Course on Computer Communication and Networks

Lecture 7
Network Layer,
Chapter 4 (6/e) - Part B
(7/e Ch5)

EDA344/DIT 420, CTH/GU

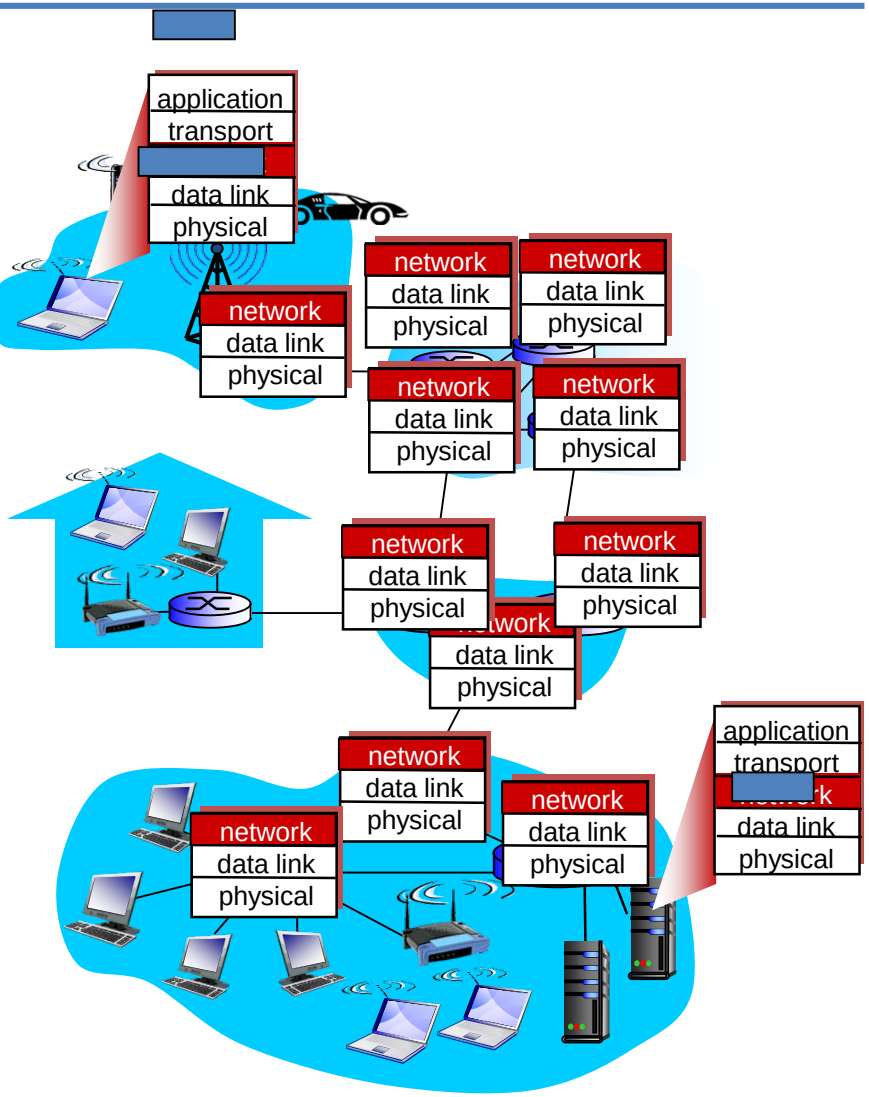
Based on the book Computer Networking: A Top Down Approach, Jim Kurose, Keith Ross, Addison-Wesley.

Network layer

Consider transporting a segment from sender to receiver

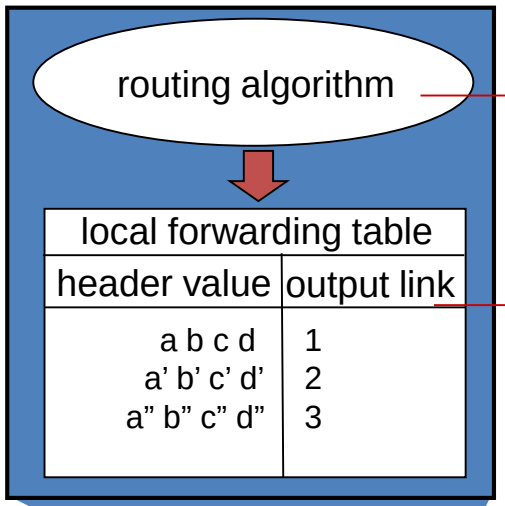
- sending side: encapsulates segments into datagrams
- receiving side: delivers segments to transport layer

- network layer protocols in *every* host, router
  - examines header fields in all datagrams passing through it



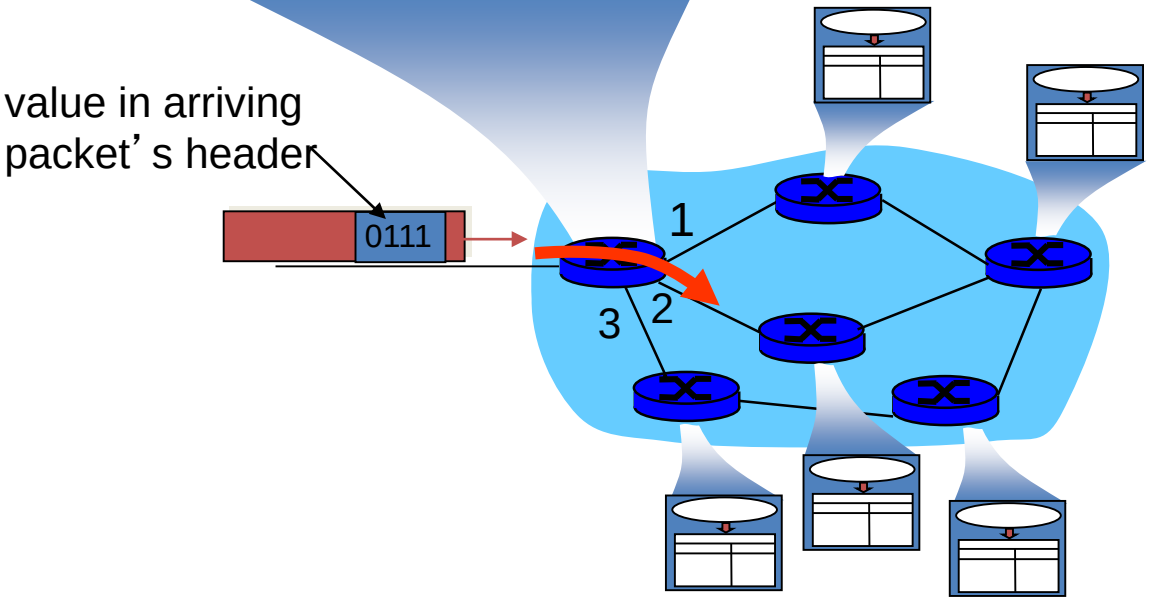
NW layer's job - routing and forwarding

Interplay between the two:



routing algorithm determines path through network (***control-plane*** functionality)

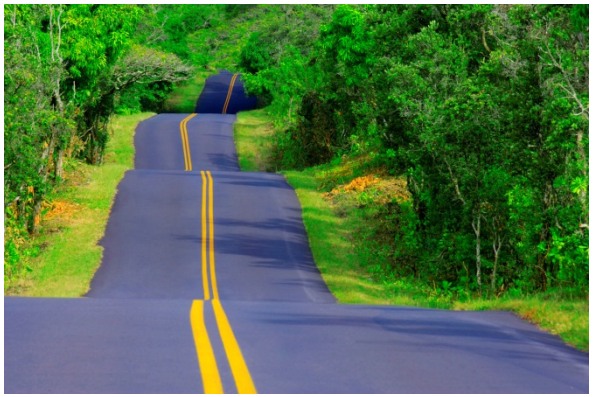
forwarding table determines local forwarding at this router (***data-plane*** functionality)



Roadmap Network Layer

PREVIOUS Lect.

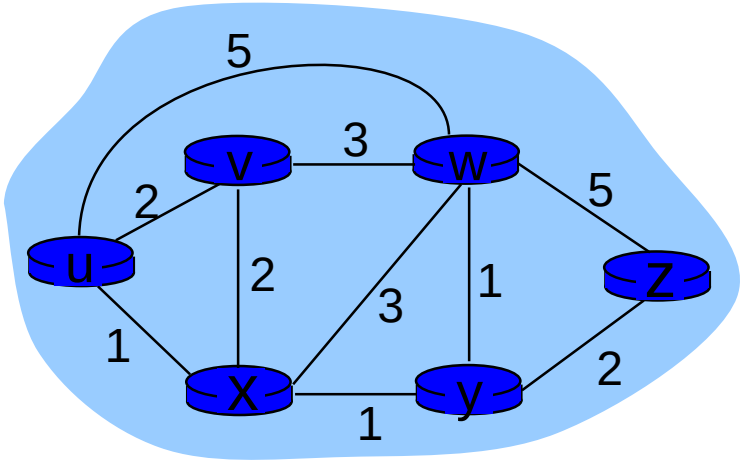
- Forwarding versus routing
- Network layer service models
  - Network layer architecture (shift): Software-Defined Networks
- Inside a router
- The Internet Network layer: IP, Addressing & related: IPv4, maskig&forwarding, obtaining an IP address, DHCP, NAT, IPv6



Now: Control, routing

- path selection/routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- instantiation, implementation in the Internet routing protocols
  - RIP
  - OSPF
  - BGP
- ICMP (control protocol)

Graph abstraction: costs



Graph: G = (N,E)

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

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

Cost of link x→w is c(x,w) = 3

Cost of link could be always 1 hop, or related directly to delay or inversely to bandwidth, or any other metric

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

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 algorithm: finds least-cost path

Routing Algorithm Classification

Global or decentralized?

