Welcome to

CS 3516: *Computer Networks*

Prof. Yanhua Li

Time: 9:00am –9:50am M, T, R, and F
Location: AK219
Fall 2018 A-term

Quiz 8 this Thursday 10/4 with bonus

On subnet, IPv6, routing algorithm

Quiz 7

Grades are ready

Project 3 due next Tue 10/9

Example log is available on Canvas.

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

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

```
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

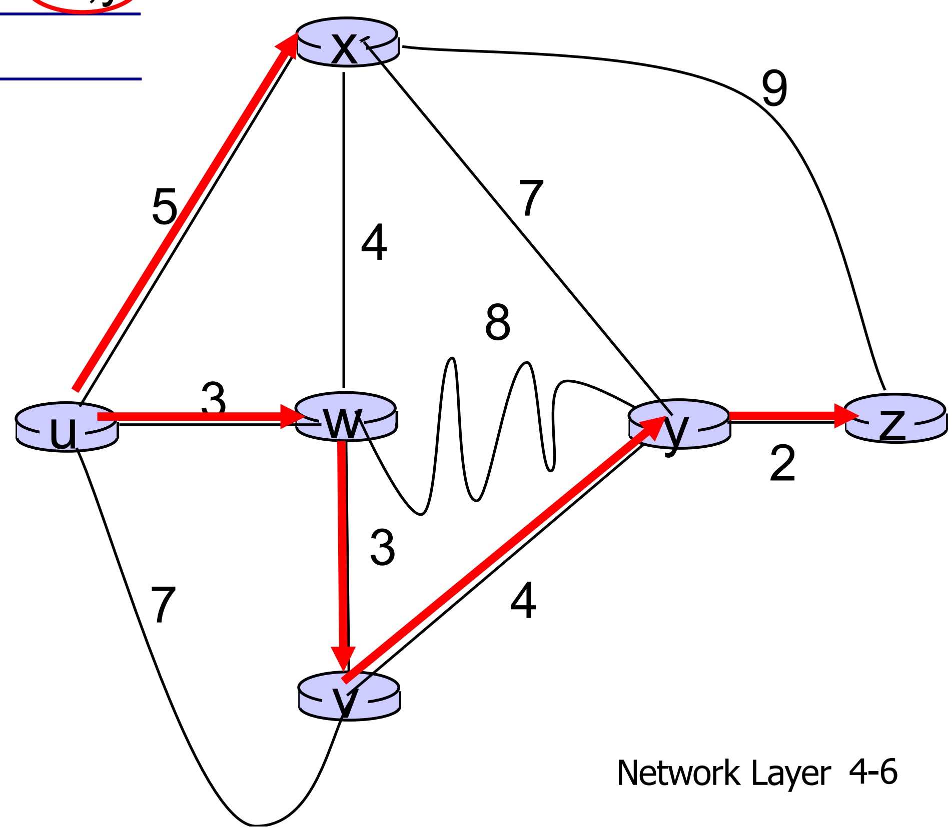
Dijkstra's algorithm: example

