This lecture will be recorded!!!

Welcome to

CS 3516: *Computer Networks*

Prof. Yanhua Li

Time: 9:00am –9:50am M, T, R, and F
Zoom Lecture
Fall 2020 A-term

Quiz 8 tomorrow 10/9

with 2 bonus points

Topics: subnet, IPv6, DV routing

Project 3 due next Thursday 10/15

Extra office hours

To be announced!

Project 2

Grading will be done by this Sat 10/10

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

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

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

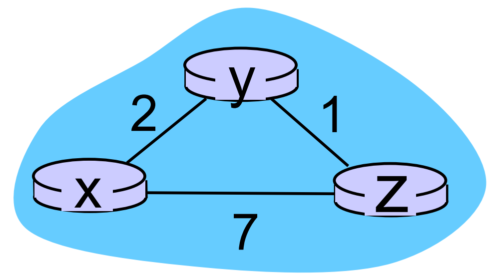
node z table

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 →



Distance vector 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:

wait for (change in local link cost or msg from neighbor)

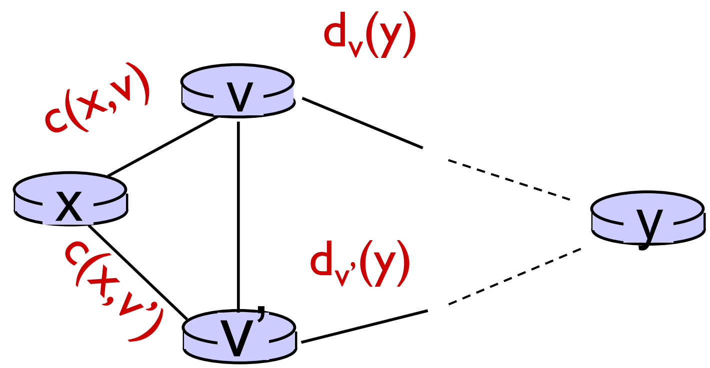
recompute estimates

if DV to any dest has changed, *notify* neighbors

Chapter 5: outline

5.2 routing algorithms

- distance vector
- link state



A Link-State Routing Algorithm

Dijkstra's algorithm

- net topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ("source") to all other nodes
  - gives *forwarding table* for that node
- iterative:
  - after k iterations, know least cost path to k nearest dest.'s

notation: given src u

- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

Dijkstra's Algorithm

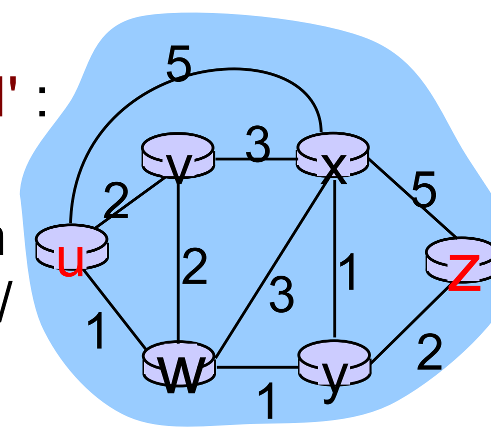
```
 Collect global topology info
 Initialization:
   N' = {u}
   for all nodes v
     if v adjacent to u
        then D(v) = c(u,v)
     else D(v) = ∞

 Loop
   find w not in N' such that D(w) is a minimum
   add w to N'
   update D(v) for all v adjacent to w and not in N' :
      D(v) = min( D(v), D(w) + c(w,v) )
   /* new cost to v is either old cost to v or known
    shortest path cost to w plus cost from w to v */
 until all nodes in N'
```

