Network:
Policy Routing Analysis, DHCP

Qiao Xiang

https://qiaoxiang.me/courses/cnns-xmuf21/index.shtml

12/09/2021

This deck of slides are heavily based on CPSC 433/533 at Yale University, by courtesy of Dr. Y. Richard Yang.

Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing computation protocols
      - Global Internet routing
        - Basic architecture
        - BGP (Border Gateway Protocol): The de facto Inter-domain routing standard
          - Basic operations
          - BGP as a policy routing framework (control interdomain routes)
          - Policy/interdomain routing analysis
            - Global preference aggregation and Arrow's Theorem
            - Local preference aggregation

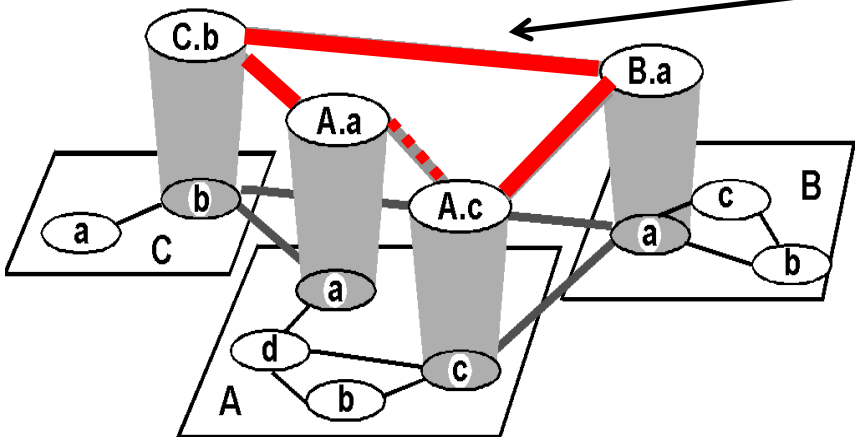
Admin

- Week 15 lab class:
  - Lab assignment 3-4 results to be announced
  - A quick review on lecture 16-27

Recap: Internet Routing Architecture

- Interdomain routing uses a path vector protocol based on AS topology
  - improves scalability, privacy, autonomy
- Only a small # of routers (gateways) from each AS in the interdomain level
  - improves scalability
- Autonomous systems have flexibility to choose their own intradomain routing protocols
  - allows autonomy

Recap: Routing with Autonomous Systems



Inter-AS routers form an overlay

Gateway routers of same AS share learned external routes using iBGP.

Gateway routers participate in intradomain to learn internal routes.

AS C
(RIP intra routing)

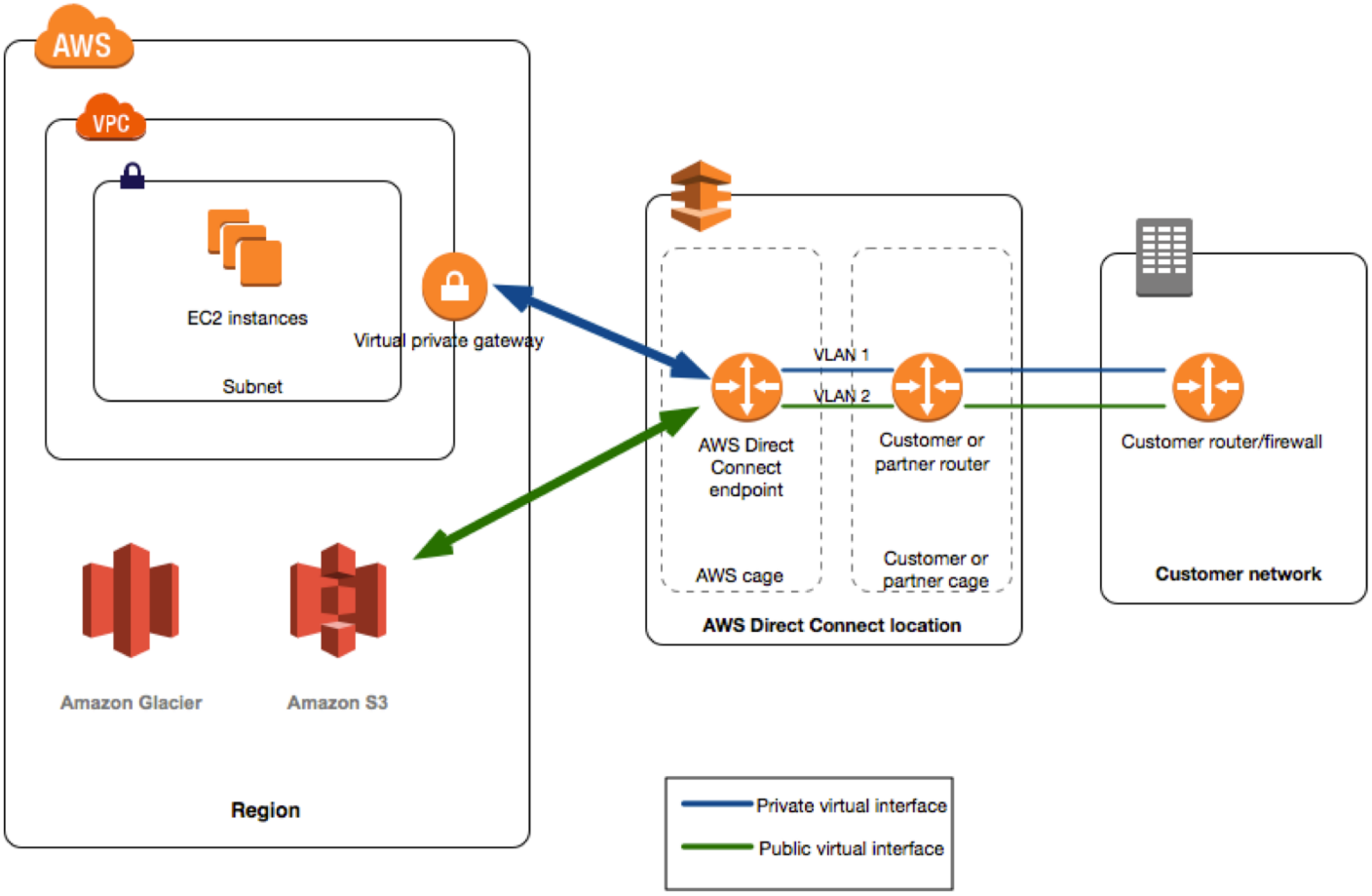
AS A
(OSPF intra routing)

AS B
(OSPF intra routing)

a
b
eBGP
e
iBGP
d
c
f
g
i
h

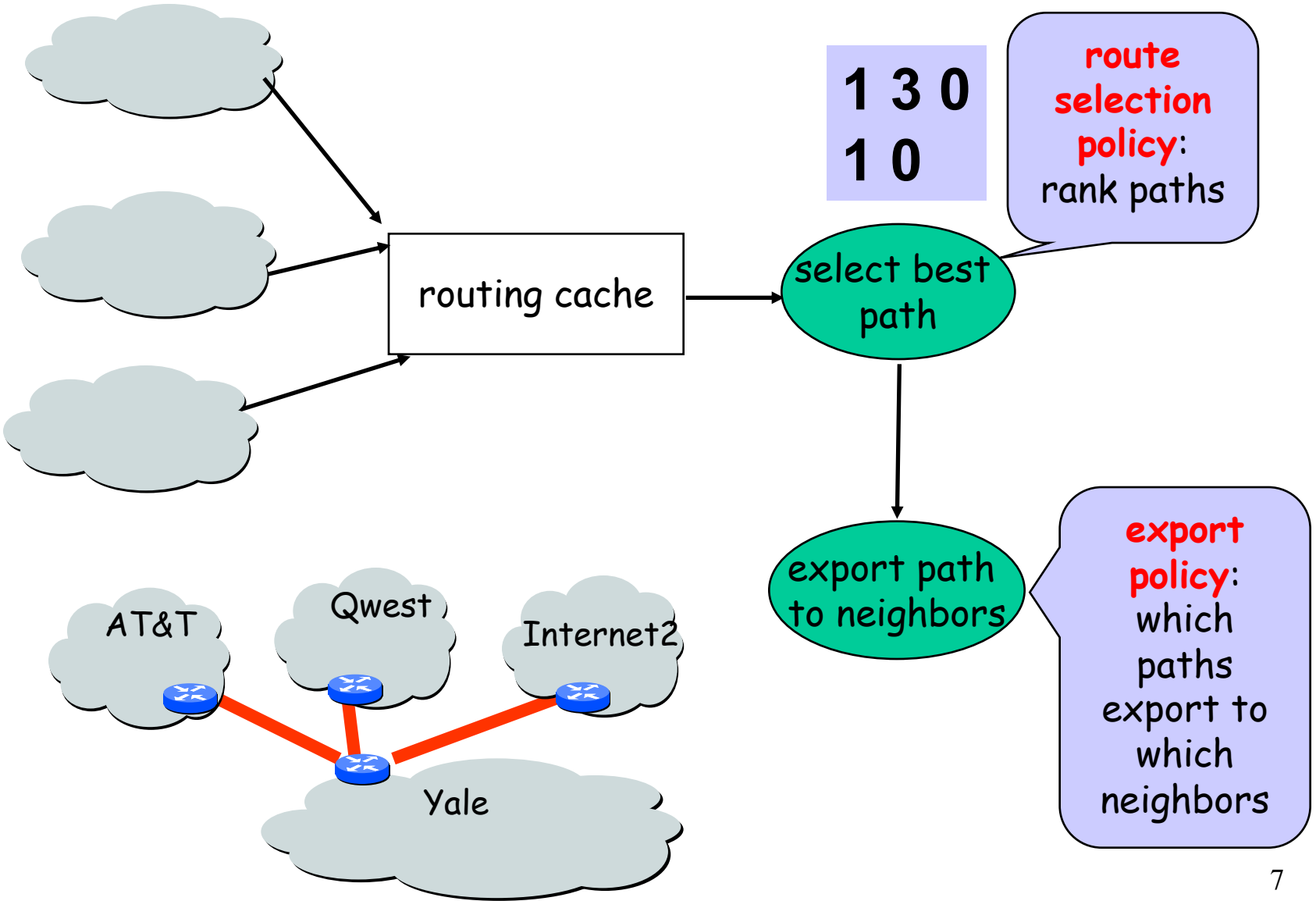
Gateway routers of diff. auto. systems exchange routes using eBGP

Example: AWS Direct Connect



http://docs.aws.amazon.com/directconnect/latest/UserGuide/Welcome.html

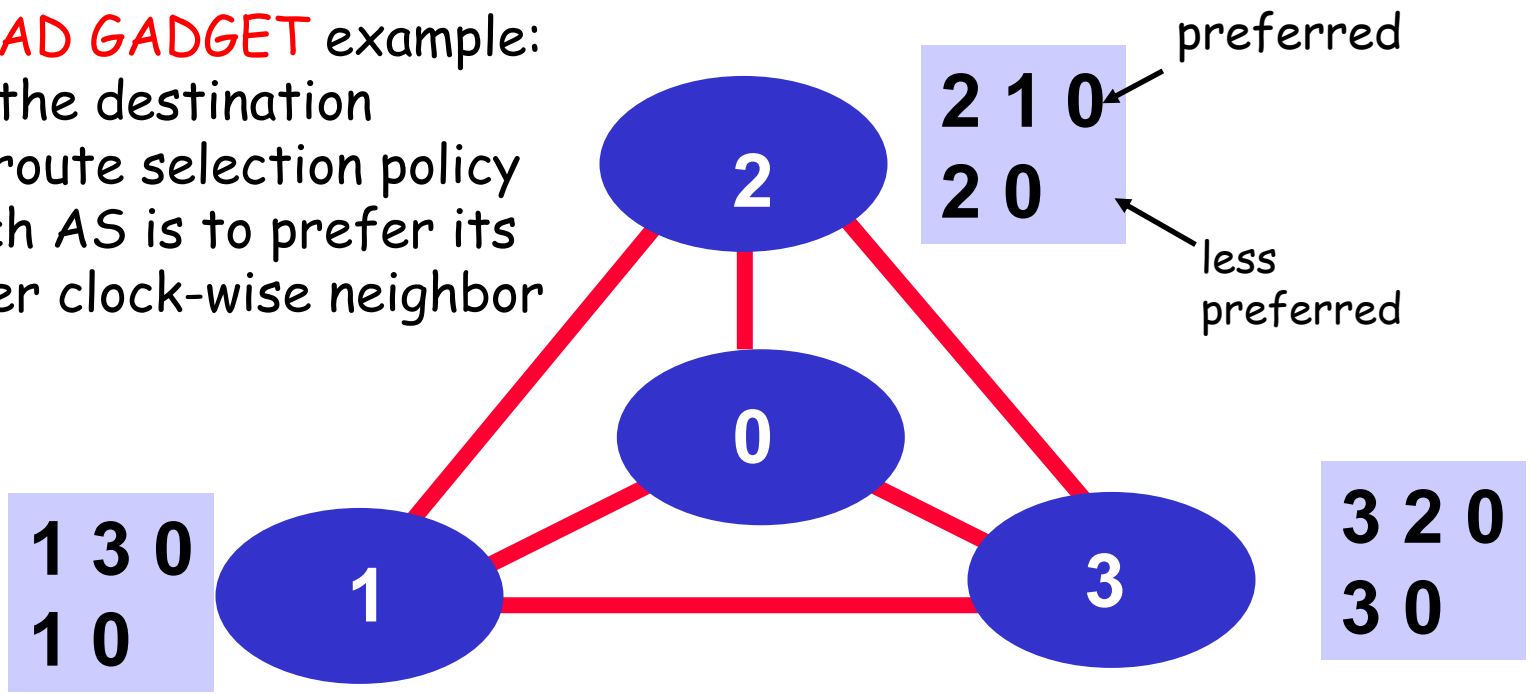
Recap: BGP as a Policy Routing Framework



Recap: Policy Routing as a Preference Aggregation System

- A policy routing system can be considered as a system to aggregate individual preferences, but aggregation may not be always successful.

