Network:
Policy Routing Analysis, DHCP

Y. Richard Yang

http://zoo.cs.yale.edu/classes/cs433/

12/04/2018

Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing protocols (distance vector, link state)
      - Global Internet routing
        - Basic architecture
        - Policy routing analysis
    - Address assignment

Admin

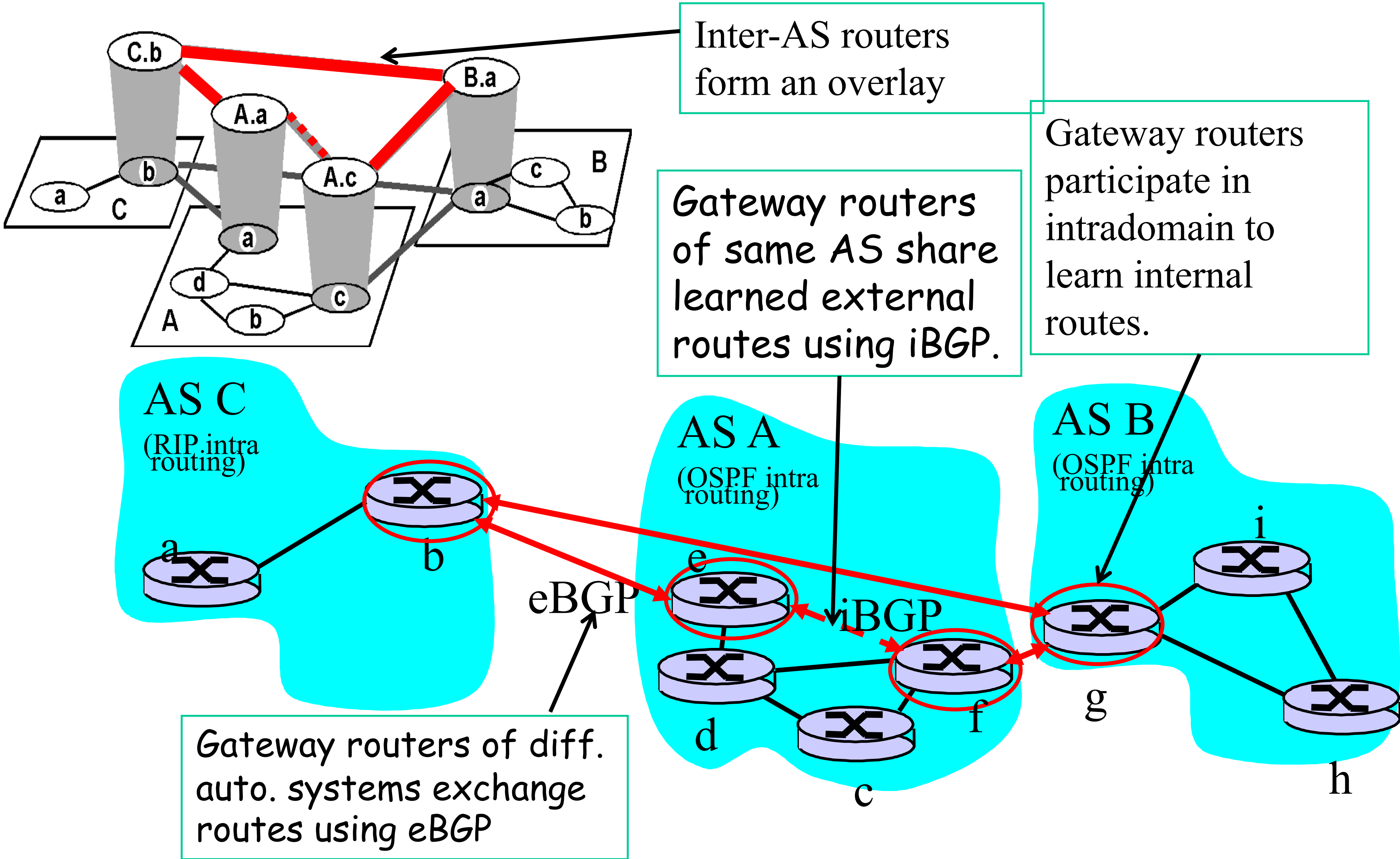
- Assignment five (written assignment) questions

- Exam 2 reminder: 7:00-8:30 pm Dec. 11 at AKW 200 and AKW 300

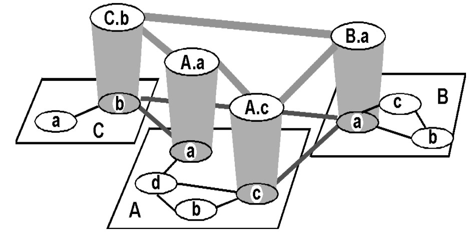
Recap: Basic Routing Protocols

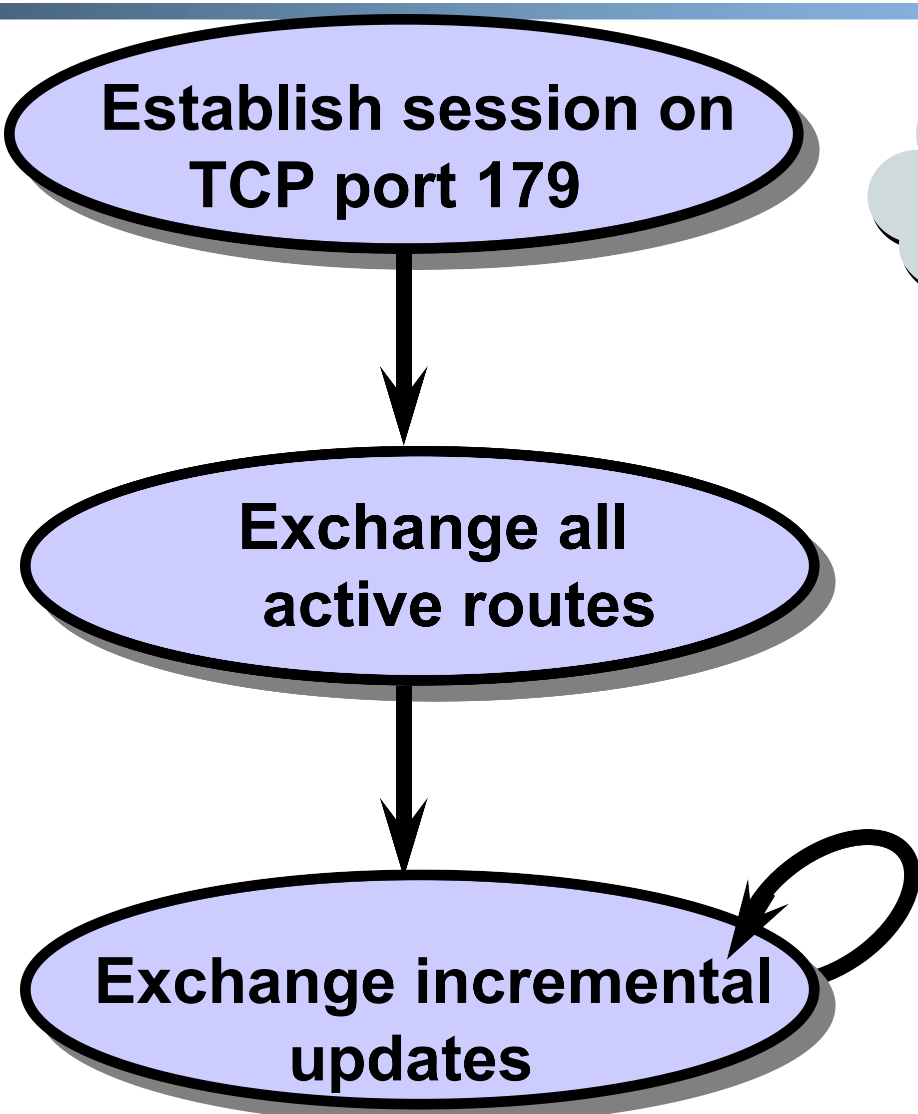
- Safe, Distributed DV protocols:
  - Issue: Counting-to-infinity
  - Basic idea: use local, sufficient conditions to enforce global no loop condition
  - Analytical techniques: use global invariants to gain understanding of distributed protocols
- Link state protocols
  - Instead of distributed computing, use distributed state synchronization to avoid the churns of distributed computation