1 hop

k hops

notation: given src u

- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known



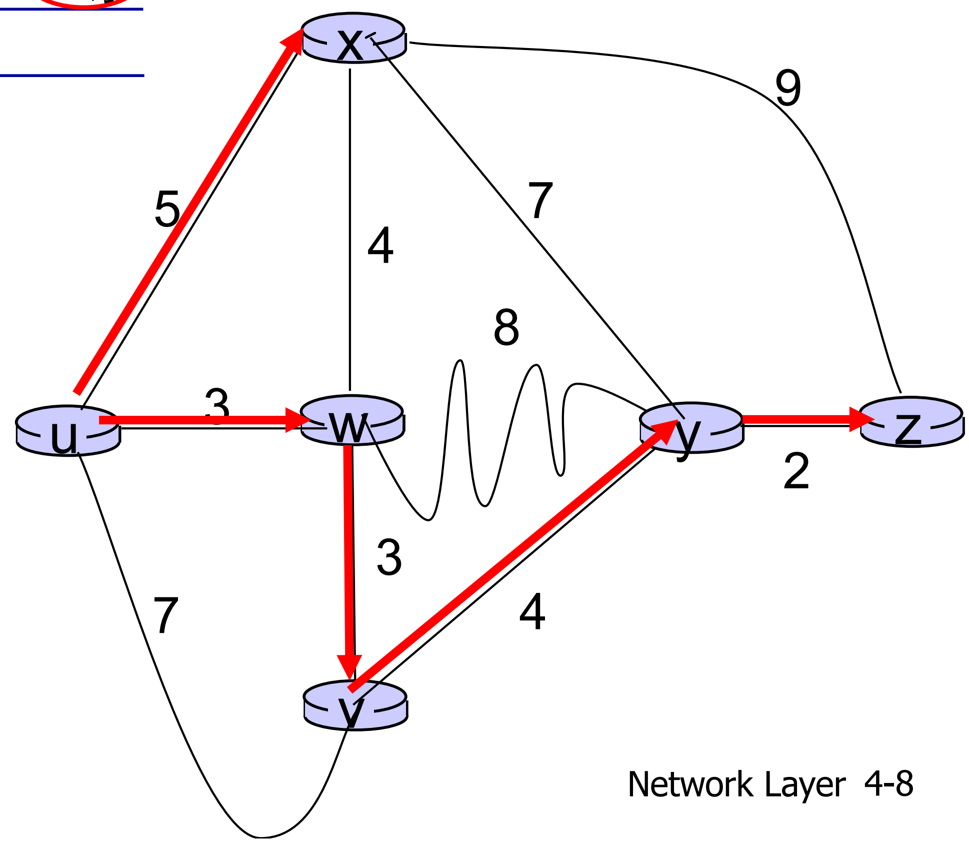
Dijkstra's algorithm: example

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

$D(v) = \min(D(v), D(w) + c(w,v))$

notes:

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



Chapter 4-5: outline

4.1 introduction

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

IPv6: motivation

- *initial motivation:* 32-bit address space soon to be completely allocated.
- additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate Quality of Services (QoS)

IPv6 address

128 bits (16 bytes)