The BAD GADGET example:
- 0 is the destination
- the route selection policy of each AS is to prefer its counter clock-wise neighbor



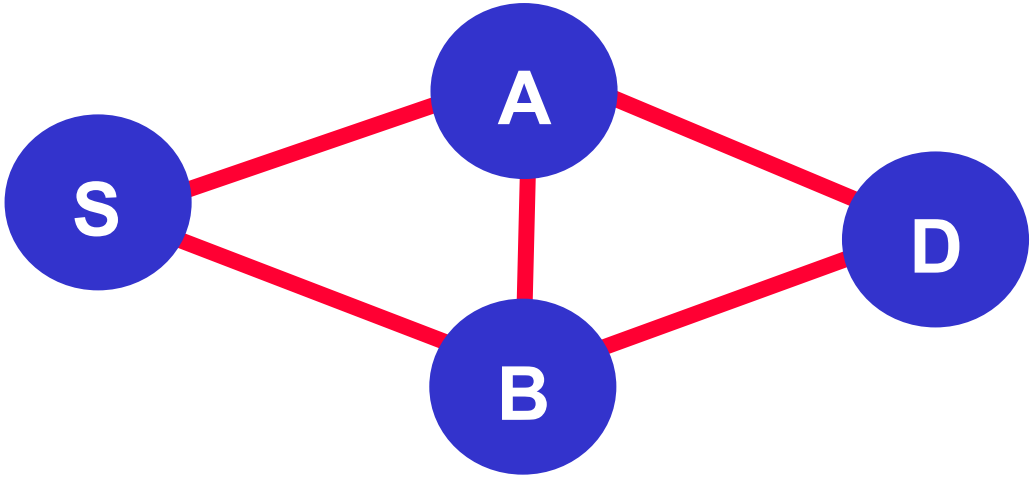
Policy (preferences) aggregation fails: routing instability !

Recap: General Framework of Preference Aggregation

- Also called Social Choice
  - Given individual preferences, define a framework to aggregate individual preferences:
    - A set of choices: a, b, c, …
    - A set of voters 1, 2, …
      - Each voter has a preference (ranking) of all choices, e.g.,
        - » voter 1: a > b > c
        - » voter 2: a > c > b
        - » voter 3: a > c > b
    - A well-specified aggregation rule (protocol) computes an aggregation of ranking, e.g.,
      - Society (network): a > b > c

For more details: see Semih Salihoglu (2007). Interdomain Routing as a Social Choice.

Example: Aggregation of Global Preference



- Choices (for S->D route): SAD, SBD, SABD, SBAD
- Voters S, A, B, D
- Each voter has a preference, e.g.,
  - S: SAD > SBD > SABD > SBAD
  - ...

Recap: Arrow's Aggregation Framework

- Axioms:
  - Transitivity
    - if a > b & b > c, then a > c
  - Unanimity:
    - If all participants prefer a over b (a > b) => a > b
  - Independence of irrelevant alternatives (IIA)
    - Social ranking of a and b depends only on the relative ranking of a and b among all participants
- Result:
  - Arrow's Theorem: Any constitution that respects transitivity, unanimity and IIA is a dictatorship.

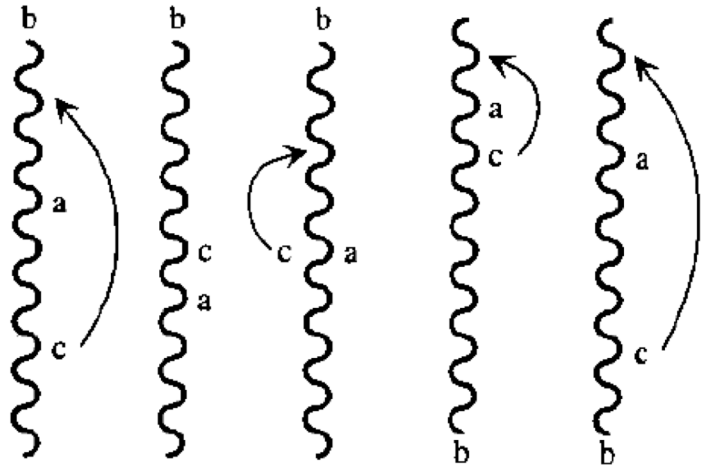
Proofs of Arrow's Theorem

- There are quite a few proofs, and the six-page paper linked on the Schedule page gives three simple proofs.

- Below, I give the key insight of the proof using approach 1.

The Extremal Lemma

- Let choice b be chosen arbitrarily. Assume that every voter puts b at the very top or the very bottom of his ranking. Then society must as well (even if half voters put b at the top and half at the bottom)
- Proof: by contradiction.
  - Assume there exist a and c such that society has a >= b; b >= c.
    - By transitivity, a >= c
  - We can move c above a w/o changing ab or cb votes, leading to c > a
    - By unanimity, c > a



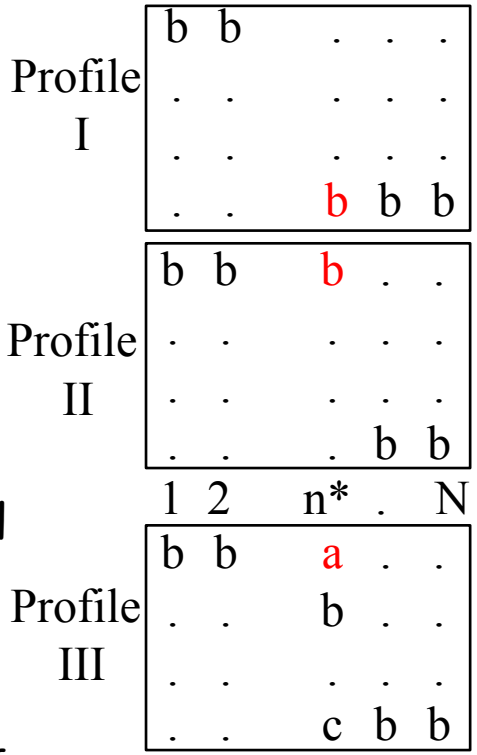
Step 1: Existence of Pivotal Voter

- Let choice b be chosen arbitrarily. There exist a voter n* = n(b) who is extremely pivotal for b in the sense that by changing his vote at some profile, he can move b from the very bottom to the very top in the social ranking.

- Proof:
  - Consider an extreme profile where b is at the bottom of each voter.
  - Consider voter from 1 to n, and we move b from bottom to top one-by-one.
  - The first voter whose change causes b to move to the top is n*
    - By unanimity, this change must happen at the latest when n*=n

Step 2: n*=n(b) is dictator of any pair ac not involving b

- Proof
  - Consider a from ac pair. We show that if a $>_{n^*}$ c, then constitution protocol has a > c
  - Let profile before n* moves b to top as profile I
  - Let profile after n* moves b to top as profile II
  - Construct profile III from II by letting n* move a above b; all others can arrange ac as they want, but leave b in extreme position
    - a>b in profile I, and a,b is equivalent by IIA in profiles I and III
    - b>c in profile II, and bc is equivalent by IIA in profiles II and III



Constitution protocol: b bottom

Constitution protocol : b top

Constitution protocol: a > b since ab same as I

b > c since bc same as II

Step 3: n* is dictator for every pair ab

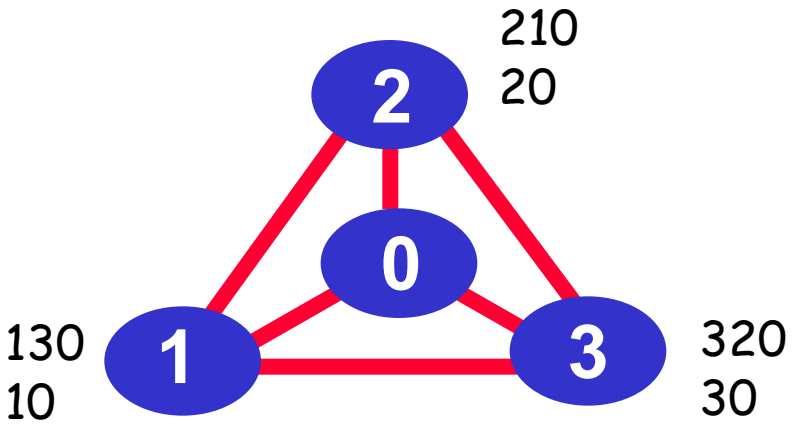
- Consider c not equal to a or b
- There exists n(c) who is a dictator of any pair not involving c, such as the pair ab, i.e.,
  - For any profile, if $a >_{n(c)} b$, $a > b$ in constitution protocol
- n(c) must be n*
  - Assume not.
    - Consider Profile I and Profile II.
    - Since n(c) is not n*, n(c) ranking of ab does not change in Profile I and Profile II.
    - When n* changes ab ranking between Profile I and Profile II, the global ranking of ab changes.
    - Contradiction.

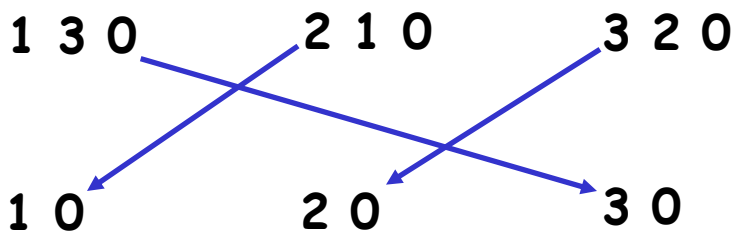
Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing computation protocols
      - Global Internet routing
        - Basic architecture
        - BGP (Border Gateway Protocol): The de facto Inter-domain routing standard
          - Basic operations
          - BGP as a policy routing framework (control interdomain routes)
          - Policy/interdomain routing analysis
            - Global preference aggregation and Arrow's Theorem
            - *Local preference aggregation*

BGP w/ Local Preference

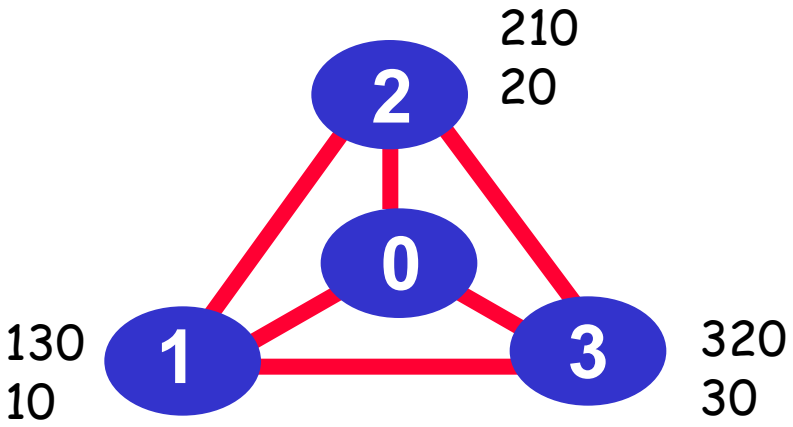
- BGP preferences are typically local (only on paths start from itself)
- Hence the preferences have dependency (priority)
  - The "closer" a node to the destination, the more "powerful" it may be