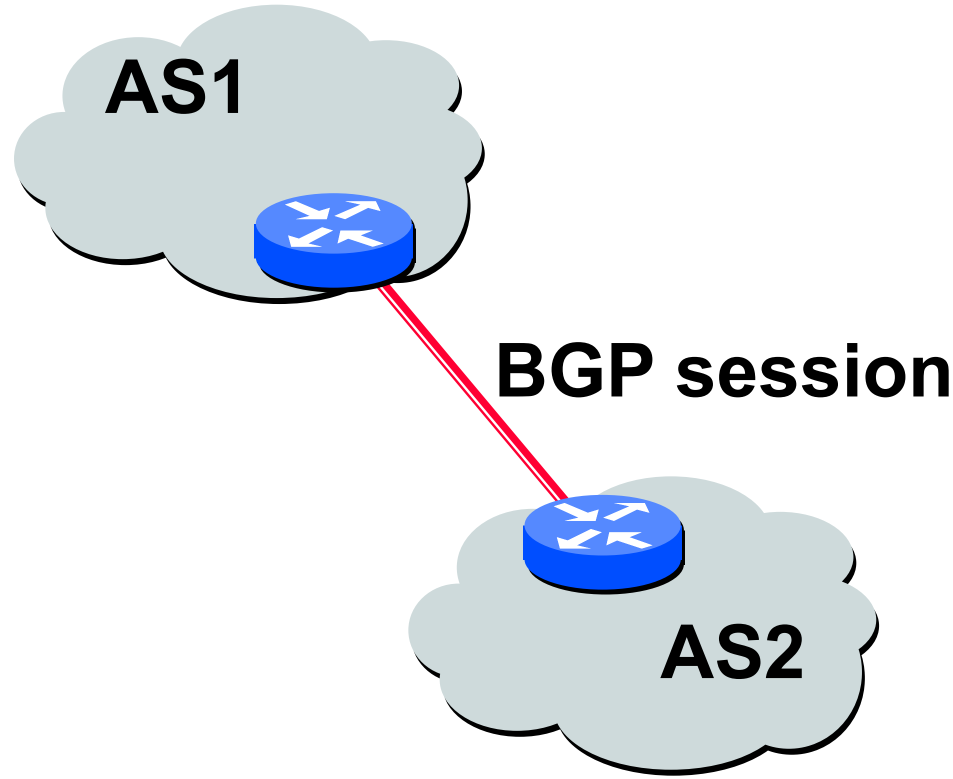
Recap: Global Internet Routing Architecture



BGP Basic Operations





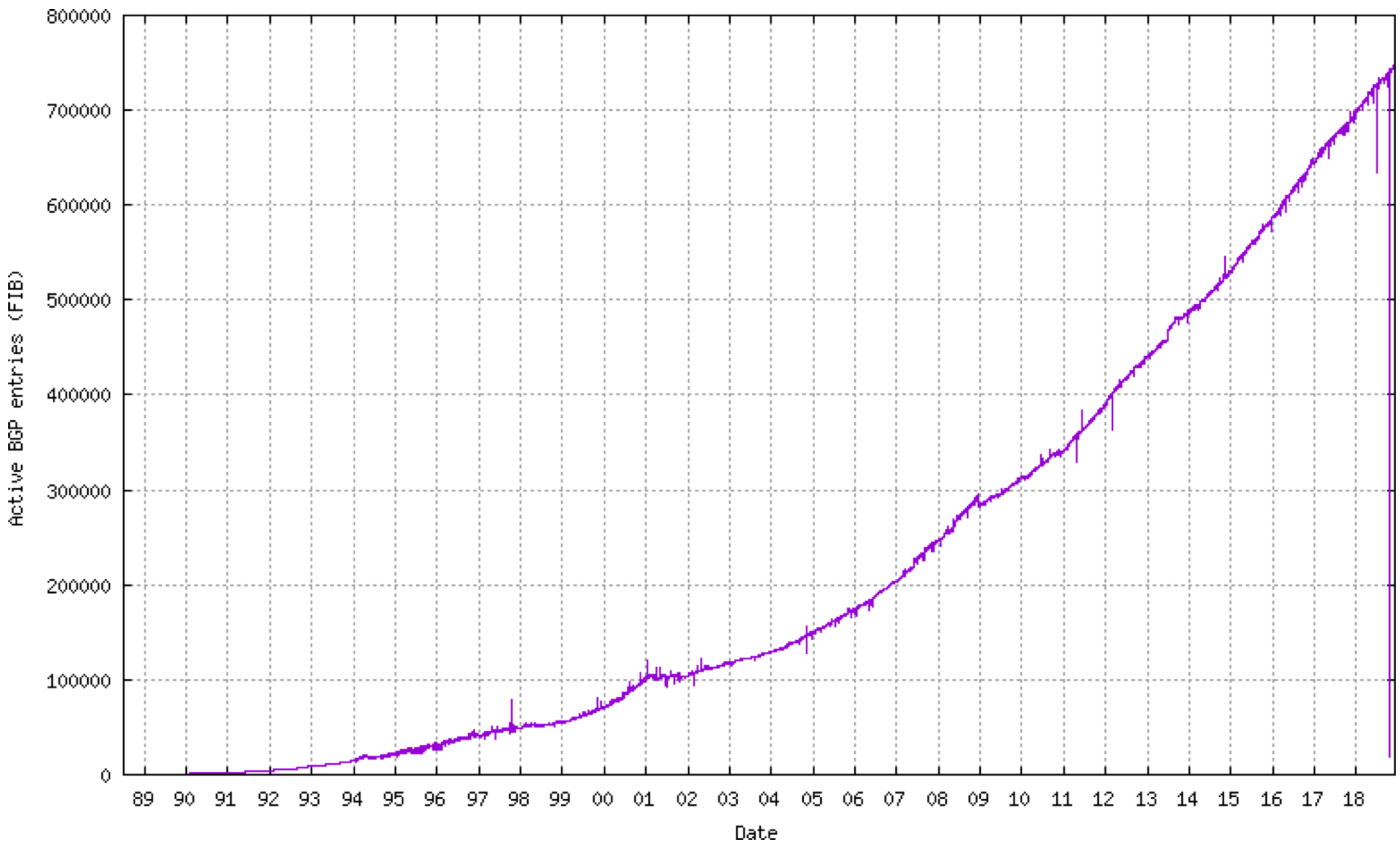


while (connection is ALIVE)
  exchange UPDATE message
  select best available route
  if route changes, export to neigh.