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					

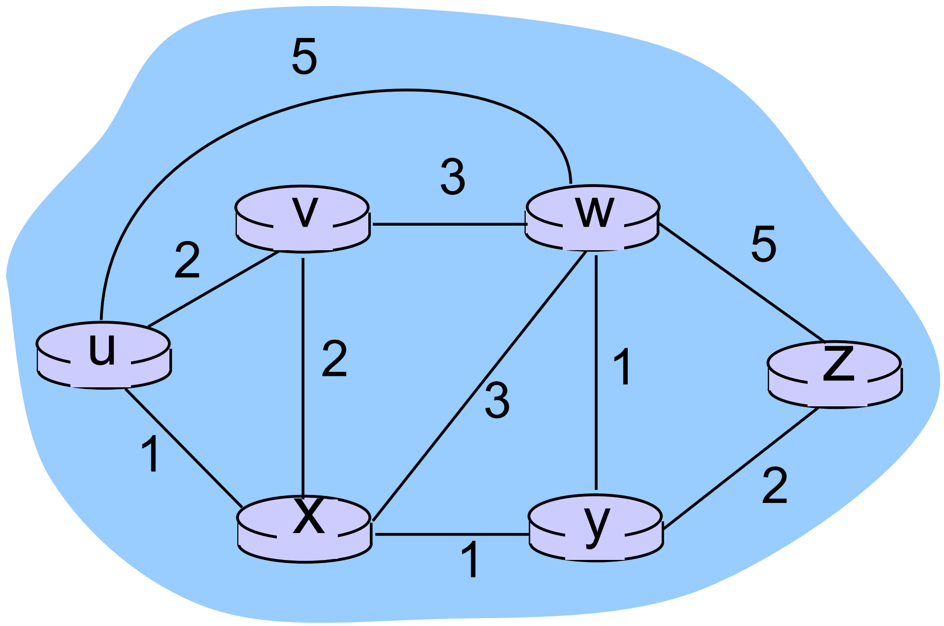
D(v) = min(D(v), D(w) + c(w,v))

notes:

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

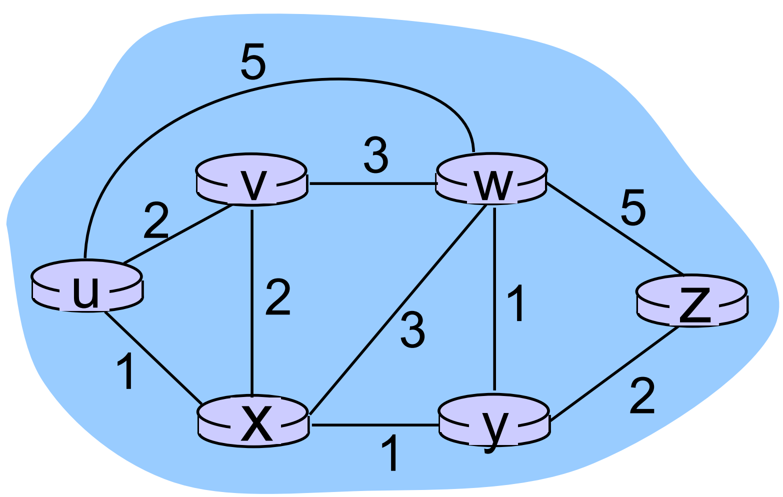


Dijkstra's algorithm: another example for you to practice offline



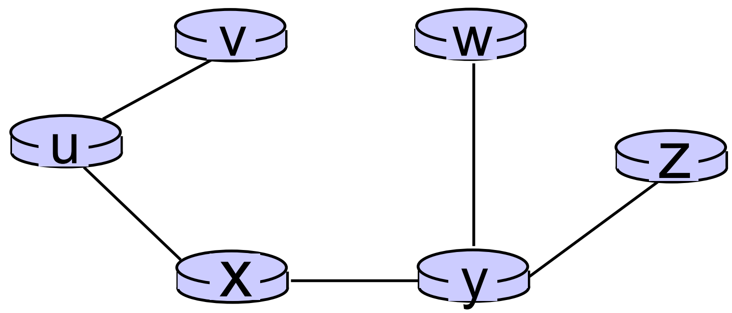
Dijkstra's algorithm: another example

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Dijkstra's algorithm: example (2)

resulting shortest-paths from u:



resulting forwarding table in u:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