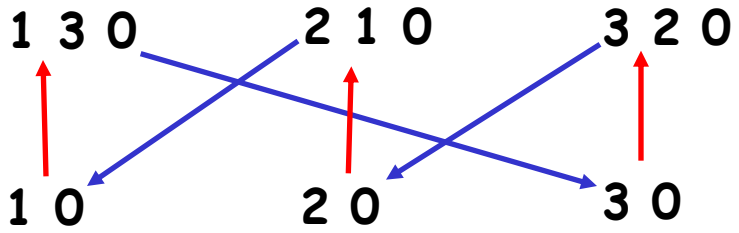




Complete Dependency: P-Graph

- Complete dependency can be captured by a structure called P-graph
- Nodes in P-graph are feasible paths
- Edges represent priority (low to high)
  - A directed edge from path N_1P_1 to P_1
    - intuition: to let N_1 choose N_1P_1, P_1 must be chosen and exported to N_1
  - A directed edge from a lower ranked path to a higher ranked path
    - intuition: the higher ranked path should be considered first

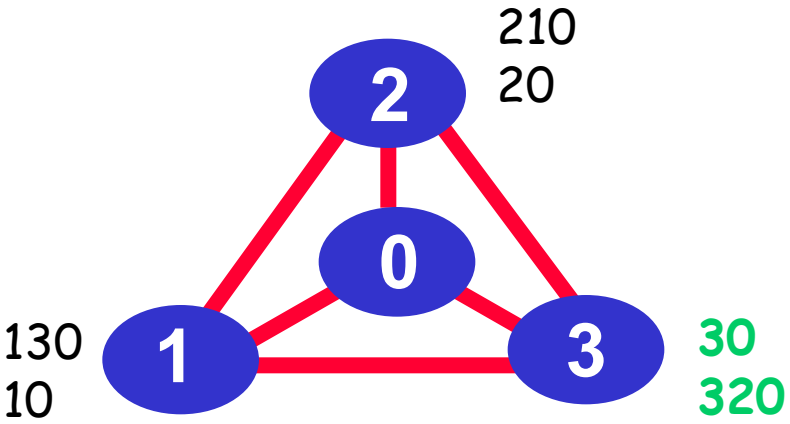


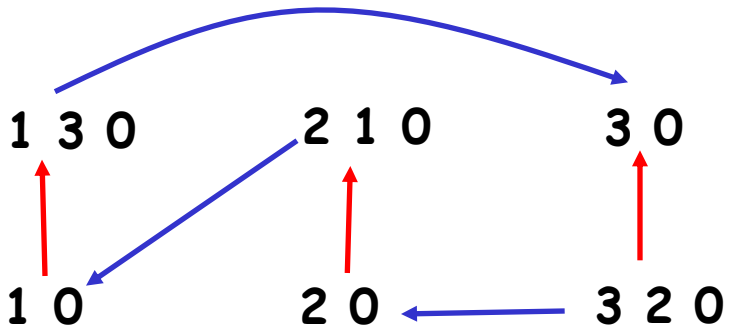


Any observation on the P-graph?

P-Graph and BGP Convergence

- If the P-graph of the networks has no loop, then policy routing converges.
  - intuition: choose the path node from the partial order graph with no out-going edge to non-fixed path nodes, fix the path node, eliminate all no longer feasible; continue
- Example: suppose we swap the order of 30 and 320



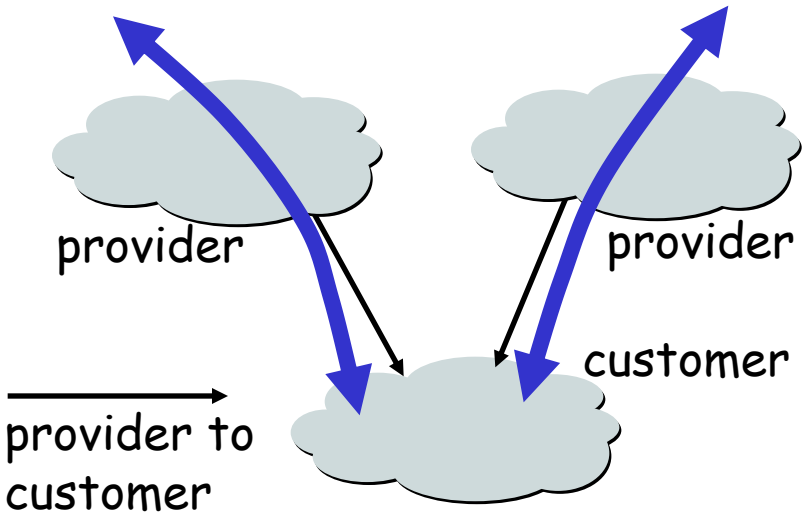


Outline

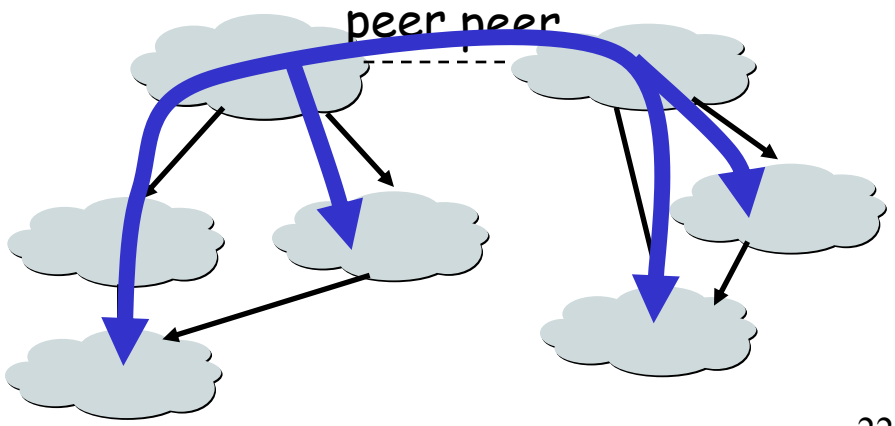
- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing computation protocols
      - Global Internet routing
        - Basic architecture
        - BGP (Border Gateway Protocol): The de facto Inter-domain routing standard
          - Basic operations
          - BGP as a policy routing framework (control interdomain routes)
          - Policy/interdomain routing analysis
            - Global preference aggregation and Arrow's Theorem
            - Local preference aggregation
            - ***Economics and interdomain routing patterns***

Internet Economy: Two Types of Business Relationship

- *Customer provider relationship*
  - a provider is an AS that connects the customer to the rest of the Internet
  - customer pays the provider for the transit service
  - e.g., Yale is a customer of AT&T and QWEST

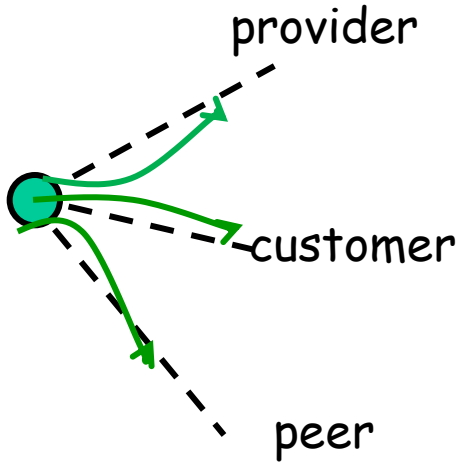


- *Peer-to-peer relationship*
  - mutually agree to exchange traffic between their respective **customers** only
  - there is no payment between peers



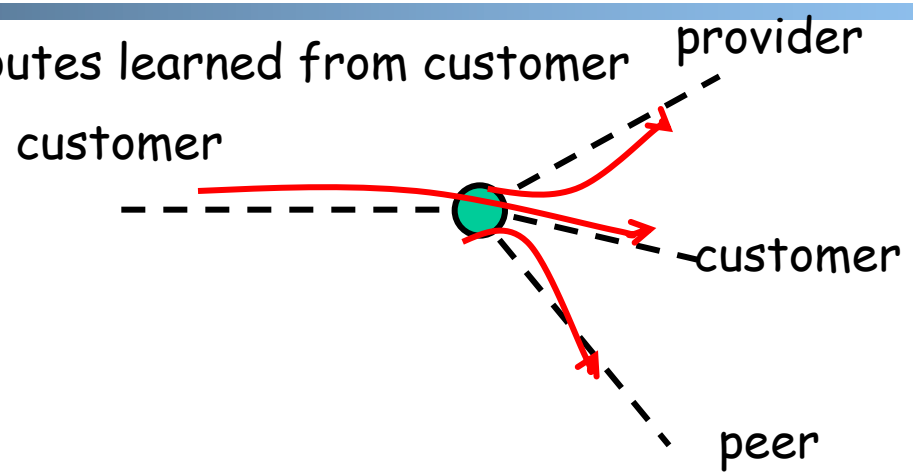
Route Selection Policies and Economics

- Route selection (ranking) policy:
  - the typical route selection policy is to prefer customers over peers/providers to reach a destination, i.e., Customer > pEer/Provider



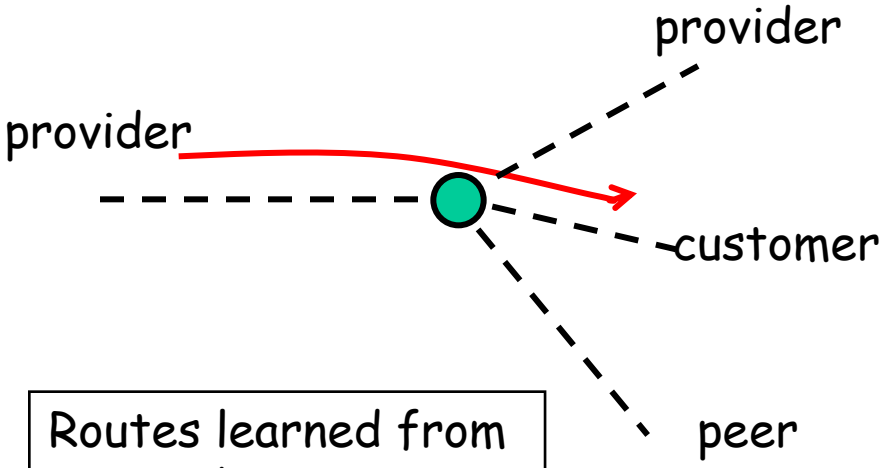
Export Policies and Economics

case 1: routes learned from customer



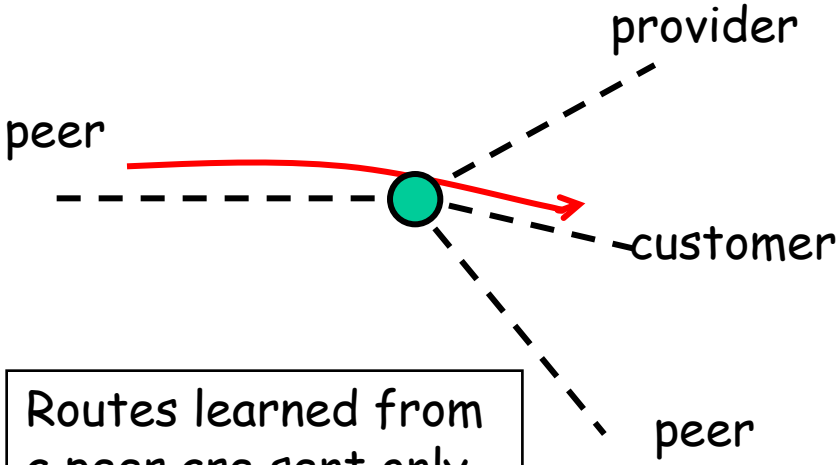
Routes learned from a customer are sent to all other neighbors