Global:

- all routers have complete and global knowledge about topology, and all link-costs

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- exchange of info with neighbors
- Iteratively calculate the least-cost paths to other routers

Static or dynamic routing?

Static:

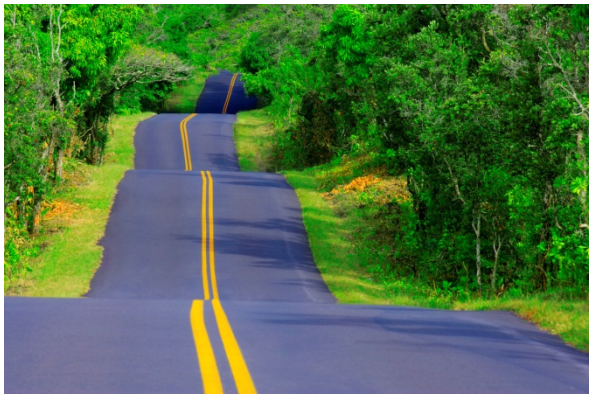
- routes change slowly over time, manually configured

Dynamic:

- routes change more quickly
  - periodic update, or
  - in response to link-cost changes

Roadmap Network Layer

- Forwarding versus routing
- Network layer service models
  - Network layer architecture (shift): Software-Defined Networks
- Inside a router
- The Internet Network layer: IP, Addressing & related



Control, routing

- path selection/routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- instantiation, implementation in the Internet routing protocols
  - RIP
  - OSPF
  - BGP
- ICMP (control protocol)

A Link-State (LS) Routing Algorithm

Dijkstra's shortest path algo

- link costs known to all nodes
  - Each node floods "link state multicasts" with costs to its neighbors etc
  - all nodes get same info
- Each node computes least cost paths from itself to all other nodes
  - gives forwarding table for that node
- iterative: after k iterations, knows least cost path to k destinations

Dijsktra's Algorithm at node u

```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v adjacent (directly attached
5          then D(v) = c(u,v)
6      else D(v) = ∞
7
8  Loop
9      find node w not in N' such that D(w) is a minimum
10     add node w to N'
11     update D(v) for all v adjacent to w and not in N' :
12        D(v) = min{ D(v), D(w) + c(w,v) }
13     /* new cost to v is either old cost to v or known
14      shortest path cost to w plus cost from w to v */
15 until all nodes N in N'
```

c(x,y): link cost from x to y. Initially cost(x,y) = ∞ if not direct neighbors

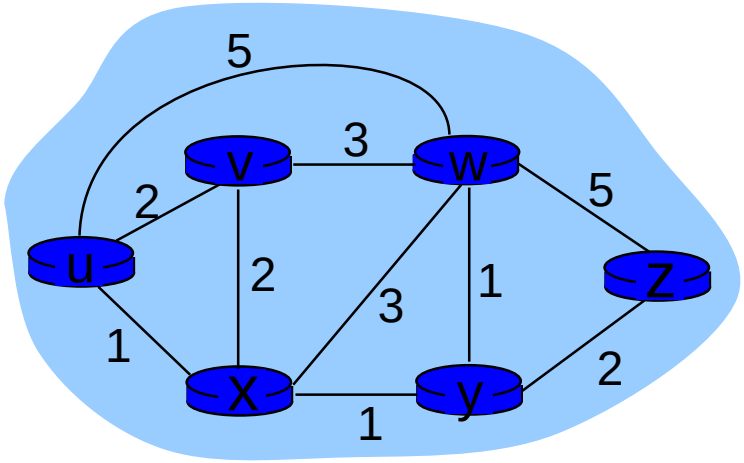
D(v): **D**istance; current value of cost of path from source to destination v

p(v): **p**redecessor node, i.e. previous node that is neighbor of v along current path from the source to node v

N': set of **N**odes whose least cost path definitively known

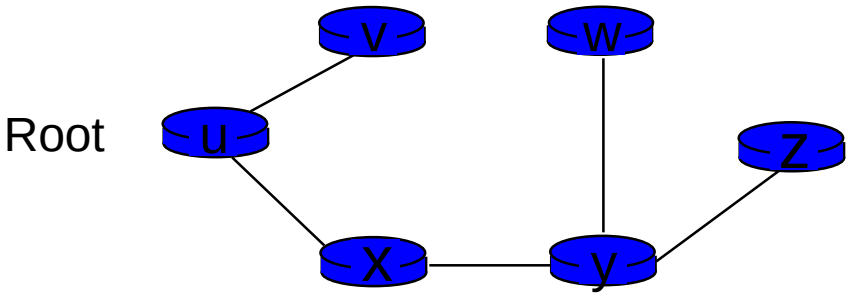
Dijkstra's algorithm: example node u

Step	N'	v D(v),p(v)	w D(w),p(w)	x D(x),p(x)	y D(y),p(y)	z D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxv		4,x		2,x	∞
3	uxvy		3,y			4,y
4	uxvyw					4,y
5	uxvywz					



Dijkstra's algorithm: forwarding table

Resulting shortest-path tree from node u as root:



Resulting forwarding table in u:

destination	via link	cost
v	(u,v)	**2**
x	(u,x)	**1**
y	(u,x)	**2**
w	(u,x)	**3**
z	(u,x)	**4**

Dijkstra's algorithm, discussion

Algorithm complexity: n nodes

- each iteration: need to check all nodes, not in N'
- n(n+1)/2 comparisons: Order of (n^2)

Oscillations possible:

e.g., if link cost = delay-based or traffic-based, dynamically variable metric

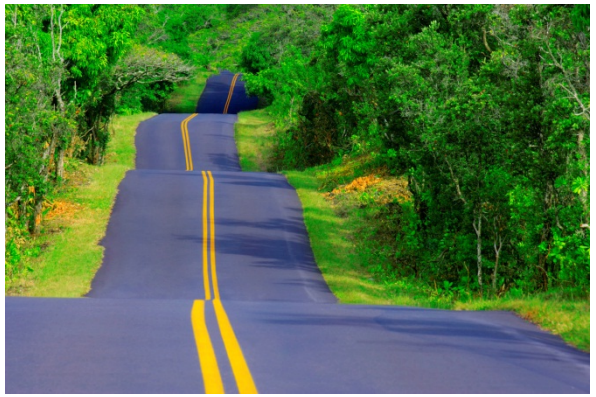
must avoid these metrics

But:

- Good for small networks
- Link-cost changes are not frequent, more stable network
- Faster to converge when changes in link-costs

Roadmap Network Layer

- Forwarding versus routing
- Network layer service models
  - Network layer architecture (shift): Software-Defined Networks
- Inside a router
- The Internet Network layer: IP, Addressing & related



Control, routing

- path selection/routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- instantiation, implementation in the Internet routing protocols
  - RIP
  - OSPF
  - BGP
- ICMP (control protocol)

Distance Vector (DV) Algorithm

Bellman-Ford Equation:

Define

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

If v is any neighbor to x with link cost c(x,v) and has $d_v(y)$ as least-cost path to y

Then the DV estimate:

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

min taken over all neighbors v of x

cost to neighbor v

cost from neighbor v to destination y

Distance vector (DV) algorithm

iterative, asynchronous:

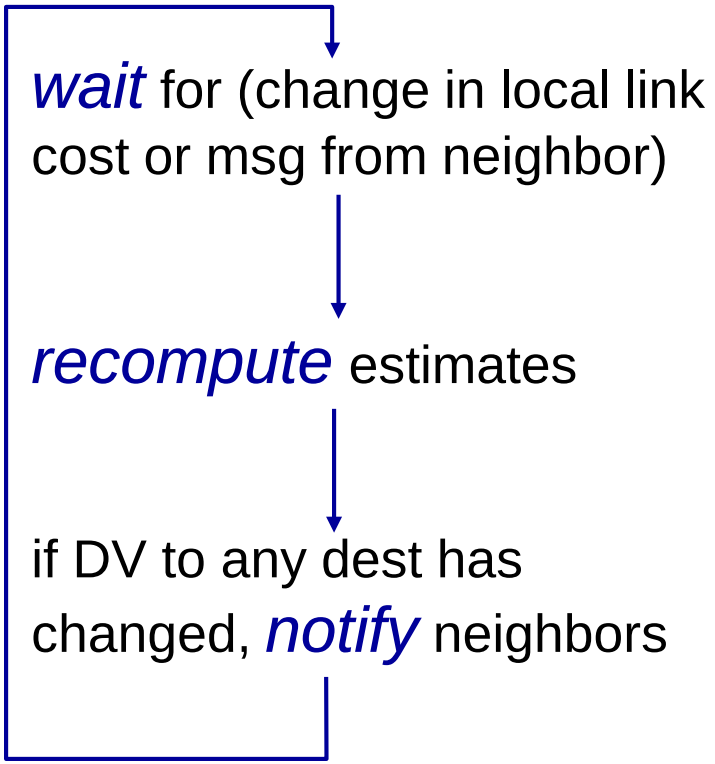
each local iteration caused by:

- local link cost change
- DV update message from neighbor

distributed:

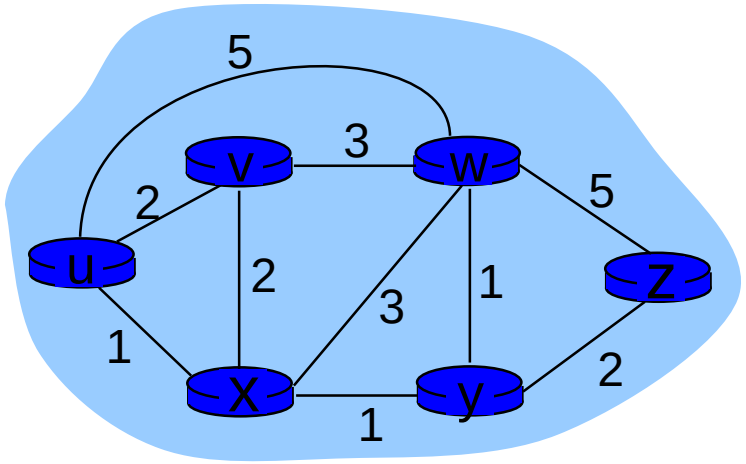
- each node notifies neighbors *only* when its DV changes
  - neighbors then notify their neighbors if necessary

each node:



Bellman-Ford: example

Nodes v, x & w are the neighbors of u



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

BF equation says:

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

Node x that achieves minimum is the next hop in least-cost path to z ➜ forwarding table

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

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

node x table

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

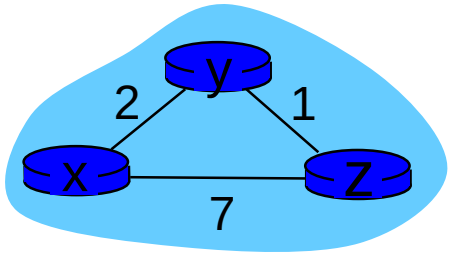
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

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



time

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

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

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

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

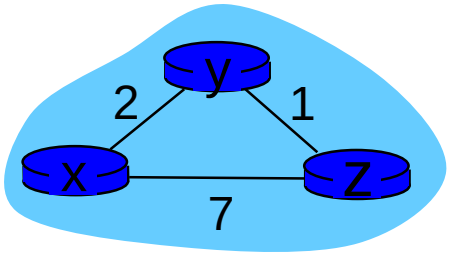
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

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

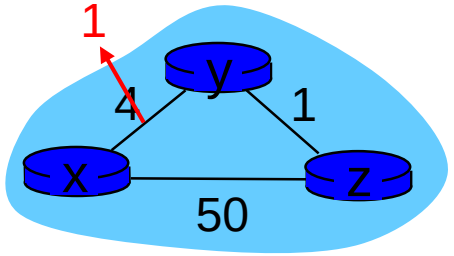


time

DV: link cost changes (good news)

Link cost changes: decreased cost

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



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

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

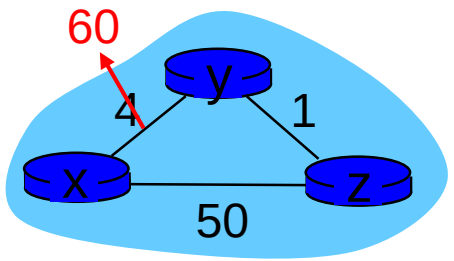
"good news travels fast"

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

DV: link cost changes (bad news)

Link cost changes: increased cost

- bad news travels slow - "count to infinity" problem!
- 44 iterations before algorithm stabilizes!
  - y already knows z has cost 5 to reach x
  - y therefore announces cost 6 to reach x
  - z announces cost is now 7, etc..



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)

"bad news travels slow"

Comparison of LS and DV algorithms

Message complexity

- LS: with n nodes, E links, O(nE) messages sent
- DV: exchange between neighbors only, until convergence

Convergence due to changes

- LS:
  - may have oscillations
  - fast convergence
- DV:
  - may be routing loops
  - count-to-infinity problem
  - slow convergence

Robustness: what happens if router malfunctions?

LS:

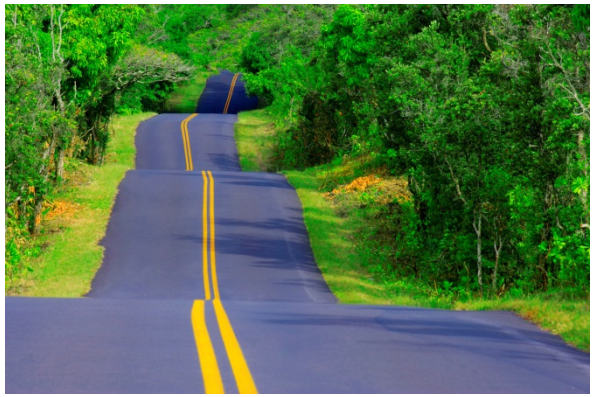
- each node computes only its *own* table
- limited damage

DV:

- node can advertise incorrect *path* cost
- each node's table used by others
  - error propagates through network

Roadmap Network Layer

- Forwarding versus routing
- Network layer service models
  - Network layer architecture (shift): Software-Defined Networks
- Inside a router
- The Internet Network layer: IP, Addressing & related



Control, routing

- path selection/routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- instantiation, implementation in the Internet routing protocols
  - RIP
  - OSPF
  - BGP
- ICMP (control protocol)

Hierarchical Routing

Our routing study thus far - idealization

- all routers identical
- network “flat”

… *not* true in practice

scale: millions of destinations!

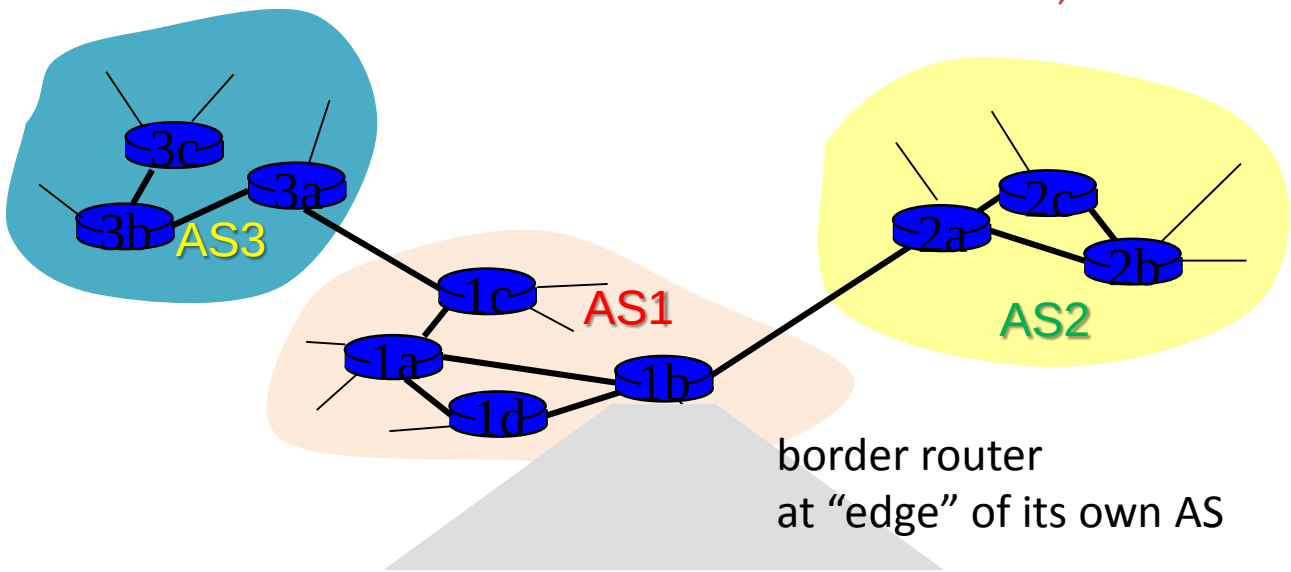
- can’t store all destinations in routing tables!
- LS routing info exchange would swamp links!
- DV would never terminate

administrative autonomy

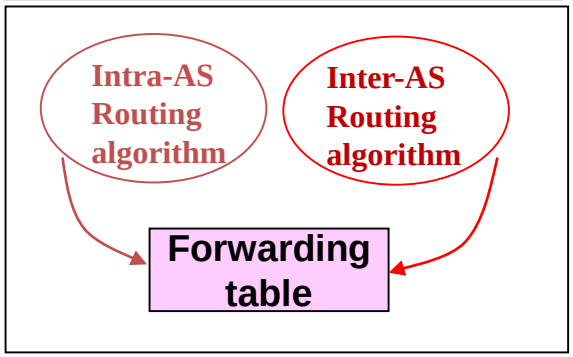
- Internet = network of networks
- each network administrator may want to control routing in its own network

Interconnected ASs

- aggregate routers into regions, "autonomous systems" (AS)
  - Internet: > 39,000 AS



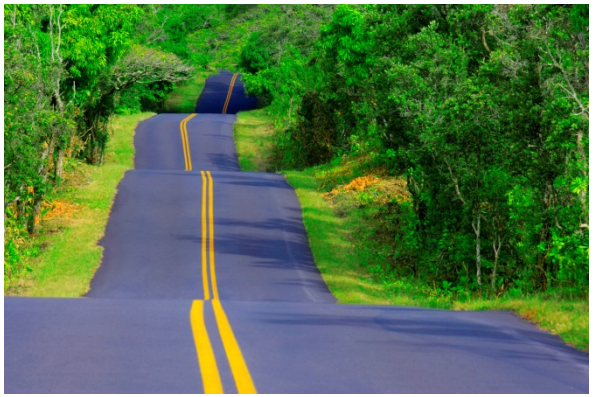
Forwarding table configured by both intra- and inter-AS routing algorithms