BGP Messages

- Four types of messages
  - OPEN: opens TCP connection to peer and authenticates sender
  - UPDATE: advertises new path vectors (or withdraws old)
  - KEEPALIVE keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

Recap: CIDR Address Aggregation to Reduce # Detinations

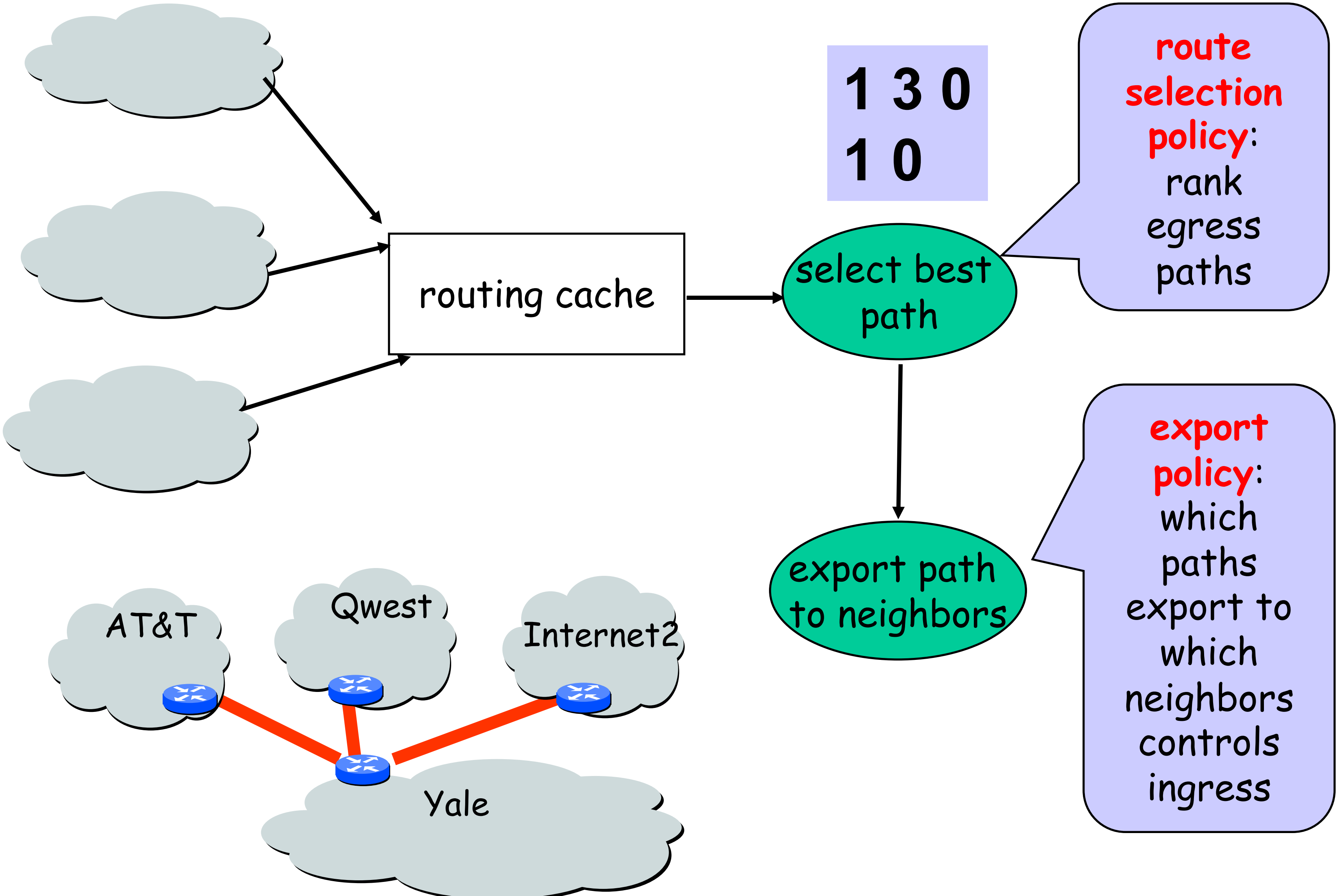


Active BGP Entries (http://bgp.potaroo.net/as1221/bgp-active.html)
Internet Growth (http://www.caida.org/research/topology/as_core_network/historical.xml)

Recap: Features of Global Routing Architecture Design

- Scalability
  - Only a small # of routers (gateways) from each AS in the interdomain level
  - CIDR aggregation reduces amt data to be carried
- Privacy
  - Interdomain routing carries only path vector, not internal network path
- Autonomy
  - Autonomous systems have flexibility to choose their own intradomain routing protocols
  - Each network chooses interdomain path according to its own policy

Recap: BGP as a Policy Routing Framework



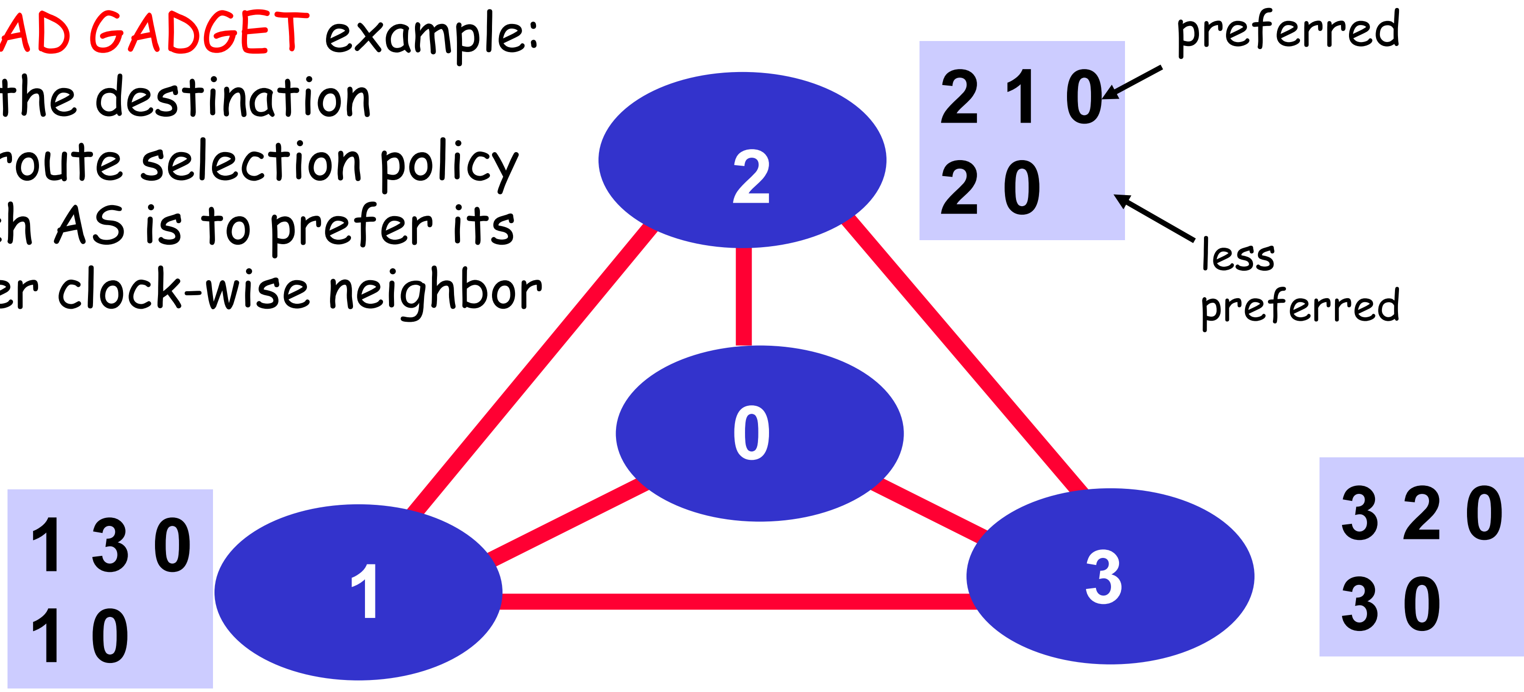
Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing protocols (distance vector, link state)
      - Global Internet routing
        - Basic architecture
        - Policy routing analysis