case 2: routes learned from provider



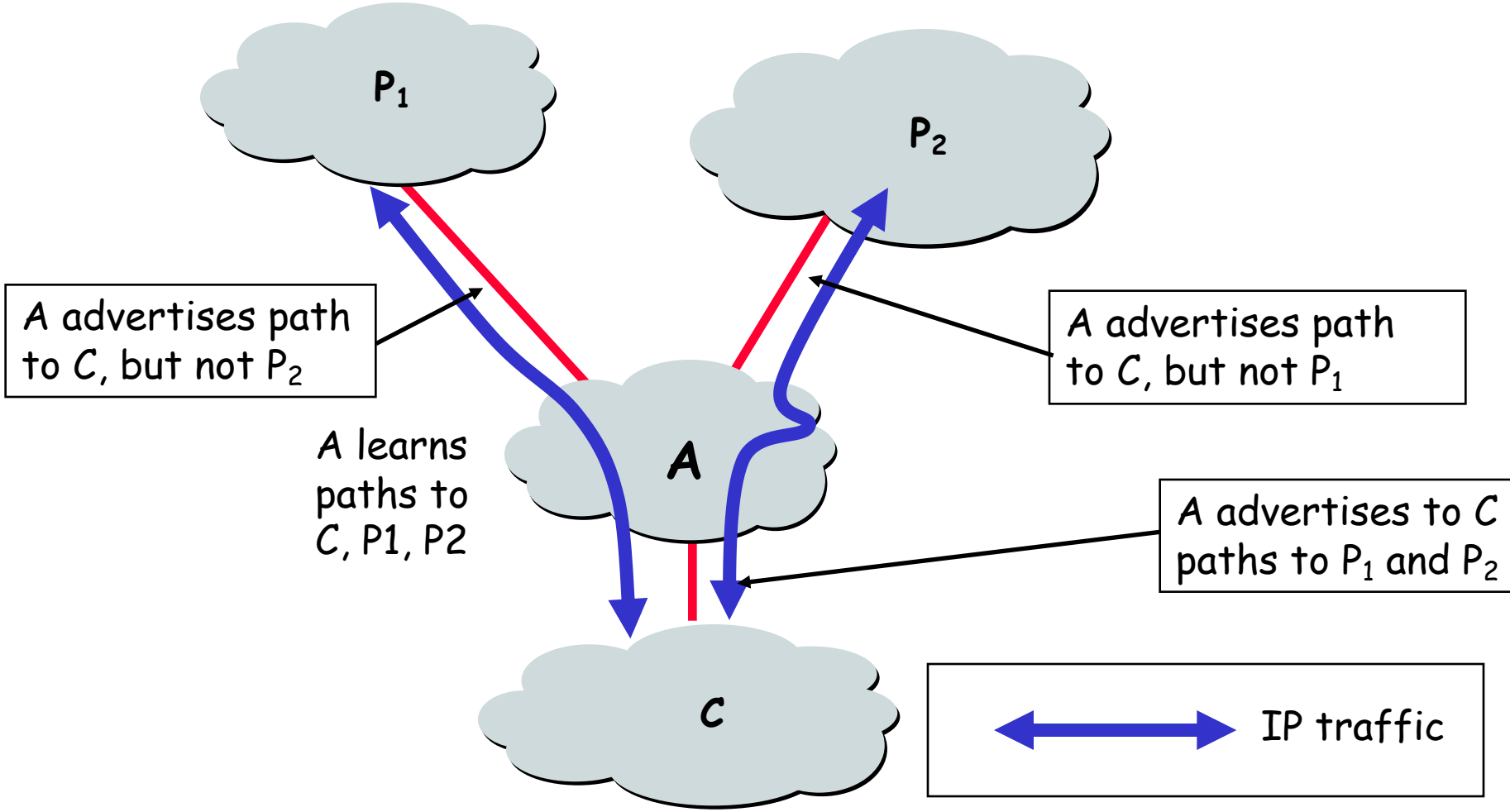
Routes learned from a provider are sent only to customers

case 3: routes learned from peer



Routes learned from a peer are sent only to customers

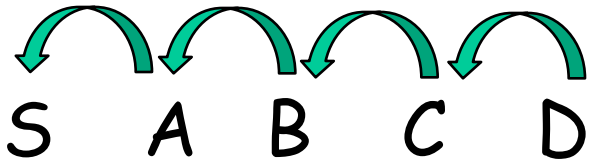
Example: Typical Export -> No-Valley Routing



Suppose P_1 and P_2 are providers of A; A is a provider of C

Typical Export Policies Route Patterns

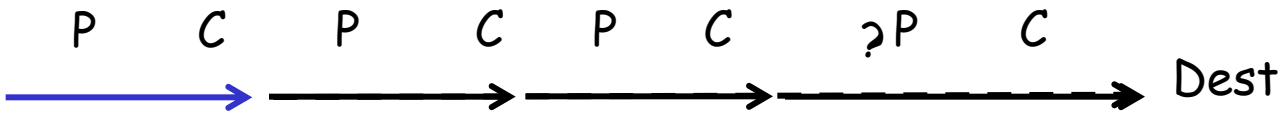
- Assume a BGP path SABCD to destination AS D. Consider the business relationship between each pair:



- Three types of business relationships:
  - PC (provider-customer)
  - CP (customer-provider)
  - PP (peer-peer)

Typical Export Policies Route Patterns

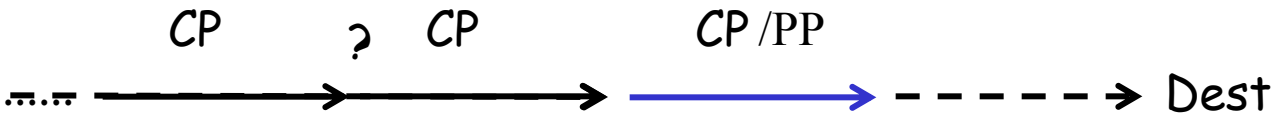
- Invariant 1 of valid BGP routes (with labels representing business relationship)



Reasoning: only route learned from customer is sent to provider; thus after a PC, it is always PC to the destination

Typical Export Policies Route Patterns

- Invariant 2 of valid BGP routes (with labels representing business relationship)



Reasoning: routes learned from peer or provider are sent to only customers; thus all relationship before is CP.

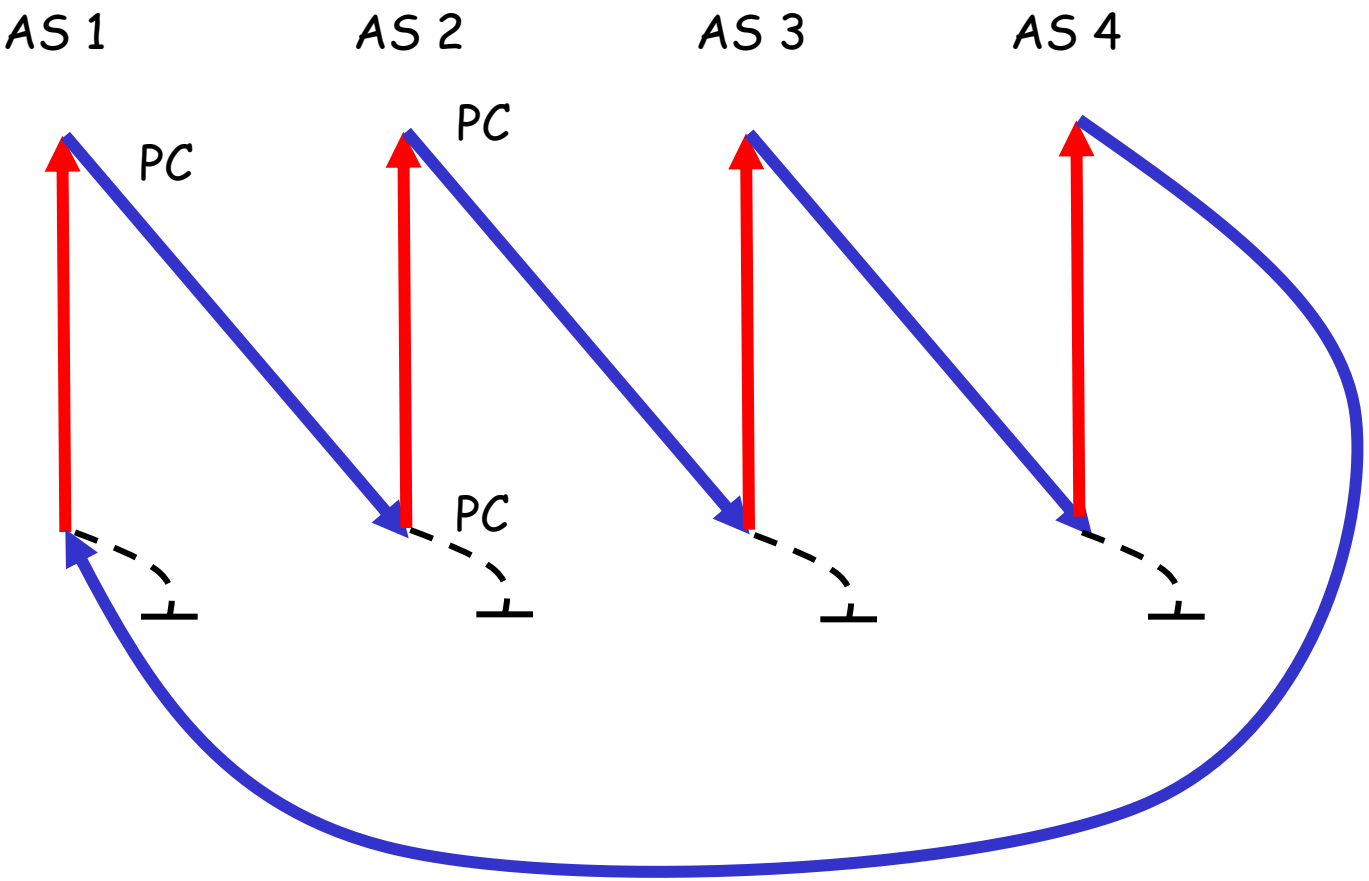
Stability of BGP Policy Routing

- Suppose
  1. there is no loop formed by provider-customer relationship in the Internet
  2. each AS uses typical route selection policy:
     C > E/P
  3. each AS uses the typical export policies

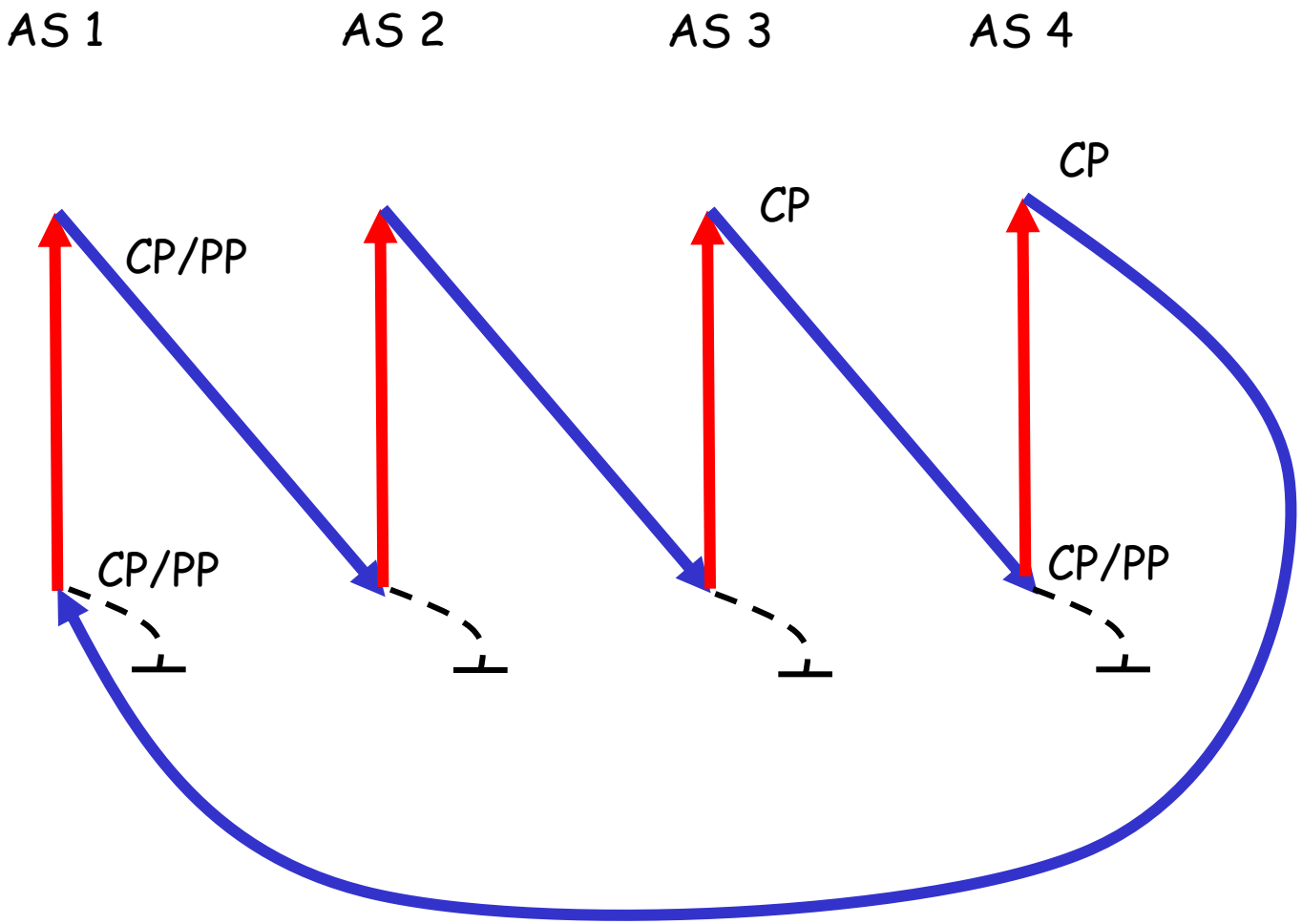
- Then policy routing always converges (i.e., is stable).