- intra-AS sets entries for internal destinations
- inter-AS sets entries for external destinations

Roadmap Network Layer

- Forwarding versus routing
- Network layer service models
  - Network layer architecture (shift): Software-Defined Networks
- Inside a router
- The Internet Network layer: IP, Addressing & related



Control, routing

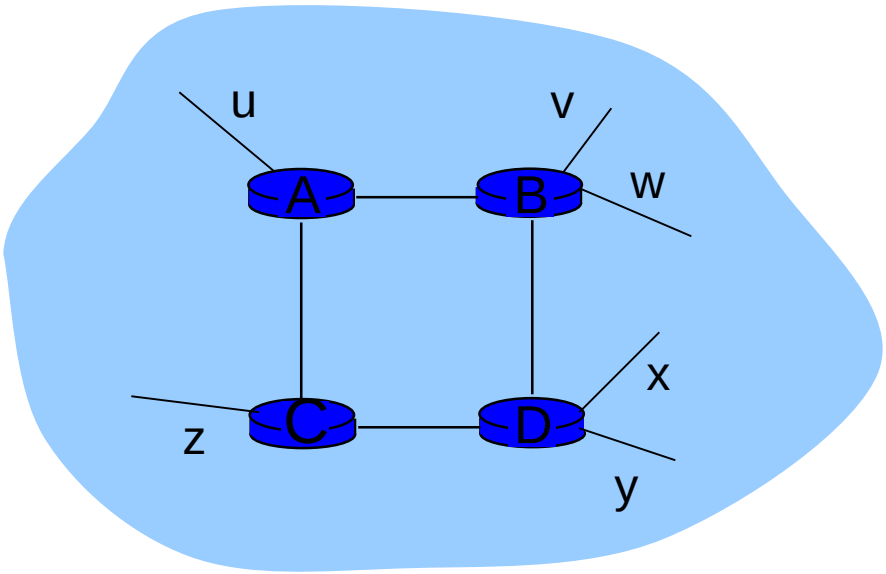
- path selection/routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- instantiation, implementation in the Internet routing protocols
  - RIP
  - OSPF
  - BGP
- ICMP (control protocol)

Intra-AS Routing

- also known as Interior Gateway Protocols (IGP)
- most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol [DV]
  - OSPF: Open Shortest Path First [LS]
  - EIGRP: Enhanced Interior Gateway Routing Protocol (Cisco proprietary)

RIP (Routing Information Protocol)

- distance vector algorithm
- included in BSD-UNIX Distribution 4.3 in 1982
- distance metric: number of hops (max = 15 hops)
- Version 1 classful and version 2 classless

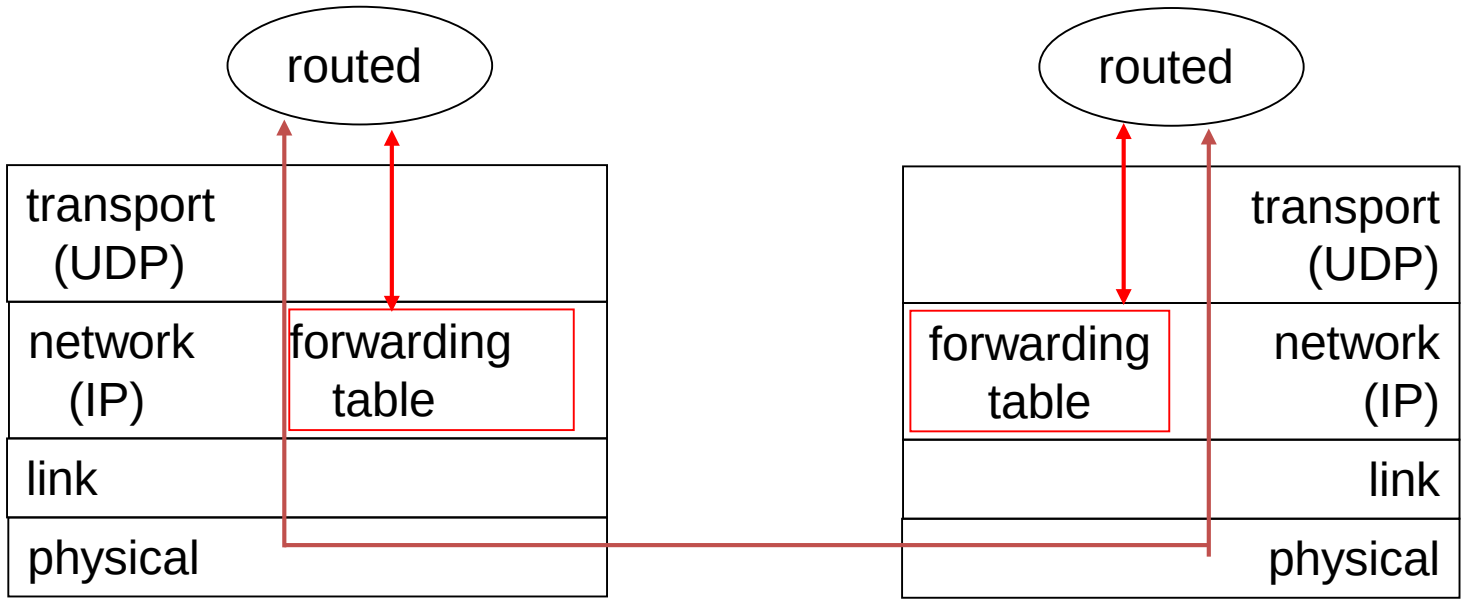


From router A to subnets:

destination	hops
u	0
v	1
w	1
x	2
y	2
z	1

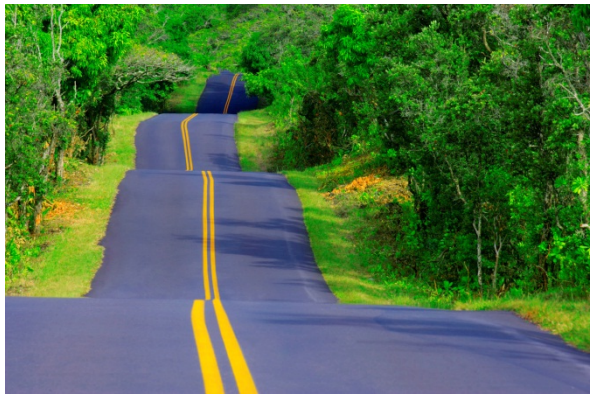
RIP Table processing

- RIP routing tables managed by **application-level** process called ***routed*** (route daemon)
- advertisements periodically sent in UDP packets (port 520) using broadcast (or multicast, RIP v.2)



Roadmap Network Layer

- Forwarding versus routing
- Network layer service models
  - Network layer architecture (shift): Software-Defined Networks
- Inside a router
- The Internet Network layer: IP, Addressing & related



Control, routing

- path selection/routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- instantiation, implementation in the Internet routing protocols
  - RIP
  - OSPF
  - BGP
- ICMP (control protocol)

OSPF (Open Shortest Path First)

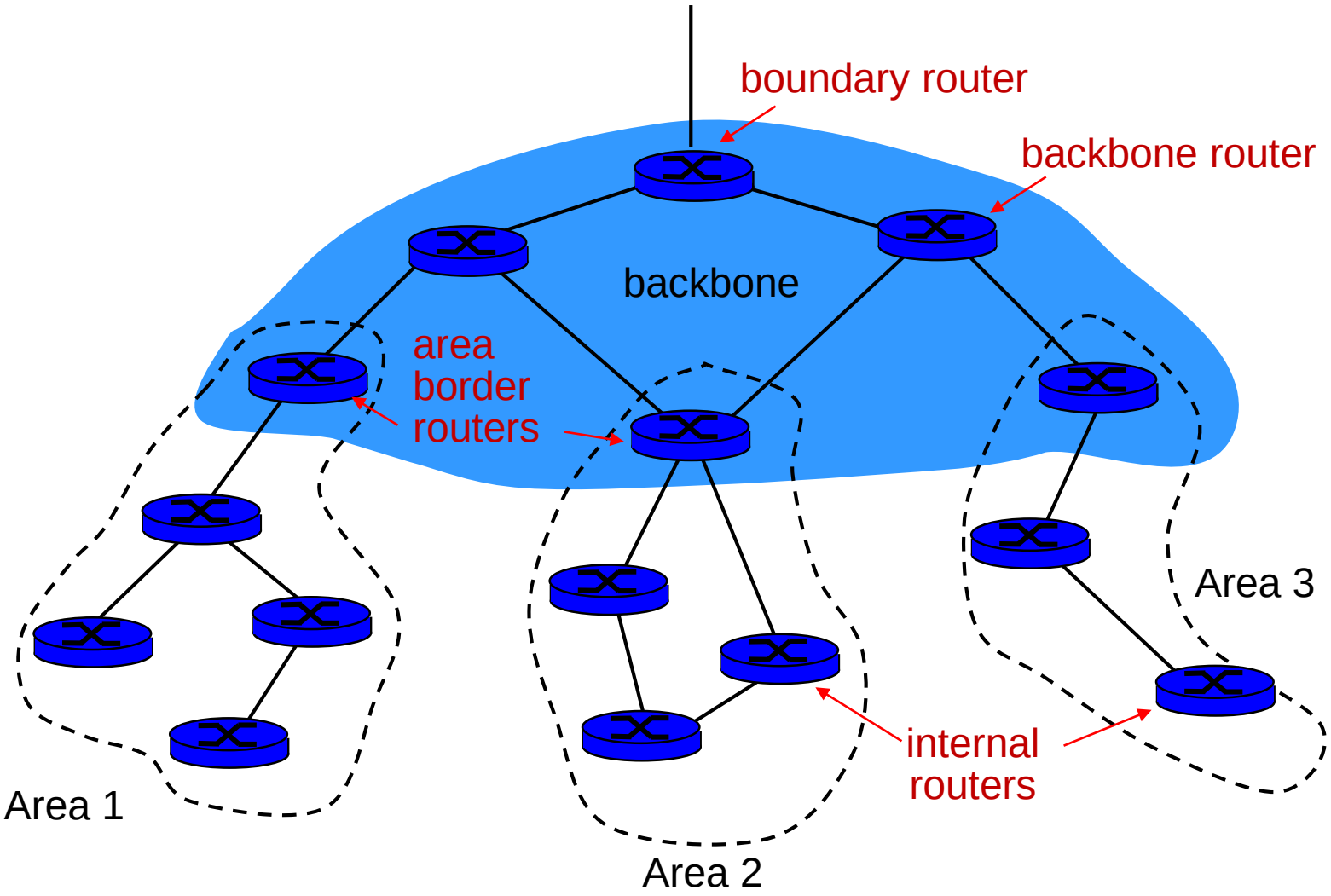
- "open": just means publicly available (RFC 2328)
- uses Link State algorithm
  - complete topology map built at each node
  - route computation using Dijkstra's algorithm
  - works in larger networks (hierarchical structure with areas)