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
- hierarchical routing

4.6 routing in the Internet
- RIP
- OSPF
- BGP

OSPF (Open Shortest Path First)

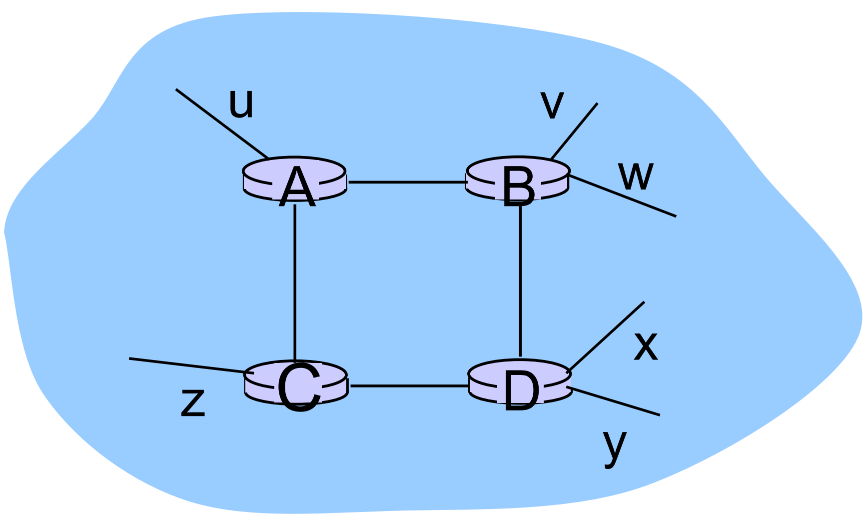
- "open": publicly available

- uses link state algorithm
  - LS packet dissemination; topology map at each node
  - Dijkstra's algorithm

- OSPF advertisement
- advertisements flooded to *entire* network (or AS)
  - carried in OSPF messages directly over IP (rather than TCP or UDP
- Multiple same-cost paths allowed