Case 1: A Link is PC

Proof by contradiction. Assume a loop in P-graph. Consider a fixed link. in the loop

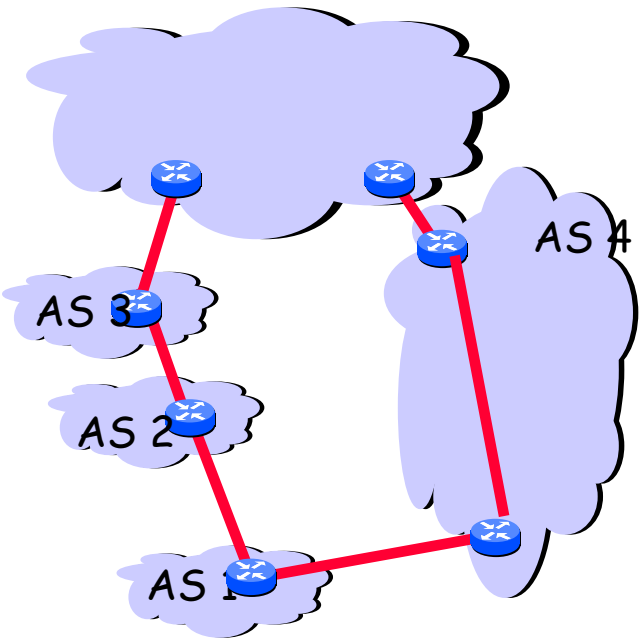


Case 2: Link is CP/PP

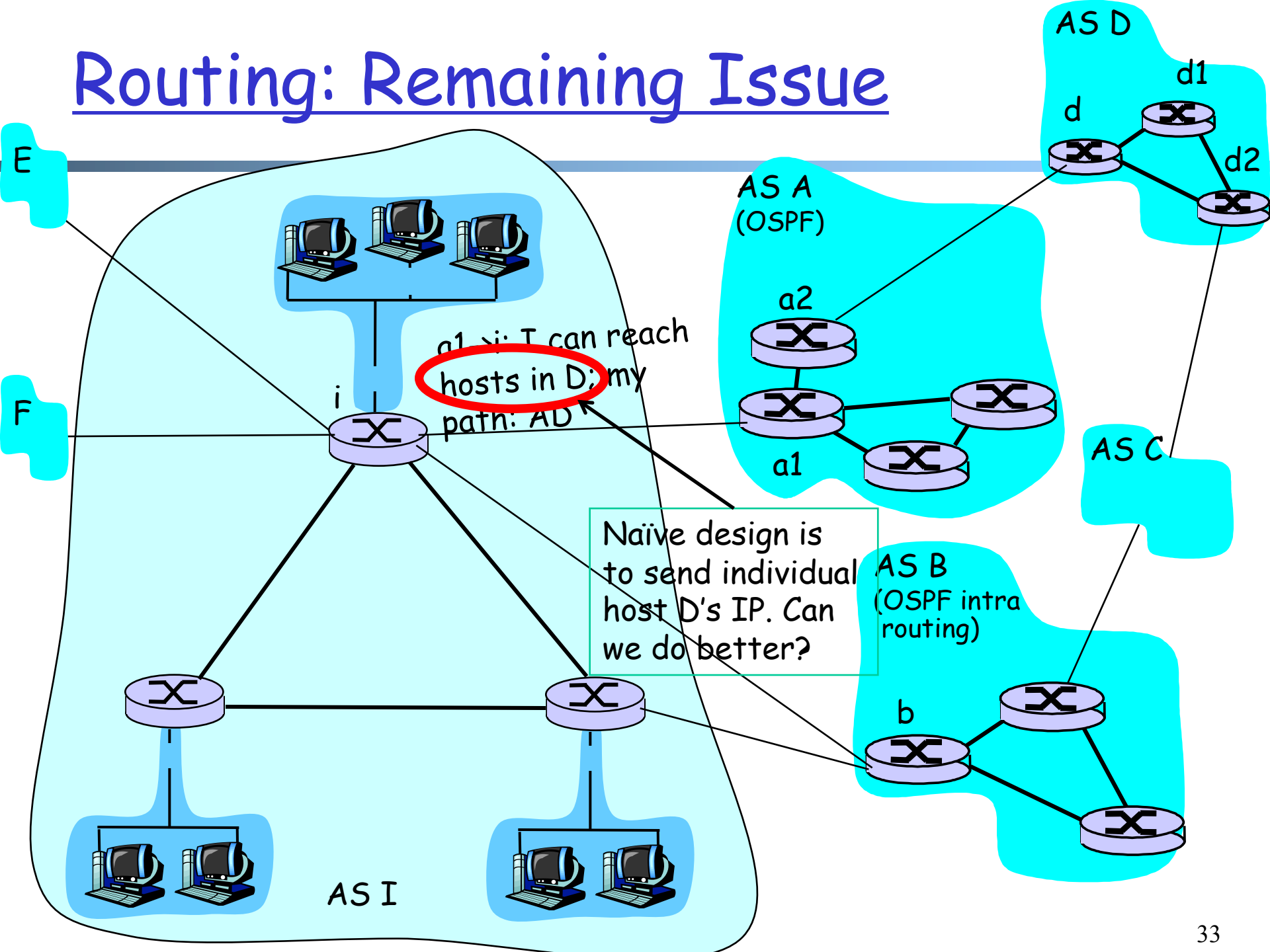


Summary: BGP Policy Routing

- Advantage
  - satisfies current demand
- Issues
  - policy dispute can lead to instability
    - current Internet economy provides a stability framework, but if the framework changes, we may see instability
  - Hierarchical routing can be inefficient



Routing: Remaining Issue



Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing computation protocols
      - Global Internet routing
        - Basic architecture
        - BGP (Border Gateway Protocol): The de facto Inter-domain routing standard
          - Basic operations
          - BGP as a policy routing framework (control interdomain routes)
          - Policy/interdomain routing analysis
            - Global preference aggregation and Arrow's Theorem
            - Local preference aggregation
            - Economics and interdomain routing patterns
          - ***IP addresses for Interdomain routing***

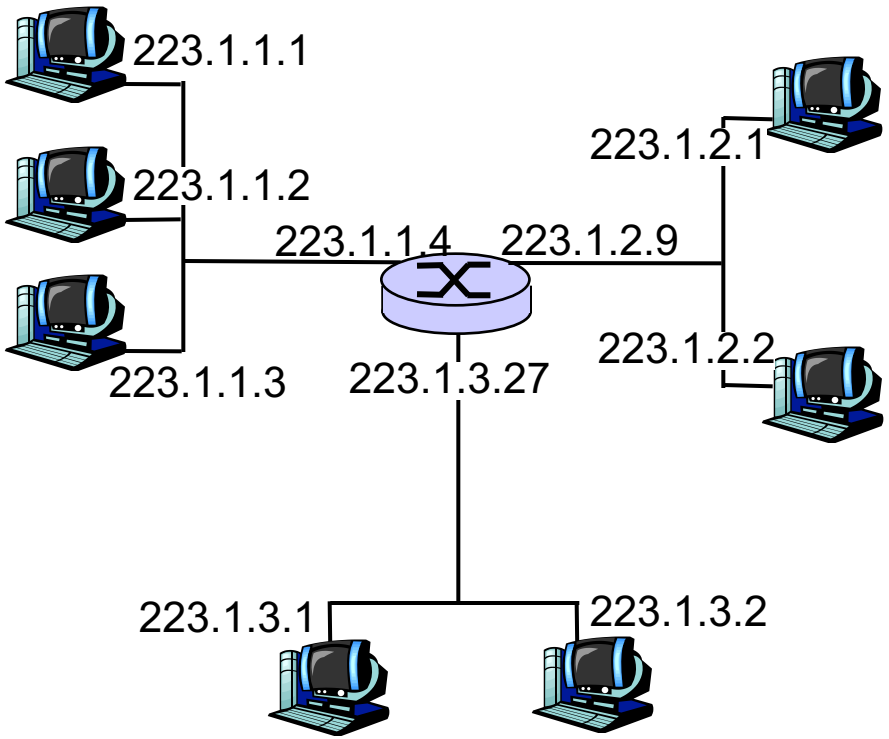
IP Addressing Scheme: Requirements

- Uniqueness: We need an address to uniquely identify each destination
- Aggregability : Routing scalability needs flexibility in aggregation of destination addresses
  - we want to aggregate as a large set of destinations as possible in BGP announcements
- Current: the unit of routing in the Internet is a classless interdomain routing (CIDR) address

IP Address: Uniqueness

- IPv4 address: A 32-bit unique identifier for an *interface*

- *interface:*
  - routers typically have multiple interfaces
  - host may have multiple interfaces



223.1.3.2 = 11011111 00000001 00000011 00000010

223 1 3 2

```
%/sbin/ifconfig -a
```

```
e.g., /etc/sysconfig/network-scripts/ifcfg-enp0s25
%ifup
```

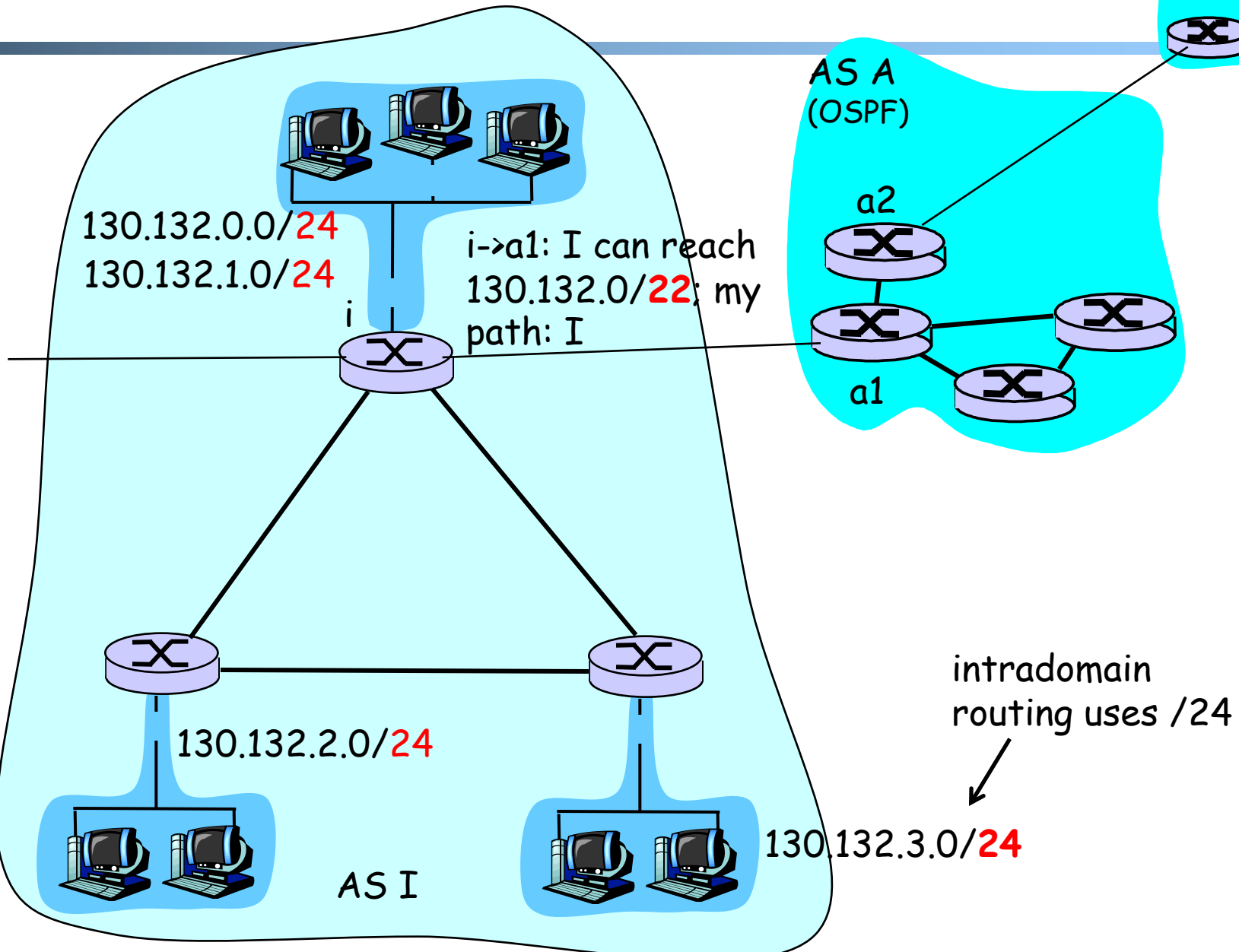
Classless InterDomain Routing (CIDR) Address: Aggregation

- A CIDR address partitions an IP address into two parts
  - A prefix representing the network portion, and the rest (host part)
  - address format: a.b.c.d/x, where x is # bits in network portion of address