Policy Routing Instability

- 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



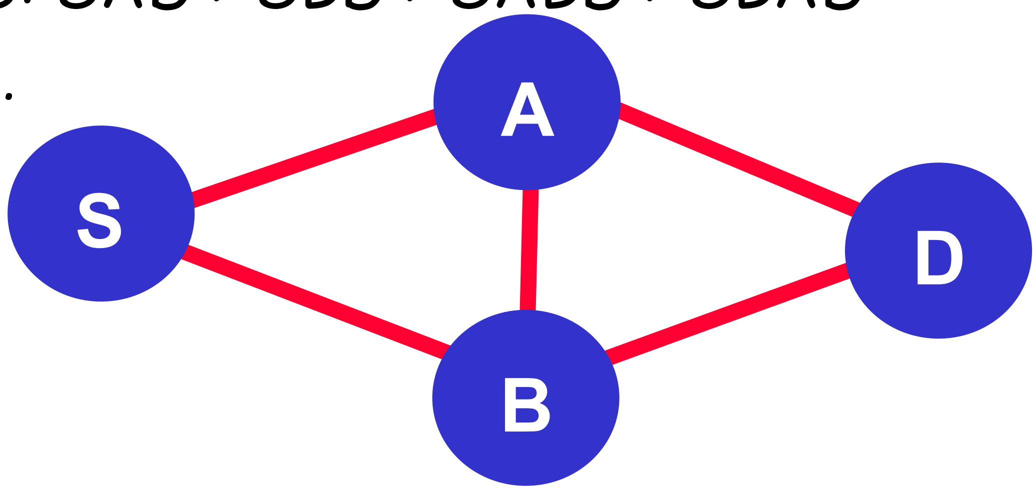
Policy (preferences) aggregation fails: routing instability !

General Framework of Preference Aggregation

- Also called Social Choice
  - Given individual preferences, define a framework (constitution) 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 > c > b

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
  - …



Global Aggregation Framework

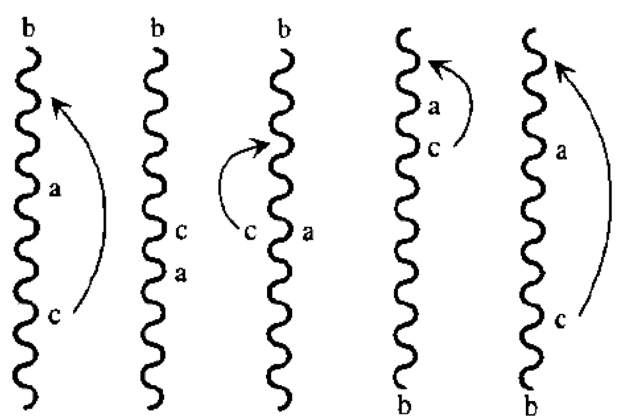
- Axioms:
  - Transitivity
    - if a > b & b > c, then a > c
  - Unanimity:
    - If all participants prefer a over b (a > b) => a > b
  - Independence of irrelevant alternatives (IIA)
    - Social ranking of a and b depends only on the relative ranking of a and b among all participants
- Result:
  - Arrow's Theorem: Any constitution (protocol) that respects transitivity, unanimity and IIA must be 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 constitution 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 constitution has a >= b; b >= c.
    - We can move c above a w/o changing a-b or b-c votes



Step 1: Existence of Pivotal Voter

- Let choice b be chosen arbitrarily. There exists 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*

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 society 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

Profile I

1	2	n*	.	N
b	b	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	b	b	b

constitution: b bottom

Profile II

1	2	n*	.	N
b	b	b	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	b	b

constitution : b top

Profile III

1	2	n*	.	N
b	b	a	.	.
.	.	b	.	.
.	.	.	.	.
.	.	c	b	b

constitution : $a > b$ since ab same as I

$b > c$ since bc same as II

Step 3: n* is dictator for every pair ab

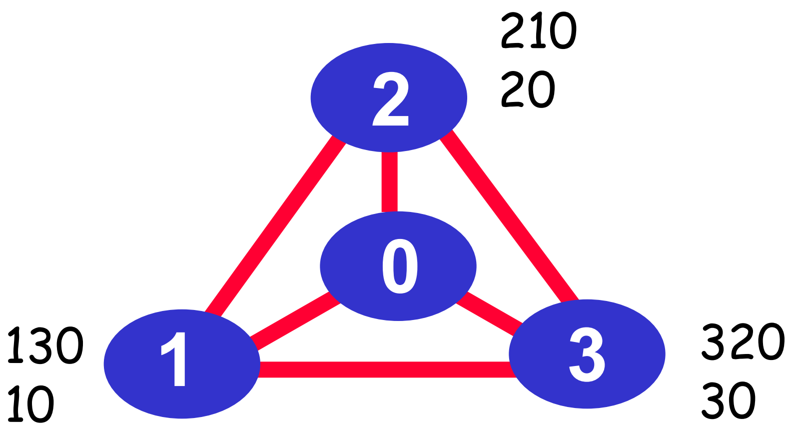
- Consider c not equal to a or b
- There exists n(c) who is a dictator of any pair not involving c, such as the pair ab, i.e.,
  - For any profile, if $a >_{n(c)} > b$, $a > b$ for society
- n(c) must be n*
  - Assume not.
    - Consider Profile I and Profile II.
    - Since n(c) is not n*, n(c) ranking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

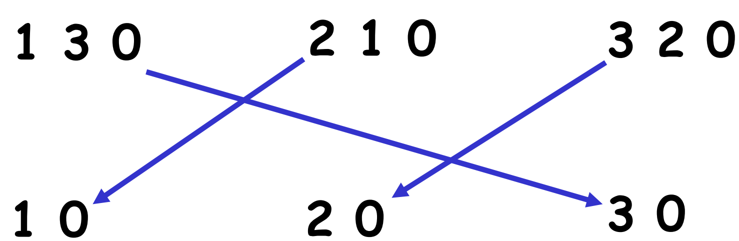
Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing protocols (distance vector, link state)
      - Global Internet routing
        - Basic architecture
        - Policy routing analysis
          - Issue: Instability
          - Global aggregation and dictatorship
          - Local aggregation and P-graph

