming packet with its arrival time (**`AT`**)
  - record departure time (**`DT`**)
  - when link-level ACK arrives, compute

    `Delay = (DT - AT) + Transmit + Latency`
  - if timeout, reset **`DT`** to departure time for retransmission
  - link cost = average delay over some time period
- Fine Tuning
  - compressed dynamic range
  - replaced **`Delay`** with link utilization

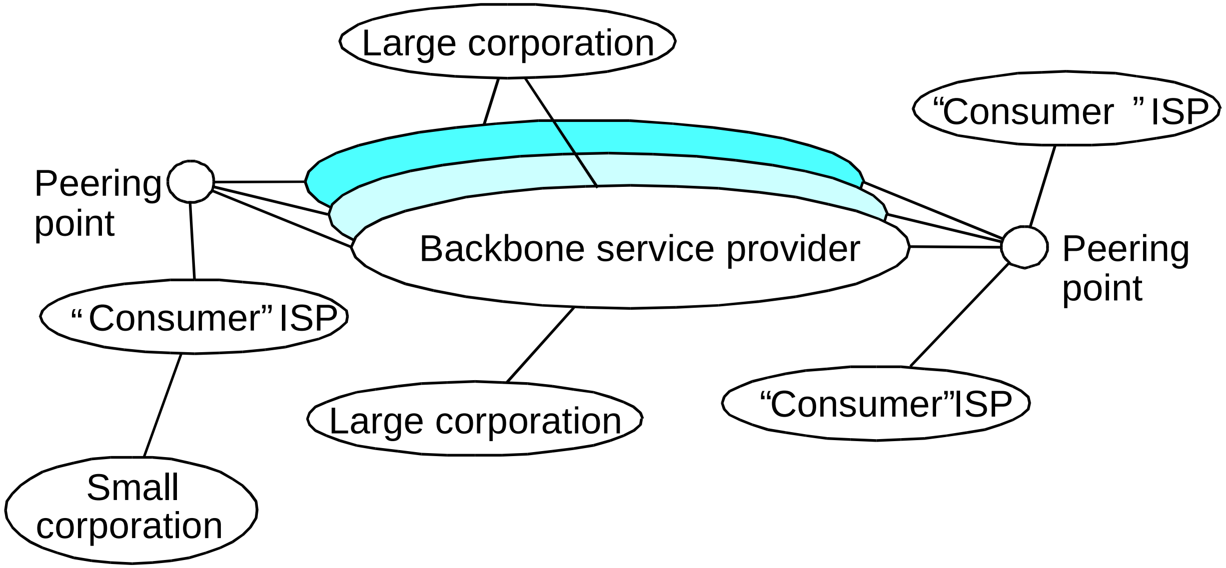
Internet Structure

Recent Past



Internet Structure

Today



Route Propagation

- Know a smarter router
  - hosts know local router
  - local routers know site routers
  - site routers know core router
  - core routers know everything
- Autonomous System (AS)
  - corresponds to an administrative domain
  - examples: University, company, backbone network
  - assign each AS a 16-bit number
- Two-level route propagation hierarchy
  - interior gateway protocol (each AS selects its own)
  - exterior gateway protocol (Internet-wide standard)

Popular Interior Gateway Protocols

- RIP: Route Information Protocol
  - developed for XNS
  - distributed with Unix
  - distance-vector algorithm
  - based on hop-count
- OSPF: Open Shortest Path First
  - recent Internet standard
  - uses link-state algorithm
  - supports load balancing
  - supports authentication

Lecture 6 and 7 (Feb 5 and 10, 2004)

Outline

Exterior Gateway Protocol

- Border Gateway Protocol – BGPv4

EGP: Exterior Gateway Protocol

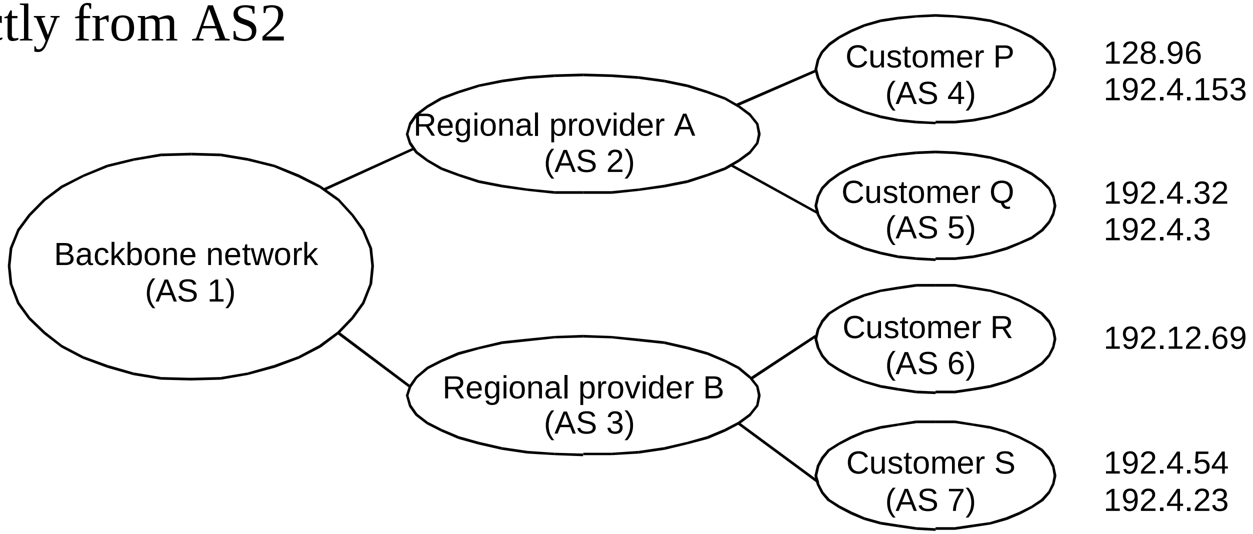
- Overview
  - designed for tree-structured Internet
  - concerned with *reachability*, not optimal routes
- Protocol messages
  - neighbor acquisition: one router requests that another be its peer; peers exchange reachability information
  - neighbor reachability: one router periodically tests if the another is still reachable; exchange HELLO/ACK messages; uses a k-out-of-n rule
  - routing updates: peers periodically exchange their routing tables (distance-vector)

BGP-4: Border Gateway Protocol

- AS Types
  - stub AS: has a single connection to one other AS
    - carries local traffic only
  - multihomed AS: has connections to more than one AS
    - refuses to carry transit traffic
  - transit AS: has connections to more than one AS
    - carries both transit and local traffic
- Each AS has:
  - one or more border routers
  - one BGP *speaker* that advertises:
    - local networks
    - other reachable networks (transit AS only)
    - provides *path* information

BGP Example

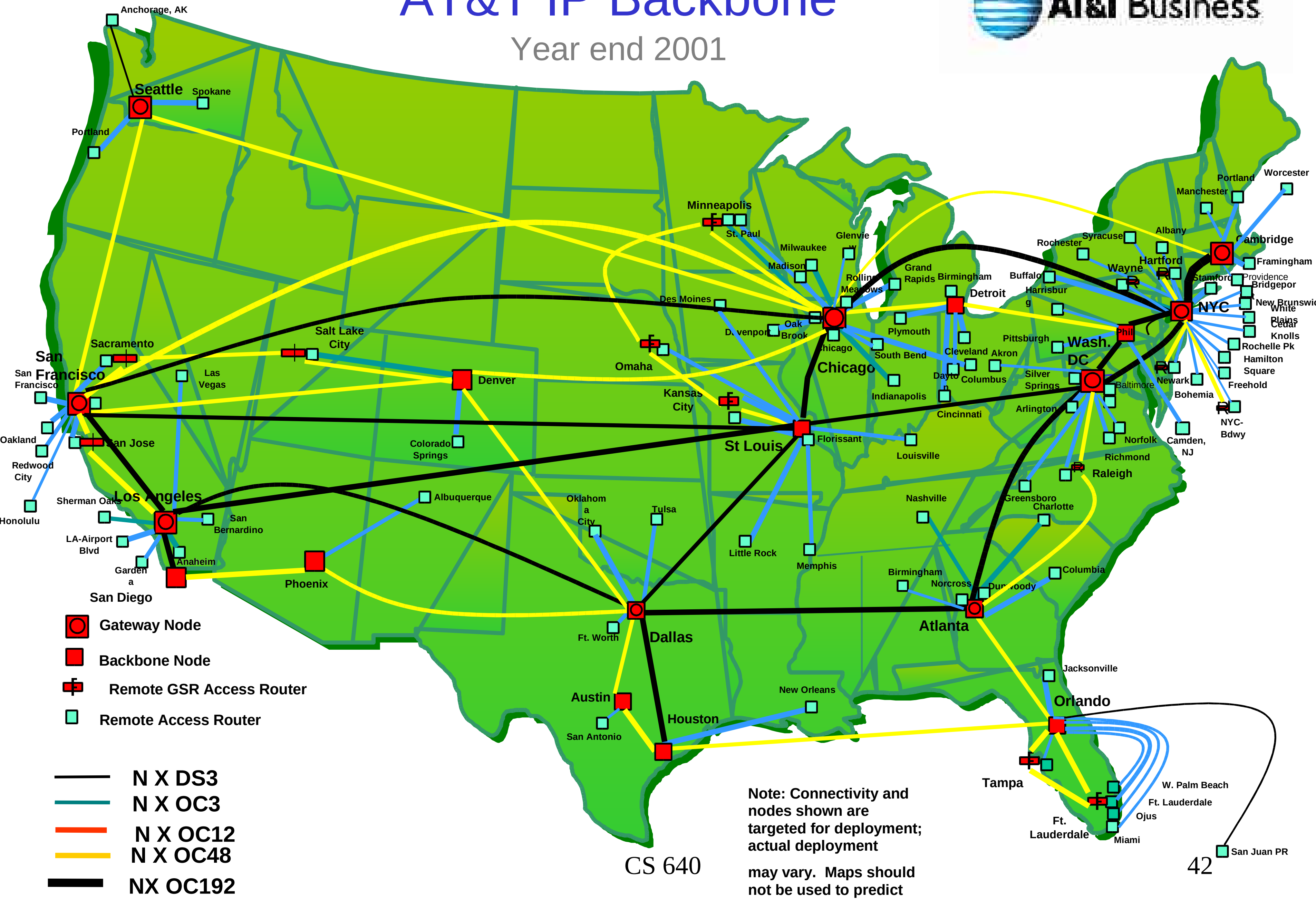
- Speaker for AS2 advertises reachability to P and Q
  - network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS2



- Speaker for backbone advertises
  - networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path (AS1, AS2).
- Speaker can cancel previously advertised paths

AT&T IP Backbone

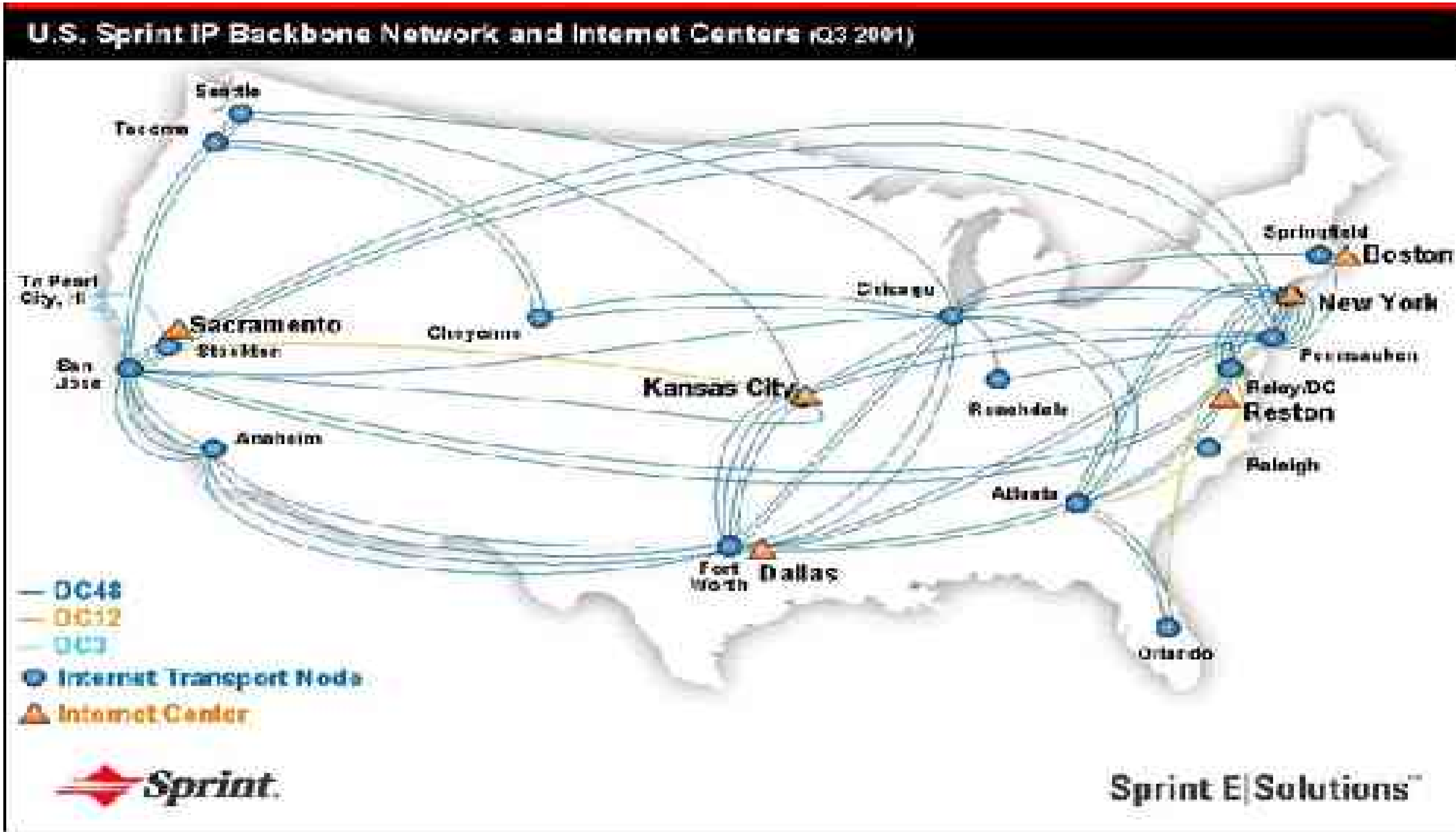
Year end 2001



Note: Connectivity and nodes shown are targeted for deployment; actual deployment may vary. Maps should not be used to predict service availability.