0123 : 4567 : 89ab : cdef : 0123 : 4567 : 89ab : cdef

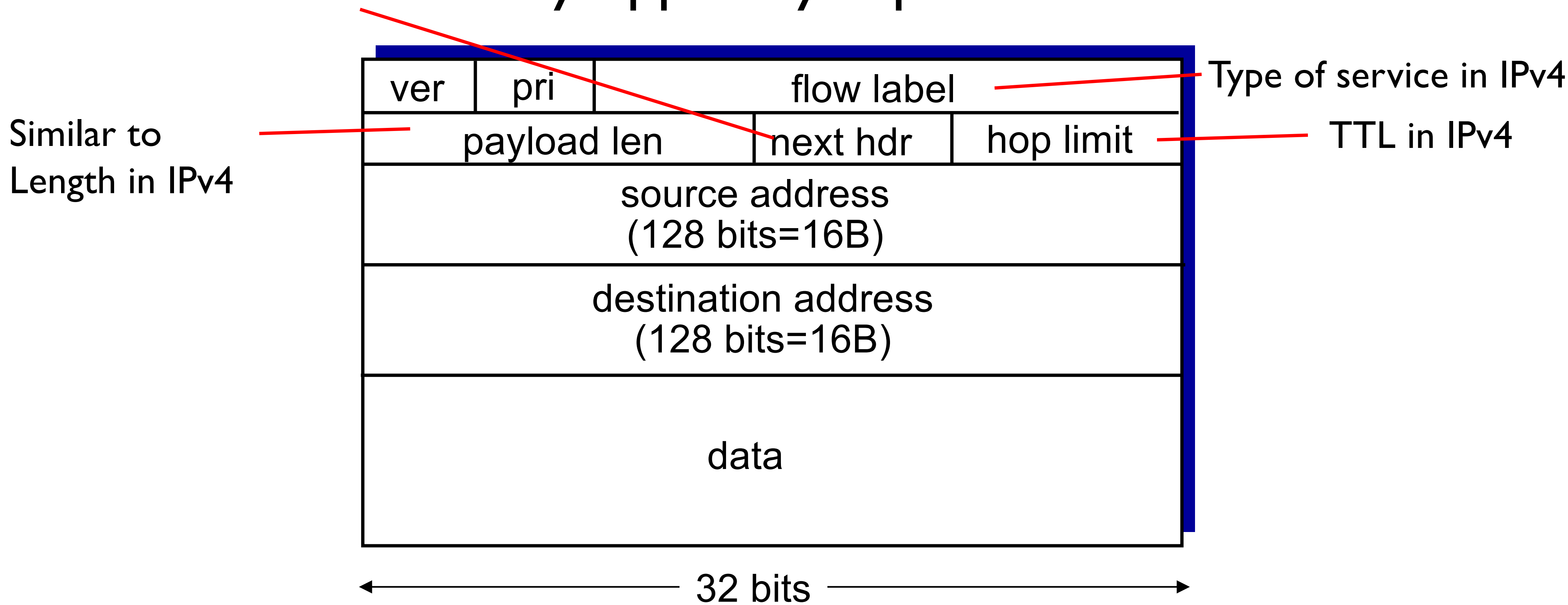
Hexadecimal notation

IPv6 datagram format:

- fixed-length 40 byte header

IPv6 datagram format (40 bytes)

priority: identify priority among datagrams in flow
flow Label: identify datagrams in same "flow."
(concept of "flow" not well defined).
next header: identify upper layer protocol for data



Other changes from IPv4

- *checksum:* removed entirely, because
  - To enable **fast processing** of IP datagrams at the network layer
    - TTL change leads to change of checksum each router
  - To reduce redundancy, since checksums are available at other layers.

Quiz 9 next Wed 10/14

Any 15 mins during 8AM-5PM

with 1 bonus point

Consider taking Quiz 9 during TA offices so you can ask questions via Zoom:

9:30AM-11AM by Heshan,

1-2:30PM by Menghai

Final Exam Friday 10/16

50 mins at 9-9:50AM (only cover topics after the mid-term)

with 32 points + 2 bonus points

We will provide you sample questions and answers in the weekend.

