Chapter 5: outline

5.1 introduction

5.2 routing protocols
- link state
- distance vector

5.3 intra-AS routing in the Internet

5.4 inter-AS routing: BGP

5.5 The SDN control plane

5.6 ICMP: The Internet Control Message Protocol

5.7 Network management and SNMP

Network-layer functions

Recall the two network-layer functions:

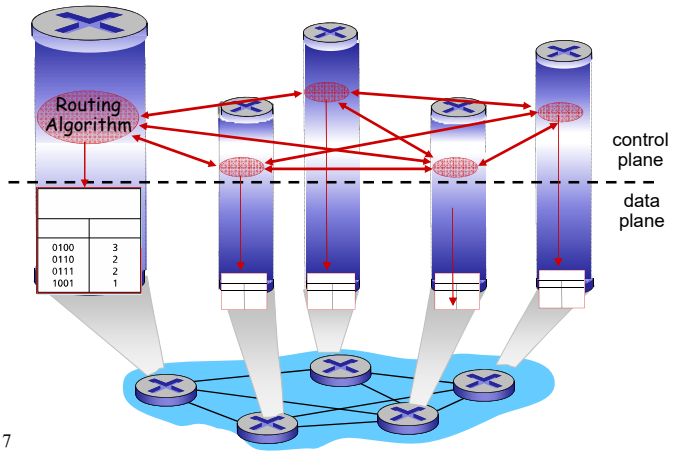
- *forwarding:* move packets from device inputs to device outputs — *data plane*
- *routing:* determine route taken by each packet from its source to destination — *control plane*

Two approaches to structure network control plane:

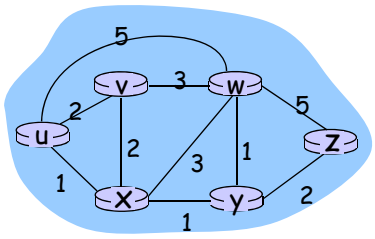
- per-router control (routers exchange messages)
- logically centralized control (SDN)

Per-router control plane

Individual processes *in routers* interact with each other by message exchange and compute forwarding tables



Graph abstraction



Graph: G = (N,E)

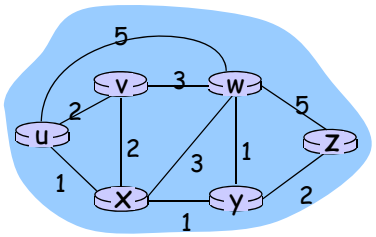
N = set of routers = { u, v, w, x, y, z }

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

Remark: Graph abstraction is also useful in other network contexts

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

Graph abstraction: link costs



- c(x,x') = cost of link (x,x')
- cost could be 1, or inversely proportional to bandwidth, etc.

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

Routing protocol tries to find least-cost paths

- cost of path computation is *ad hoc* if the link cost metric is not additive
- what if links have asymmetric costs for opposite directions? example – if queueing delays are included, then use a directed graph as model

Routing Algorithm classification

Global or decentralized information?

Global info:

- all routers have complete topology, link costs
- **link state** protocols

Decentralized info:

- router knows physically-connected neighbors, link costs to neighbors
- **distance vector** protocols

Static or dynamic?

- Static – update only after topology change
- Dynamic
  - periodic update
  - in response to link cost changes
  - may result in *route flaps*

Chapter 5: outline

5.1 introduction

5.2 routing protocols
- link state
- distance vector

5.3 intra-AS routing in the Internet

5.4 routing among the ISPs: BGP

5.5 The SDN control plane

5.6 ICMP: The Internet Control Message Protocol

5.7 Network management and SNMP

A Link-State Routing protocol

- net topology, link costs known to every node
  - accomplished **via link state broadcast**
  - all nodes have same info

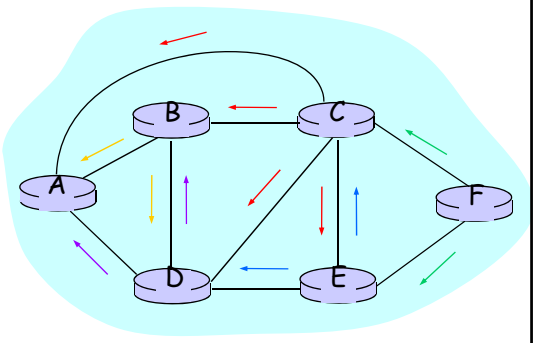
Dijkstra's algorithm *(you should have learned it)*