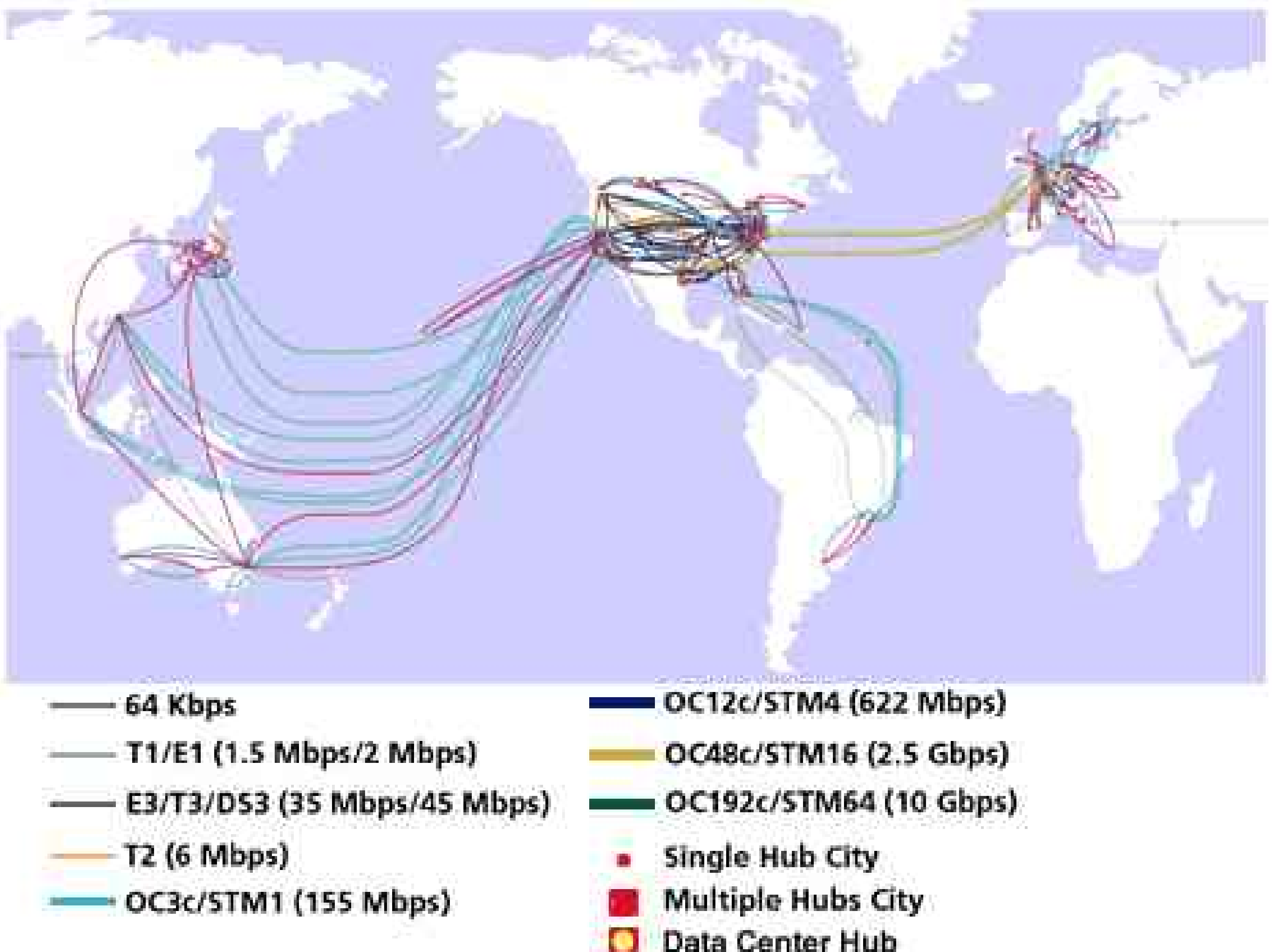
Rev. 6-4-01

Sprint, USA

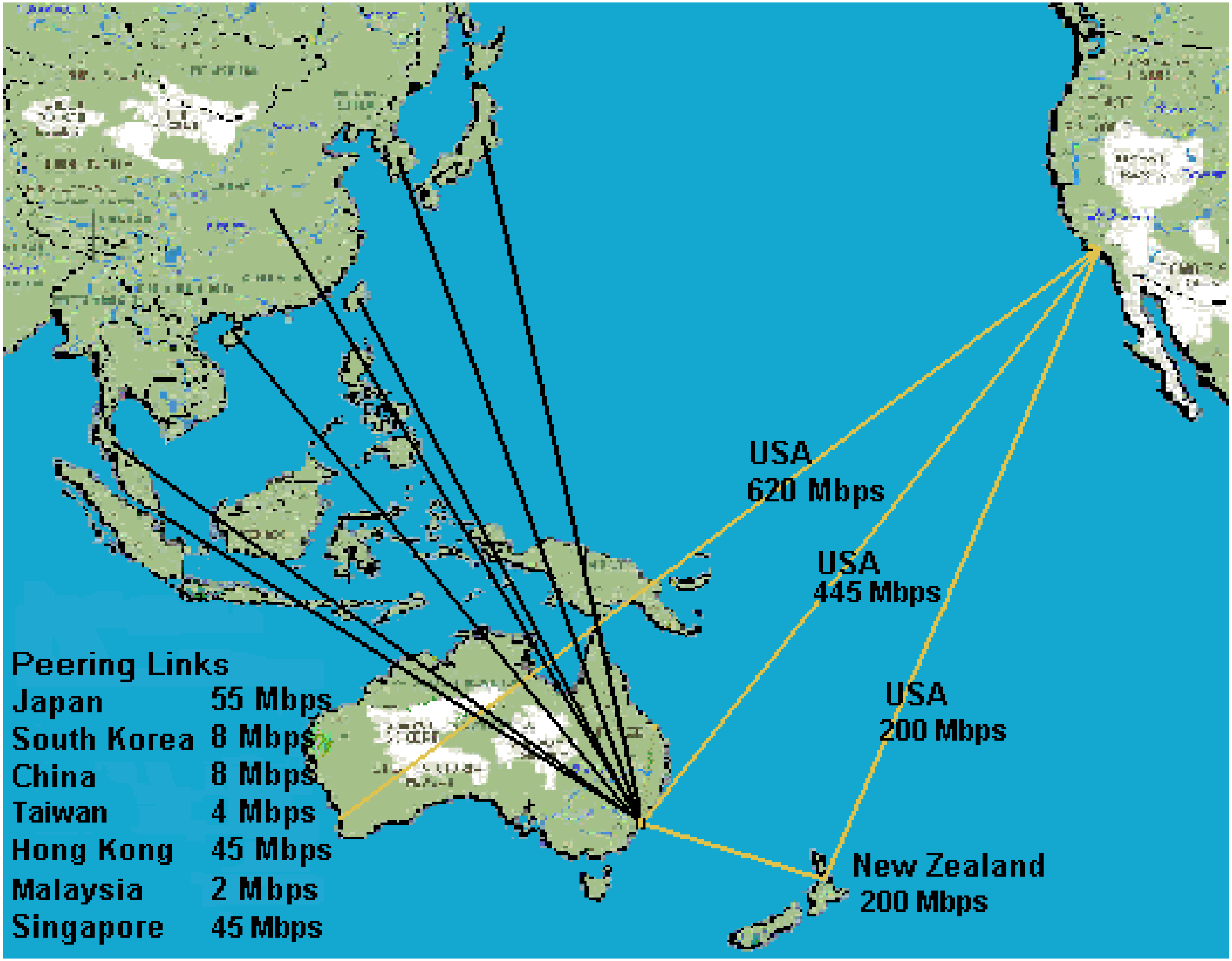


U.S. Sprint IP Backbone Network and Internet Centers (Q3 2001)
Seattle
Tacoma
Sacramento
Stockton
San Jose
Anaheim
Cheyenne
Kansas City
Chicago
Roachdale
Springfield
Boston
New York
Pennsauken
Relay/DC
Reston
Raleigh
Atlanta
Fort Worth
Dallas
Orlando
OC48
OC12
OC3
Internet Transport Node
Internet Center
Sprint
Sprint E|Solutions

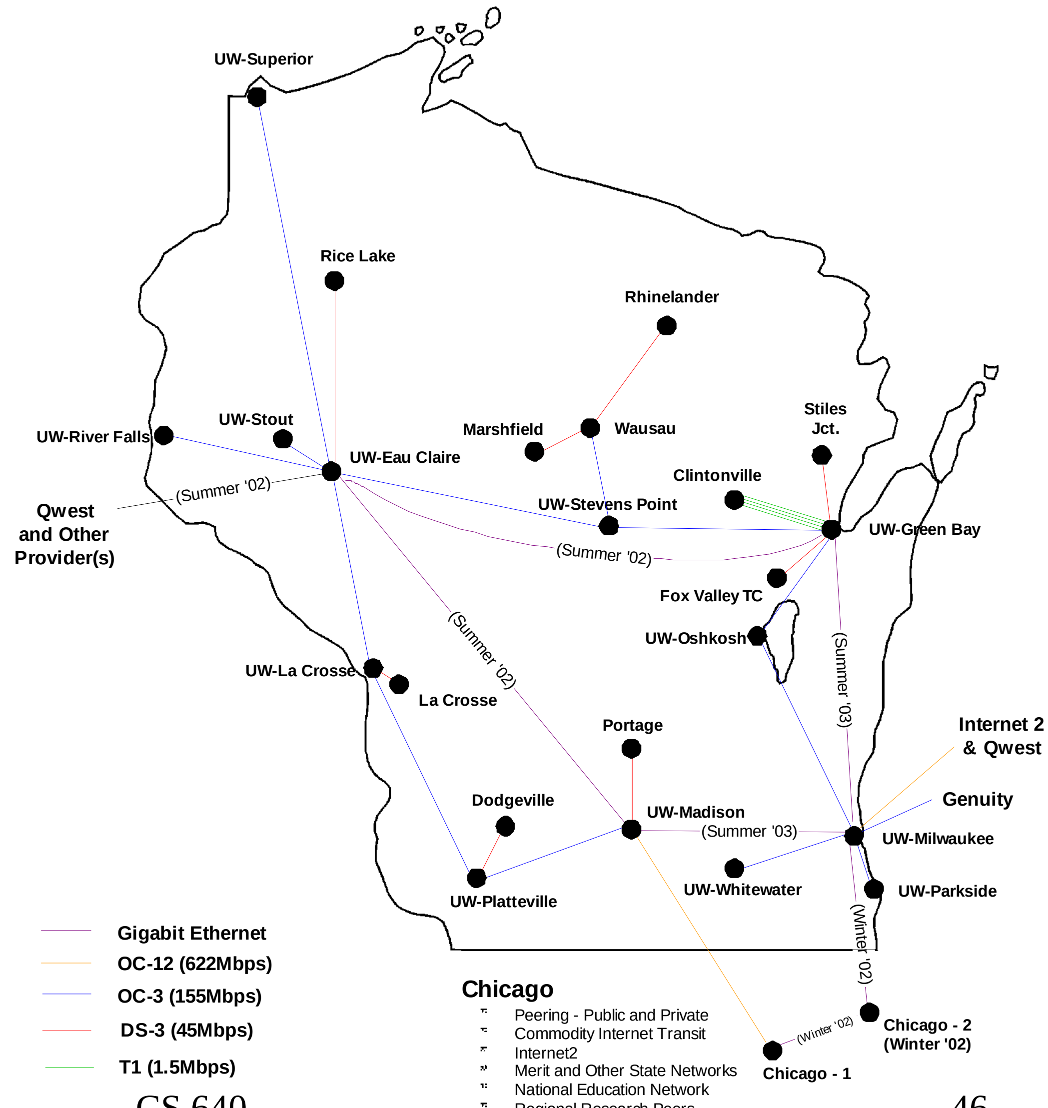
WorldCom (UUNet)