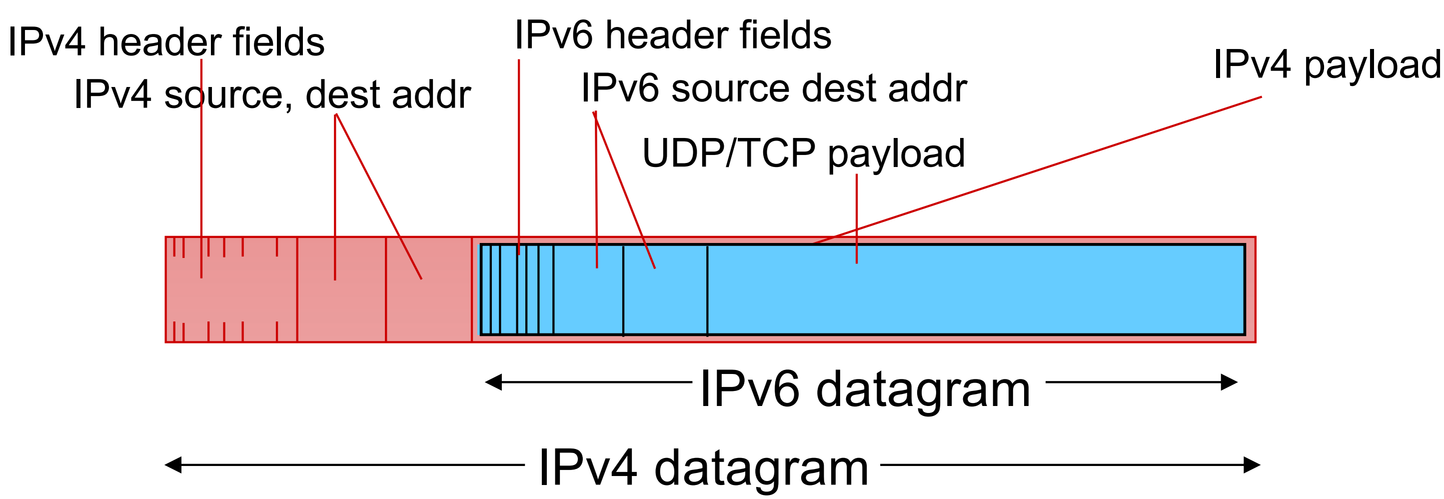
We will also post an offline video for final exam review next Tuesday 10/13.

Optional in-person Q&A session in AK219 at 9-9:50AM Thursday 10/15

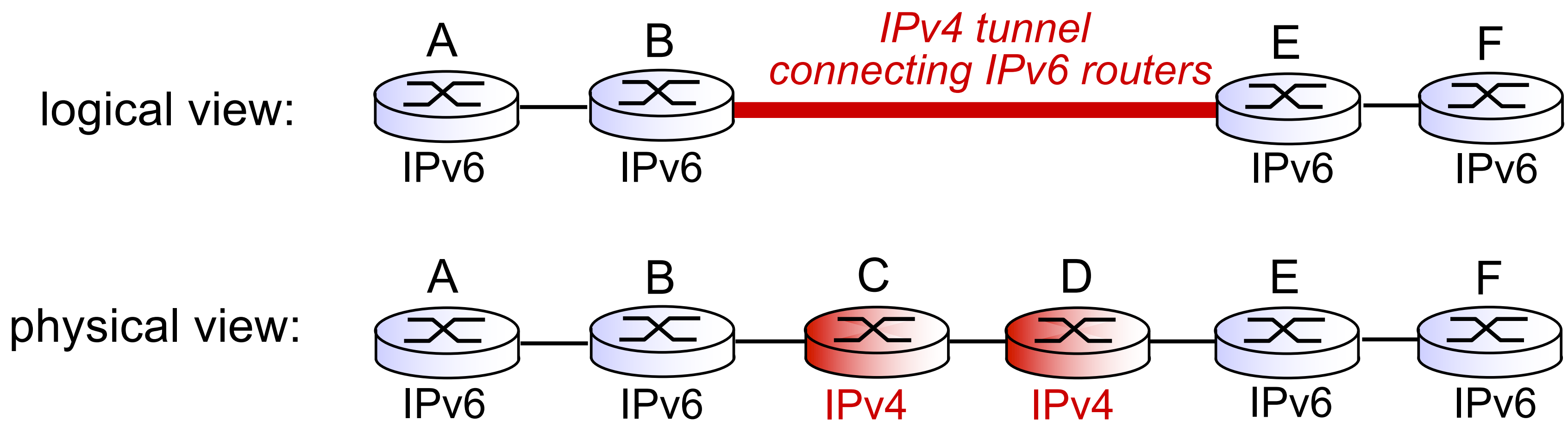
We will send a survey next week for whether you plan to attend this session. We will plan the session arrangement based on the survey results.