- OSPF advertisements sent within area via flooding.
  - carried in OSPF messages directly over IP with protocol number 89 (no UDP- or TCP-transport)
  - sent at least every 30 minutes

OSPF features

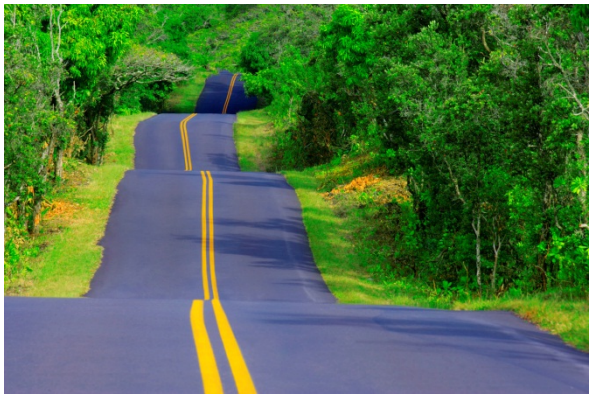
- security: all OSPF messages can be authenticated (to prevent malicious intrusion)
- multiple same-cost paths allowed
- Send HELLO messages to establish adjacencies with neighbors to check operational links
- hierarchical OSPF in large domains.

Hierarchical OSPF



Roadmap Network Layer

- Forwarding versus routing
- Network layer service models
  - Network layer architecture (shift): Software-Defined Networks
- Inside a router
- The Internet Network layer: IP, Addressing & related



Control, routing

- path selection/routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- instantiation, implementation in the Internet routing protocols
  - RIP
  - OSPF
  - BGP
- ICMP (control protocol)

Internet inter-AS routing: BGP

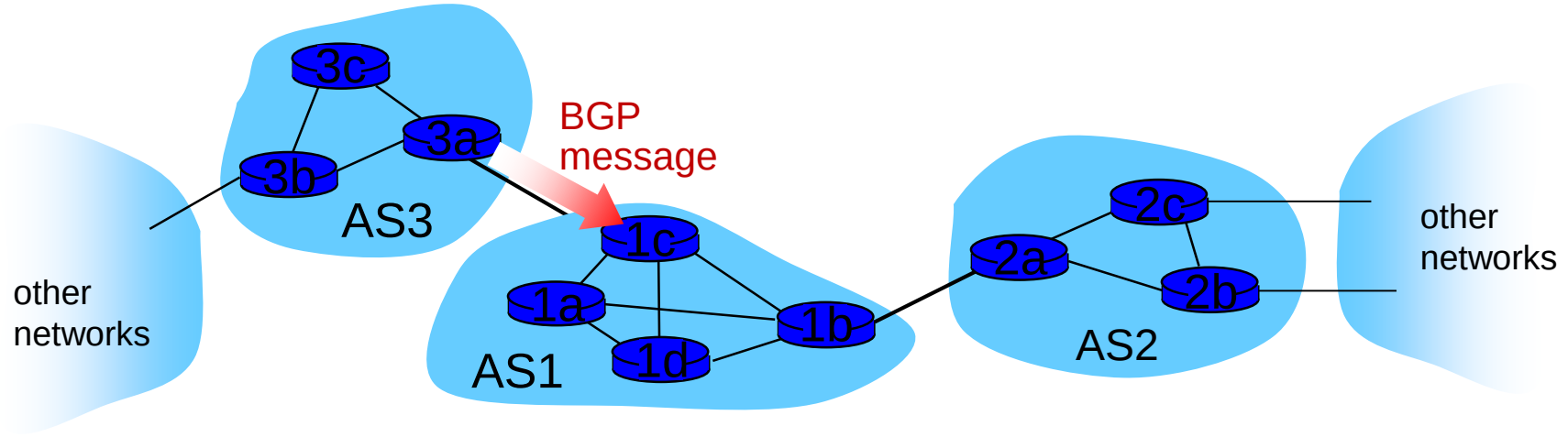
- BGP (Border Gateway Protocol)
  - *the* de facto standard routing protocol on the Internet
  - Complex protocol
  - Communicates over TCP port 179 with authentication

- BGP provides each AS a means to:
  - Obtain prefix reachability information from neighboring ASs.
  - Propagate reachability information to all AS-internal routers.
  - Determine "good" routes to prefixes based on reachability information and policy.

BGP basics

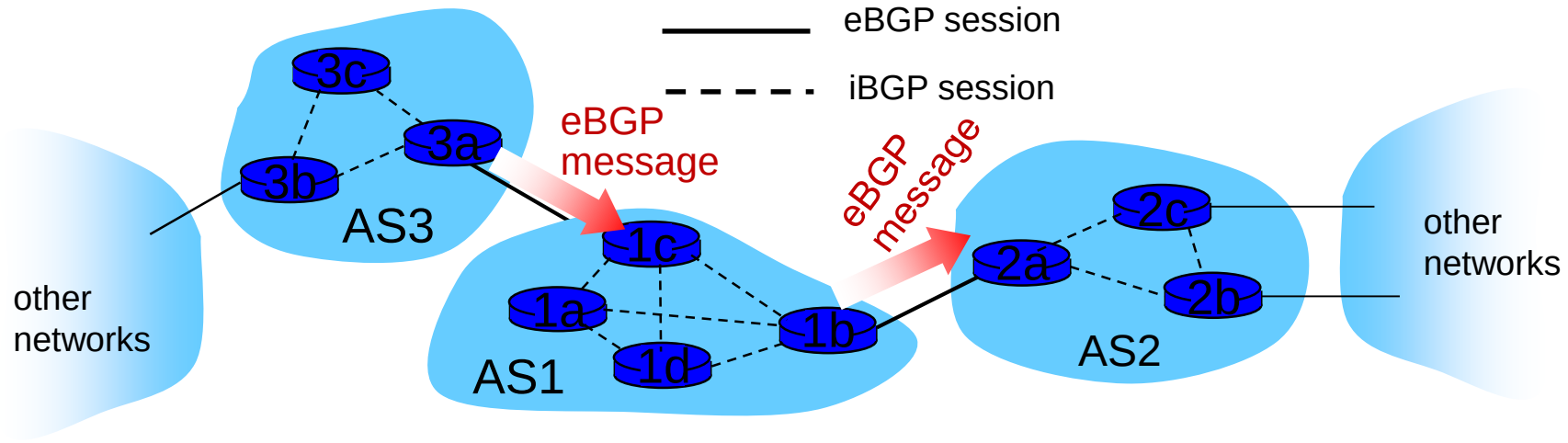
- pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: BGP sessions
  - BGP sessions need not correspond to physical links.
  - advertising *paths* to different destination network prefixes ("path vector" protocol)

- when AS3 advertises a prefix to AS1:
  - AS3 *promises* it will forward datagrams towards that prefix.
  - AS3 can aggregate prefixes in its advertisement