Telstra international

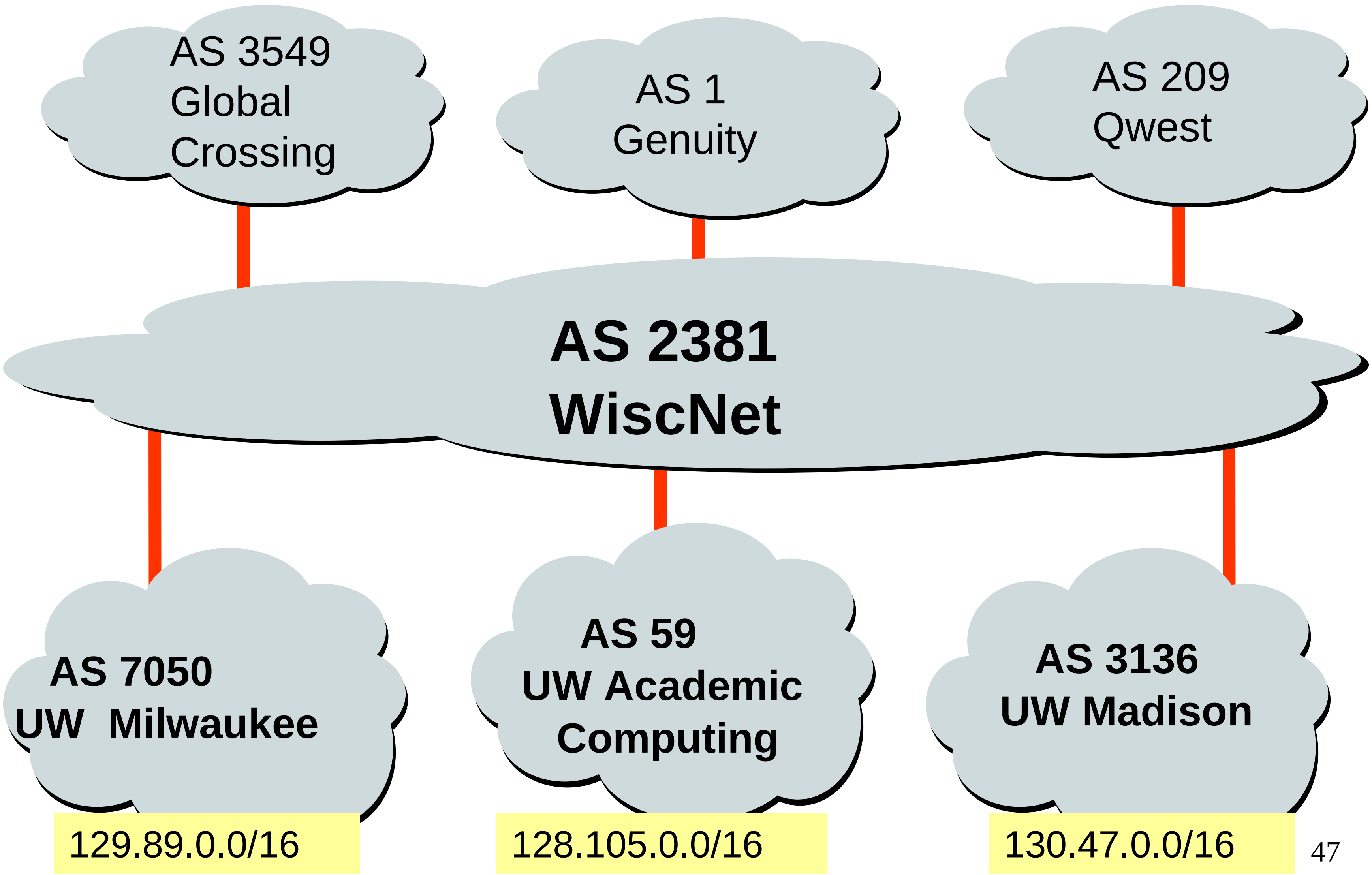


wiscnet.net

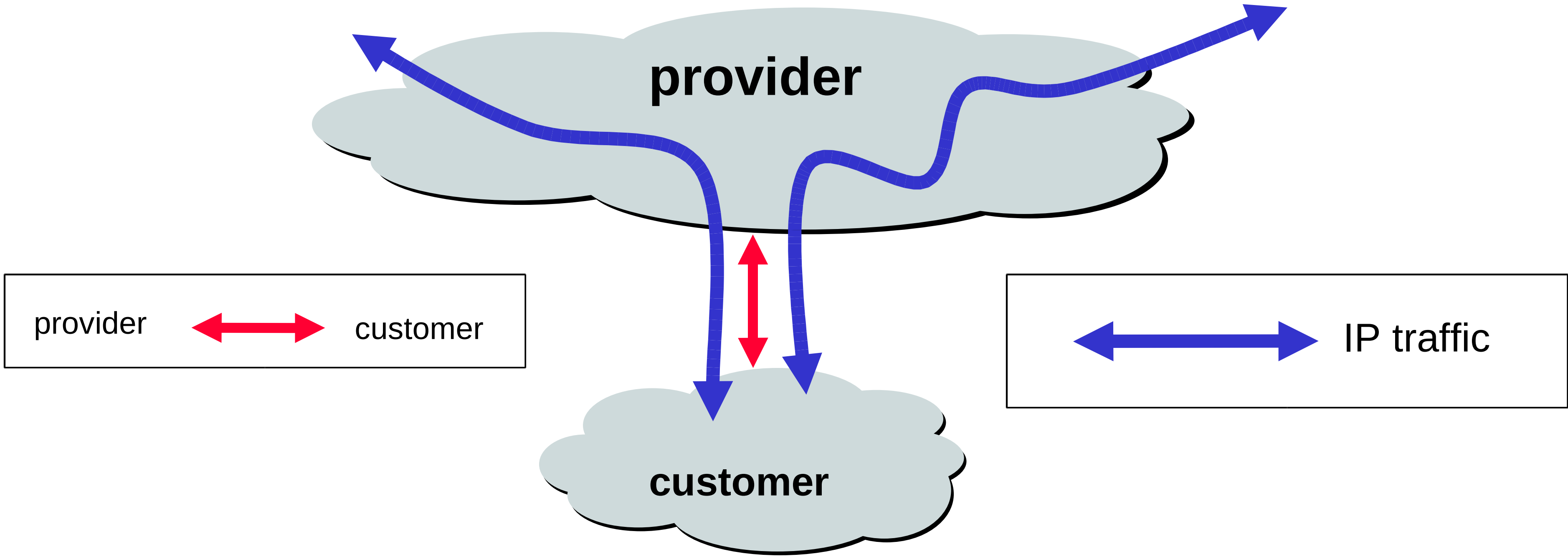


UW-Superior
Rice Lake
Rhinelander
UW-River Falls
UW-Stout
UW-Eau Claire
Marshfield
Wausau
Stiles
Jct.
Clintonville
UW-Stevens Point
UW-Green Bay
(Summer '02)
Qwest
and Other
Provider(s)
(Summer '02)
Fox Valley TC
UW-Oshkosh
(Summer '02)
(Summer '03)
UW-La Crosse
La Crosse
Portage
Internet 2
& Qwest
Genuity
Dodgeville
UW-Madison
(Summer '03)
UW-Milwaukee
UW-Whitewater
UW-Parkside
UW-Platteville
(Winter '02)
Gigabit Ethernet
OC-12 (622Mbps)
OC-3 (155Mbps)
DS-3 (45Mbps)
T1 (1.5Mbps)
Chicago
Peering - Public and Private
Commodity Internet Transit
Internet2
Merit and Other State Networks
National Education Network
Regional Research Peers
(Winter '02)
Chicago - 1
Chicago - 2
(Winter '02)

Partial View of cs.wisc.edu Neighborhood

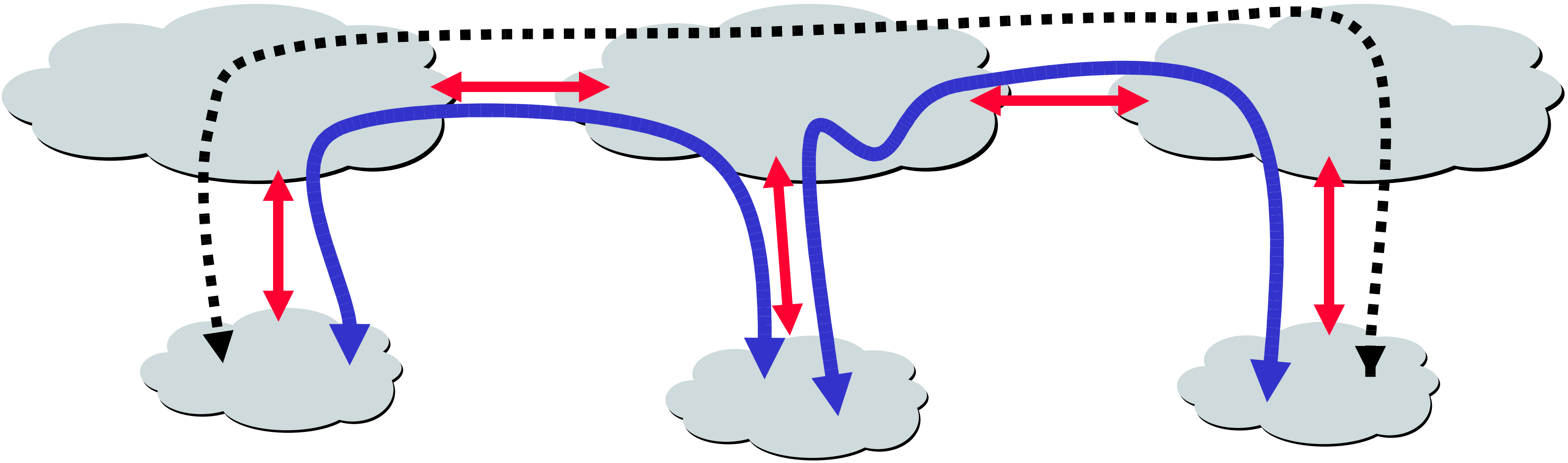


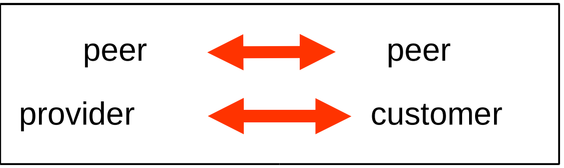
Customers and Providers

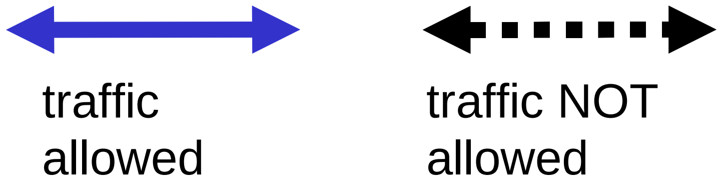


Customer pays provider for access to the Internet

The “Peering” Relationship





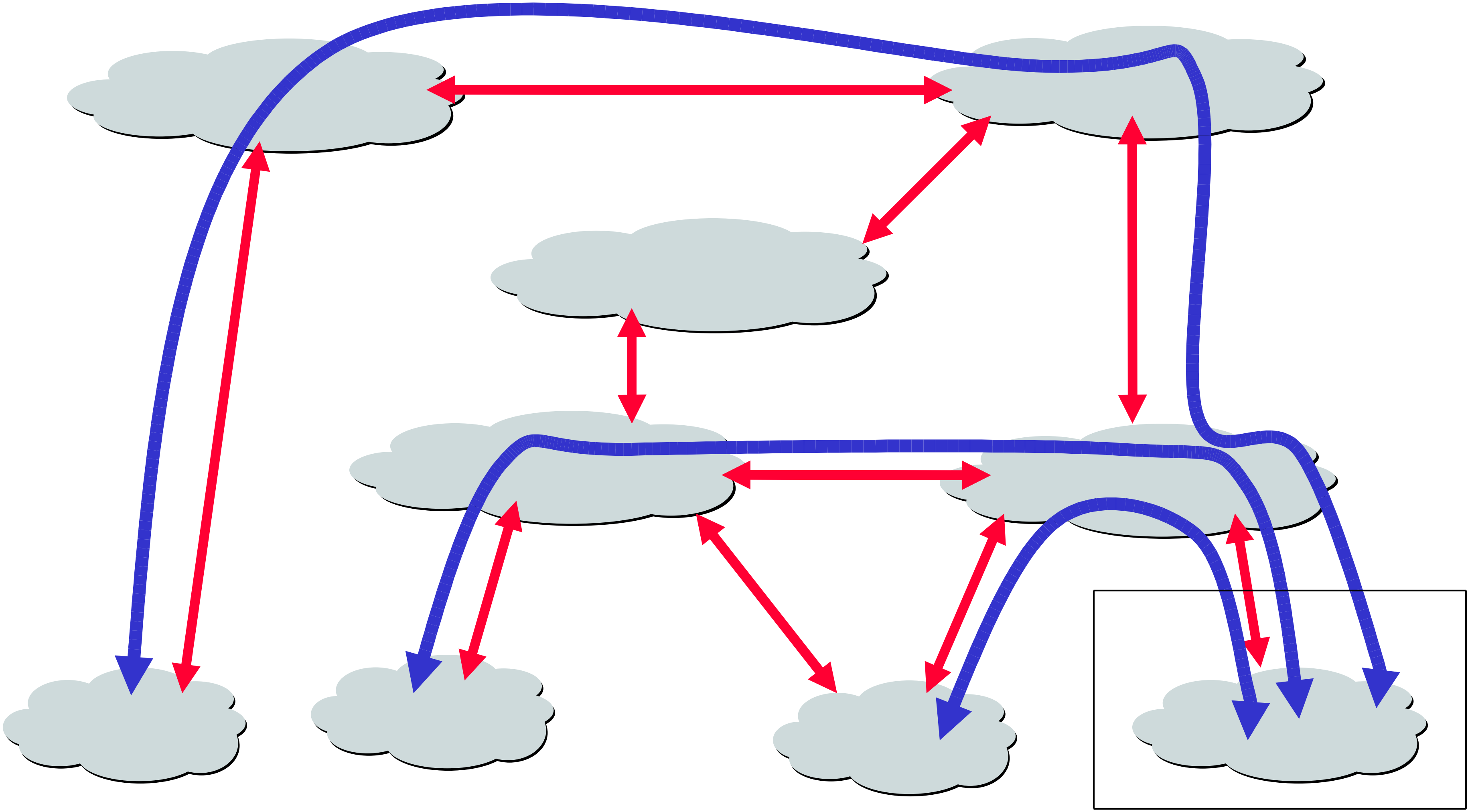


Peers provide transit between
their respective customers

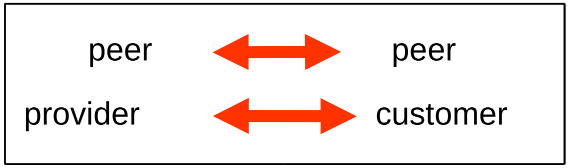
Peers do not provide transit
between peers

Peers (often) do not exchange $$$

Examples of Peering



Peering also allows connectivity between the customers of “Tier 1” providers.



To Peer or Not to Peer?

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.

Autonomous Systems (ASes)

An autonomous system is an autonomous routing domain
that has been assigned an Autonomous System Number (ASN).

… the administration of an AS appears to other ASes to have a single coherent interior routing plan and presents a consistent picture of what networks are reachable through it.