Transition from IPv4 to IPv6

- not all routers can be upgraded simultaneously
  - no "flag days"
  - how will network operate with mixed IPv4 and IPv6 routers?
- *tunneling:* IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers

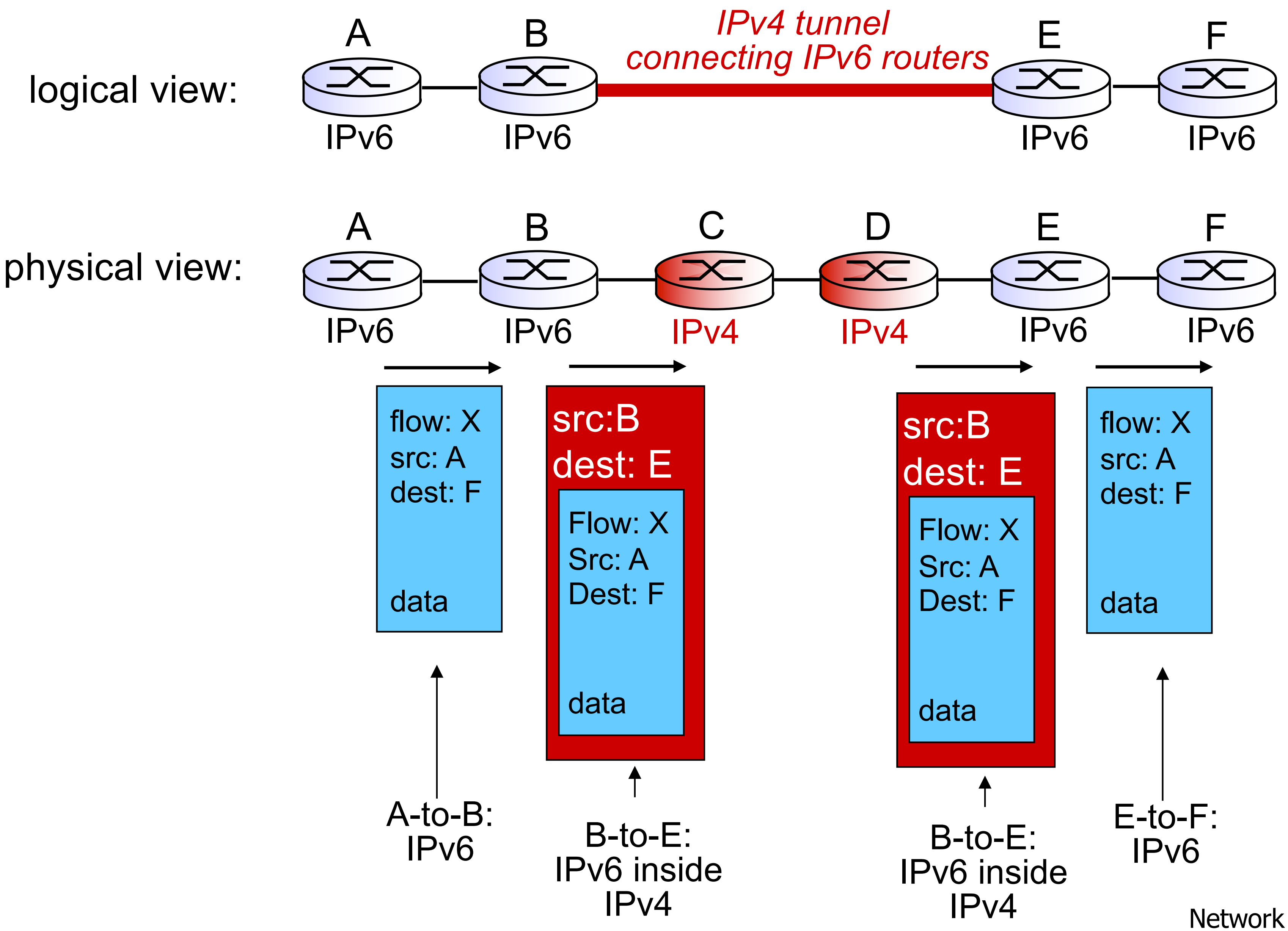


Tunneling



logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

Tunneling



IPv4 tunnel
connecting IPv6 routers
logical view:
A
B
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
A-to-B:
IPv6
B-to-E:
IPv6 inside
IPv4
B-to-E:
IPv6 inside
IPv4
E-to-F:
IPv6