BGP w/ Local Preference

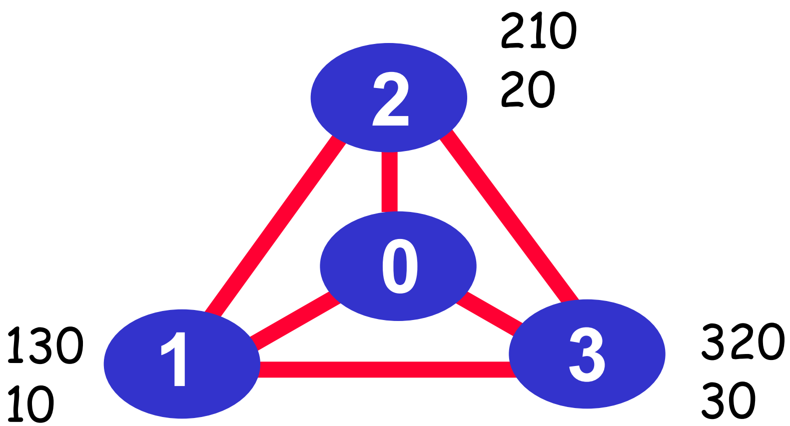
- BGP preferences are typically local, on egress routing (only on paths starting from itself)
- Hence the preferences have dependency (priority)
  - The "closer" a node to the destination, the more "powerful" it may be

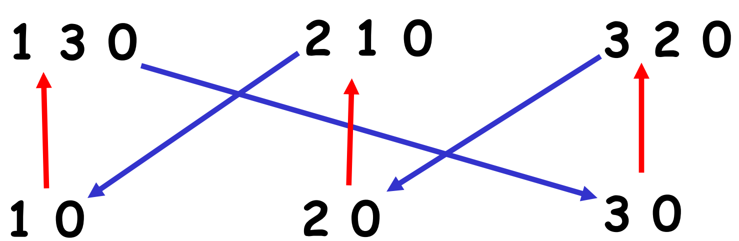




Complete Dependency: P-Graph

- 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

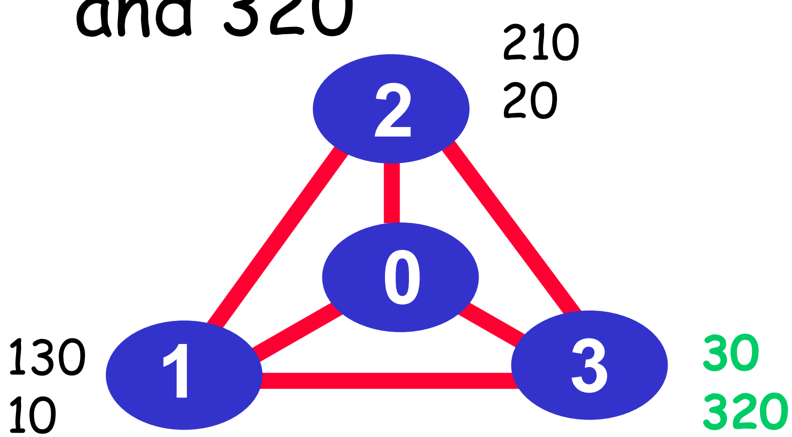


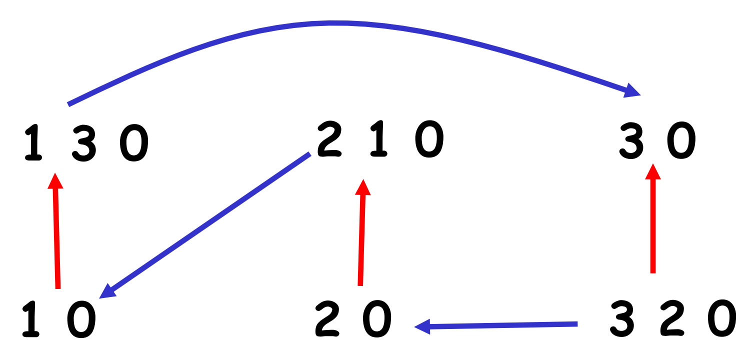


Any observation on the P-graph?

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





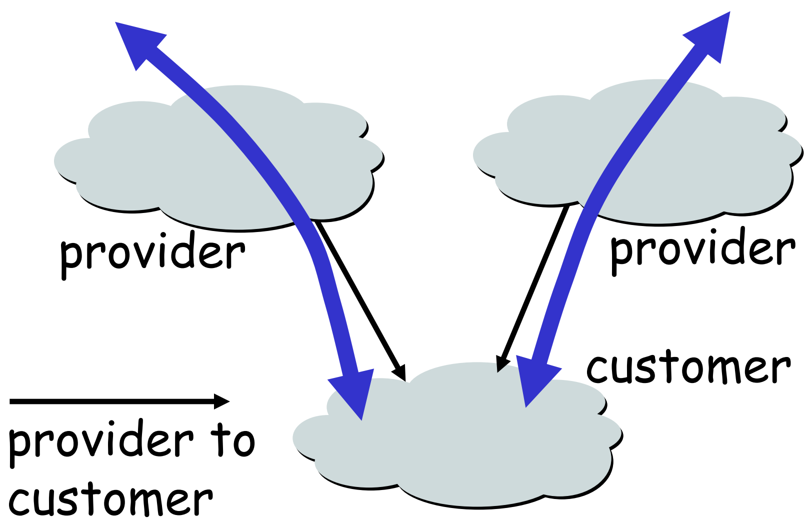
Outline

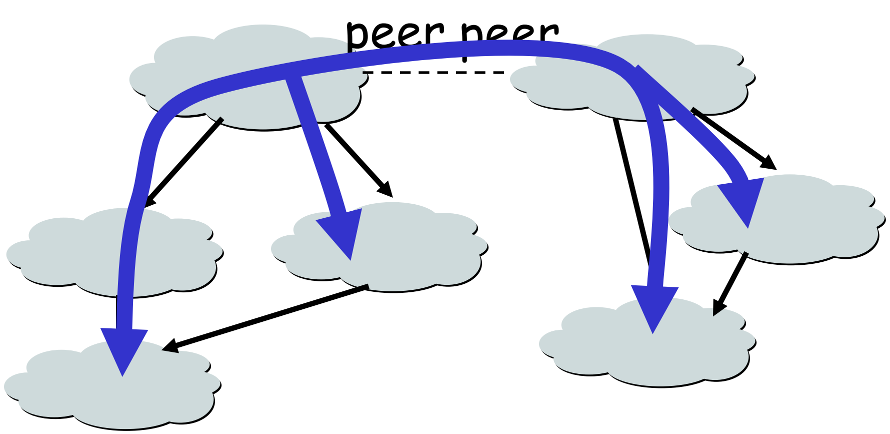
- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing protocols (distance vector, link state)
      - Global Internet routing
        - Basic architecture
        - Policy routing analysis
          - Issue: Instability
          - Global aggregation and dictatorship
          - Local aggregation and P-graph
          - ***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