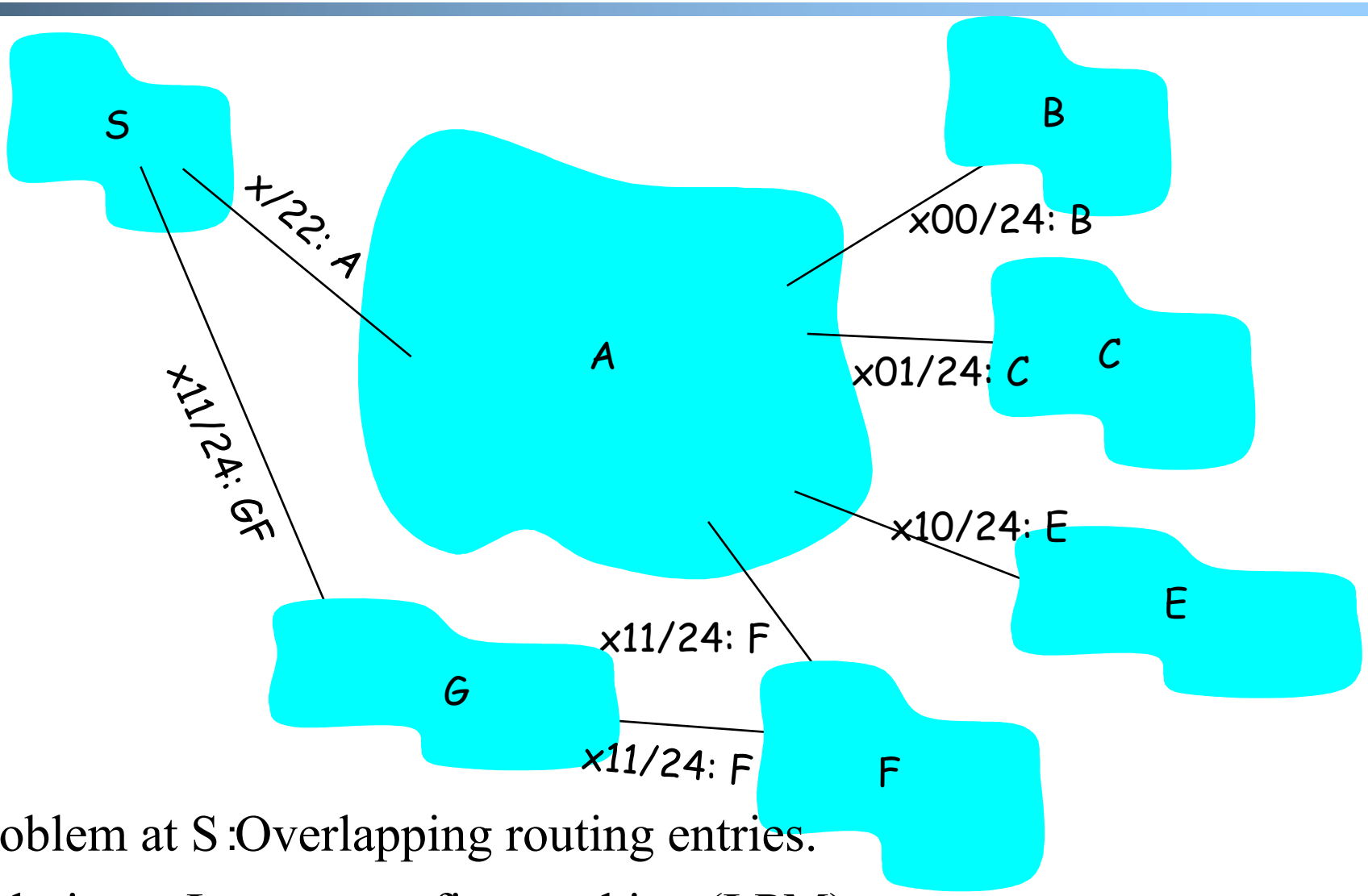
200.23.16.0/23

Some systems use mask (1's to indicate network bits), instead of the /x format

CIDR Aggregation in BGP



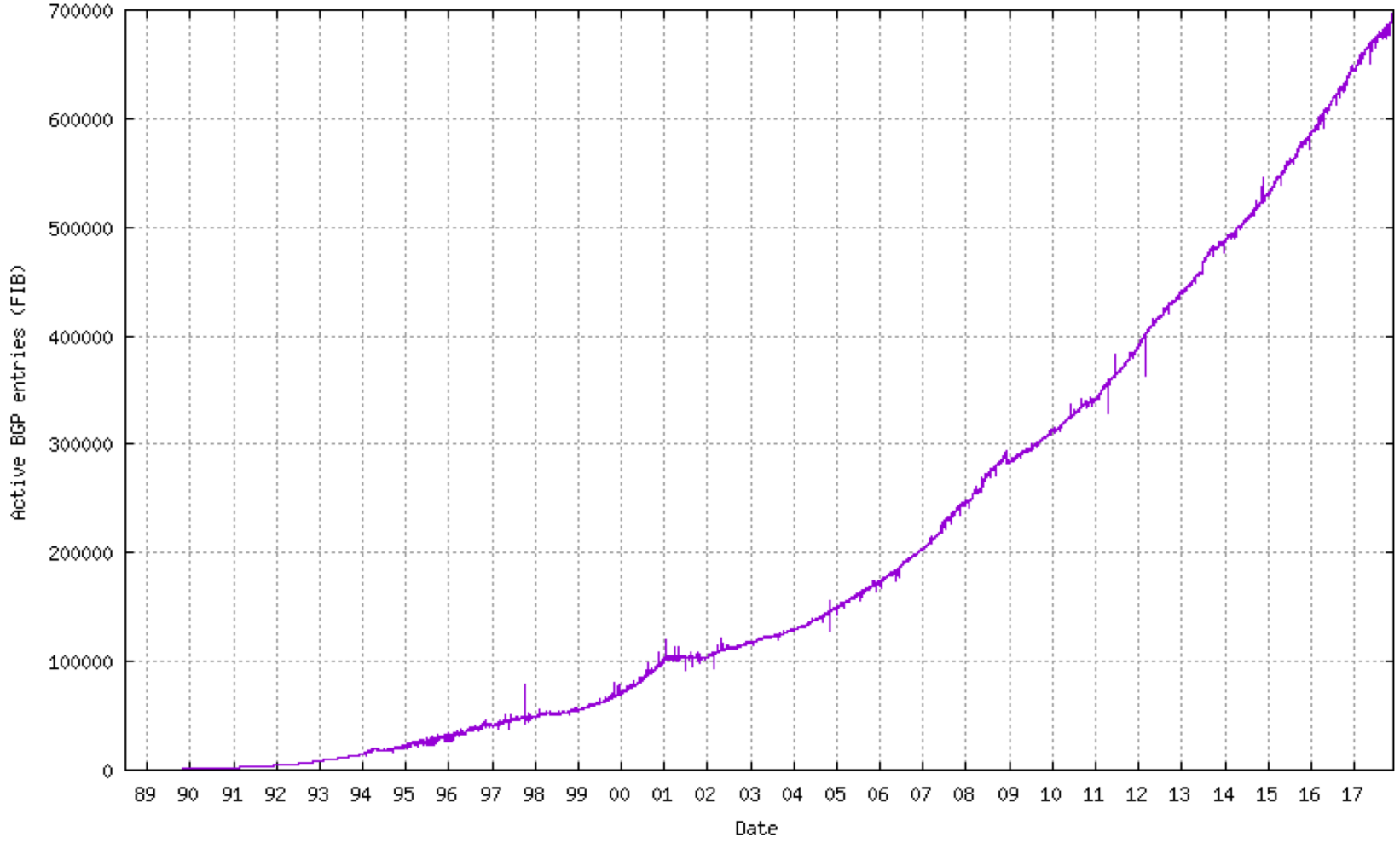
CIDR Aggregation in BGP



Problem at S :Overlapping routing entries.

Solution : Longest prefix matching (LPM)

Routing Table Size of BGP (number of globally advertised, aggregated entries)



Active BGP Entries (http://bgp.potaroo.net/as1221/bgp-active.html)
Internet Growth (http://www.caida.org/research/topology/as_core_network/historical.xml)

IP Addressing: How to Get One?

Q: How does an ISP get its block of addresses?

A: Local Internet Registry (LIR) or National Internet Registry (NIR)

https://www.iana.org/numbers

https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml

Use
%whois <IP address>
to check who is allocated the given address.

IP addresses: How to Get One?

Q: How does a *host* get an IP address?

A:

- Static configured
  - unix:
    %/sbin/ifconfig eth0 inet 192.168.0.10 netmask 255.255.255.0

- DHCP: Dynamic Host Configuration Protocol (RFC2131):
  dynamically get address from a DHCP server

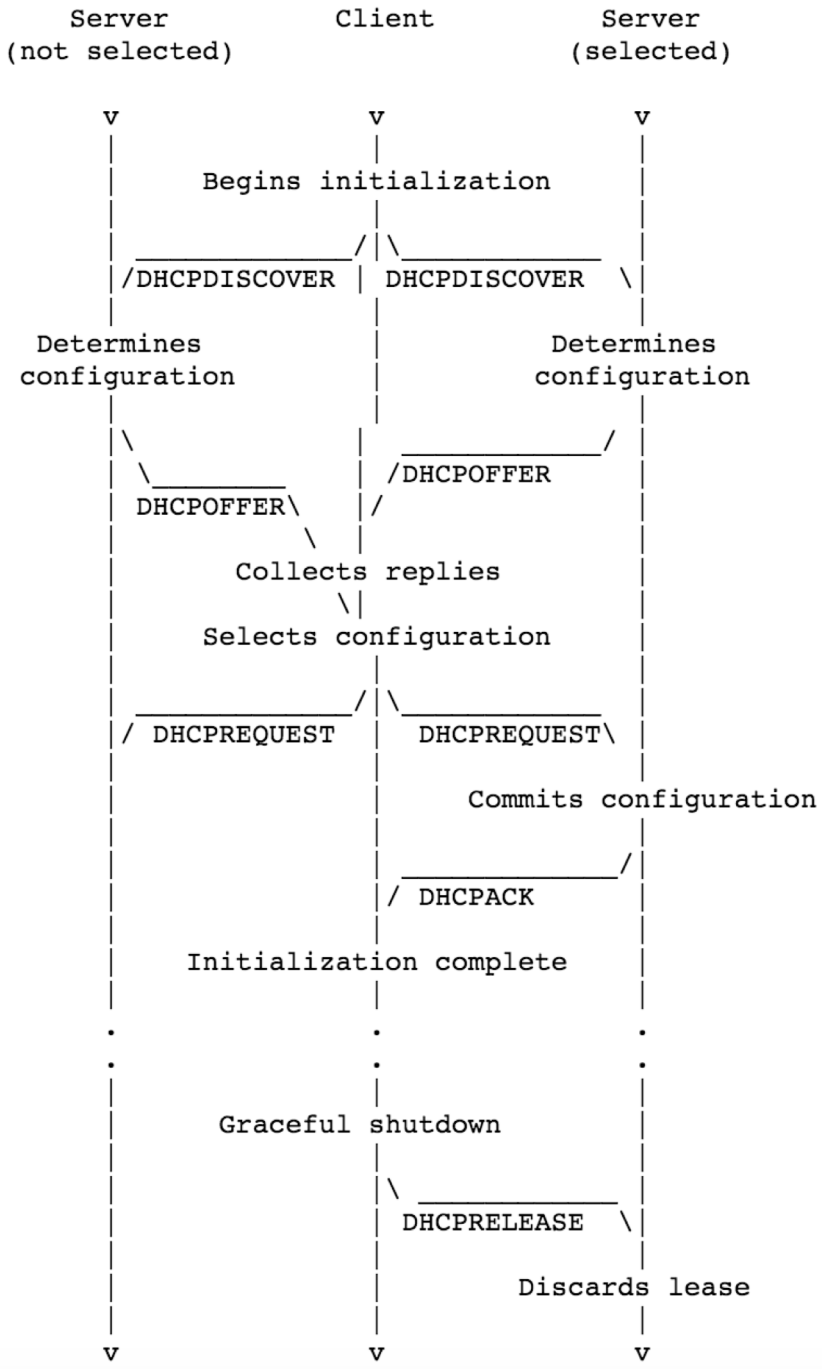
DHCP Goal and History

- Goal: allow host to *dynamically* obtain its IP address from network server when it joins network
- History
  - 1984 Reverse ARP (RFC903): obtain IP address, but at link layer, and hence requires a server at each network link
  - 1985 Bootstrap Protocol (BOOTP; RFC951): introduces the concept of a relay agent to forward across networks
  - 1993 DHCP (RFC1531): based on BOOTP but can dynamically allocate and reclaim IP addresses in a pool, as well as delivery of other parameters
  - 1993 Errors in editorials led to immediate reissue as RFC1541
  - 1997 DHCP (RFC2131): add DHCPINFORM