- computes least cost paths from one node ("source") to all other nodes in a graph
  - *iterative*: after k iterations, source knows least-cost paths to k destinations
  - yields **forwarding table** for source node

Link State Broadcast

Flooding

- Source node of "link state" sends packets to all neighbors
- Intermediate node resends to neighbors except where packet arrived
- Many duplicates which must be recognized by nodes



Distance Vector Algorithm basis

Bellman-Ford Equation (dynamic programming)

Define

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

Then

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

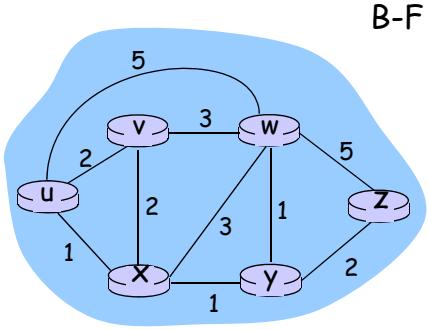
where min is taken over all neighbors v of x

Bellman-Ford example

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

B-F equation says:

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



The node that achieves minimum is next hop in shortest path ➜ put it in forwarding table

Distance Vectors Protocol (1)

- Node x
  - knows cost to each neighbor v: c(x,v)
  - sends its own distance vector (DV) ***estimate*** $[D_x(y): y \in N]$ to its neighbors periodically where $D_x(y)$ denotes ***estimate*** of least cost from x to y
- From each neighbor v, x receives $[D_v(y): y \in N]$

Distance Vector Protocol (2)

- When a node x receives a new DV estimate from a neighbor, it updates its own DV estimate using B-F equation:

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

*If the v that achieves least cost to y is **new**, node x updates its forwarding table and DV*

- Eventually, assuming that link costs and topology *do not change*, the estimate $D_x(y)$ *converges* to the actual least cost $d_x(y)$ for all x, y

Distance Vector Protocol - summary

Distributed, iterative, asynchronous

Initially, $D_x(y)$ = c(x,y) if x and y are direct neighbors; otherwise, $D_x(y) = \infty$

Each node:

waits for a change in local link cost *or* a msg from a neighbor

↓

recomputes estimates

↓

if DV estimate for any dest has changed, updates its own state and ***notifies*** its neighbors

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

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

node x table

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

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

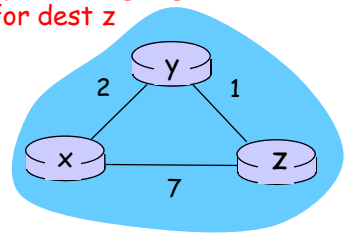
node y table

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

node z table

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

update outgoing interface for dest z



- Each row in a table is a distance vector
- Assume synchronous operations for this example – y and z also received their DV updates

time

node x table

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

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

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

node y table

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

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

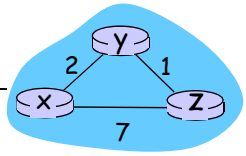
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

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

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

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



No more change

time

Distance Vector: *good news* travels fast

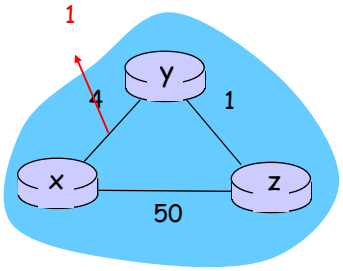
y detects a lower link cost to x, updates its DV, and sends new DV to node **z**.

z receives **y**'s updated DV, updates its own DV, and sends new DV to its neighbors.

later, **y** receives **z**'s updated DV. **y**'s least costs do not change.

A similar interaction between nodes x and z.

The DV protocol converges quickly for good news



Distance Vector: "count to infinity" problem

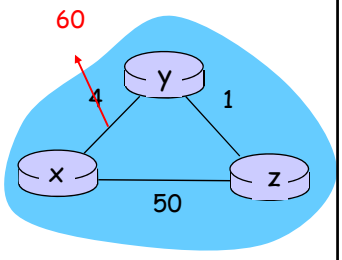
Link cost increase:

- Y still has *stale* information saying that it can go to X via Z in 6
- 44 messages exchanged between y and z before protocol 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?

bad news travels slowly!