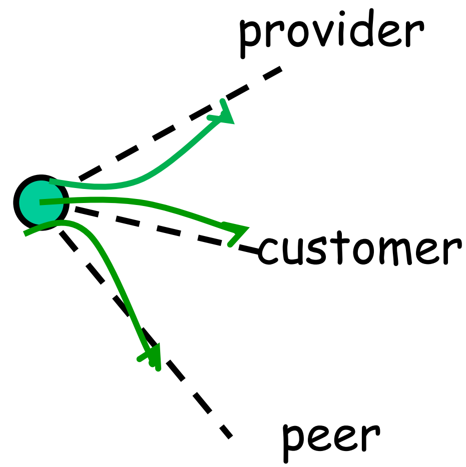
- *Peer-to-peer relationship*
  - mutually agree to exchange traffic between their respective **customers** only
  - there is no payment between peers





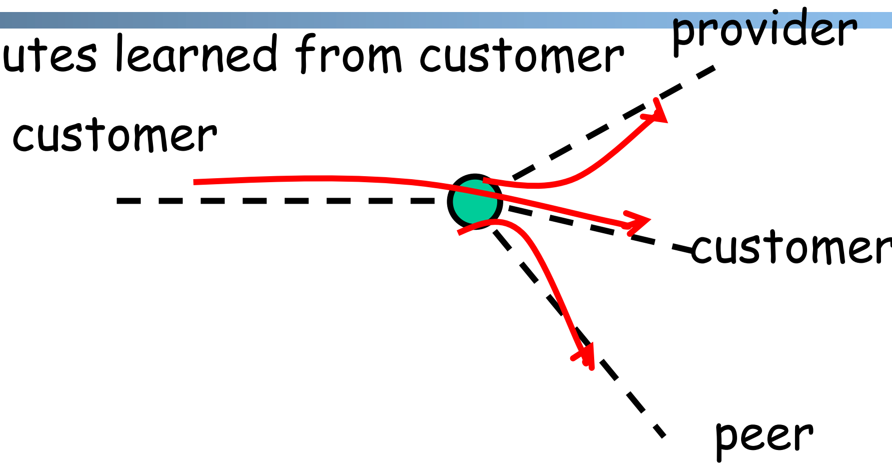
Route Selection Policies and Economics

- 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



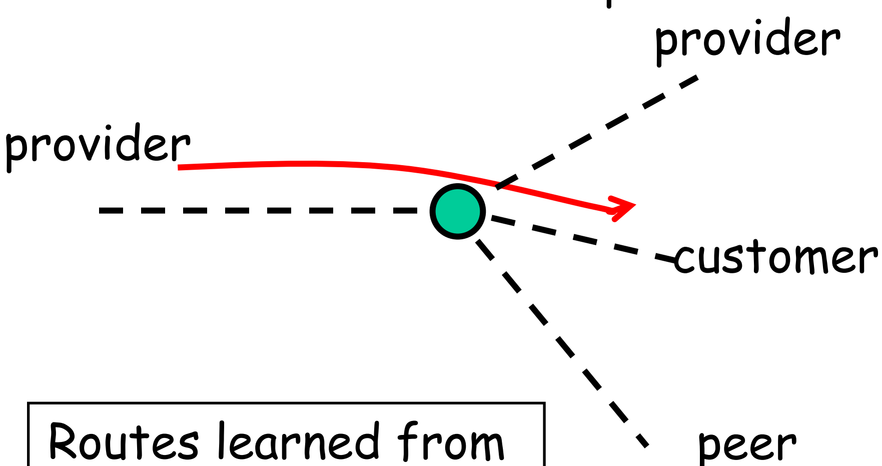
Export Policies and Economics

case 1: routes learned from customer



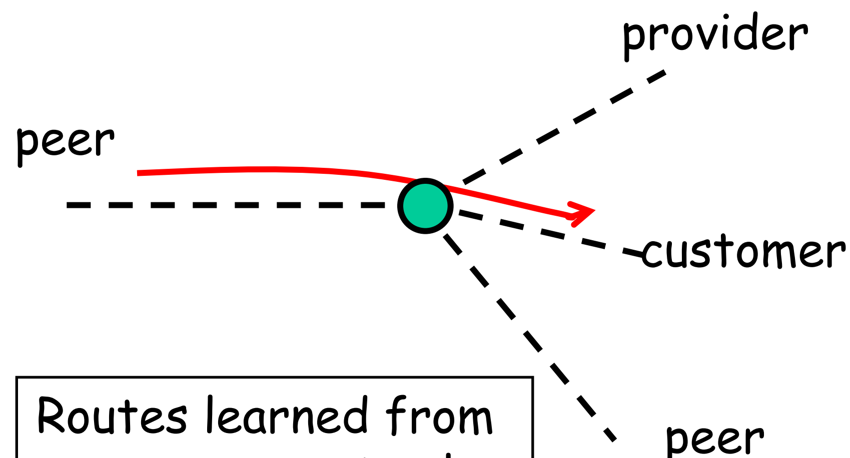
Routes learned from a customer are sent to all other neighbors

case 2: routes learned from provider



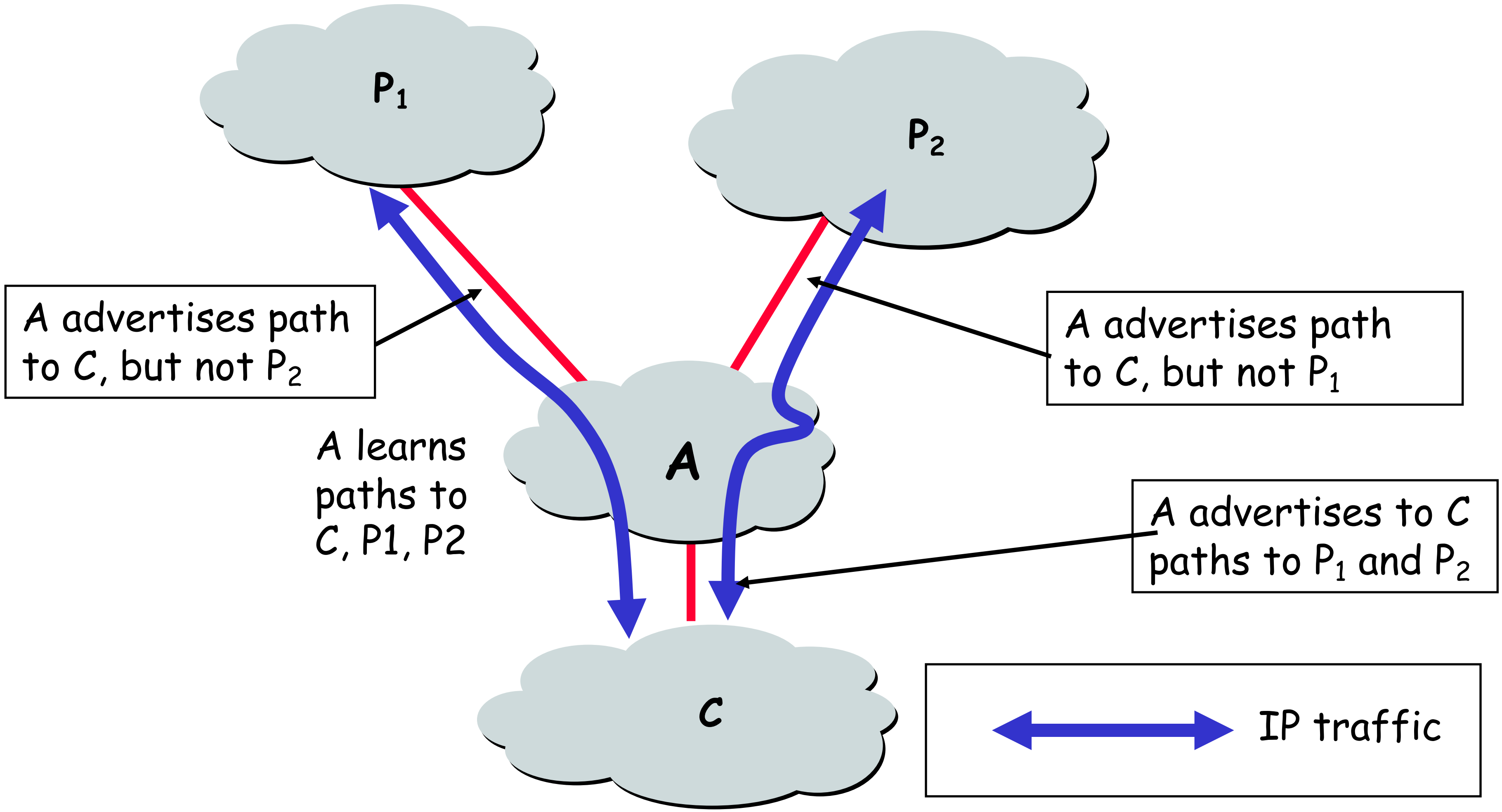
Routes learned from a provider are sent only to customers