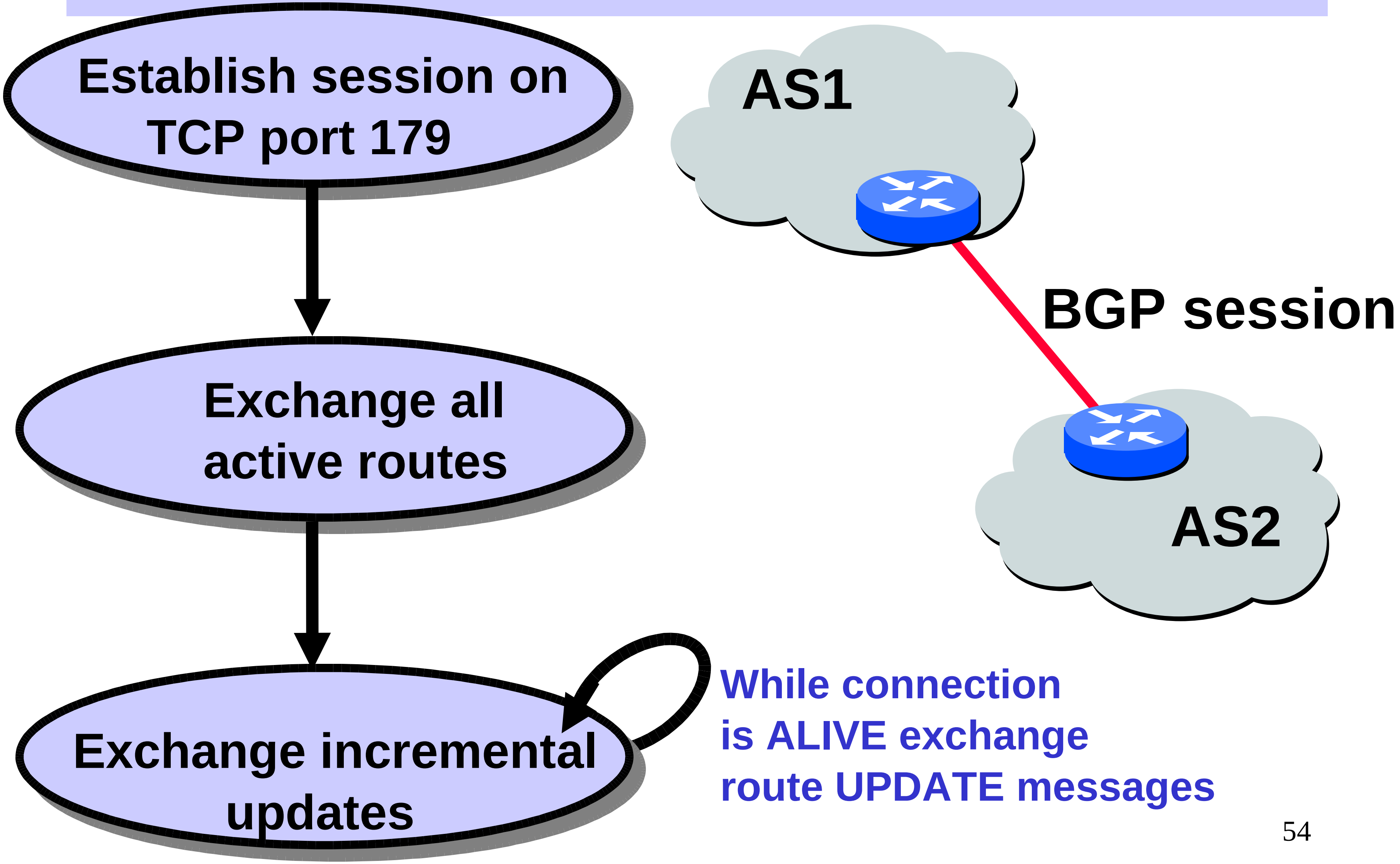
RFC 1930: Guidelines for creation, selection, and registration of an Autonomous System

BGP-4

- **BGP** = **<u>B</u>**order **<u>G</u>**ateway **<u>P</u>**rotocol
- Is a **<u>Policy-Based</u>** routing protocol
- Is the **<u>de facto EGP</u>** of today's global Internet
- Relatively simple protocol, but configuration is complex and the entire world can see, and be impacted by, your mistakes.

- 1989 : BGP-1 [RFC 1105]
  - Replacement for EGP (1984, RFC 904)
- 1990 : BGP-2 [RFC 1163]
- 1991 : BGP-3 [RFC 1267]
- 1995 : BGP-4 [RFC 1771]
  - Support for Classless Interdomain Routing (CIDR)

BGP Operations (Simplified)

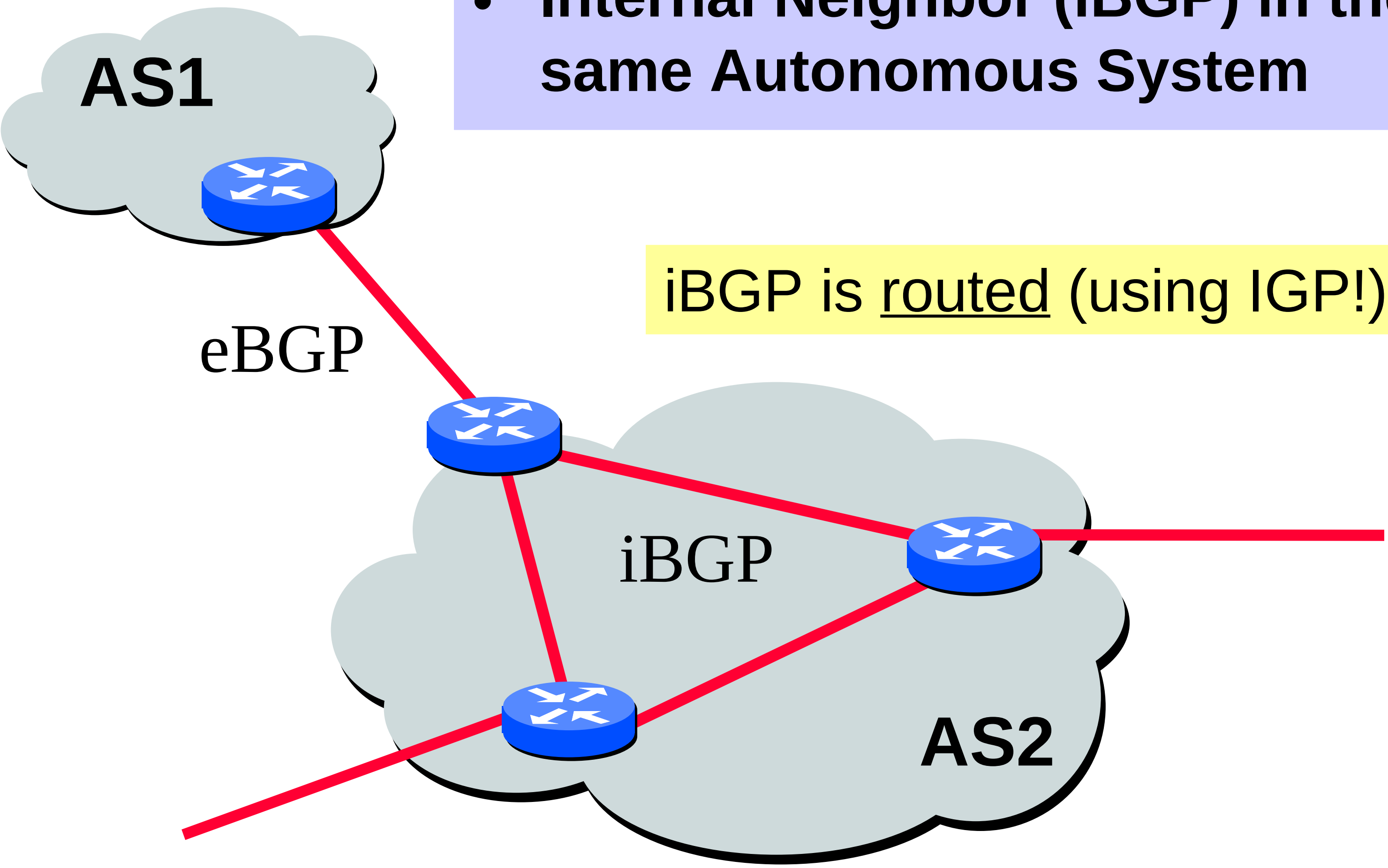


While connection is ALIVE exchange route UPDATE messages

Two Types of BGP Neighbor Relationships

- **External Neighbor (eBGP) in a different Autonomous Systems**
- **Internal Neighbor (iBGP) in the same Autonomous System**

iBGP is <u>routed</u> (using IGP!)



Four Types of BGP Messages

- **Open** : Establish a peering session.
- **Keep Alive** : Handshake at regular intervals.
- **Notification** : Shuts down a peering session.
- **Update** : Announcing new routes or withdrawing previously announced routes.

announcement

=

prefix + attributes values

BGP Attributes

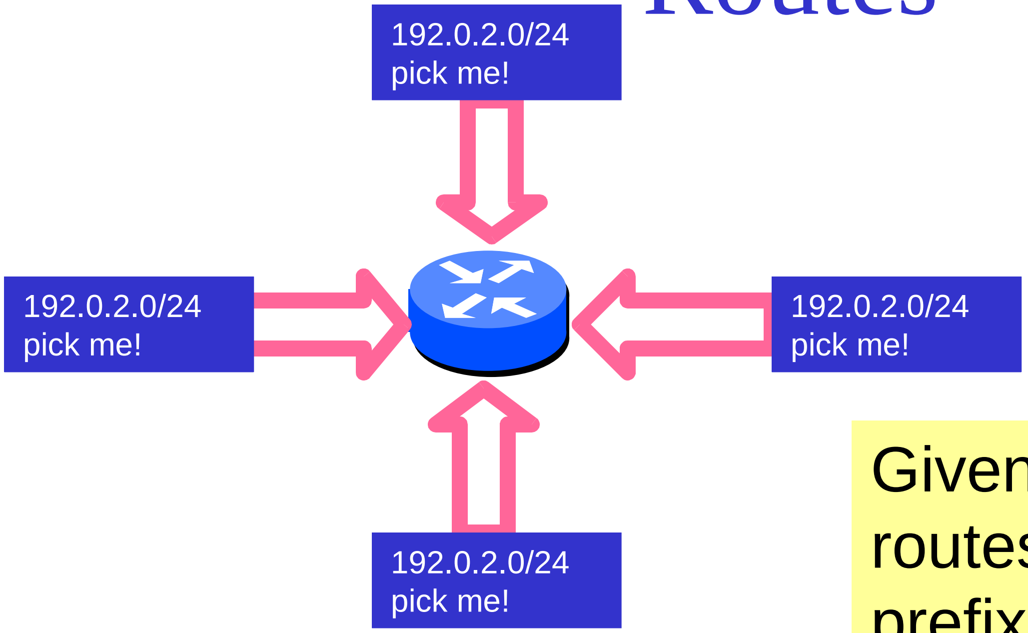
Value	Code	Reference
1	ORIGIN	[RFC1771]
2	AS_PATH	[RFC1771]
3	NEXT_HOP	[RFC1771]
4	MULTI_EXIT_DISC	[RFC1771]
5	LOCAL_PREF	[RFC1771]
6	ATOMIC_AGGREGATE	[RFC1771]
7	AGGREGATOR	[RFC1771]
8	COMMUNITY	[RFC1997]
9	ORIGINATOR_ID	[RFC2796]
10	CLUSTER_LIST	[RFC2796]
11	DPA	[Chen]
12	ADVERTISER	[RFC1863]
13	RCID_PATH / CLUSTER_ID	[RFC1863]
14	MP_REACH_NLRI	[RFC2283]
15	MP_UNREACH_NLRI	[RFC2283]
16	EXTENDED COMMUNITIES	[Rosen]
...		
255	reserved for development	

Most important attributes

From IANA: http://www.iana.org/assignments/bgp-parameters

Not all attributes need to be present in every announcement

Attributes are Used to Select Best Routes



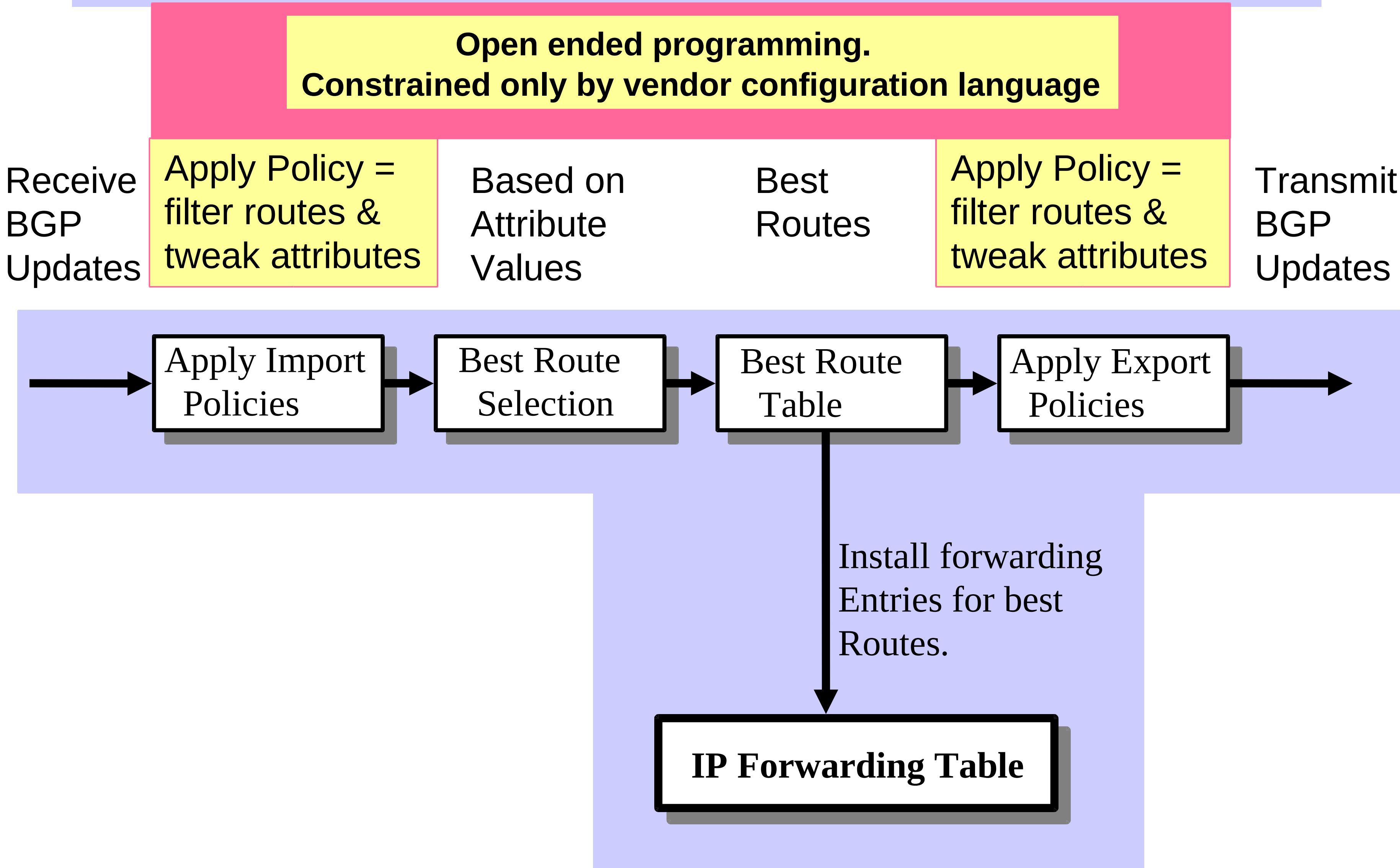
Given multiple routes to the same prefix, a BGP speaker must pick at most <u>one</u> best route

(Note: it could reject them all!)