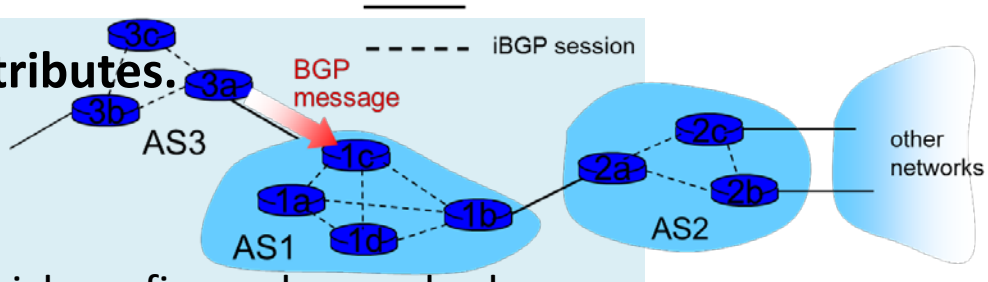


Distributing Reachability Info

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

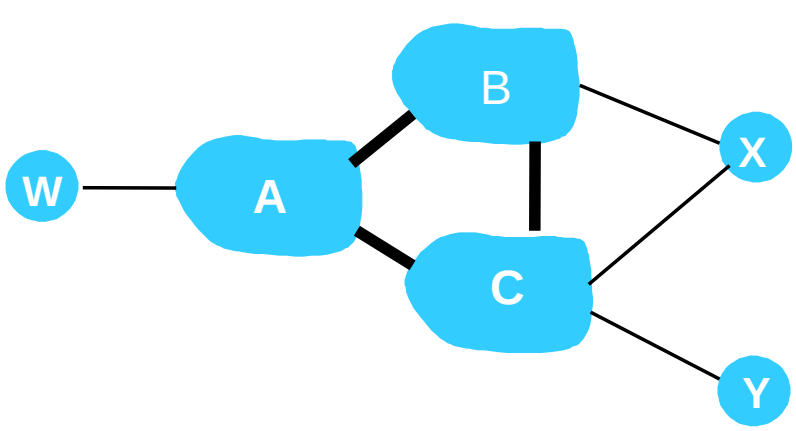


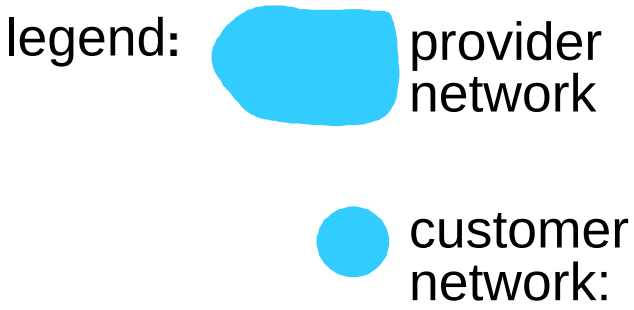
Path attributes & BGP routes



- **advertised prefix includes BGP attributes.**
  - prefix + attributes = "route"
- **two important attributes:**
  - AS-PATH: contains ASs through which prefix can be reached: e.g., AS3, AS1
  - NEXT-HOP: indicates specific AS router to next-hop AS. E.g. 1b
- **when gateway router receives route advertisement, uses import policy to accept or decline.**
  - May or may not accept/announce everything to/from peers
- **Router may learn about more than 1 route to some prefix. Router selects route based on:**
  - Policy decision
  - Shortest AS_PATH
  - Closest NEXT_HOP router

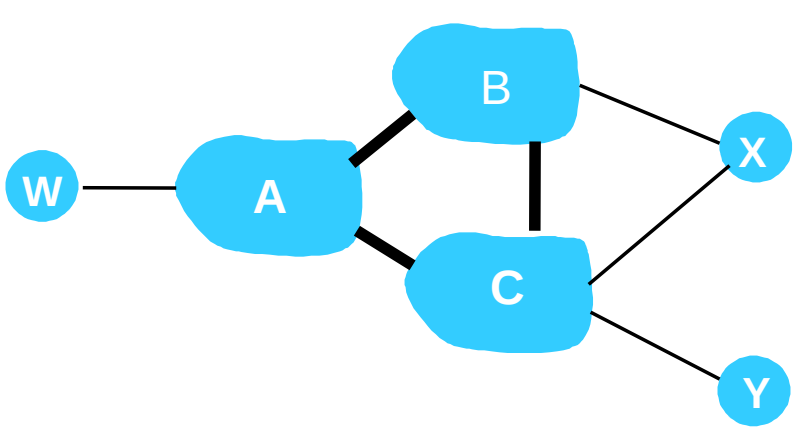
BGP routing policy example (1)

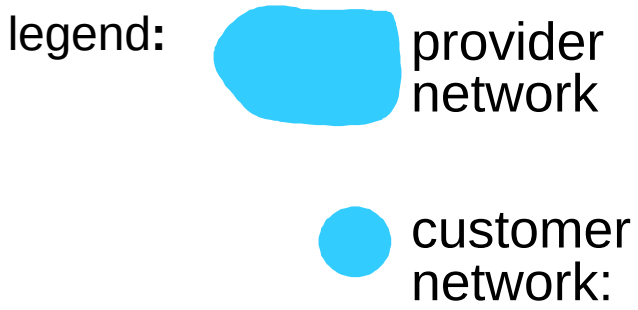




- A,B,C are provider networks
- x,w,y are customers (of provider networks)
- x is dual-homed: attached to two networks
  - x does not want to route from B via x to C
  - .. so x will not advertise to B a route to C

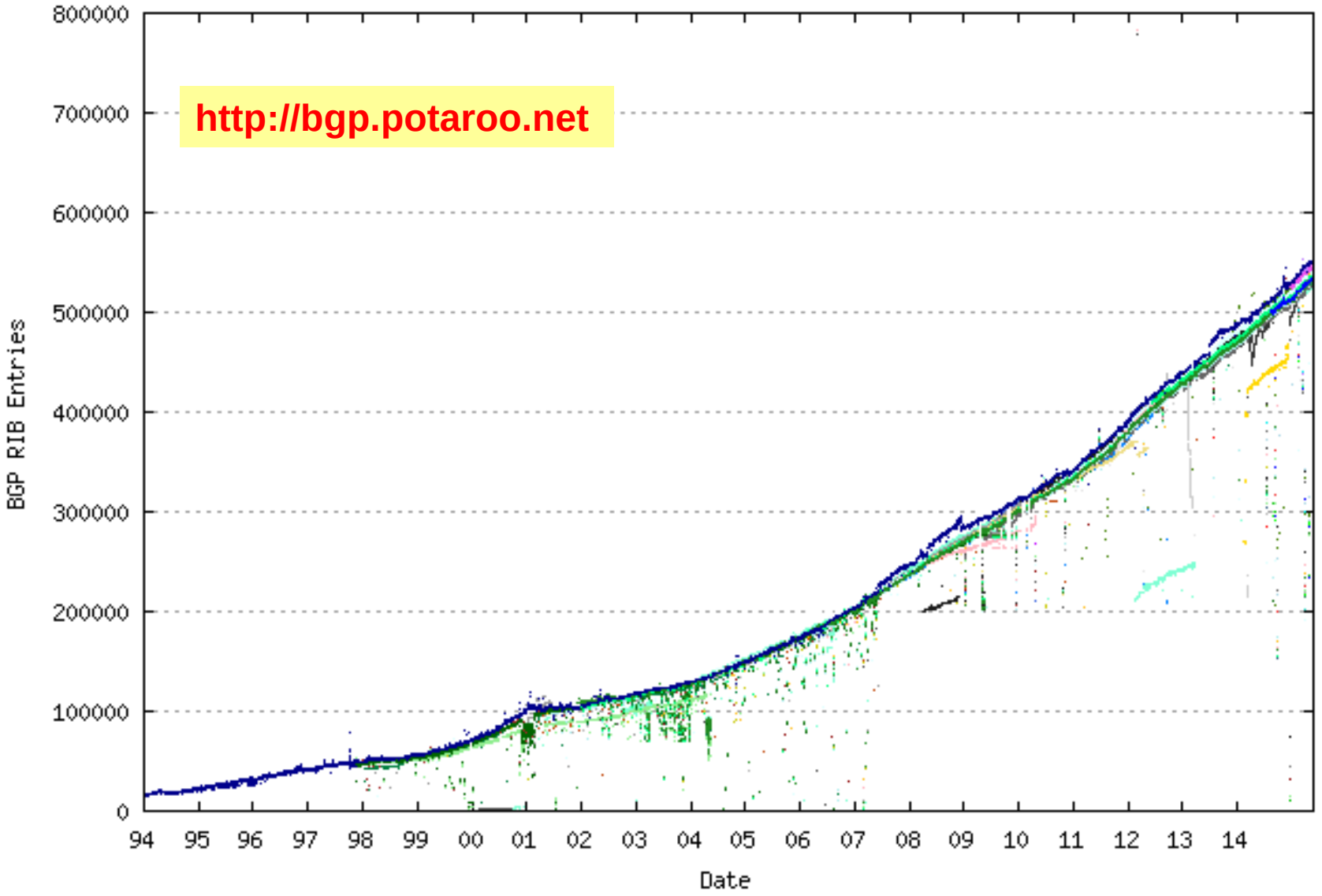
BGP routing policy example (2)





- A advertises to B path A-w
- B advertises to x path B-A-w
- Should B advertise path B-A-w to C? no
  - B gets no “revenue” for routing C-B-A-w since neither w nor C are B’s customers
  - B wants to force C to route to w via A
  - B wants to route *only* to/from its customers…

Growth of the BGP table: 1994 to Present



Why different Intra- & Inter-AS routing?