RIP (Routing Information Protocol)

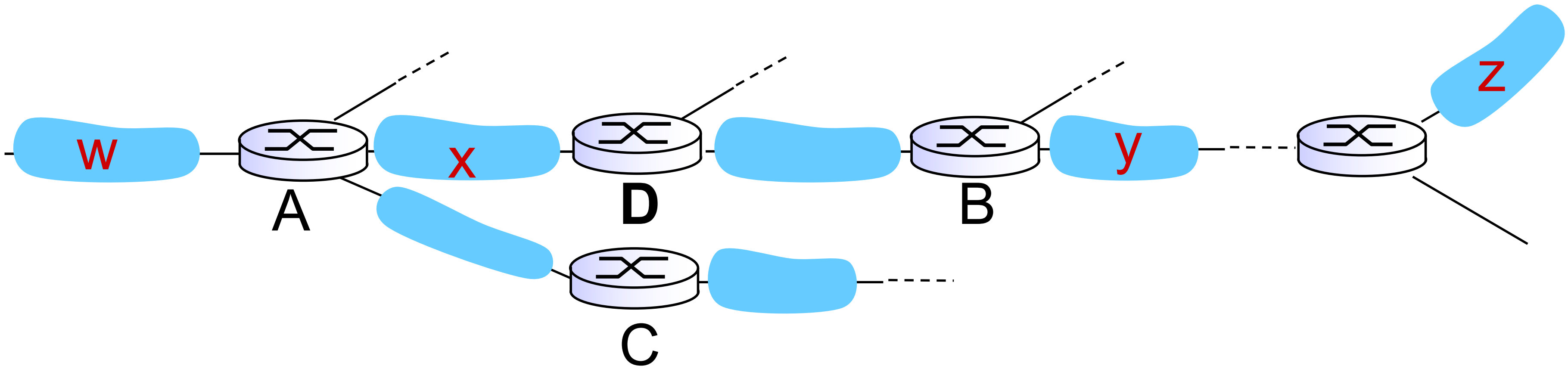
- *distance vector algorithm*
  - distance metric: # hops (*max = 15 hops*),
  - each link has cost 1
  - DVs exchanged with neighbors every 30 sec in response message (aka advertisement)
  - Failure: if no advertisement heard after 180 sec --> neighbor/link declared dead



from router A to destination *subnets:*

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

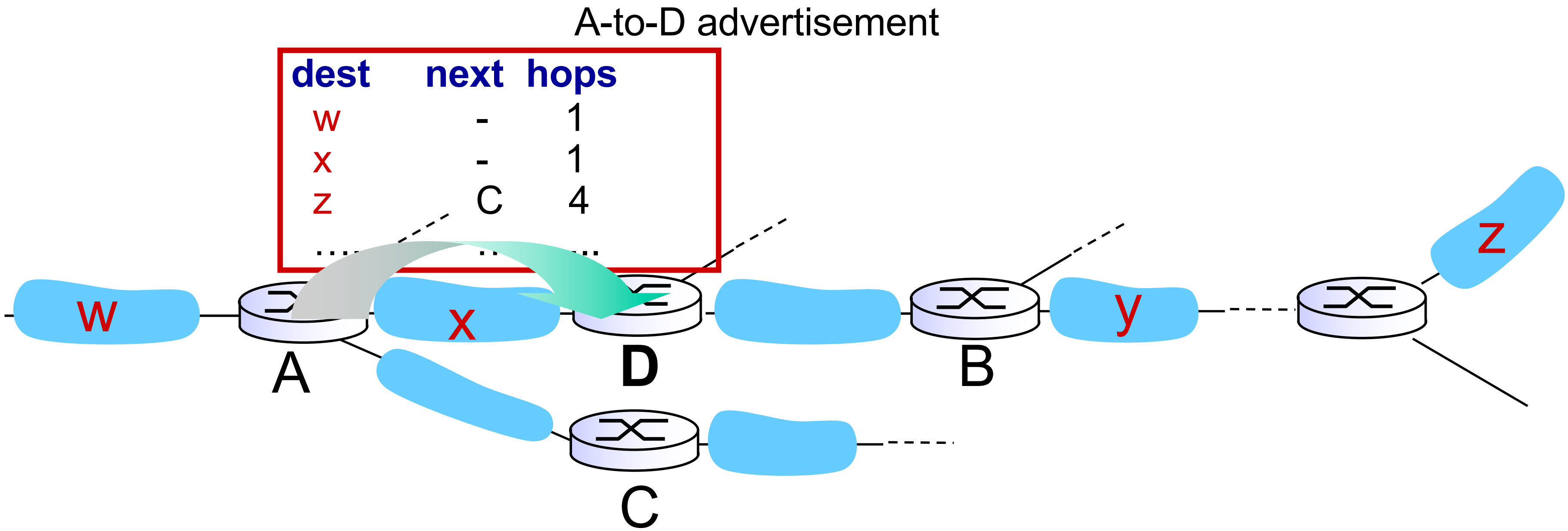
RIP: example



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