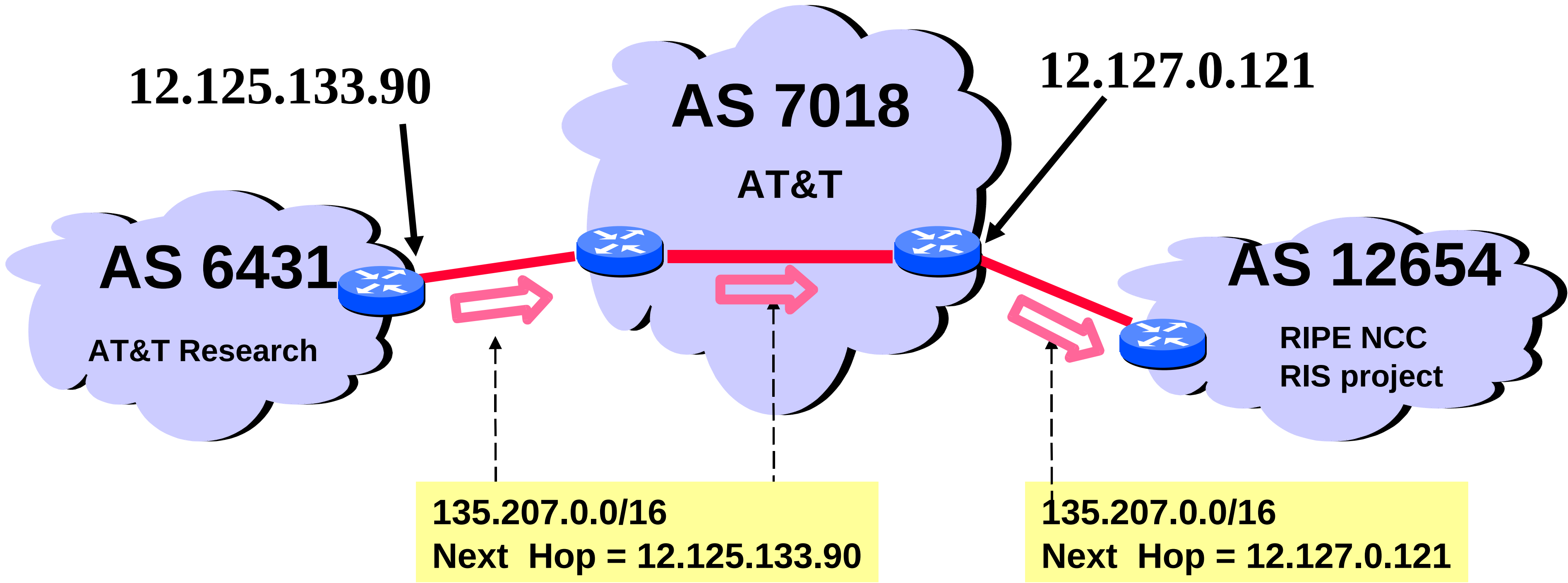
Route Selection Summary

Highest Local Preference	**Enforce relationships**
Shortest ASPATH **Lowest MED** **i-BGP < e-BGP** **Lowest IGP cost to BGP egress**	**traffic engineering**
Lowest router ID	**Throw up hands and break ties**

BGP Route Processing

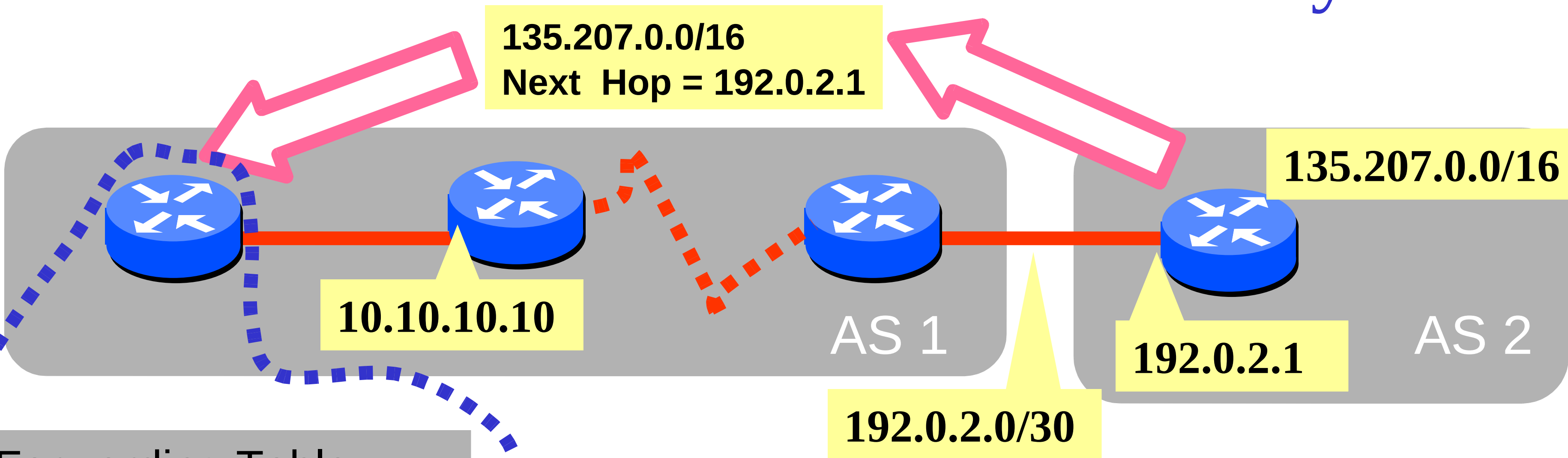


BGP Next Hop Attribute



Every time a route announcement crosses an AS boundary, the Next Hop attribute is changed to the IP address of the border router that announced the route.

Join EGP with IGP For Connectivity



Forwarding Table

destination	next hop
192.0.2.0/30	10.10.10.10

+

EGP

destination	next hop
135.207.0.0/16	192.0.2.1

Forwarding Table

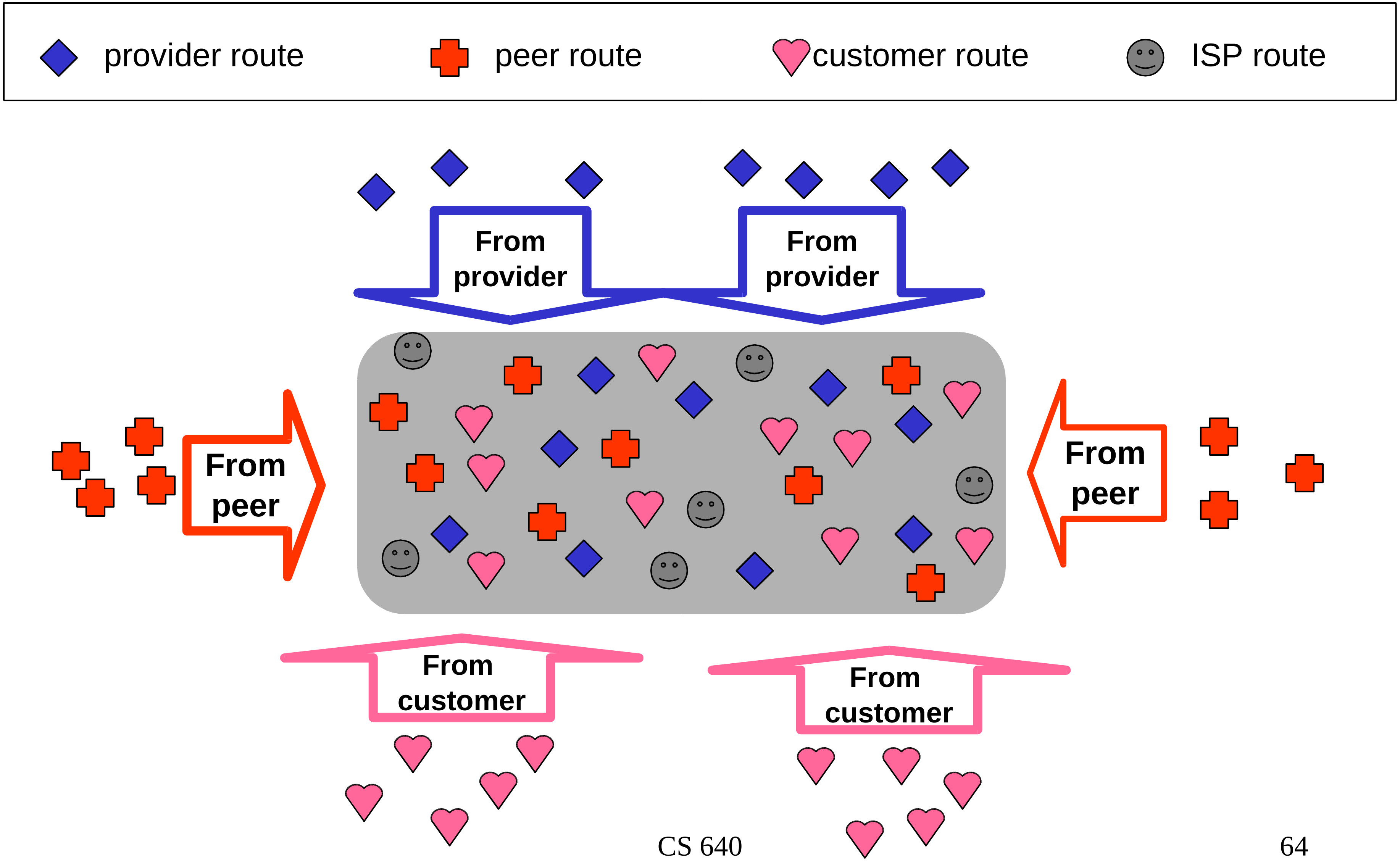
destination	next hop
135.207.0.0/16	10.10.10.10
192.0.2.0/30	10.10.10.10

Implementing Customer/Provider and Peer/Peer relationships

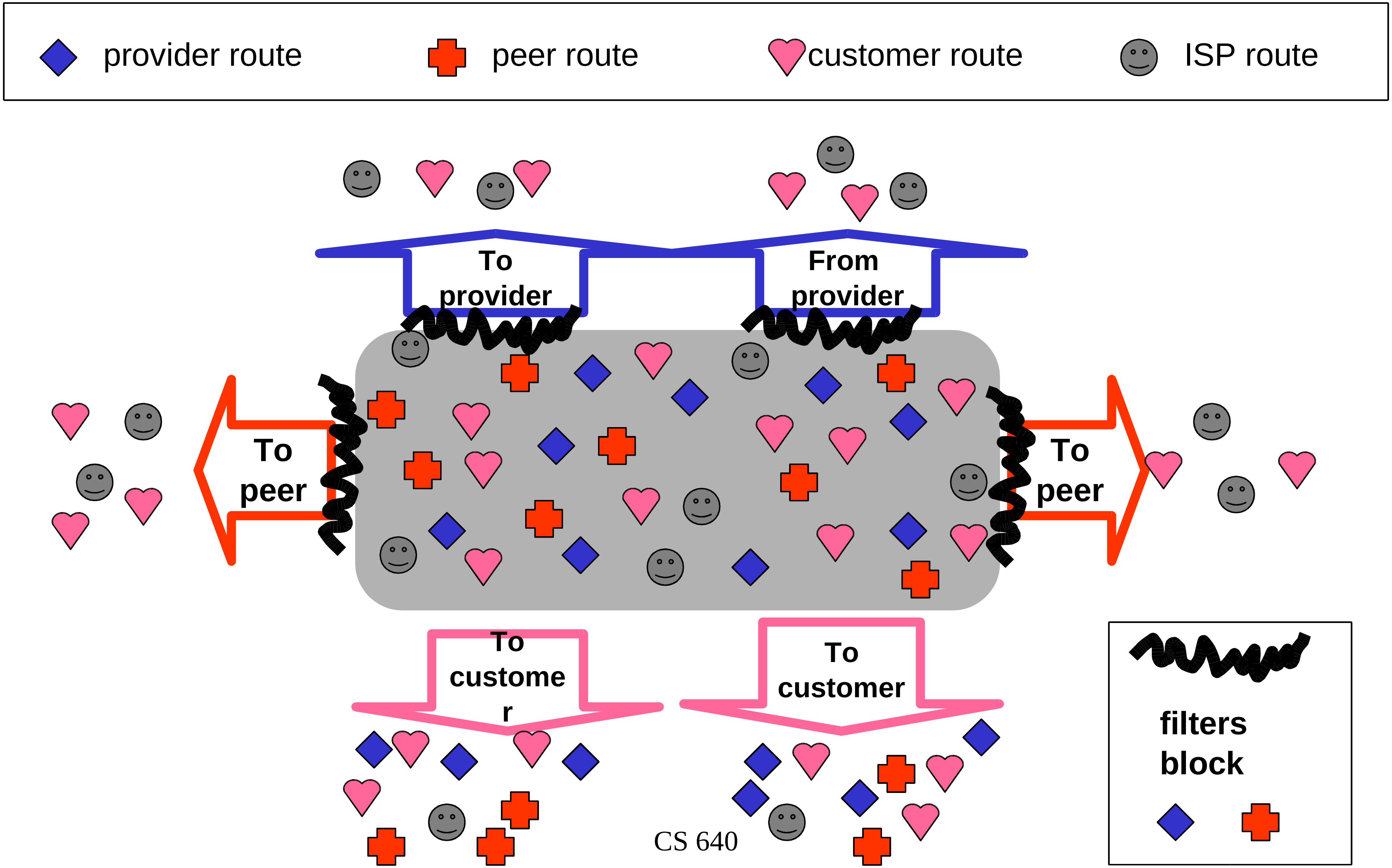
Two parts:

- Enforce transit relationships
  - Outbound route filtering
- Enforce order of route preference
  - provider < peer < customer