Policy:

- Inter-AS: admin wants control over how its traffic routed, who routes through its net.
- Intra-AS: single admin, so no policy decisions needed

Scale:

- hierarchical routing saves routing table size, reduced update traffic

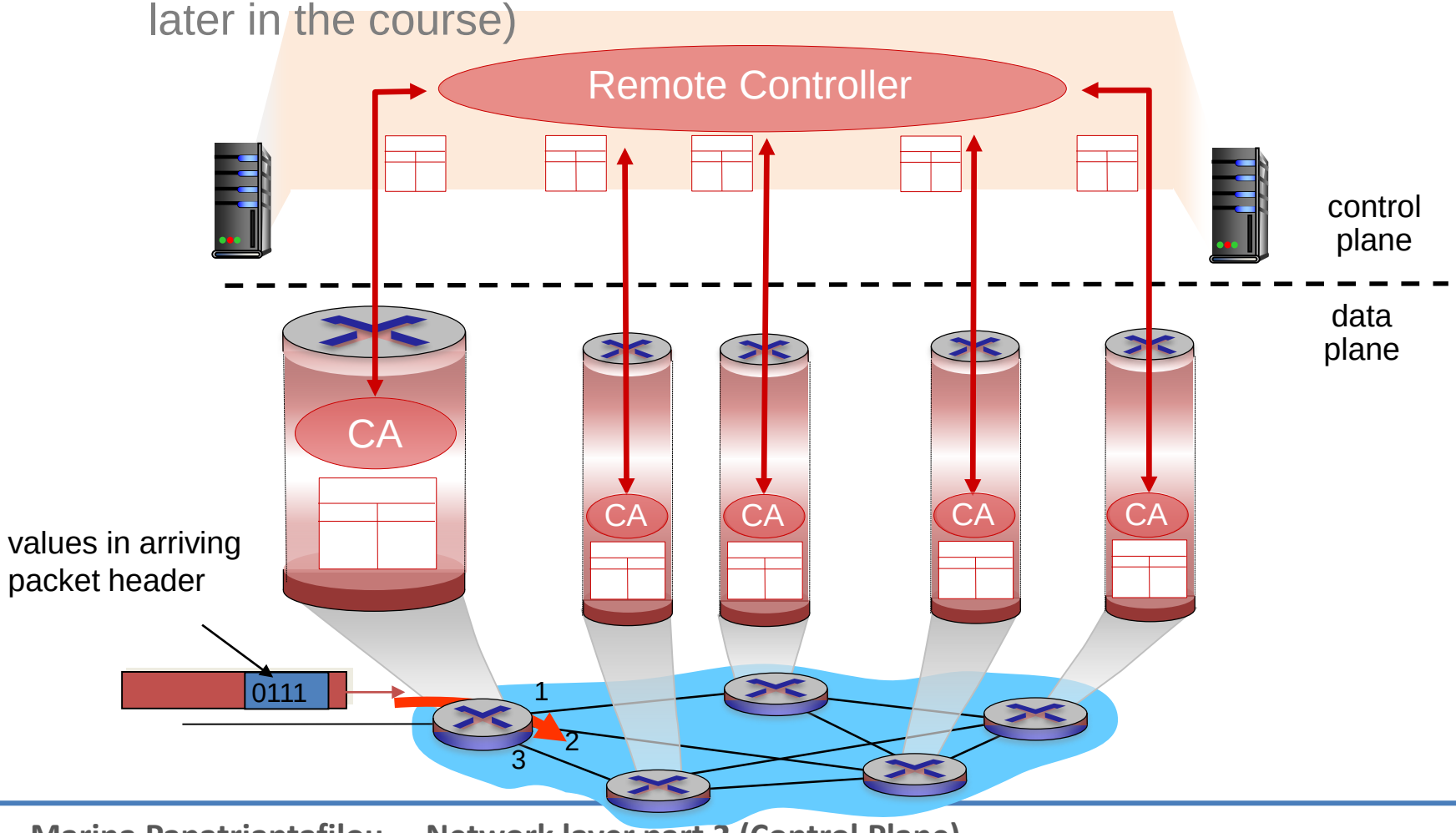
Performance:

- Inter-AS: policy may dominate over performance
- Intra-AS: can focus on performance

Recall SDN: Logically organized control plane

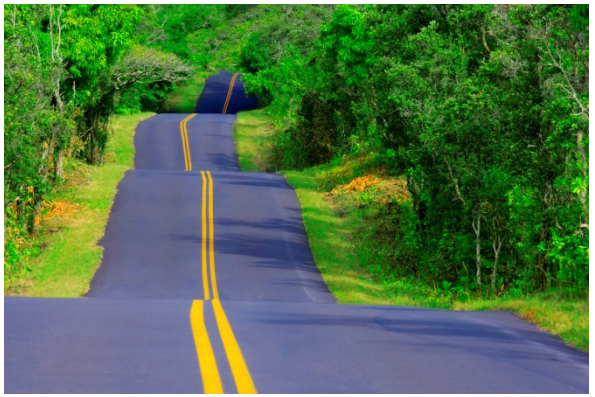
A distinct (can be remote/distributed) controller interacts with local control agents (CAs)

- this architecture (SDN) can enable new functionality (will be studied later in the course)



Roadmap Network Layer

- Forwarding versus routing
- Network layer service models
  - Network layer architecture (shift): Software-Defined Networks
- Inside a router
- The Internet Network layer: IP, Addressing & related



Control, routing

- path selection/routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- instantiation, implementation in the Internet routing protocols
  - RIP
  - OSPF
  - BGP
- ICMP (control protocol)

ICMP: Internet Control Message Protocol

- **Control and error messages** from network layer.
- All IP implementations must have ICMP support.
- ICMP messages carried in IP datagrams
- used by hosts & routers to communicate network-level control information and error reporting
  - Error reporting: e.g., unreachable network, host, ..
  - Example: (used by **ping** command)
    - Sends ICMP echo request
    - Receives ICMP echo reply
- Any ICMP error message may **never** generate a new one.

ICMP: internet control message protocol

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

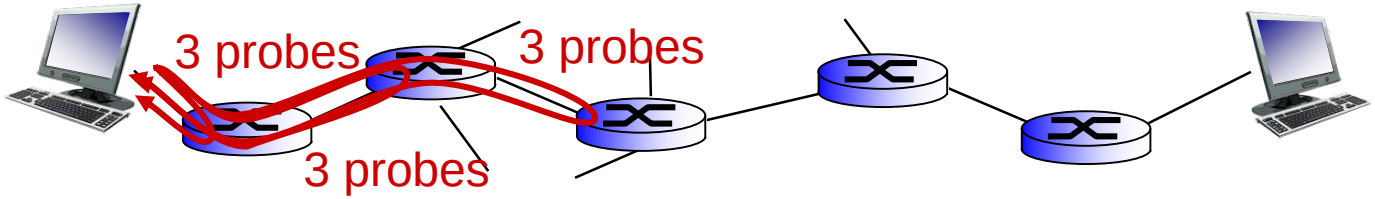
Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement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

Traceroute and ICMP

- source sends series of UDP segments (=probes) to destination
  - first set has TTL =1,second TTL=2, etc.
- when datagram in *n*th set arrives to n-th router:
  - router discards it and reports error (TTL expired) sends source ICMP message (type 11, code 0)
  - ICMP message include name of router & IP address
- when ICMP message arrives, source records RTTs

stopping criteria:

- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops



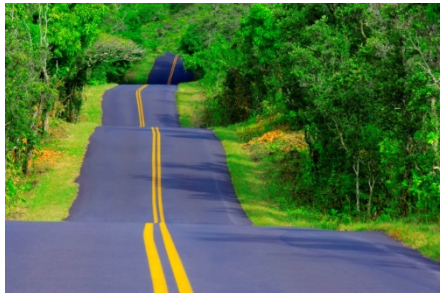
Summary Network Layer

- Forwarding versus routing
- Network layer service models
  - Network layer architecture (shift): Software-Defined Networks
- Inside a router
- The Internet Network layer: IP, Addressing & related

Control, routing

- path selection/routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- instantiation, implementation in the Internet routing protocols
  - RIP
  - OSPF
  - BGP
- ICMP (control protocol)

- **NEXT: Link Layer**



Reading instructions Network Layer (incl. prev. lecture)

- **KuroseRoss book**

Careful	Quick
5/e,6/e: 4.1-4.6 7/e: 4.1-4.3, 5.2-5.4, 5.5, 5.6, *[new- SDN, data and control plane 4.4, 5.5: in subsequent lectures, connecting to multimedia/streaming Study material through the pingpong-system]*	5/e,6/e: 4.7, 7/e: 5.7

Some complementary material /video-links

- IP addresses and subnets
  http://www.youtube.com/watch?v=ZTJIkjgyuZE&list=PLE9F3F05C381ED8E8&feature=plcp

- How does PGP choose its routes
  http://www.youtube.com/watch?v=RGe0qt9Wz4U&feature=plcp

Some taste of layer 2: no worries if not all details fall in place, need the lectures also to grasp them.

- Hubs, switches, routers
  http://www.youtube.com/watch?v=reXS_e3fTAk&feature=related
-
- What is a broadcast + MAC address
  http://www.youtube.com/watch?v=BmZNcjLtmwo&feature=plcp

- Broadcast domains:
  http://www.youtube.com/watch?v=EhJO1TCQX5I&feature=plcp

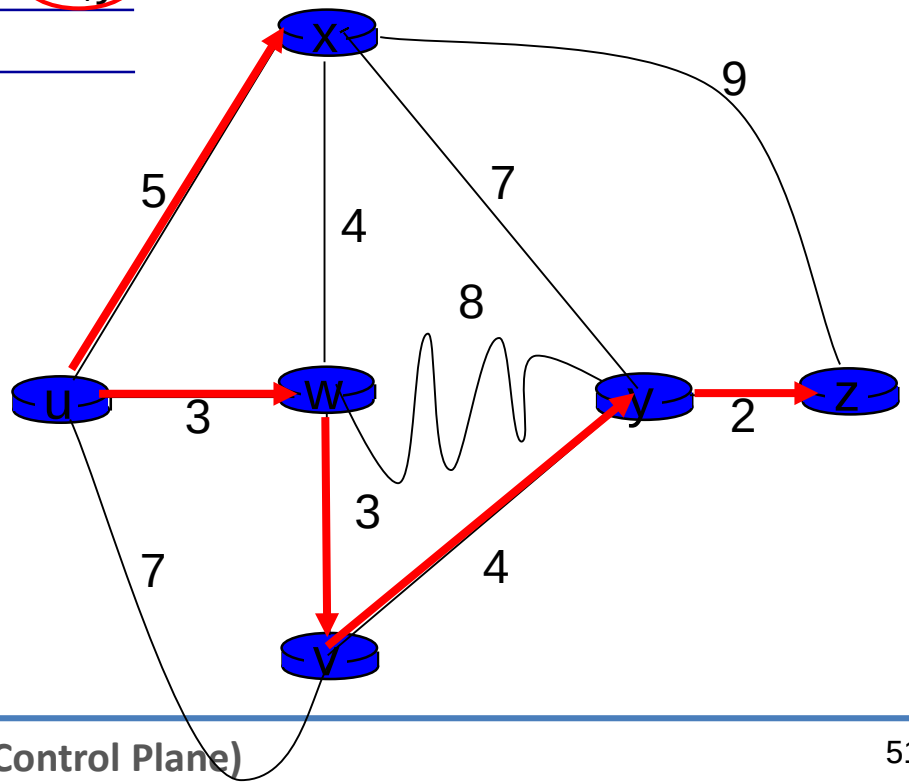
Extra slides

Dijkstra's algorithm: example

Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

notes:

- construct shortest path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)



Distance vector (DV) algorithm

Basic idea:

- distributed, asynchronous, iterative
- from time-to-time, each node sends to neighbors only its own distance vector DV estimate
- When a node x receives new DV estimate from neighbor v, it updates its own DV using Bellman-Ford equation:

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

- Under normal conditions, when information comes in about new link costs:
  - The estimate $d_x(y)$ *converge to the actual least cost*
  - Routing table recalculated
  - New results sent out to all neighbors

RIP advertisements

- distance vectors are exchanged among neighbors every 30 sec via Response Message (also called advertisement)
- each advertisement: list of up to 25 destination subnets within AS
- If no advertisement heard after 180 sec → neighbor or link declared dead (unreachable).
  - Routes via neighbor invalidated
  - New advertisements sent to other neighbors
  - Link failure info propagates to entire network
  - Poisoned reverse used with max hop count 15
  - Infinite distance is 16 hops
- RIP v.2 also supports route aggregation (1998)

Hierarchical OSPF

- two-level hierarchy: local areas, one backbone (area 0).
  - Link-state advertisements only in area
  - each node has detailed area topology; only knows direction (shortest path) to subnets in other areas.
- *area border routers:* "summarize" subnets in own area, advertise to other area border routers.
- *backbone routers:* run OSPF routing limited to backbone.
- *boundary routers:* connect to other AS's.

BGP messages

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

Getting a datagram from source to dest.

Getting a datagram from source to dest.

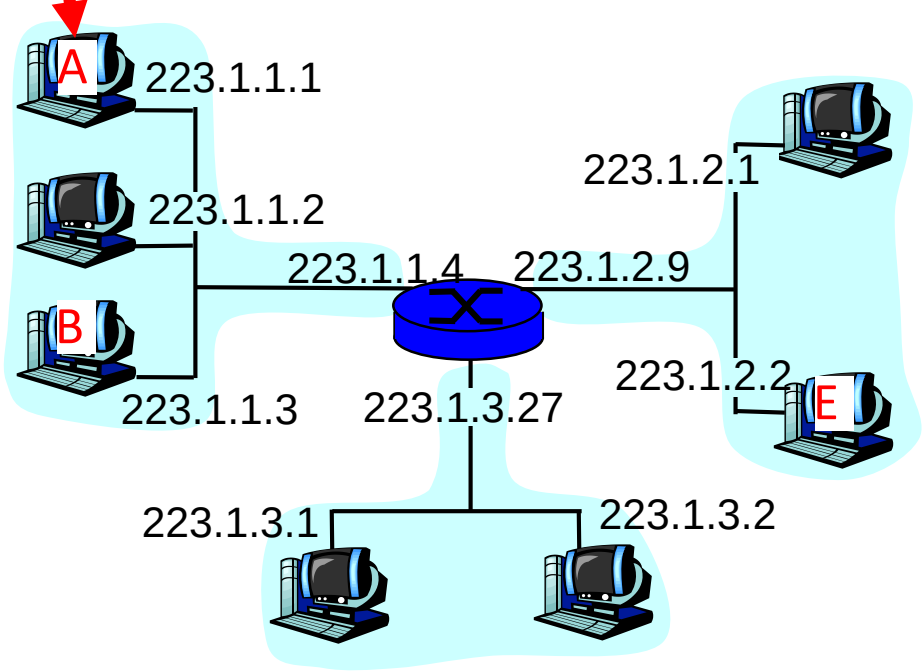
forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

IP datagram:

misc fields	source IP addr	dest IP addr	data

- datagram remains unchanged, as it travels source to destination
- addr fields of interest here



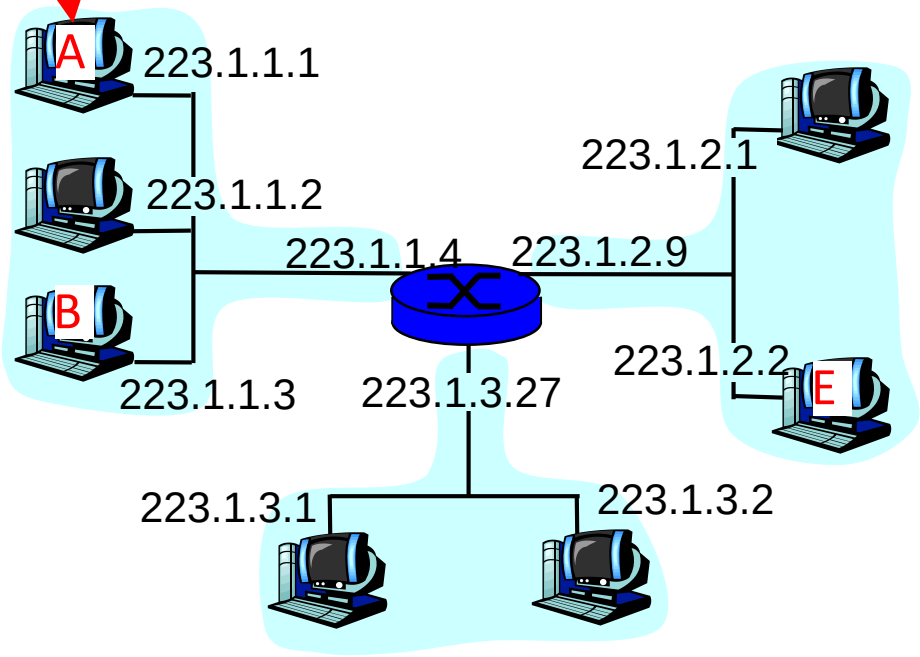
Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.1.3	data

Starting at A, given IP datagram addressed to B:

- look up net. address of B
- find B is on same net. as A (B and A are directly connected)
- link layer will send datagram directly to B (inside link-layer frame)

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



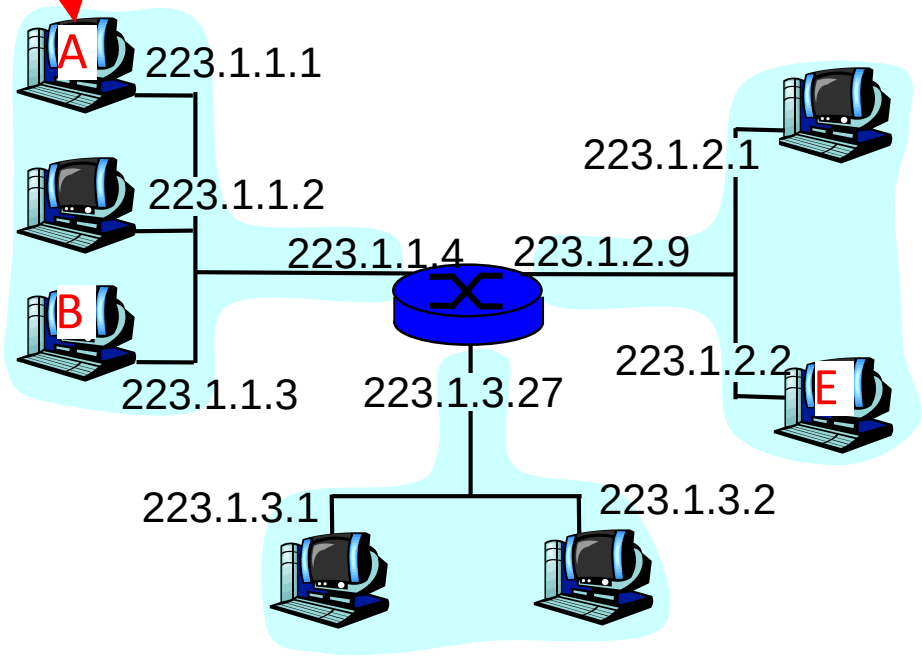
Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.2.3	data

Starting at A, dest. E:

- look up network address of E
- E on *different* network
- routing table: next hop router to E is 223.1.1.4
- link layer is asked to send datagram to router 223.1.1.4 (inside link-layer frame)
- datagram arrives at 223.1.1.4
- continued…..

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.2.3	data

Arriving at 223.1.4, destined for 223.1.2.2

- look up network address of E
- E on *same* network as router's interface 223.1.2.9
  - router, E directly attached
- link layer sends datagram to 223.1.2.2 (inside link-layer frame) via interface 223.1.2.9
- datagram arrives at 223.1.2.2!!! (hooray!)

Dest. network	next router	Nhops	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27