case 3: routes learned from peer



Routes learned from a peer are sent only to customers

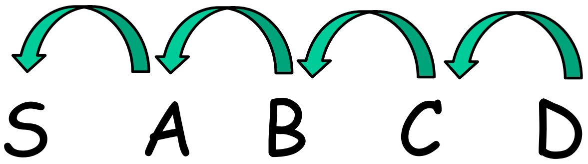
Example: Typical Export -> No-Valley Routing



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

Typical Export Policies Route Patterns

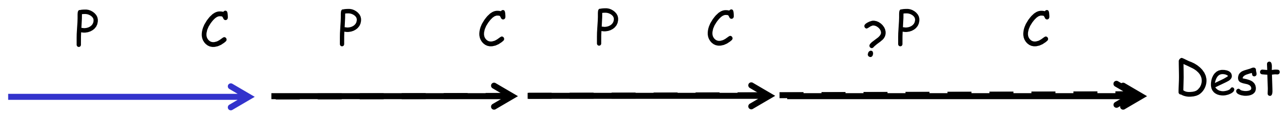
- Assume a BGP path SABCD to destination AS D. Consider the business relationship between each pair:



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

Typical Export Policies Route Patterns

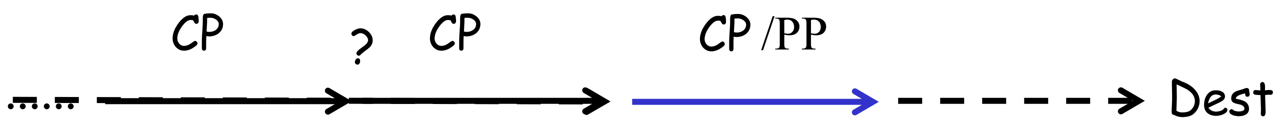
- Invariant 1 of valid BGP routes (with labels representing business relationship)



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)



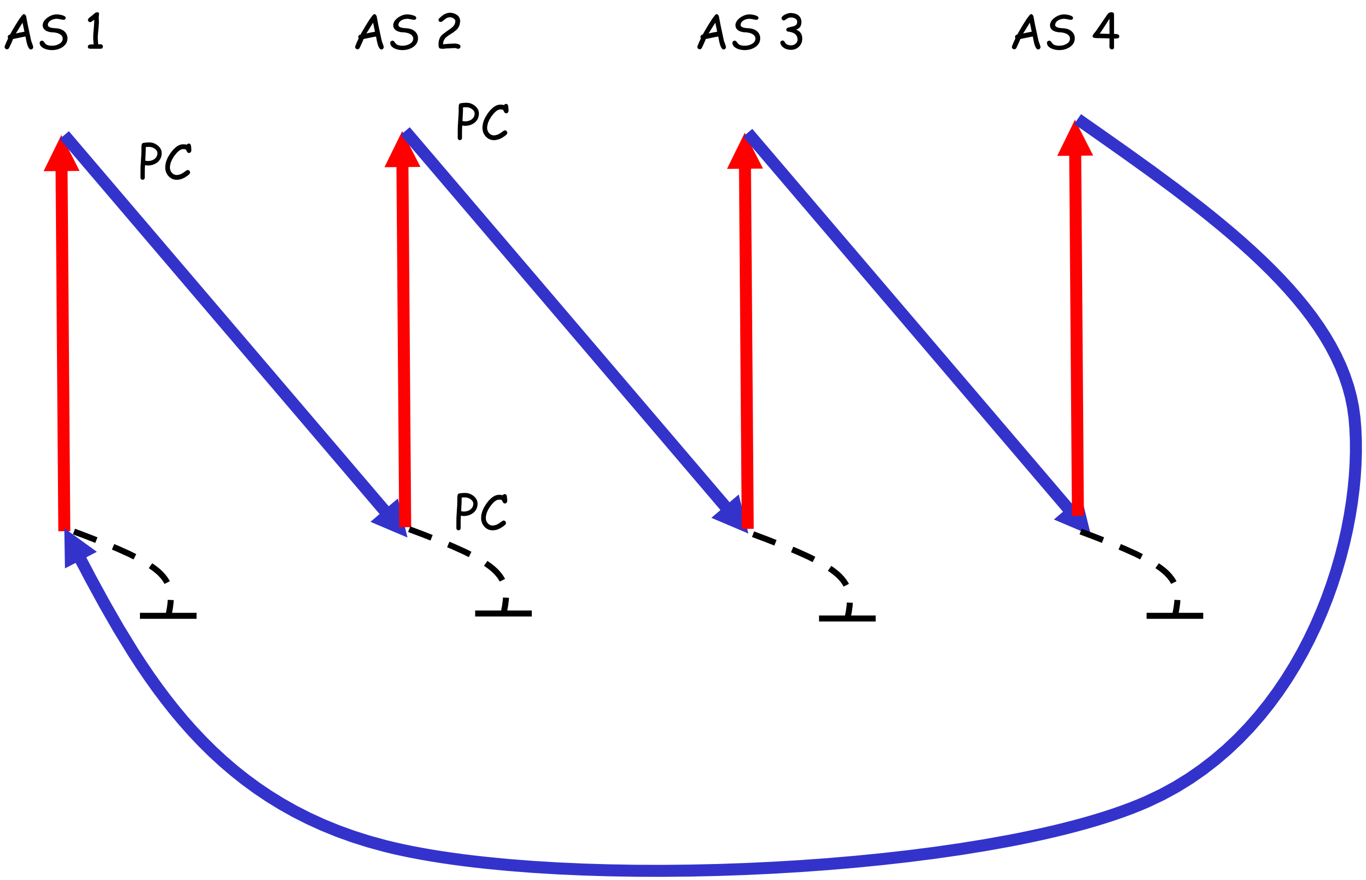
Reasoning: routes learned from peer or provider are sent to only customers; thus all relationship before is CP.