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

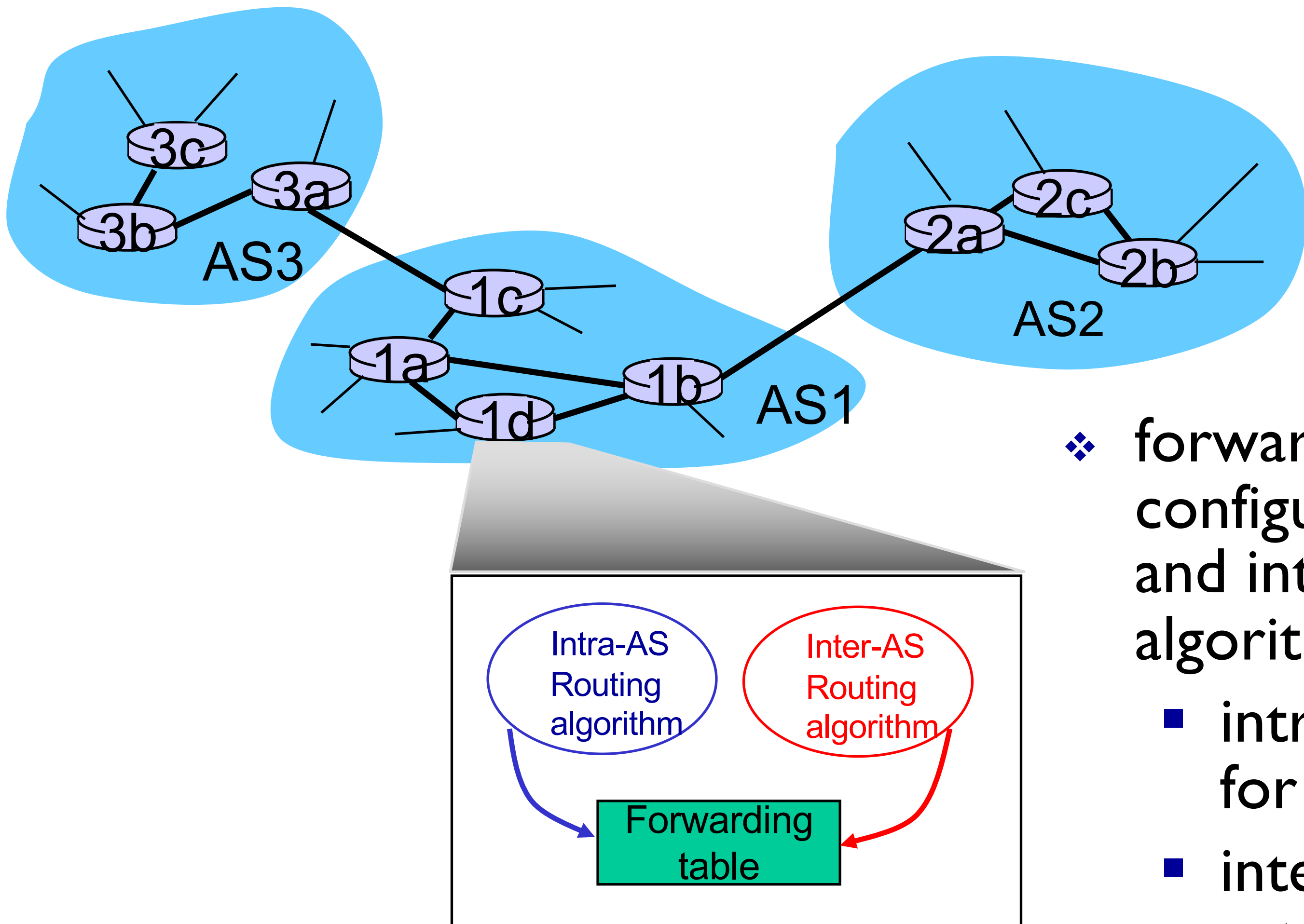
Hierarchical routing

- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol
  - 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

Interconnected ASes



- forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS sets entries for internal dests
  - inter-AS & intra-AS sets entries for external dests

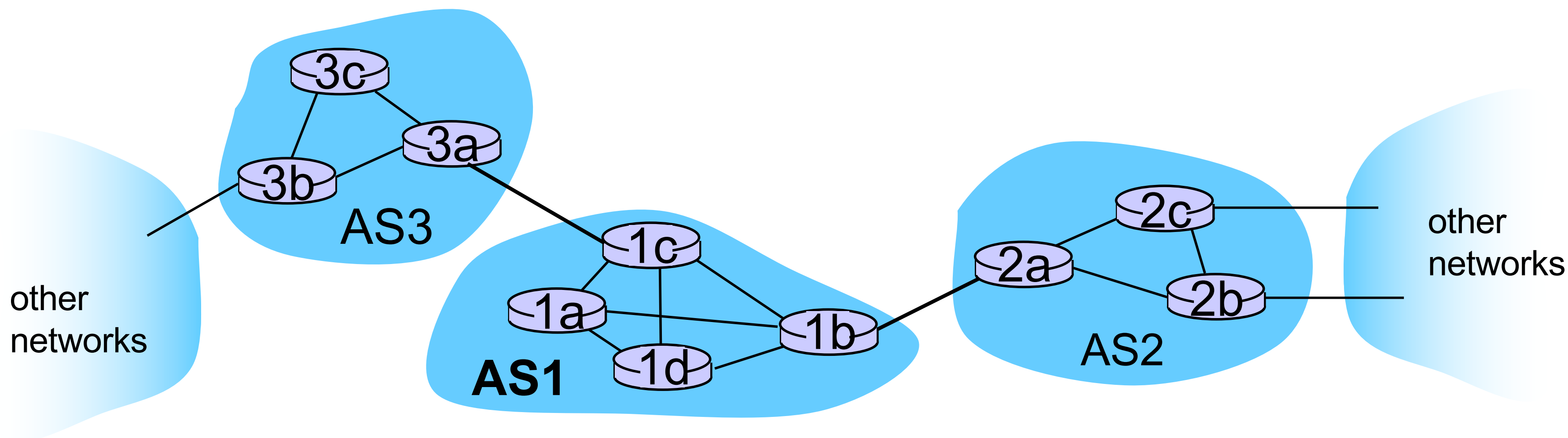
Inter-AS tasks

- suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router, but which one?