IPv6: adoption

- US National Institutes of Standards estimate:
  - ~3% of industry IP routers
  - ~11% of US gov't routers
- *Long (long!) time for deployment, use*
  - 20 years and counting!
  - think of application-level changes in last 20 years: WWW, Facebook, …

Chapter 4: outline

4.1 introduction

4.4 IP: Internet Protocol

- datagram format
- IPv4 addressing
- ICMP
- IPv6

4.5 routing algorithms

- link state
- distance vector

4.6 routing in the Internet

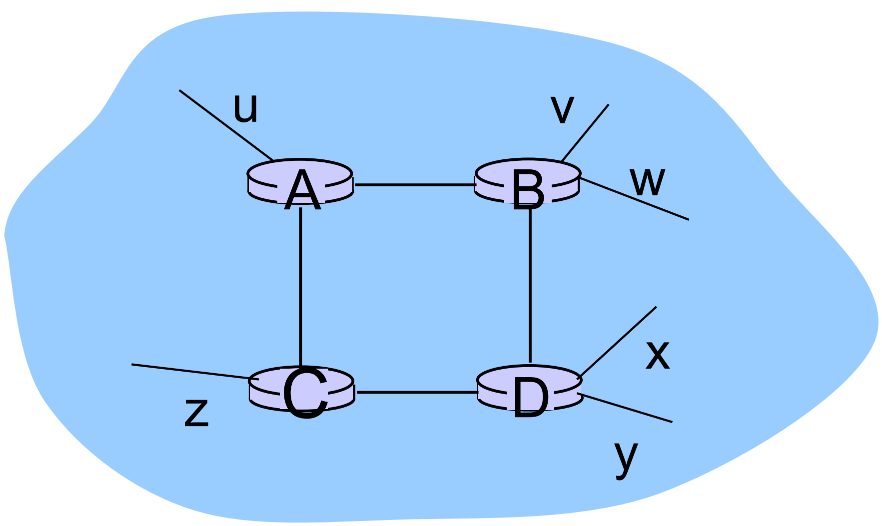
- RIP
- OSPF
- BGP

OSPF (Open Shortest Path First)

- "open": publicly available
- uses link state algorithm
  - LS packet dissemination; topology map at each node
  - Dijkstra's algorithm
- OSPF advertisement
- advertisements flooded to *entire* network (or Autonomous System, AS)
  - carried in OSPF messages directly over IP (rather than TCP or UDP
- Multiple same-cost paths allowed

RIP (Routing Information Protocol)