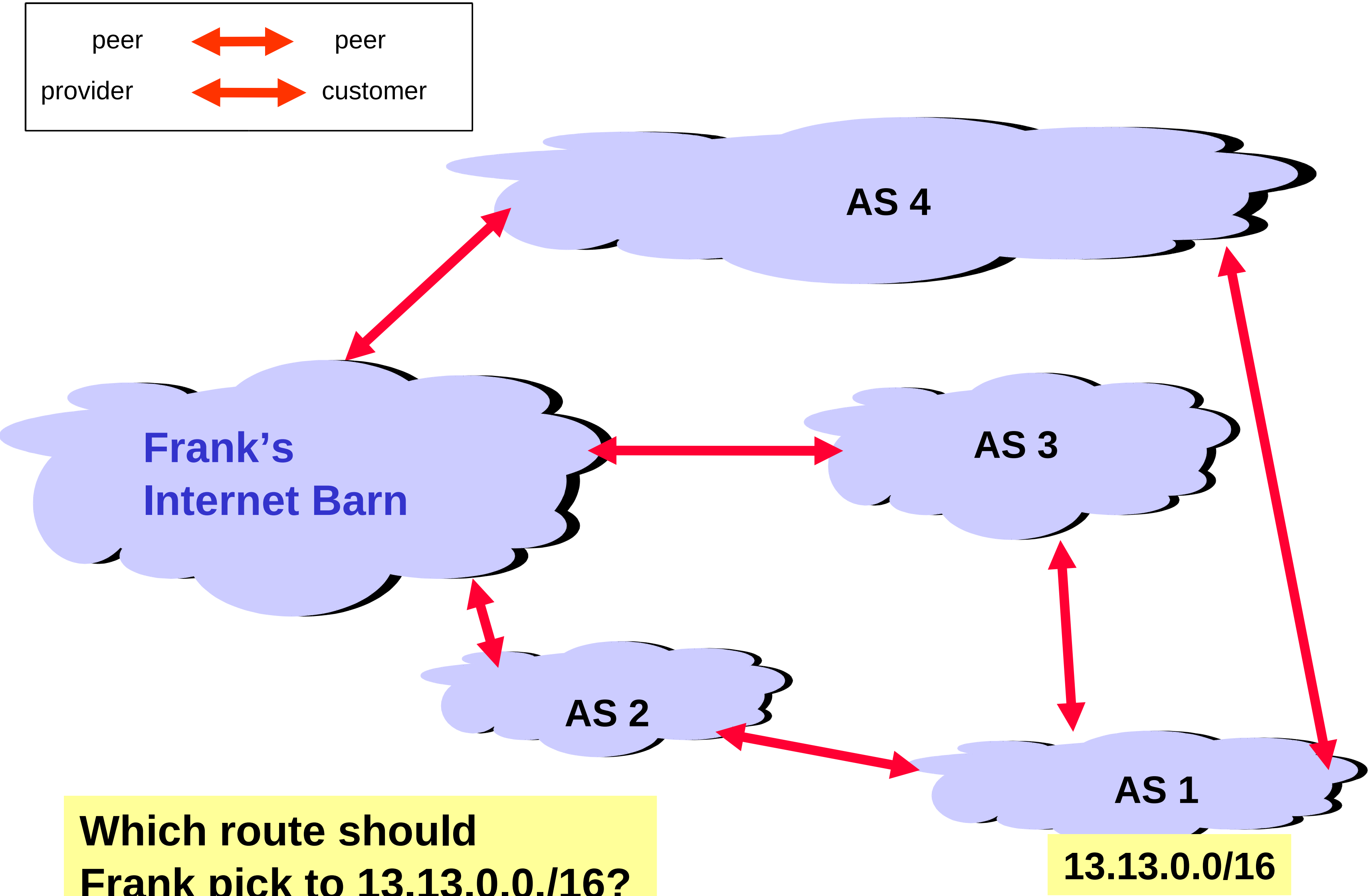
Import Routes



Export Routes

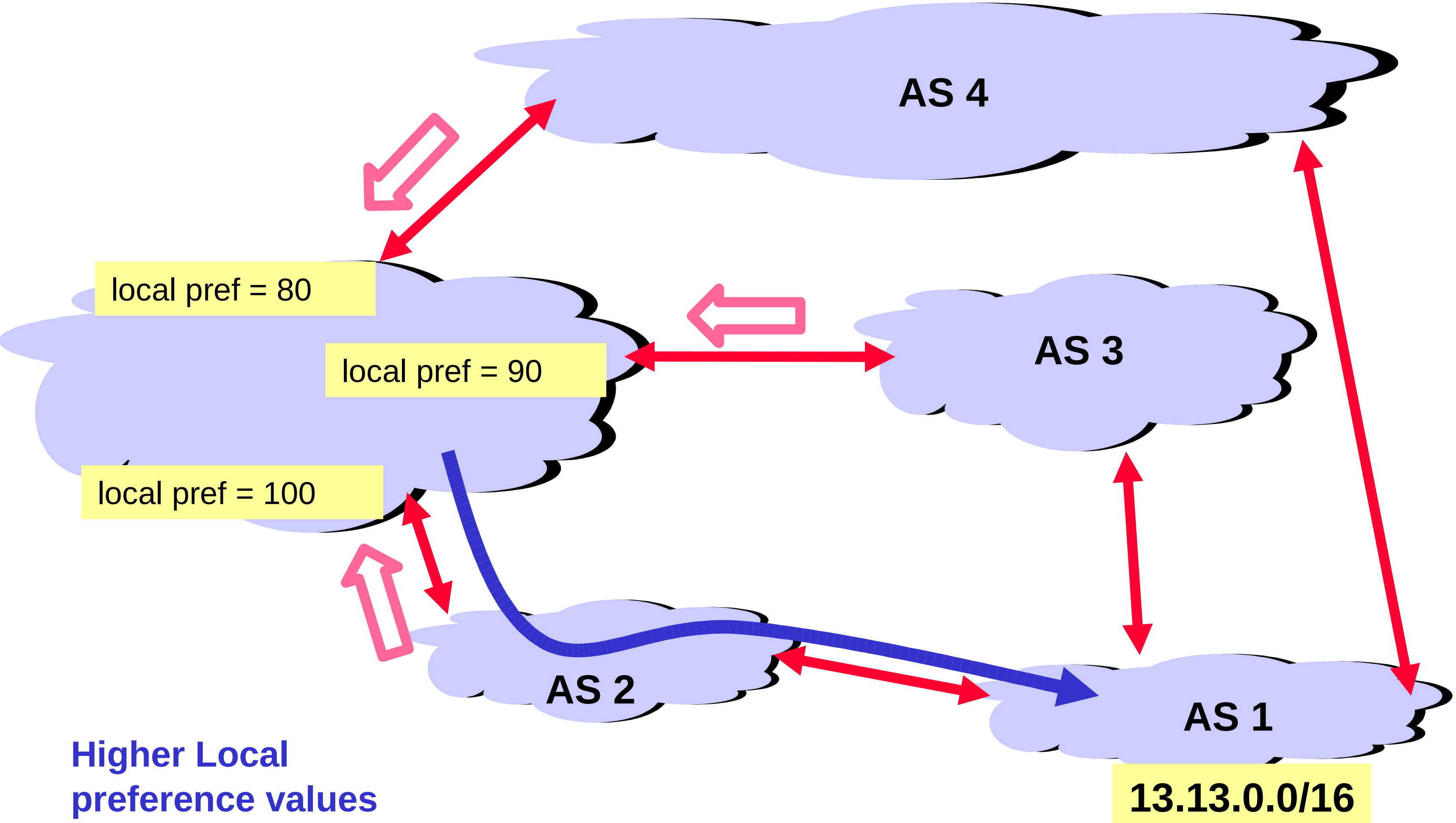


So Many Choices



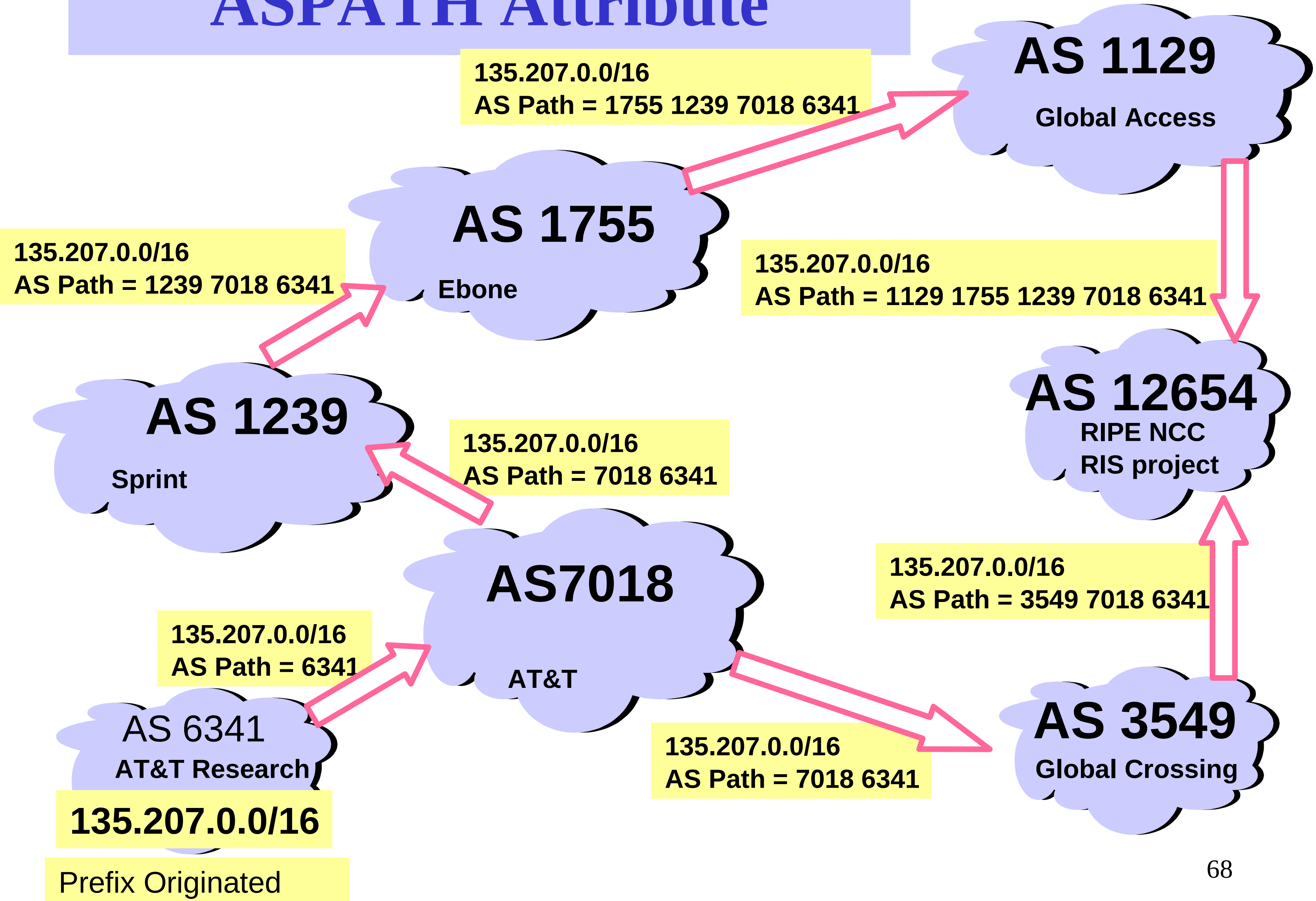
Which route should Frank pick to 13.13.0.0./16?

LOCAL PREFERENCE

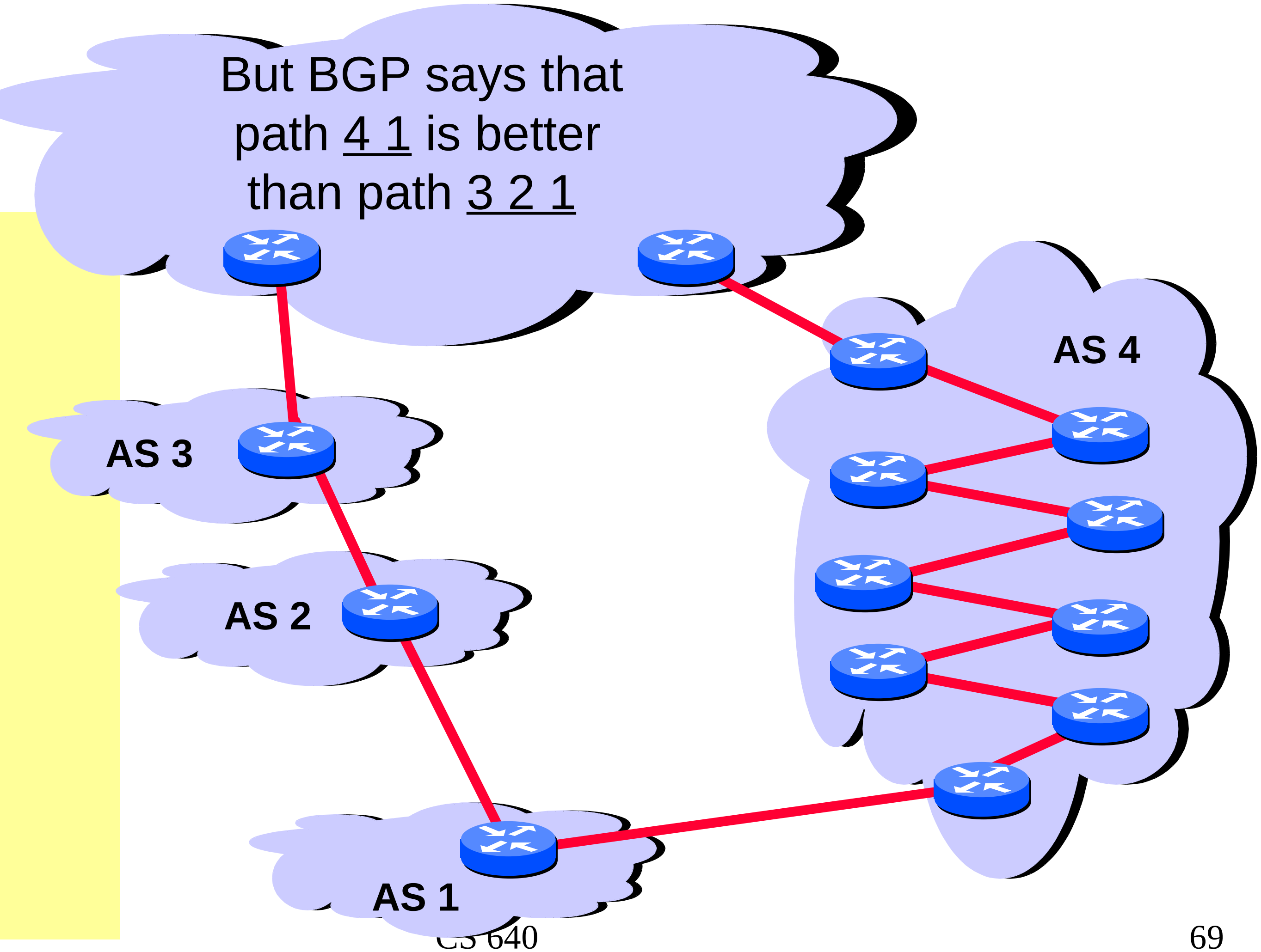


Higher Local preference values are more preferred

ASPATH Attribute



Shorter Doesn’t Always Mean Shorter

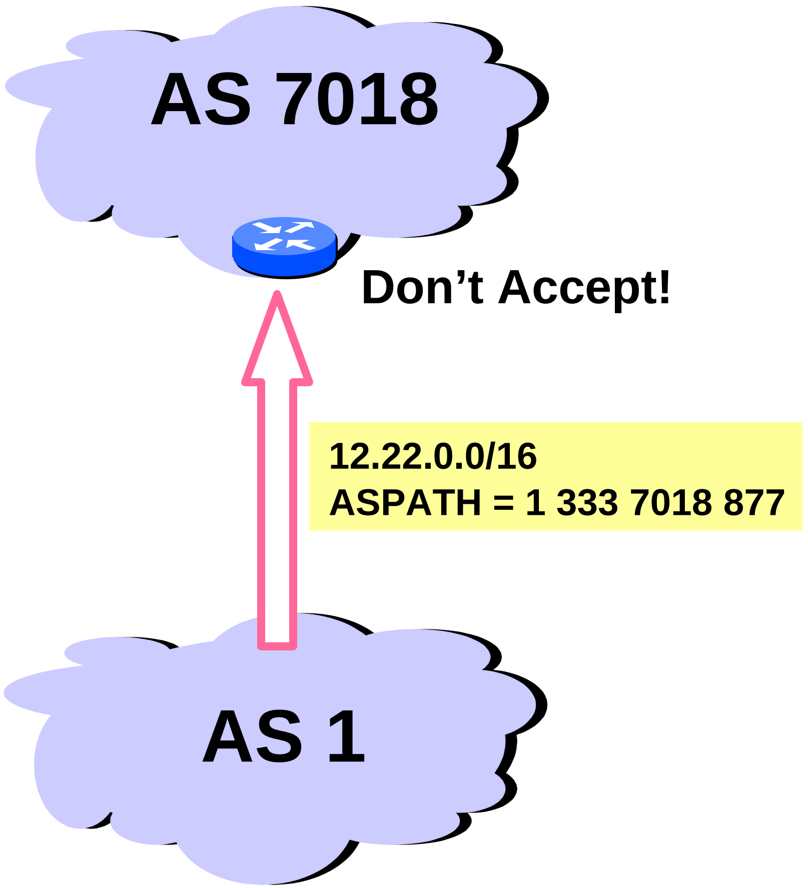


In fairness: could you do this “right” and still scale?