AS1 must:

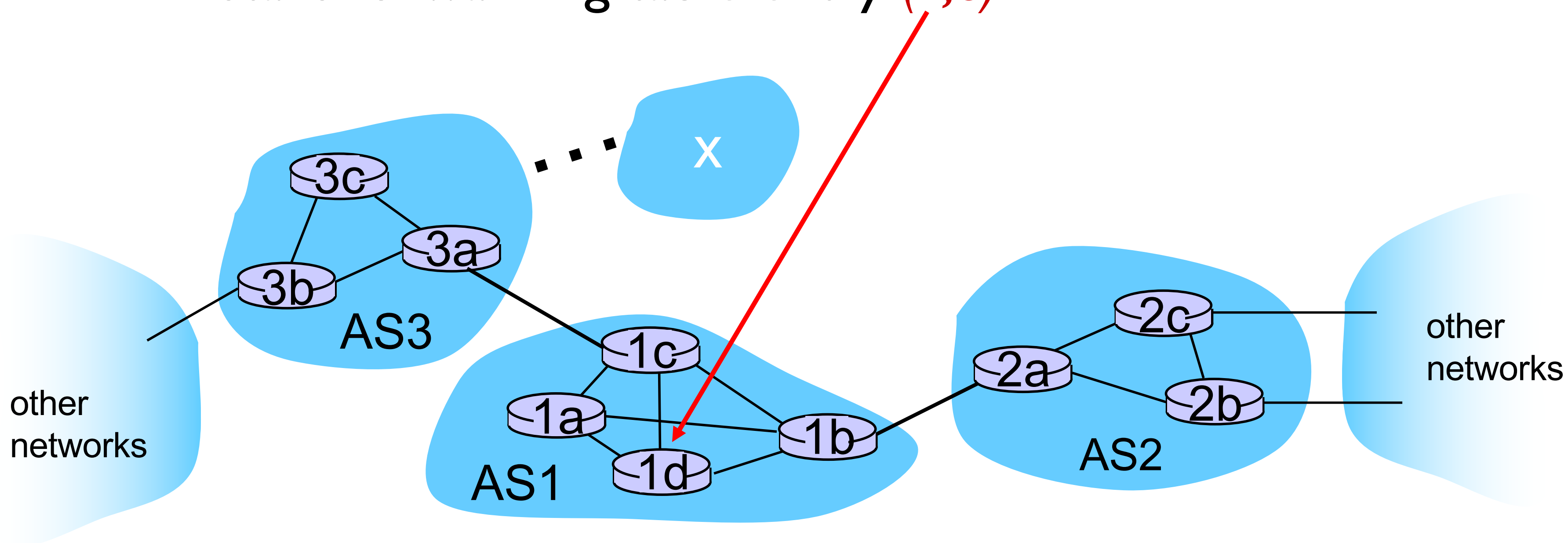
1. learn which dests are reachable through AS2, which through AS3
2. propagate this reachability info to all routers in AS1

job of inter-AS routing!



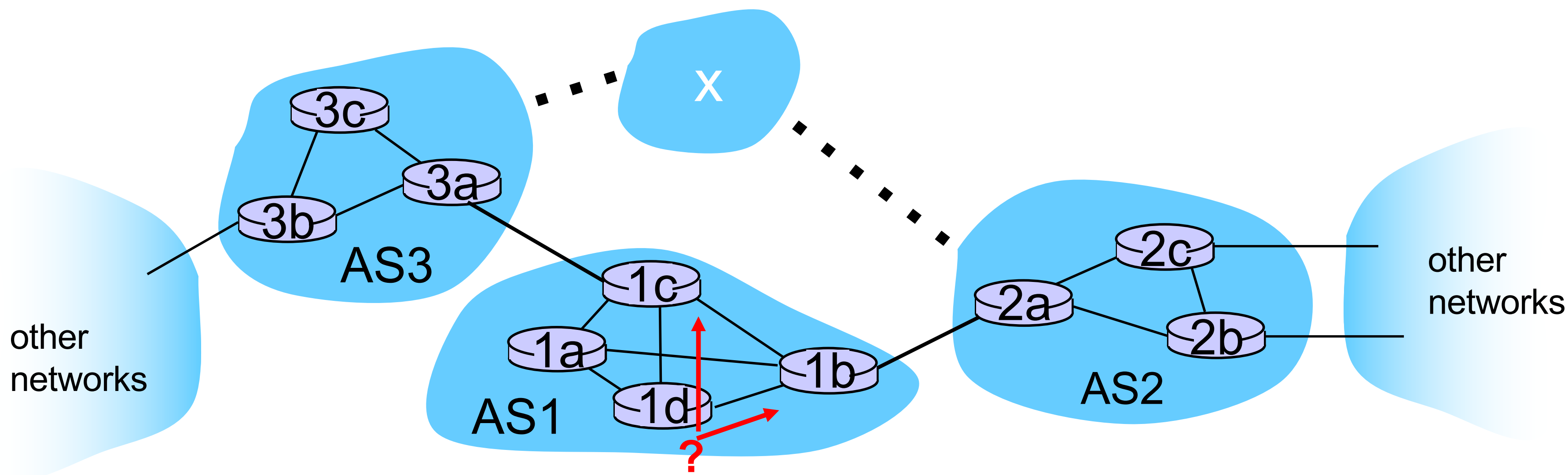
Example: setting forwarding table in router 1d

- suppose AS1 learns (via inter-AS protocol) that subnet *x* reachable via AS3 (gateway 1c), but not via AS2
  - inter-AS protocol propagates reachability info to all internal routers
- router 1d determines from intra-AS routing info that its interface e is on the least cost path to 1c
  - installs forwarding table entry *(x,e)*



Example: choosing among multiple ASes

- now suppose AS1 learns from inter-AS protocol that subnet *x* is reachable from AS3 *and* from AS2.