DHCP: Dynamic Host Configuration Protocol

The often used DORA model (4 messages)

- host broadcasts "DHCP discover" msg
- DHCP server responds with "DHCP offer" msg
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg



https://www.ietf.org/rfc/rfc2131.txt

igure 3: Timeline diagram of messages exchanged between DHCP client and servers when allocating a new network address

DHCPDISCOVER message

UDP Src=0.0.0.0 sPort=68 Dest=255.255.255.255 dPort=67			
OP	**HTYPE**	**HLEN**	**HOPS**
0x01	0x01	0x06	0x00
XID			
0x3903F326			
SECS		**FLAGS**	
0x0000		0x8000	
CIADDR (Client IP address)			
0x00000000			
YIADDR (Your IP address)			
0x00000000			
SIADDR (Server IP address)			
0x00000000			
GIADDR (Gateway IP address)			
0x00000000			
CHADDR (Client hardware address)			
0x00053C04			
0x8D590000			
0x00000000			
0x00000000			
192 octets of 0s, or overflow space for additional options. BOOTP legacy			
Magic cookie			
0x63825363			
DHCP Options			
DHCP option 53: DHCP Discover			
DHCP option 50: 192.168.1.100 requested			
DHCP option 55: Parameter Request List: Request Subnet Mask (1), Router (3), Domain Name (15), Domain Name Server (6)			

DHCPOFFER message

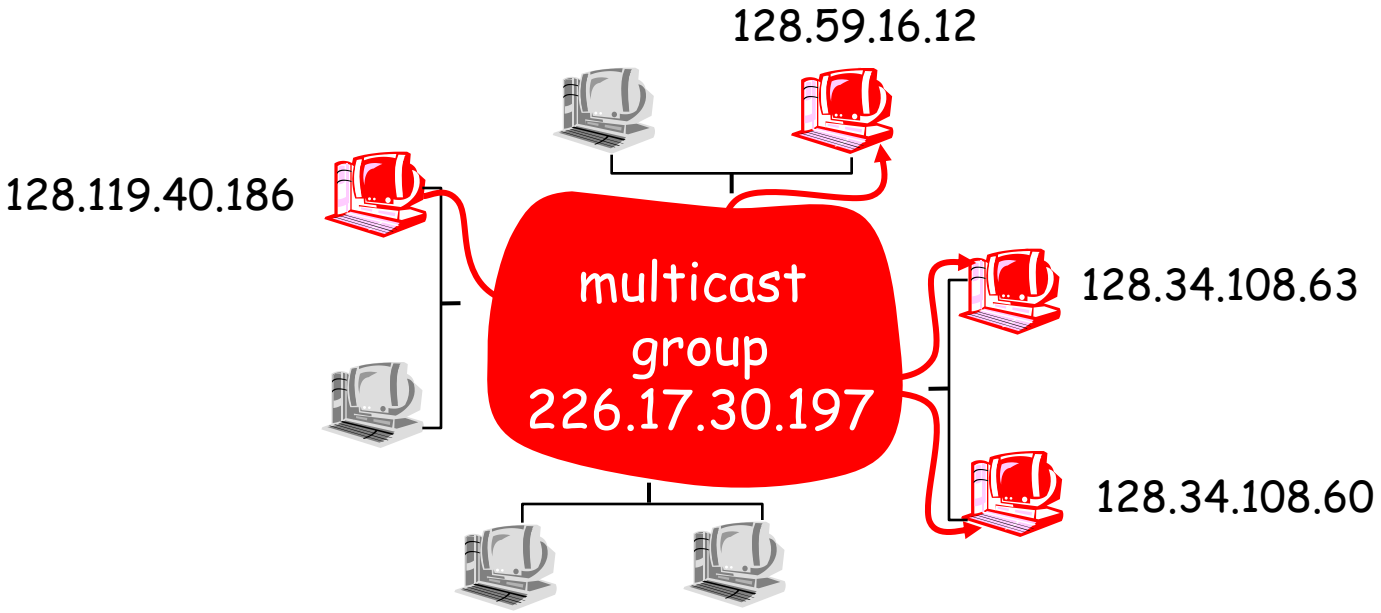
UDP Src=192.168.1.1 sPort=67 Dest=255.255.255.255 l dPort=68			
OP	**HTYPE**	**HLEN**	**HOPS**
0x02	0x01	0x06	0x00
XID			
0x3903F326			
SECS		**FLAGS**	
0x0000		0x0000	
CIADDR (Client IP address)			
0x00000000			
YIADDR (Your IP address)			
0xC0A80164 (This translates to 192.168.1.100)			
SIADDR (Server IP address)			
0xC0A80101 (This translates to 192.168.1.1)			
GIADDR (Gateway IP address)			
0x00000000			
CHADDR (Client hardware address)			
0x00053C04			
0x8D590000			
0x00000000			
0x00000000			
192 octets of 0s. BOOTP legacy			
Magic cookie			
0x63825363			
DHCP Options			
DHCP option 53: DHCP Offer			
DHCP option 1: 255.255.255.0 subnet mask			
DHCP option 3: 192.168.1.1 router			
DHCP option 51: 86400s (1 day) IP address lease time			
DHCP option 54: 192.168.1.1 DHCP server			
DHCP option 6: DNS servers 9.7.10.15, 9.7.10.16, 9.7.10.18			

Exercise

- DHCP lease renew

Optional Read: IP Multicast

IP Multicast: Service Model



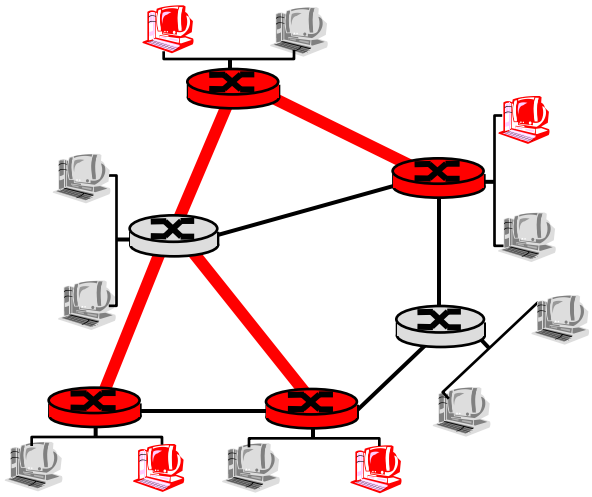
- Multicast group concept: use of indirection
  - A group is identified by a location-independent logical address (class D IP address: prefix 1110)

- Open group model
  - Anyone can send packets to the "logical" group address
  - Anyone can join a group and receive packets
- Normal, best-effort delivery semantics of IP

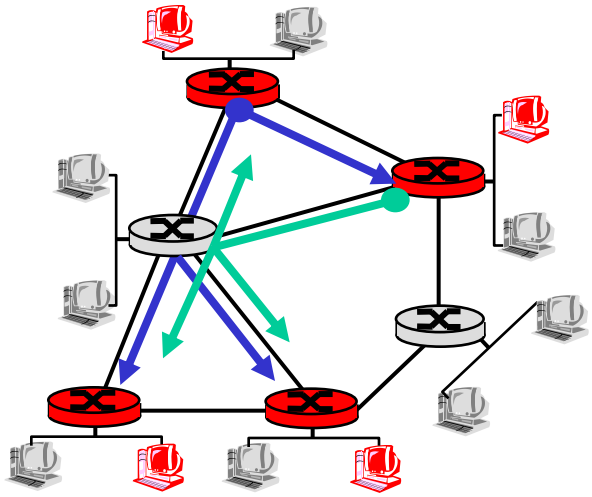
Needed: infrastructure to deliver mcast-addressed datagrams to all hosts that have joined that multicast group

Multicast Across LANs

- *Goal:* find a tree (or trees) connecting routers having local mcast group members
  - *source-based:* different tree from sender to each receiver
    - Distance-vector multicast routing protocol (DVMRP)
    - Protocol-independent multicast-dense mode (PIM-DM)
  - *shared-tree:* same tree used by all group members
    - Core-Based Tree (CBT)
    - Protocol-independent multicast-sparse mode (PIM-SM)



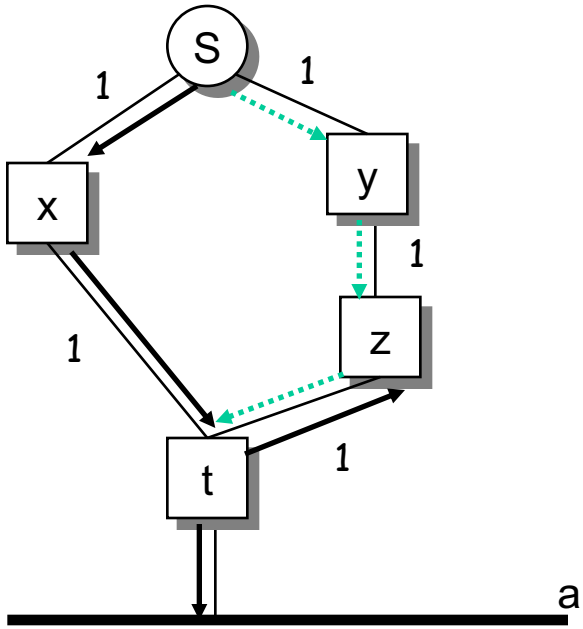
shared tree



source-based trees

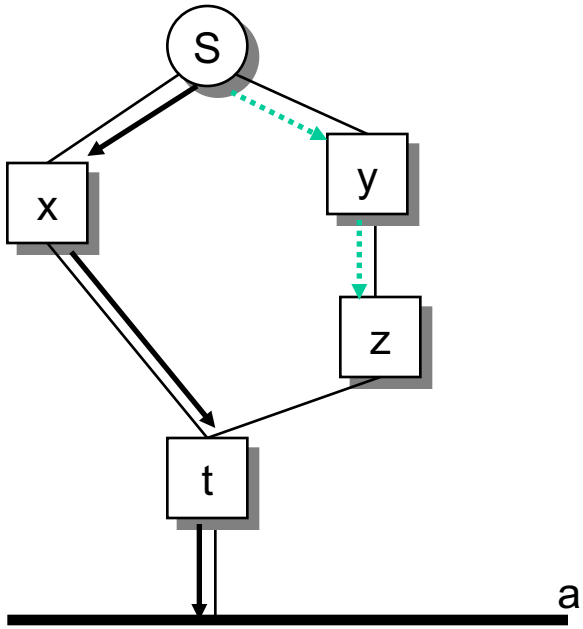
Source Tree: Reverse Path Flooding (RPF)

- A router x forwards a packet from source (S) iff it arrives via neighbor y, and y is on the shortest path from x back to S
- A packet is replicated to all but the incoming interface



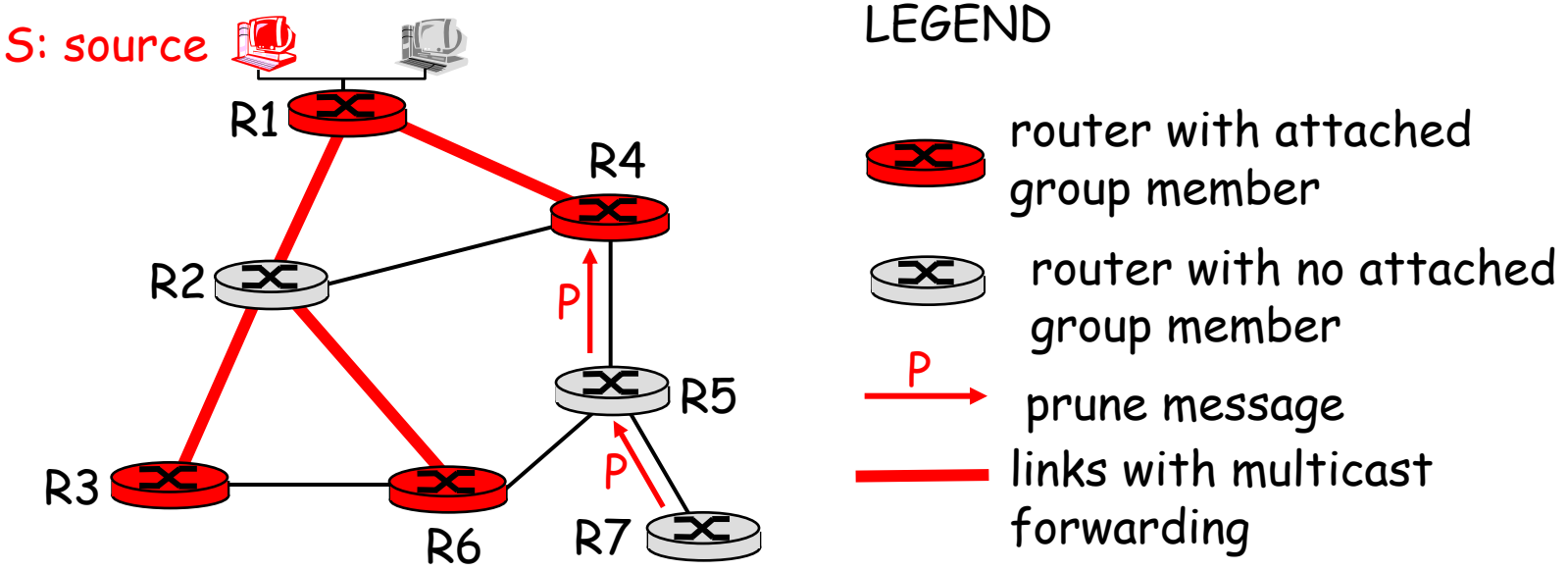
Reverse Path Forwarding: Improvement

- Basic idea: forward a packet from S only on child links for S
- A child link of router x for source S
  - a link that has x as parent on the shortest path from the link to S
  - a child x notifies its parent y (through the routing protocol) that it has selected y as its parent