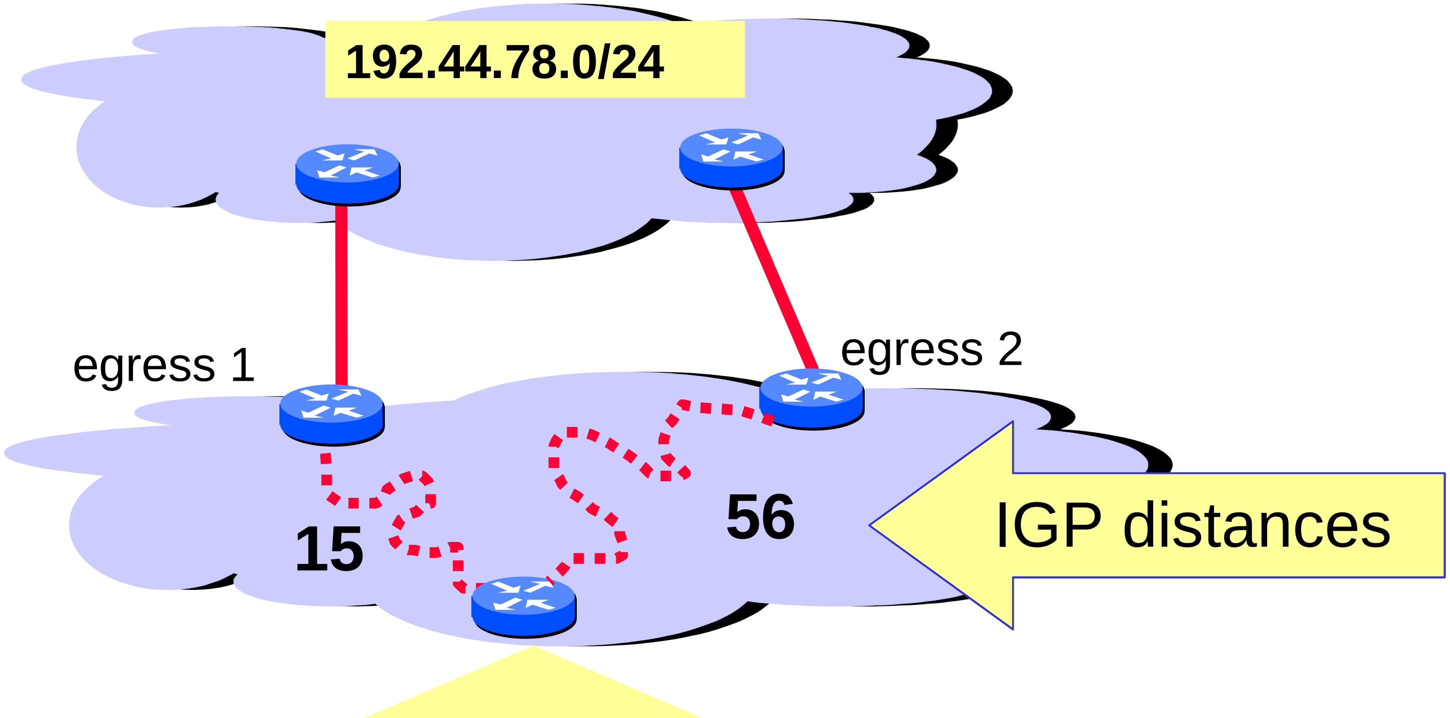
Exporting internal state would dramatically increase global instability and amount of routing state

Interdomain Loop Prevention

BGP at AS YYY will never accept a route with ASPATH containing YYY.



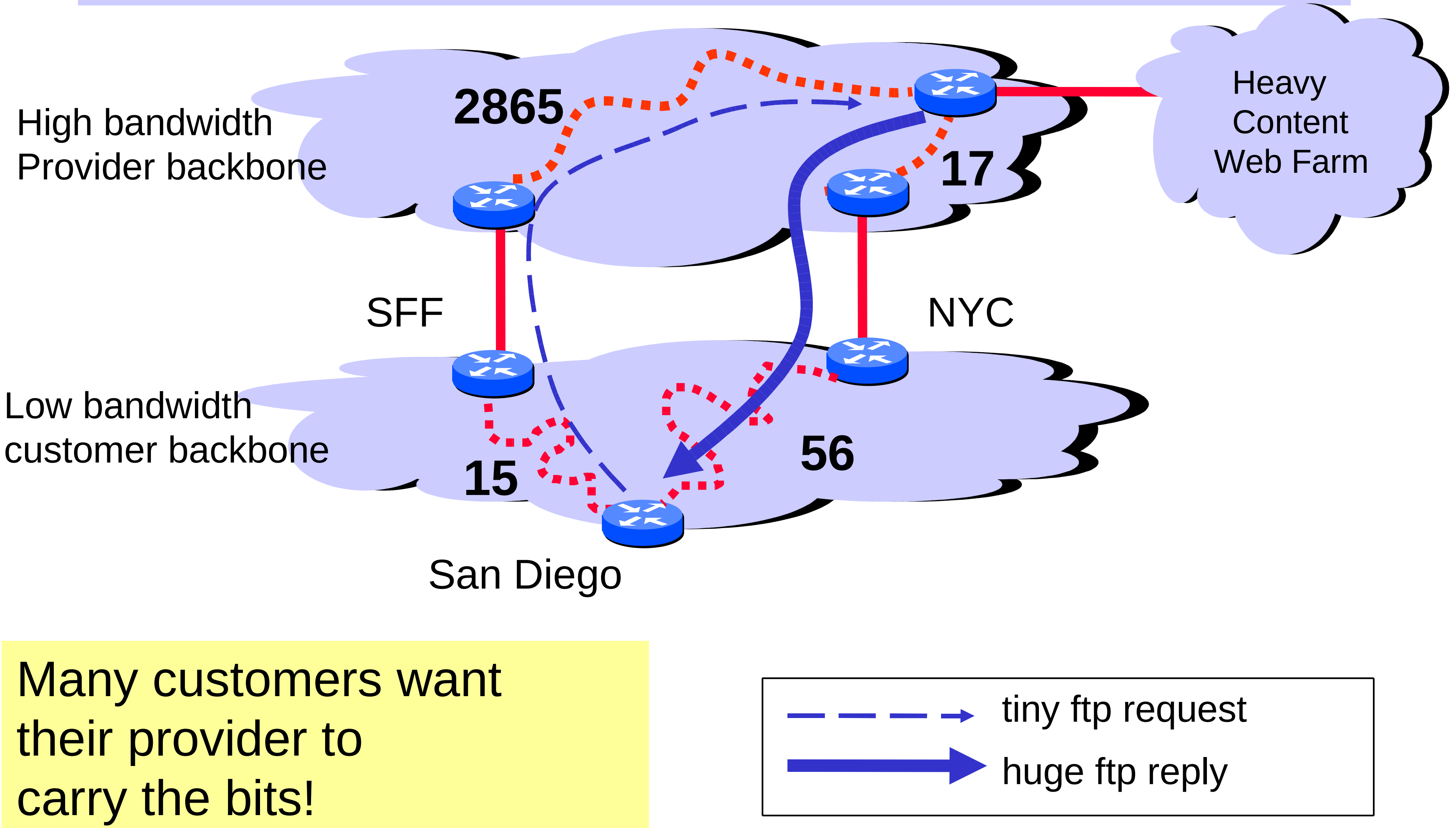
Hot Potato Routing: Go for the Closest Egress Point



This Router has two BGP routes to 192.44.78.0/24.

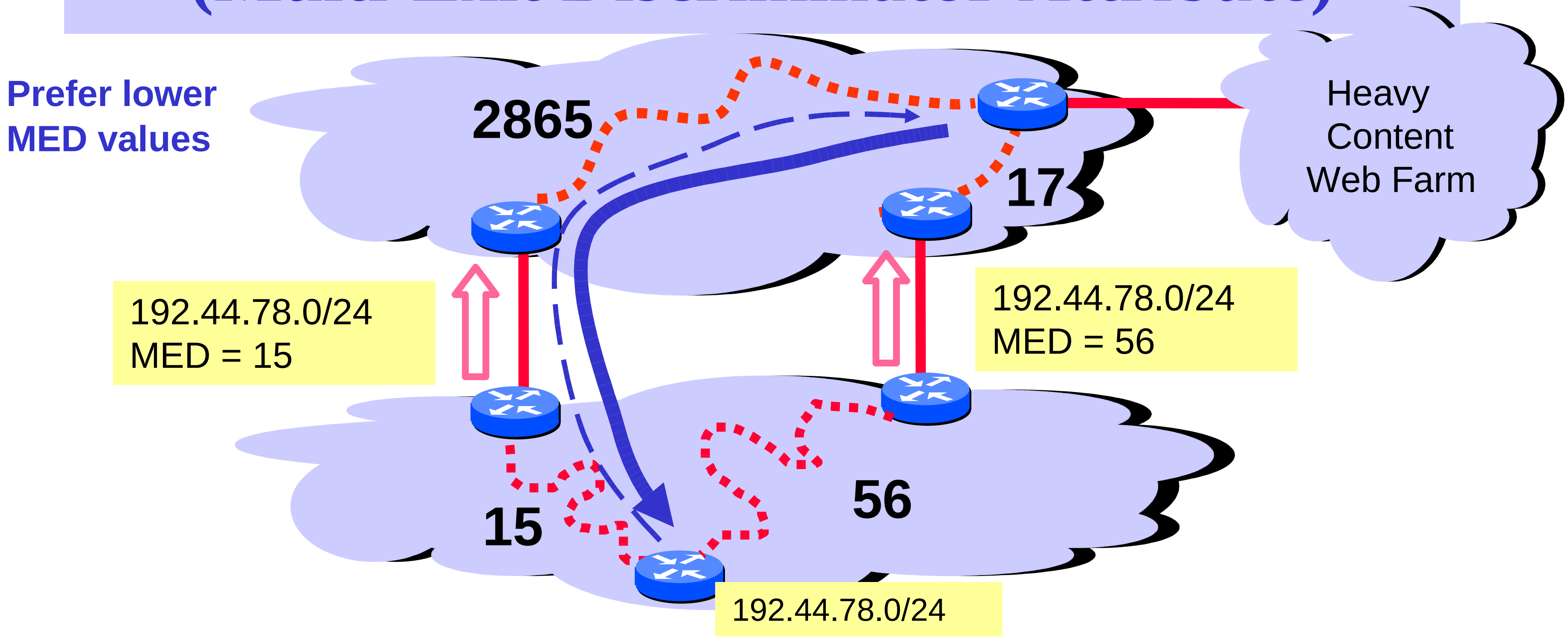
Hot potato: get traffic off of your network as
Soon as possible. Go for egress 1!

Sometimes hot potato is not enough



Many customers want their provider to carry the bits!

Cold Potato Routing with MEDs (Multi-Exit Discriminator Attribute)

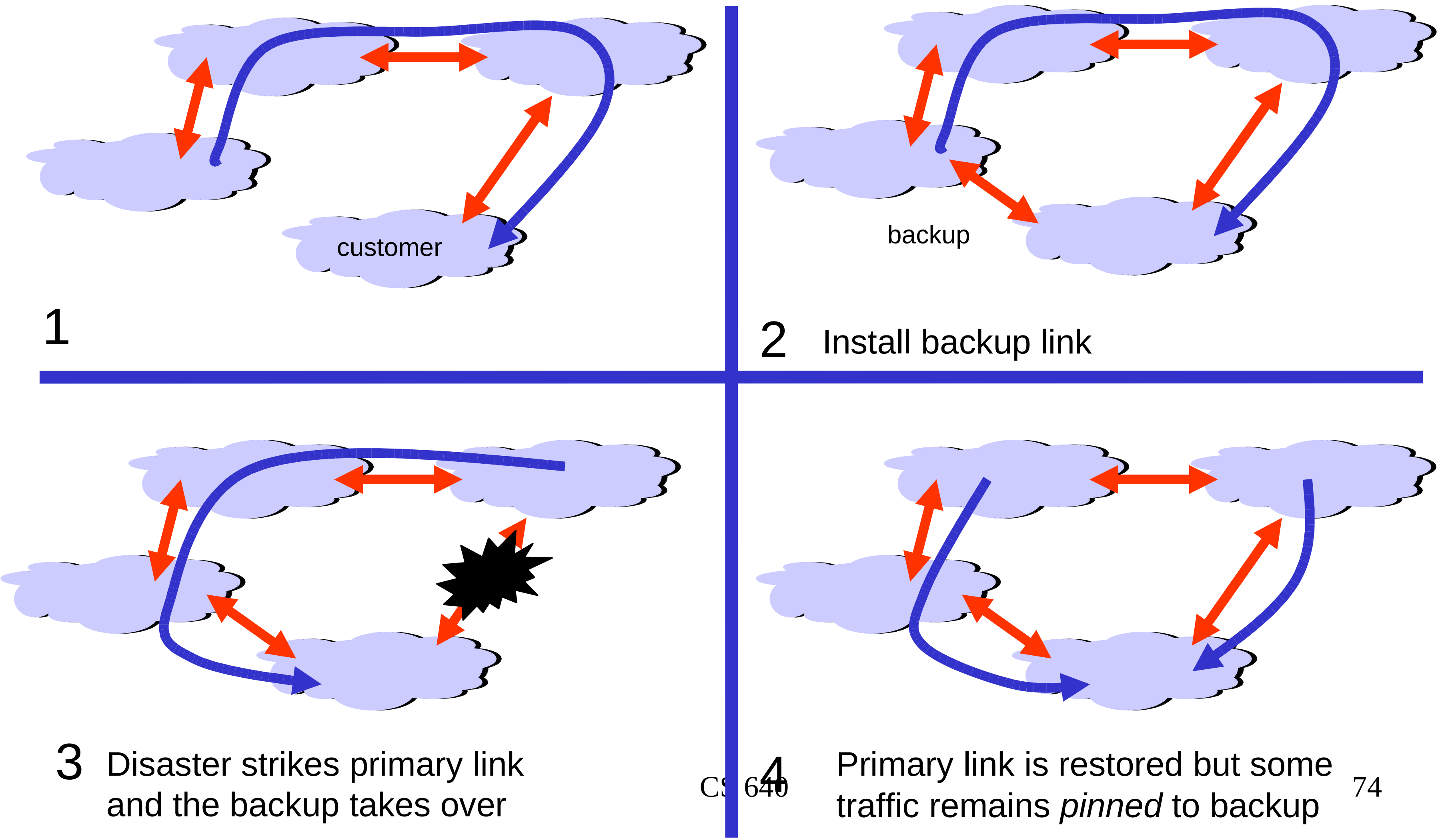


This means that MEDs must be considered BEFORE IGP distance!