Chapter 5: outline

5.1 introduction

5.2 routing protocols
- link state
- distance vector

5.3 intra-AS routing in the Internet

5.4 5.4 inter-AS routing: BGP

5.5 The SDN control plane

5.6 ICMP: The Internet Control Message Protocol

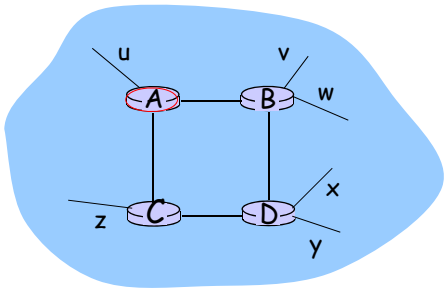
5.7 Network management and SNMP

Intra-AS Routing

- also known as Interior Gateway Protocols (IGP)
- most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - EIGRP (Cisco) – distance vector with "loop-freedom"

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 subnets:

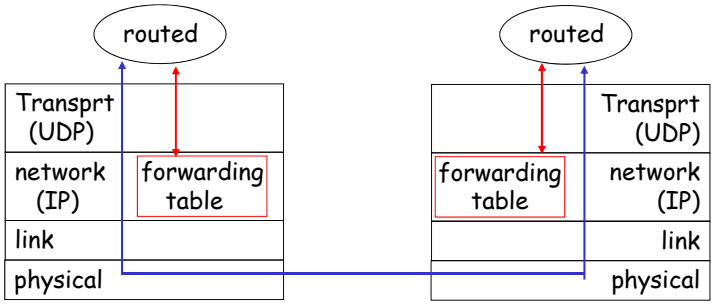
destination	hops
u	1
v	2
w	2
x	3
y	3
z	2

RIP advertisements

- *distance vectors:* exchanged with neighbors every 30 sec via Response Message (also called advertisement)
- each advertisement: list of up to 25 destination subnets within AS

RIP Table processing

- RIP routing tables managed by **application-level** process called routed (daemon)
- advertisements sent in UDP packets, periodically sent

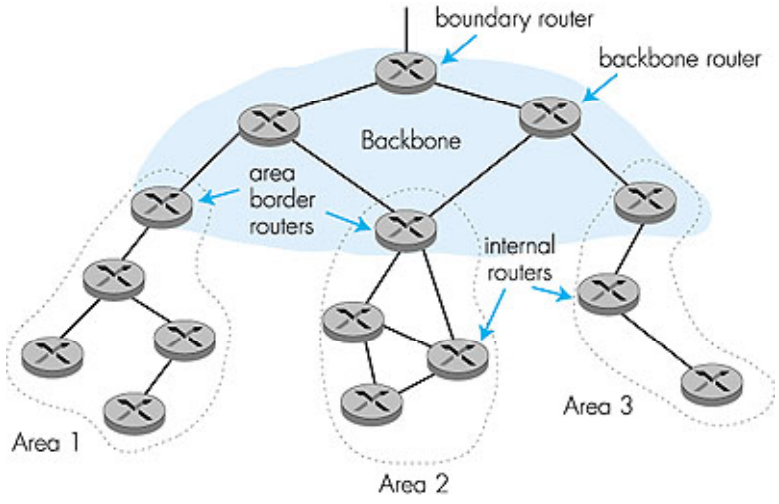


OSPF (Open Shortest Path First)

- "open": publicly available
- uses **Link State** algorithm
- OSPF advertisement carries one entry per neighbor router
- advertisements disseminated to entire AS (via flooding)
  - carried in OSPF messages directly over IP (rather than TCP or UDP)
- security: all OSPF messages authenticated
- …

Note: IS-IS routing protocol: nearly identical to OSPF

Hierarchical OSPF



Chapter 5: outline

5.1 introduction

5.2 routing protocols
- link state
- distance vector

5.3 intra-AS routing in the Internet

5.4 inter-AS routing: BGP

5.5 The SDN control plane

5.6 ICMP: The Internet Control Message Protocol

5.7 Network management and SNMP

Routing among ISPs

scale: hundreds of millions destination subnets:

- forwarding tables still too large after aggregation of prefixes
- Link State and Distance Vector do not scale

administrative autonomy

- internet is a network of networks
- each network admin wants to control routing in its own network

Hierarchical Routing

- autonomous systems (ASes)
  - stub vs. transit ASes
  - transit AS has an AS number from ICANN
- routers in an AS run the same intra-AS routing protocol
  - different ASes can run different intra-AS routing protocols

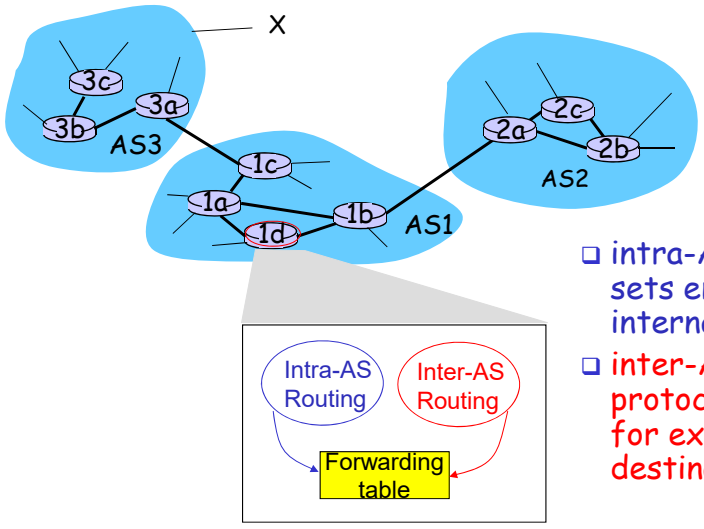
Gateway router

- has direct link to a gateway router in another AS
- for inter-AS routing

Internet inter-AS routing – basic ideas

- Gateway routers run BGP (Border Gateway Protocol): *the* de facto standard
- an AS advertises its existence using BGP to rest of Internet: *"I am here"* and
  1. gateway routers exchange reachability information with neighboring ASes (using external BGP on TCP connections) and
  2. they propagate reachability information to all internal routers of the AS (using internal BGP on TCP connections);
  3. "good" routes to other ASes selected based on reachability information and also policy

Forwarding table entries

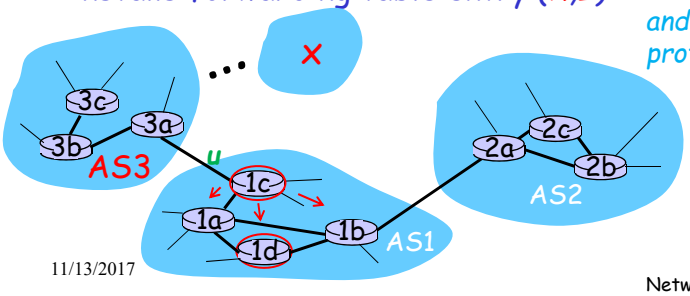


- intra-AS protocol sets entries for internal destinations
- inter-AS & intra-AS protocols set entries for external destinations

Example: Setting forwarding table in router 1d

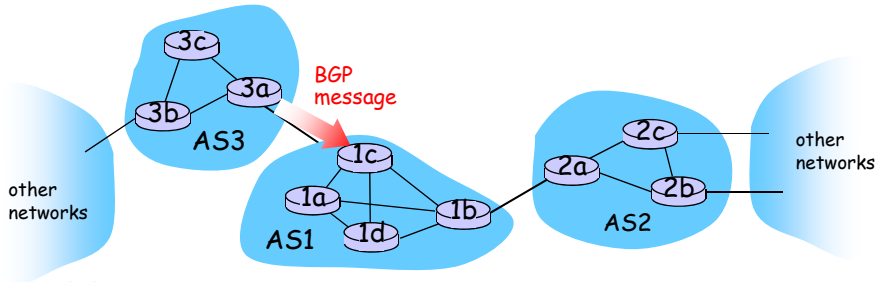
- AS1 (using eBGP) learns that subnet *x* is reachable via AS3 (gateway 1c) but not via AS2
- gateway 1c (using iBGP) propagates this reachability info to all other routers in AS1
- for subnet *x*, router 1d determines from **intra-AS** routing info that its interface *I* is on the least cost path to the *link* (*identified by subnet prefix u*) between 1c and 3a
  - installs forwarding table entry (*x*,*I*)