- to configure forwarding table, router 1d must determine which gateway it should forward packets towards for dest x
  - this is also job of inter-AS routing protocol!
  - *hot potato routing: send* packet towards closest of two routers.



Intra-AS Routing

- also known as *interior gateway protocols (IGP)*
- most common intra-AS routing protocols:
  - **RIP:** Routing Information Protocol-- DVR
  - **OSPF:** Open Shortest Path First-- LSR
  - **EIGRP:** Enhanced Interior Gateway Routing, by Cisco, DVR
  - …

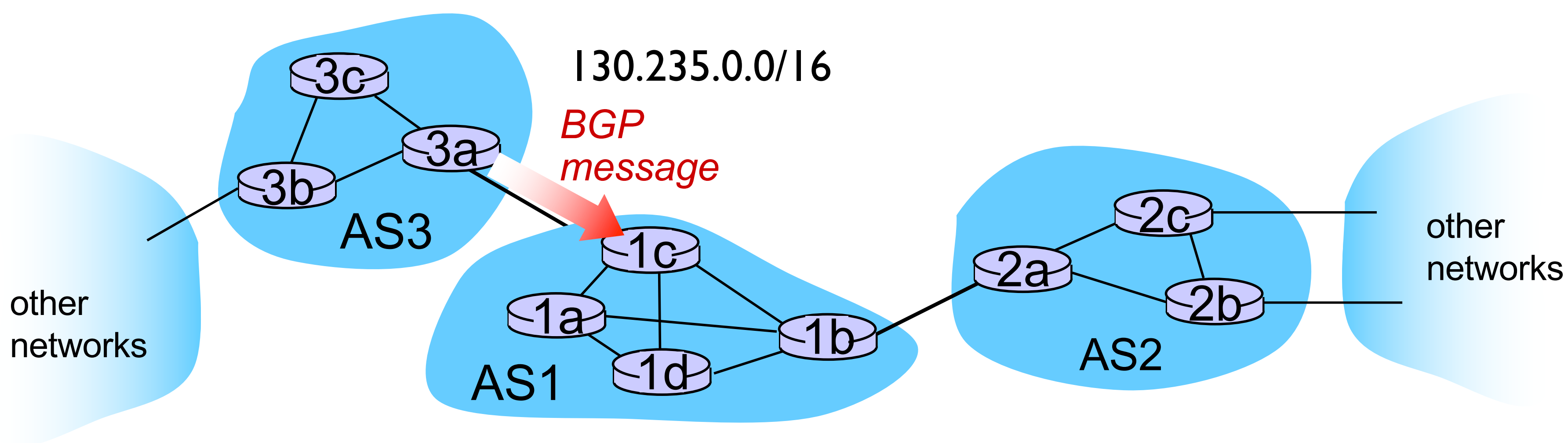
Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- BGP provides each AS a means to: (external vs interior)
  - eBGP: obtain subnet reachability information from neighboring ASs.
  - iBGP: propagate reachability information to all AS-internal routers.
  - determine "good" routes to other networks based on ***reachability information and policy***.