Note1 : some providers will not listen to MEDs

Note2 : MEDs need not be tied to IGP distance

Policies Can Interact Strangely ("Route Pinning" Example)



Network Layer - 3

- *ICMP*
- *IPv6*
- *Router architecture*
- *Switching*
  - *IP switching, Tag switching*

IP Version 6

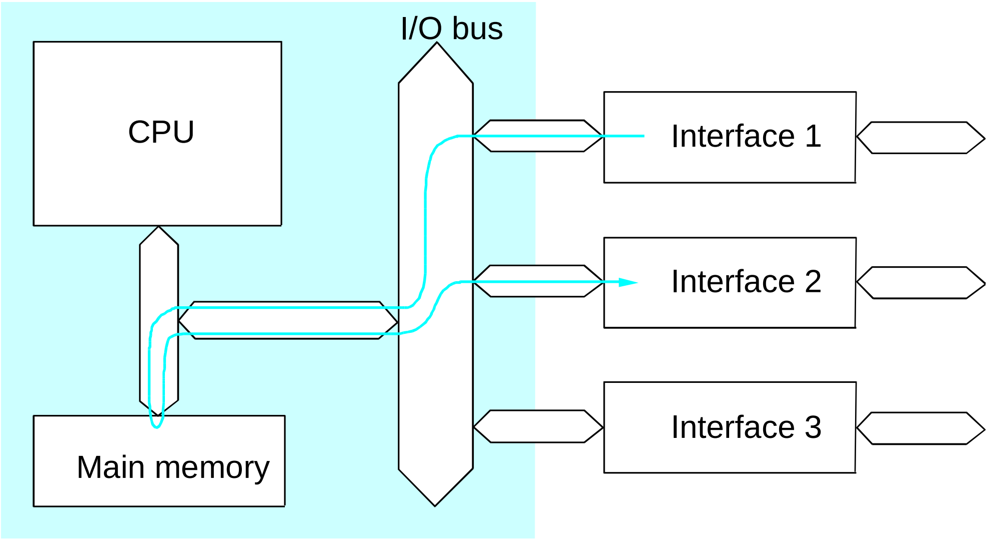
- Features
  - 128-bit addresses (classless)
  - multicast
  - real-time service
  - authentication and security
  - autoconfiguration
  - end-to-end fragmentation
  - protocol extensions
- Header
  - 40-byte "base" header
  - extension headers (fixed order, mostly fixed length)
    - fragmentation
    - source routing
    - authentication and security
    - other options

Internet Control Message Protocol (ICMP)

- Echo (ping)
- Redirect (from router to source host)
- Destination unreachable (protocol, port, or host)
- TTL exceeded (so datagrams don’t cycle forever)
- Checksum failed
- Reassembly failed
- Cannot fragment

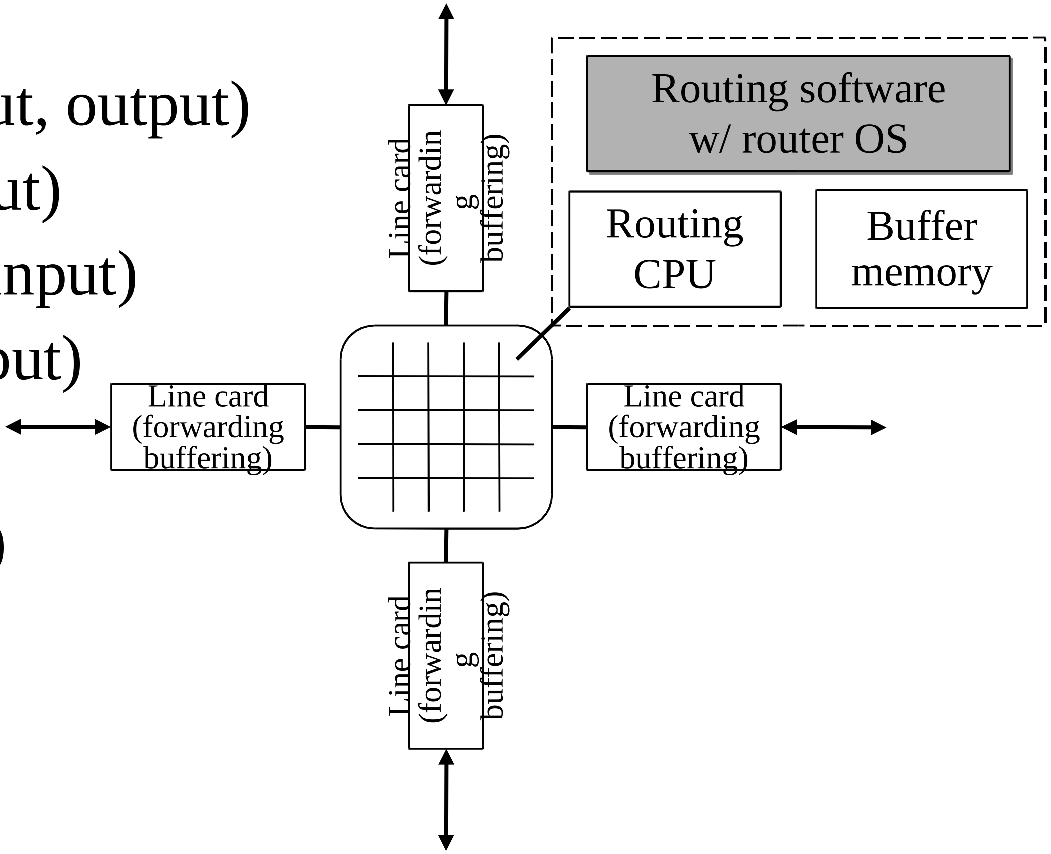
Workstation-Based

- Aggregate bandwidth
  - 1/2 of the I/O bus bandwidth
  - capacity shared among all hosts connected to switch
  - example: 1Gbps bus can support 5 x 100Mbps ports (in theory)
- Packets-per-second
  - must be able to switch small packets
  - 300,000 packets-per-second is achievable
  - e.g., 64-byte packets implies 155Mbps



High-Speed IP Router

- Switch (possibly ATM)
- Line Cards
  - link interface (input, output)
  - router lookup (input)
  - common IP path (input)
  - packet queue (output)
- Control Processor
  - routing protocol(s)
  - exceptional cases



IP Forwarding is Slow

- Problem: classless IP addresses (CIDR)
- Route by variable-length Forwarding Equivalence Classes (FEC)
  - FEC = IP address plus prefix of 1-32 bits; e.g., 172.200.0.0/16
- IP Router
  - forwarding tbl: <FEC> ⟶ <next hop, port>
  - match IP address to FEC w/ longest prefix

ATM Forwarding

- Primary goal: fast, cheap forwarding
- 1Gb/s IP router: $187,000
- 5Gb/s ATM switch: $41,000
- Create Virtual Circuit at Flow Setup
  – <in VCI> ⟶ <port, out VCI>
- Cell Forwarding
  – index, swap, switch

Cisco: Tag Switching

- Add a VCI-like tag to packets
  - <in tag> ⟶ <next hop, port, out tag>
- TSR uses ATM switch hardware
- IP routing protocols (OSPF, RIP, BGP)
  - build forwarding table from routing table
- Goal: IP router functionality at ATM switch speeds/costs

Forwarding

- *Shim* before IP header

- Tag Forwarding Information Base (TFIB)
  - <in tag> → <next hop, port, out tag>
- Just like ATM
  - index, swap, switch

Tag Binding

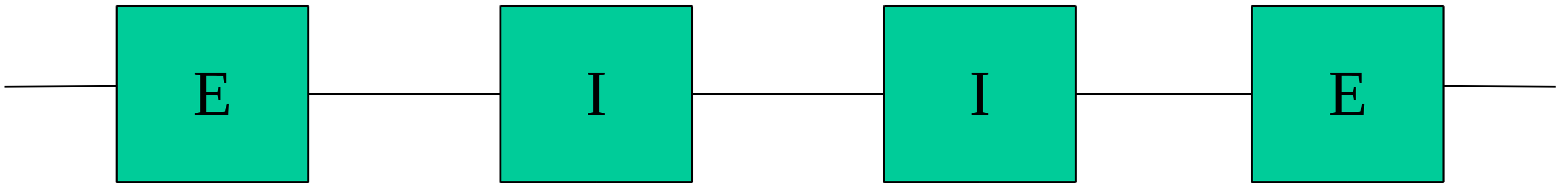
- New FEC from IP routing protocols
  - Select local tag (index in TFIB)
  - <in tag> ⟶ <next hop, port, ???>
- Need <out tag> for next hop
- Other routers need my <in tag>
- Solution: distribute tags like other routing info

Tag Distribution Protocol

- Send TDP messages to peers
  - <FEC, my tag>
- Upon receiving TDP message, check if sender is next hop for FEC
  - yes, save tag in TFIB
  - no, can discard or save for future use
- 'Control-driven' label assignment

The First Tag

- Two kinds of routers: edge vs. interior



- Edge: add shim based on IP lookup, strip at exit
- Interior: forward by tag only

Robustness Issues

- What if tag fault?
  - try to forward (default route)
  - discard packet
- Forwarding Loops
  - topology changes cause temporary loops
  - TTL field in tag, same as IP

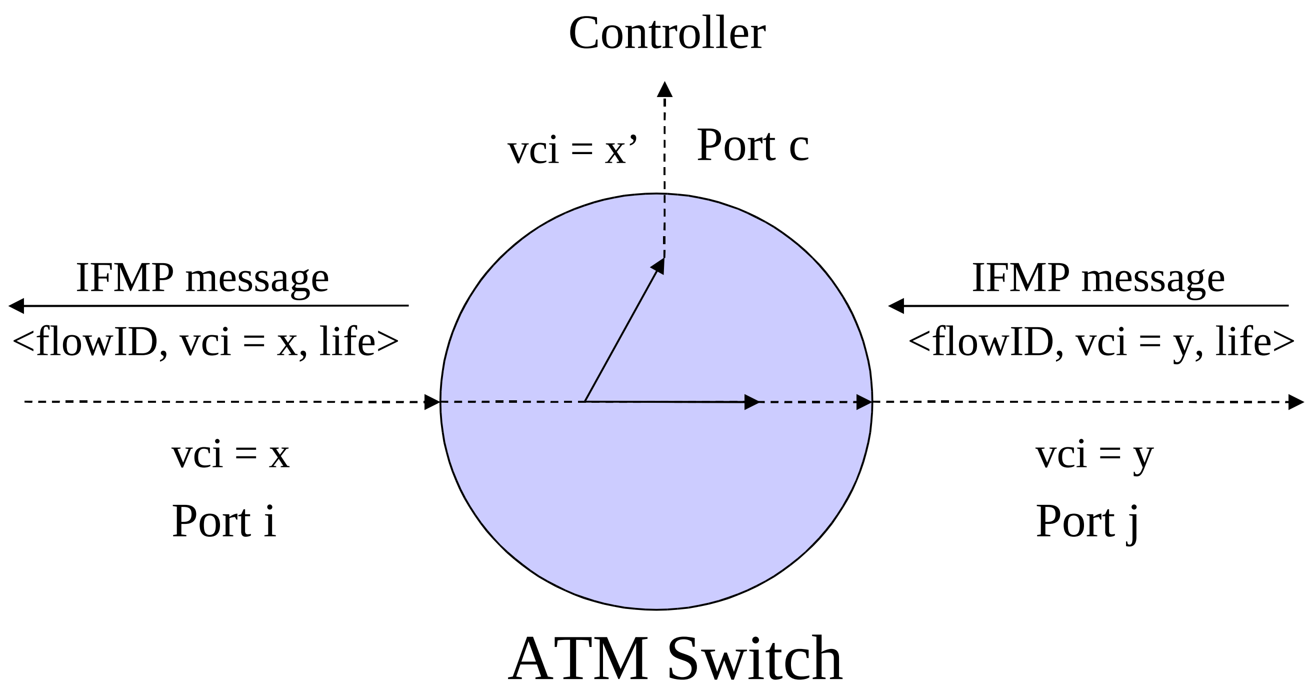
Ipsilon: IP Switching

- Run on ATM switch over ATM network
  - ATM hardware + IP switching software
- Idea: Exploit temporal locality of traffic to cache routing decisions
- Associate labels (VCI) with flows
  - forward packets as usual
  - main difference is in how labels are created, distributed to other routers

IP Switch

- Assume default ATM virtual circuits between routers
- Router runs IP routing protocol, can forward IP packets on default VCs
- Identify flows, assign flow-specific VC
  - flow = port pair or host pair
- 'Data-driven' label assignment

Flow Setup on IP Switch



- <vci = x> ⟶ <port c, vci = x’>
- Get IFMP, <vci = x> ⟶ <port j, vci = y>

Comparison

IP Switching

- Switch by flow
- Data driven
- Soft-state timeout
- Between end-hosts
- Every router can do IP lookup
- Scalable?

Tag Switching

- Switch by FEC
- Control driven
- Route changes
- Between edge TSRs
- Interior TSRs only do tag switching