Note: both inter-AS and intra-AS protocols are used



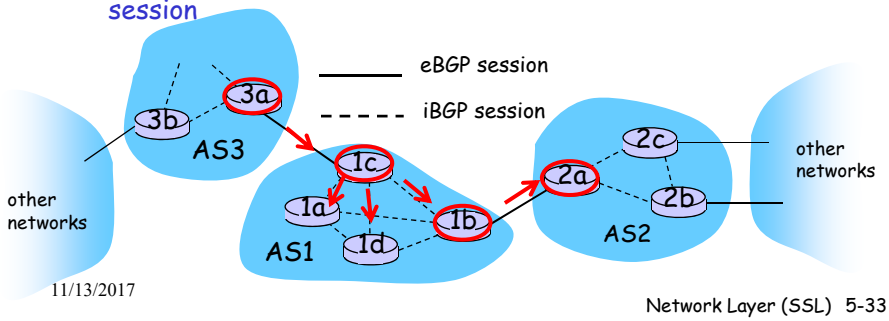
BGP - advertizing paths

- eBGP session: two BGP routers ("peers") exchange messages
  - advertising ***paths*** to various destination network prefixes ("path vector" protocol)
- when AS3 advertises a prefix to AS1, AS3 *promises* it will forward datagrams towards that prefix
  - AS3 can aggregate prefixes in its advertisement



BGP : distributing path information

- using eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
  - 1c can then use **iBGP** to distribute new prefix info to all routers in AS1
- when a router learns of a new prefix, it creates entry for prefix in its forwarding table.
  - in this example, gateway router 1b can then re-advertise such new reachability info to AS2 over 1b-to-2a eBGP session

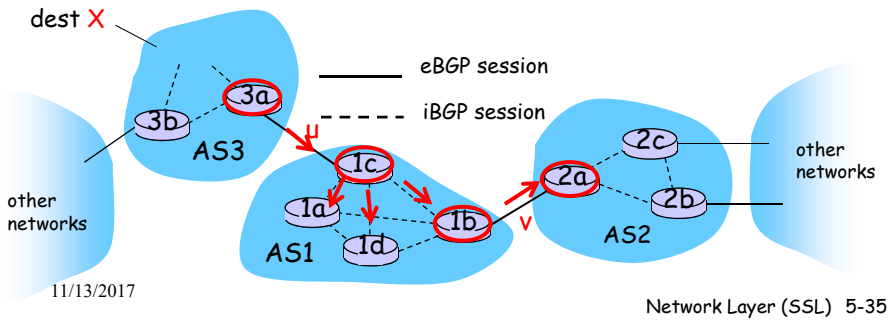


Path attributes & BGP routes

- advertised prefix includes BGP attributes.
  - prefix + attributes = "route"
- two important attributes:
  - AS-PATH: contains ASes through which prefix advertisement has passed: e.g, AS 67, AS 17
  - NEXT-HOP: the router interface (its subnet IP address) that begins the AS path
    - there may be multiple links from current AS to next-hop AS
- when a gateway router receives route advertisement, it
  - checks for loop
  - uses the AS's import policy to accept or reject route

Examples of NEXT-HOP for dest X

- For routers in AS1, next-hop is u, path vector is AS3, x
- For routers in AS2, next-hop is v, path vector is AS1, AS3, x



BGP route selection

- router may learn many routes to the same prefix. Router must select *one route*
- **Criteria**
  1. local preference value attribute (policy decision)
     - ***Import rule***: customer routes are preferred over peer routes, which are preferred over provider routes
  2. shortest AS-PATH

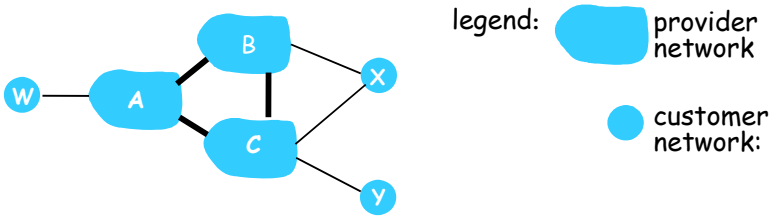
  ...

  6. closest NEXT-HOP router: hot potato routing

  ...