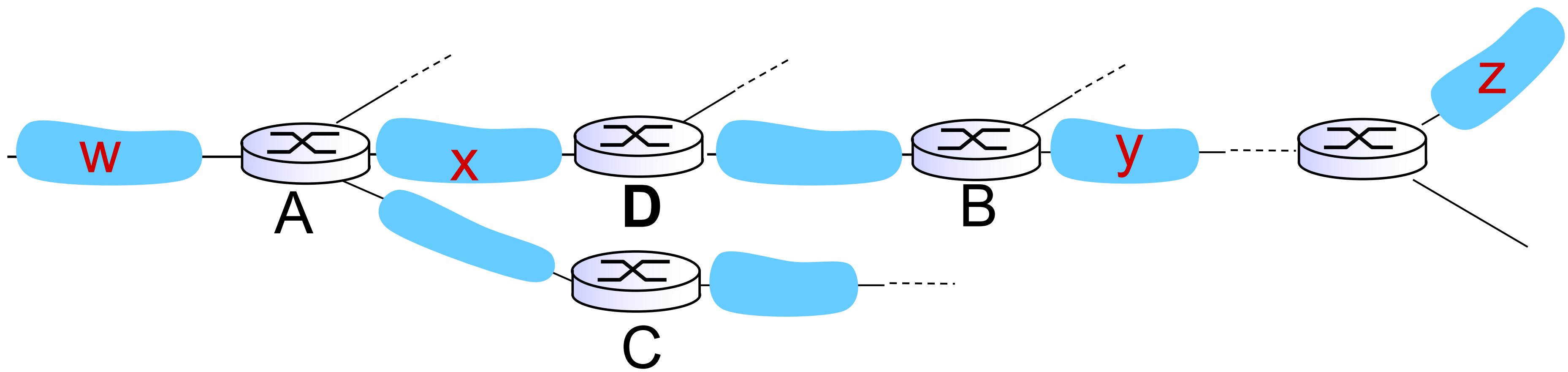
- *distance vector algorithm in an AS*
  - distance metric: # hops (*max = 15 hops*),
  - each link has cost 1
  - DVs exchanged with neighbors every 30 sec in response message (aka advertisement)
  - Failure: if no advertisement heard after 180 sec --> neighbor/link declared dead



from router A to destination *subnets:*

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

RIP: example



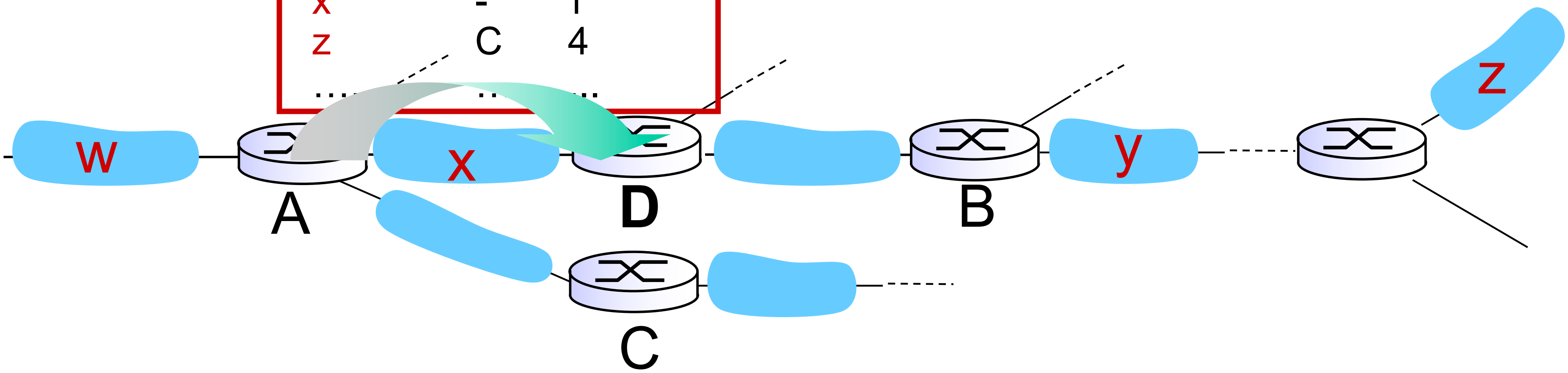
routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

RIP: example

A-to-D advertisement

dest	next	hops
w	-	1
x	-	1
z	C	4
….	…	....



routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
….	….	....

Done with intra-domain routing

4.5 routing algorithms

- link state
- distance vector

4.6 routing in the Internet

- OSPF
- RIP

Hierarchical routing

our routing study thus far - idealization
- all routers identical
- network "flat"

… *not* true in practice

scale: with 600 million destinations:
- can't store all dest's in routing tables!
- routing table exchange would swamp links!

administrative autonomy
- internet = network of networks
- each network admin may want to control routing in its own network

Hierarchical routing

- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol: RIP, OSPF.
  - routers in different AS can run different intra-AS routing protocol

gateway router:

- at "edge" of its own AS
- has link to router in another AS

Questions?