Stability of BGP Policy Routing

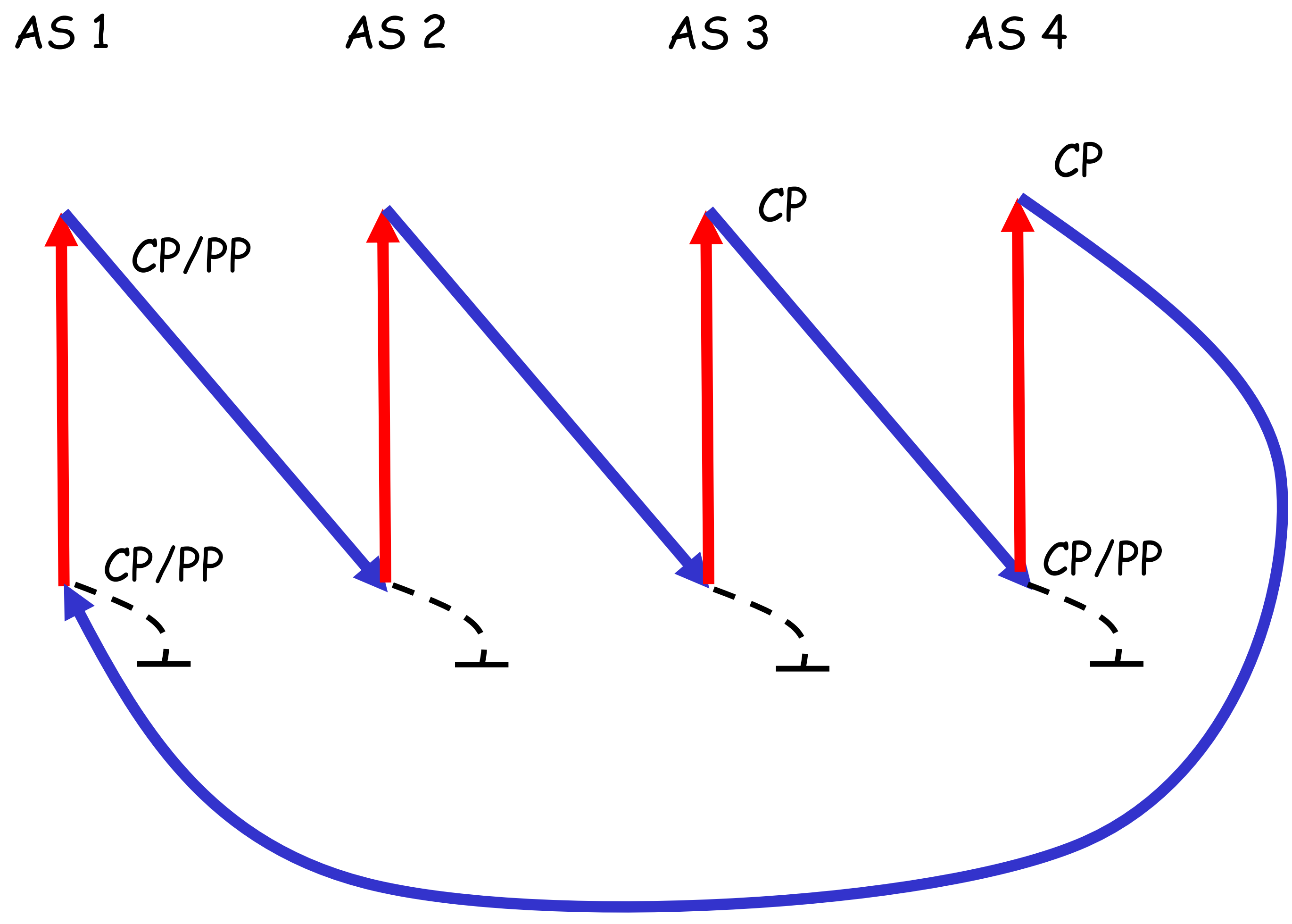
- Suppose
  1. there is no loop formed by provider-customer relationship in the Internet
  2. each AS uses typical route selection policy: C > E/P
  3. each AS uses the typical export policies
- Then policy routing always converges (i.e., is stable).

Case 1: A Link is PC

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

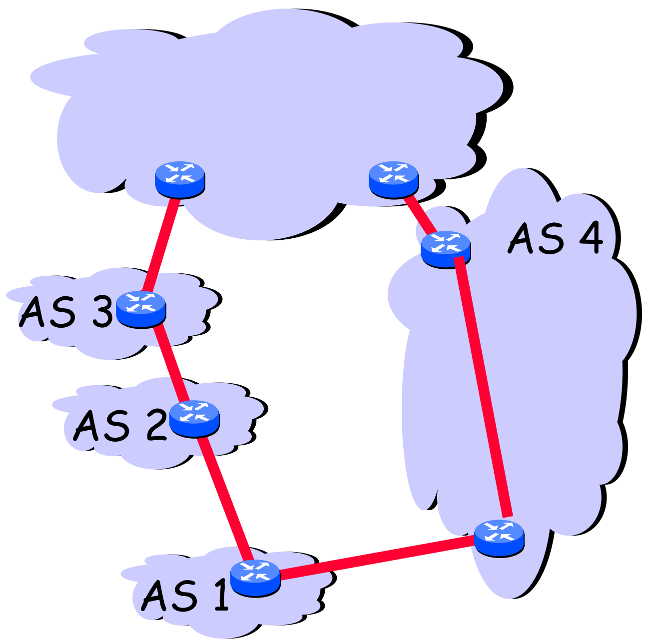


Case 2: Link is CP/PP



Summary: BGP Policy Routing 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



Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing protocols (distance vector, link state)
      - Global Internet routing
        - Basic architecture
        - Policy routing analysis
  - Address assignment

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

Exercise

- DHCP wireshark example

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

```
                Server          Client          Server
            (not selected)                    (selected)

                  v               v               v
                  |               |               |
                  |     Begins initialization     |
                  |               |               |
                  | _____________/|\____________  |
                  |/DHCPDISCOVER | DHCPDISCOVER  \|
                  |               |               |
              Determines          |          Determines
             configuration        |         configuration
                  |               |               |
                  |\             |  ____________/ |
                  | \________    | /DHCPOFFER     |
                  | DHCPOFFER\   |/               |
                  |           \  |                |
                  |       Collects replies        |
                  |             \|                |
                  |     Selects configuration     |
                  |               |               |
                  | _____________/|\____________  |
                  |/ DHCPREQUEST  |  DHCPREQUEST\ |
                  |               |               |
                  |               |     Commits configuration
                  |               |               |
                  |               | _____________/|
                  |               |/ DHCPACK      |
                  |               |               |
                  |    Initialization complete    |
                  |               |               |
                  .               .               .
                  .               .               .
                  |               |               |
                  |      Graceful shutdown        |
                  |               |               |
                  |               |\ ____________ |
                  |               | DHCPRELEASE  \|
                  |               |               |
                  |               |        Discards lease
                  |               |               |
                  v               v               v
```

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

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

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 I 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			