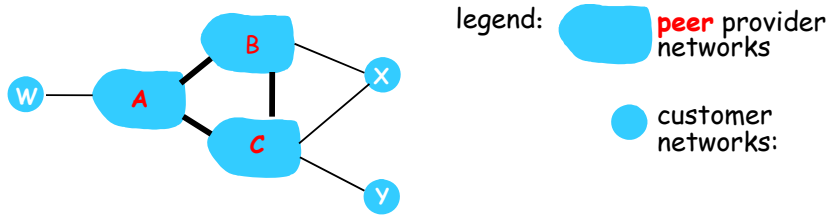
  (additional criteria) ...

BGP route export policy example (1)



- A,B,C are provider networks
- X,W,Y are customers (of provider networks)
- X is *dual-homed*: attached to two networks
  - X does not want to route from B via itself to C
  - .. so X will not advertise to B that it has a route to C

BGP route export policy example (2)



- A advertises path AW to B
- B advertises path BAW to X
- Should B advertise path BAW to C ?
  - No ! B gets no "revenue" for routing CBAW since neither C nor A nor W is a customer of B
  - B wants to route *only* to/from its customers

Export rule: peer/provider routes advertised to customers only; customer routes advertised to all neighbor ASes

Why different Intra- and Inter-AS routing ?

Scale:

- hierarchical routing reduces table size, also update traffic

Policy (including financial consideration)

- Intra-AS: single admin, so no policy decisions needed
- Inter-AS: admin wants control over how its own traffic is routed (*by import rule*), who routes through its network (*by export rule*).

Performance:

- Intra-AS: can focus on performance
- Inter-AS: policy dominates performance

Chapter 5: outline

5.1 introduction

5.2 routing protocols
- link state
- distance vector

5.3 intra-AS routing in the Internet

5.4 inter-AS routing: BGP

5.5 The SDN control plane

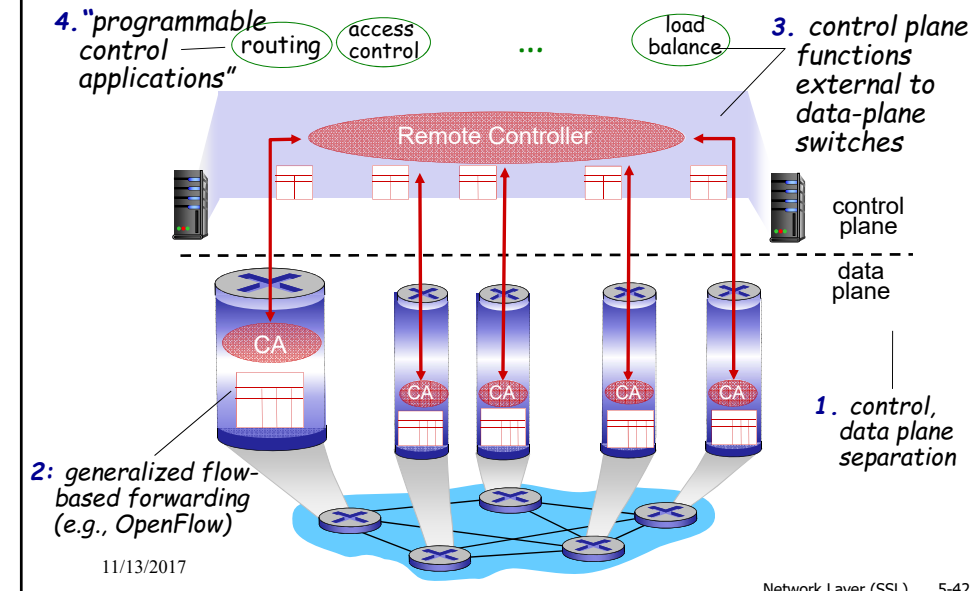
5.6 ICMP: The Internet Control Message Protocol

5.7 Network management and SNMP

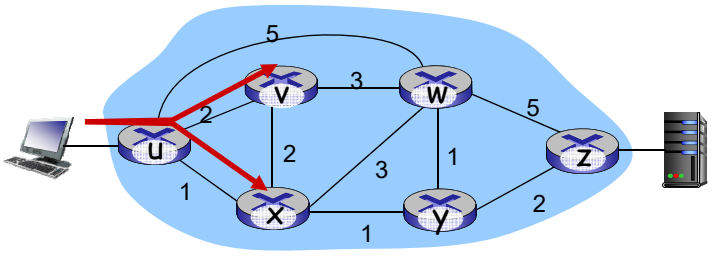
Software defined networking (SDN)