Reverse Path Forwarding: Pruning

- No need to forward datagrams down subtree with no mcast group members
- "prune" msgs sent upstream by router with no downstream group members



Pruning

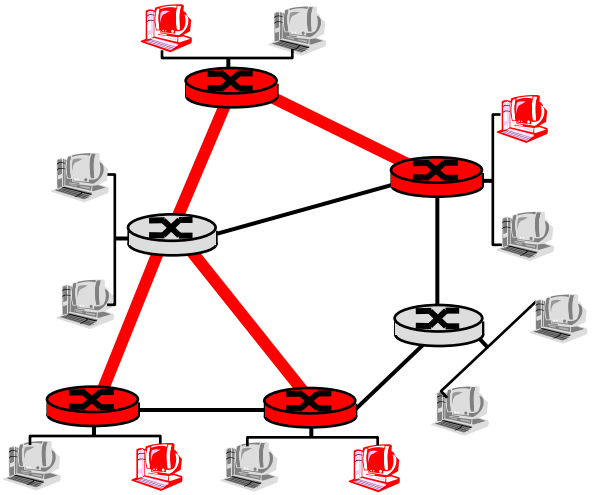
- Prune (Source, Group) at a leaf router if no members
  - send No-Membership Report (NMR) up tree
- If all children of router R prune (S,G)
  - propagate prune for (S,G) to its parent
- What do you do when a member of a group (re)joins?
  - send a Graft message to upstream parent
- How to deal with failures?
  - prune dropped
  - flow is reinstated
  - down stream routers re-prune
- Note: again a soft-state approach

Implementation of Source Trees in the Internet

- Multicast OSFP (MOSFP)
  - Membership is part of the link state distribution; calculate source specific, pre-pruned trees
- Reverse Path Forwarding
  - Distance Vector Multicast Routing Protocol (DVMRP)
  - Protocol Independent Multicast – Dense Mode (PIM-DM)
    - very similar to DVMRP
  - Difference: PIM uses any unicast routing algorithm to determine the path from a router to the source; DVMRP uses distance vector
  - Question: the state requirement of Reverse Path Forwarding

Building a Shared Tree

- Steiner Tree: minimum cost tree connecting all routers with attached group members
- A Steiner tree is not a spanning tree because you do not need to connect all nodes in the network
- Problem is NP-hard
- Excellent heuristics exists
- Not used in practice:
  - computational complexity
  - information about entire network needed
  - monolithic: rerun whenever a router needs to join/leave

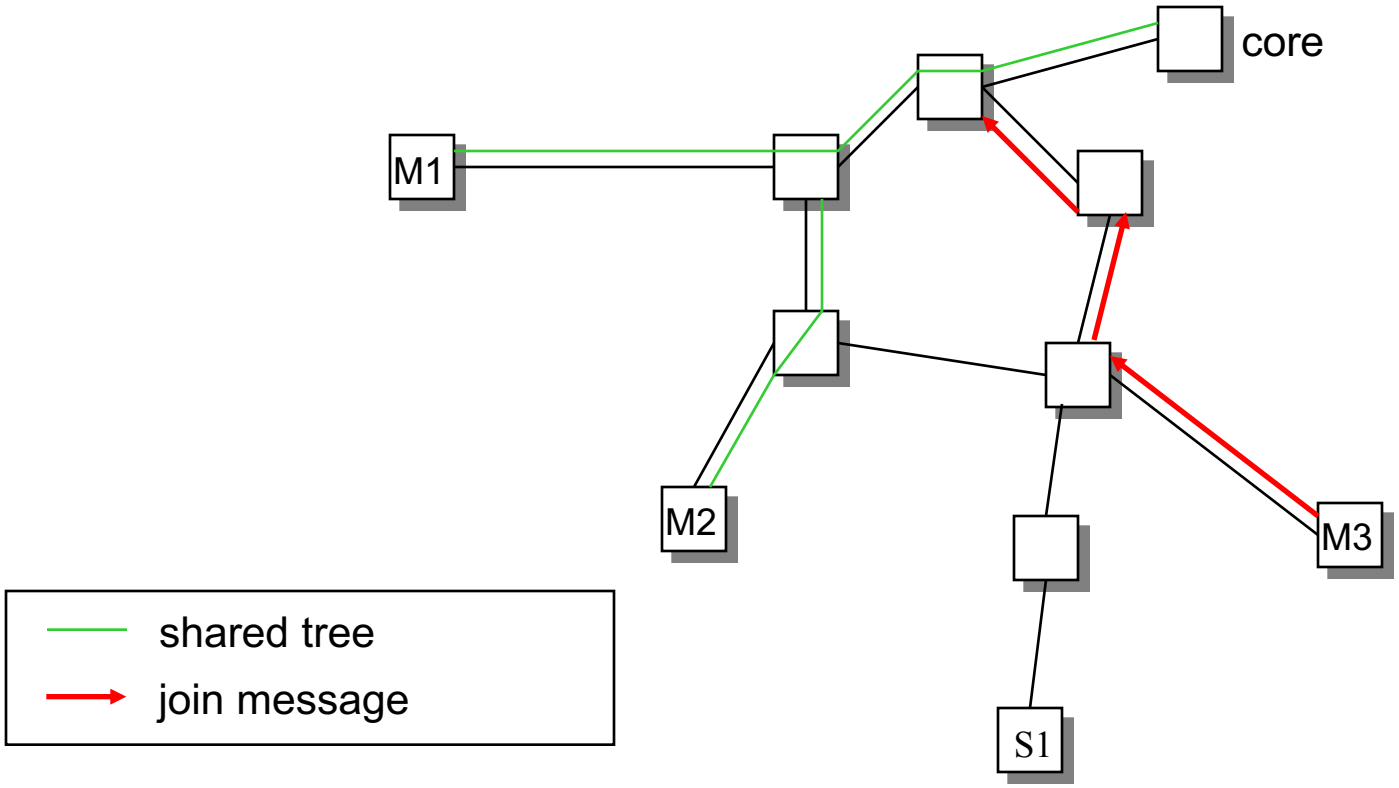


Center (Core) based Shared Tree

- Single delivery tree shared by all
- One router identified as *"center"* of tree
- Tree construction is receiver-based
  - edge router sends unicast *join-msg* addressed to center router
  - *join-msg* "processed" by intermediate routers and forwarded towards center
  - *join-msg* either hits existing tree branch for this center, or arrives at center
  - path taken by *join-msg* becomes new branch of tree for this router
- A sender unicasts a packet to center
  - The packet is distributed on the tree when it hits the tree

Example: M3 Joins

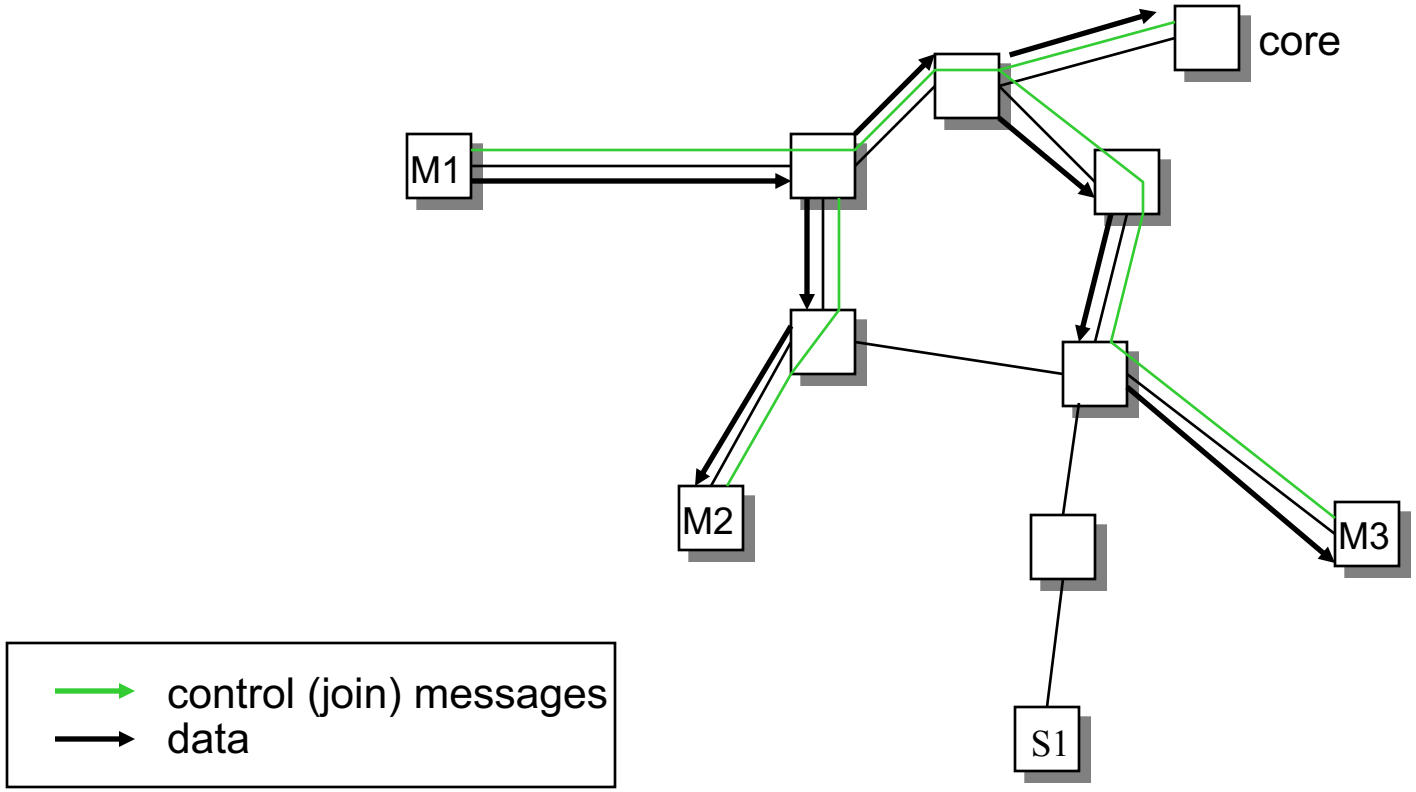
- Group members: M1, M2



Discussion: what is property of the constructed tree?

Example: M1 Sends Data

- Group members: M1, M2, M3
- M1 sends data

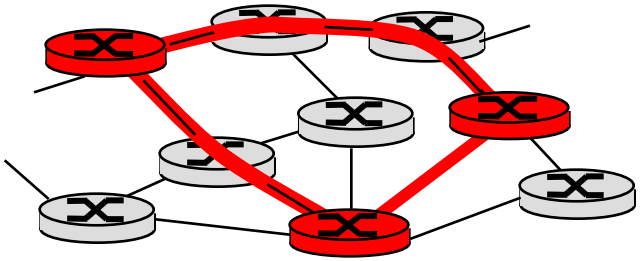


Shared Tree Protocols in the Internet

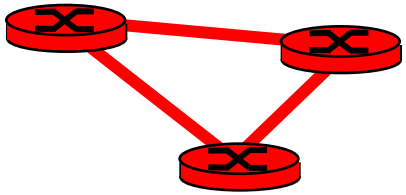
- Core Based Tree
- Protocol Independent Multicast (PIM) Sparse mode
- The catch: how do you know the center?
  - session announcement

Mbone: Tunneling

Q: How to connect "islands" of multicast routers in a "sea" of unicast routers?



physical topology



logical topology

- mcast datagram encapsulated inside "normal" (non-multicast-addressed) datagram
- normal IP datagram sent thru "tunnel" via regular IP unicast to receiving mcast router
- receiving mcast router unencapsulates to get mcast datagram