- Internet network layer historically has been implemented via distributed, per-router approach
  - *monolithic* routers run *proprietary* implementations of Internet protocol standards (IP, RIP, IS-IS, OSPF, BGP) in *proprietary* router OS (e.g., Cisco IOS), forwarding packets
- Other devices in data plane
  - different "middle boxes" for other network functions: firewalls, load balancers, NAT boxes, ..
  - switches for layer-2 forwarding
- ~2005: rethinking separation of network control plane from data plane

Software defined networking (SDN)



ISP Traffic engineering



Q: what if network operator wants to split u-to-z traffic along uvwz *and* uxyz (*load balancing*)?

A: can't do it using current protocols based upon shortest-path computation

Layers in SDN architecture

"Network-control apps"

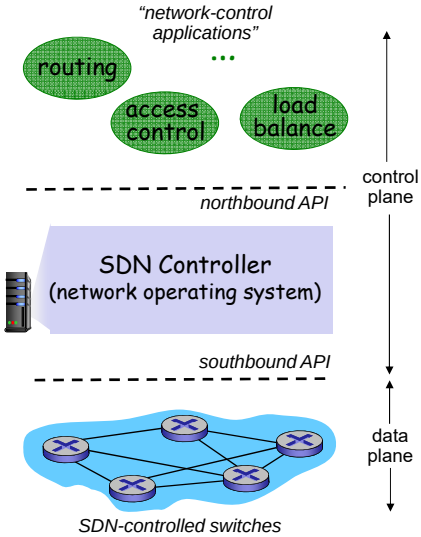
- *use network state information provided by SDN controller*

SDN controller (net OS)

- maintains network state information, statistics, and flow tables
- may be implemented as a distributed system for performance and fault-tolerance

Data plane switches

- fast, simple, commodity switches
- communicate with controller to provide state information and receive flow tables



Software defined networking (SDN)

- Logically centralized controller
  - controller provides accurate network state information to *"network-control applications"*
  - centralized computation of flow tables is easier than distributed computation using protocol messages
  - greater flexibility and better control of traffic flows
- Open standards allow "unbundling" of network functionality
  - data plane boxes, SDN controllers can be provided by different vendors

Observation: SDN is (mainly) for networks under the same administrative control e.g., Google SDN uses both inter-AS and intra-AS routing protocols: BGP between datacenters and IS-IS for intra-datacenter

Chapter 5: outline

5.1 introduction

5.2 routing protocols
- link state
- distance vector

5.3 intra-AS routing in the Internet

5.4 inter-AS routing: BGP

5.5 The SDN control plane

5.6 ICMP: The Internet Control Message Protocol

5.7 Network management and SNMP

ICMP: Internet Control Message Protocol

- "above" IP in network layer
  - ICMP msgs carried in IP datagrams
  - error reporting: unreachable network, host, port, protocol
  - echo request/reply (used by ping)
- ICMP message
  - type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	**0**	**echo reply (to ping)**
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	**3**	**dest port unreachable**
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - *not used*)
8	**0**	**echo request (ping)**
9	0	route advertisement
10	0	router discovery
11	**0**	**TTL expired**
12	0	bad IP header

Traceroute uses ICMP messages

- Source sends series of UDP segments to dest
  - First has TTL =1
  - Second has TTL=2,
    …, each with **unlikely port number**
- When *n*th datagram arrives to *n*th router:
  - Router discards datagram and
  - sends to source a "TTL expired" message with name of router & IP address
- When "TTL expired" message arrives, source calculates RTT
- Traceroute does this 3 times for each TTL value

Stopping criterion for source

- Such a UDP segment arrives at destination host
- Destination returns msg "dest port unreachable" packet
- Upon receipt of this msg, source stops.

Chapter 5: outline

5.1 introduction

5.2 routing protocols
- link state
- distance vector

5.3 intra-AS routing in the Internet

5.4 inter-AS routing: BGP

5.5 The SDN control plane

5.6 ICMP: The Internet Control Message Protocol

5.7 Network management and SNMP

End of Chapter 5