- allows subnet to advertise its existence to rest of Internet: *"I am here"*

BGP basics

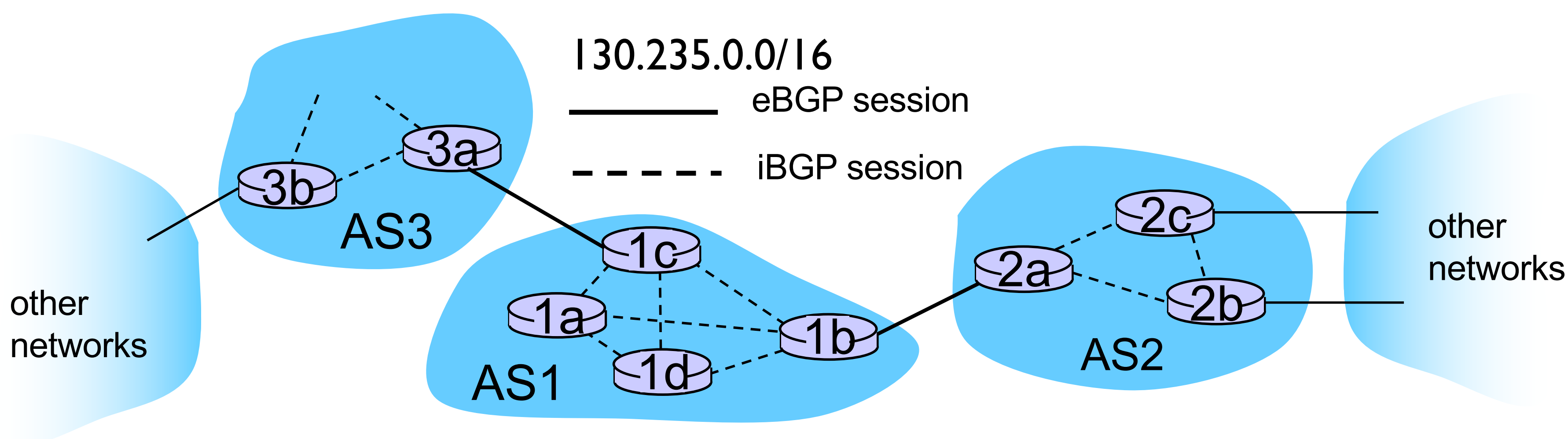
- BGP session: two BGP routers ("peers") exchange BGP messages:
  - advertising *paths* to different destination network prefixes

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



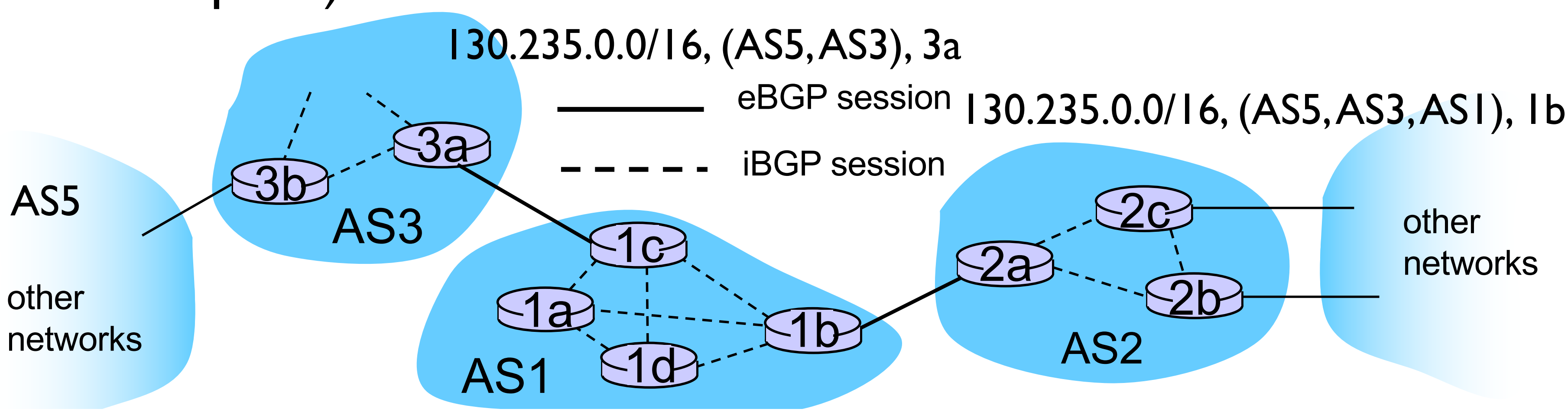
BGP basics: distributing path information

- using eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
  - 1c can then use iBGP do 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 and BGP routes

- advertised prefix includes BGP attributes
  - prefix + attributes = "route"
- two important attributes:
  - AS-PATH: contains ASs through which prefix advertisement has passed: e.g., AS 67, AS 17;
  - AS numbers are maintained by ICANN;
  - NEXT-HOP: indicates specific internal-AS router to next-hop AS. (may be multiple links from current AS to next-hop-AS)



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 datagram format:

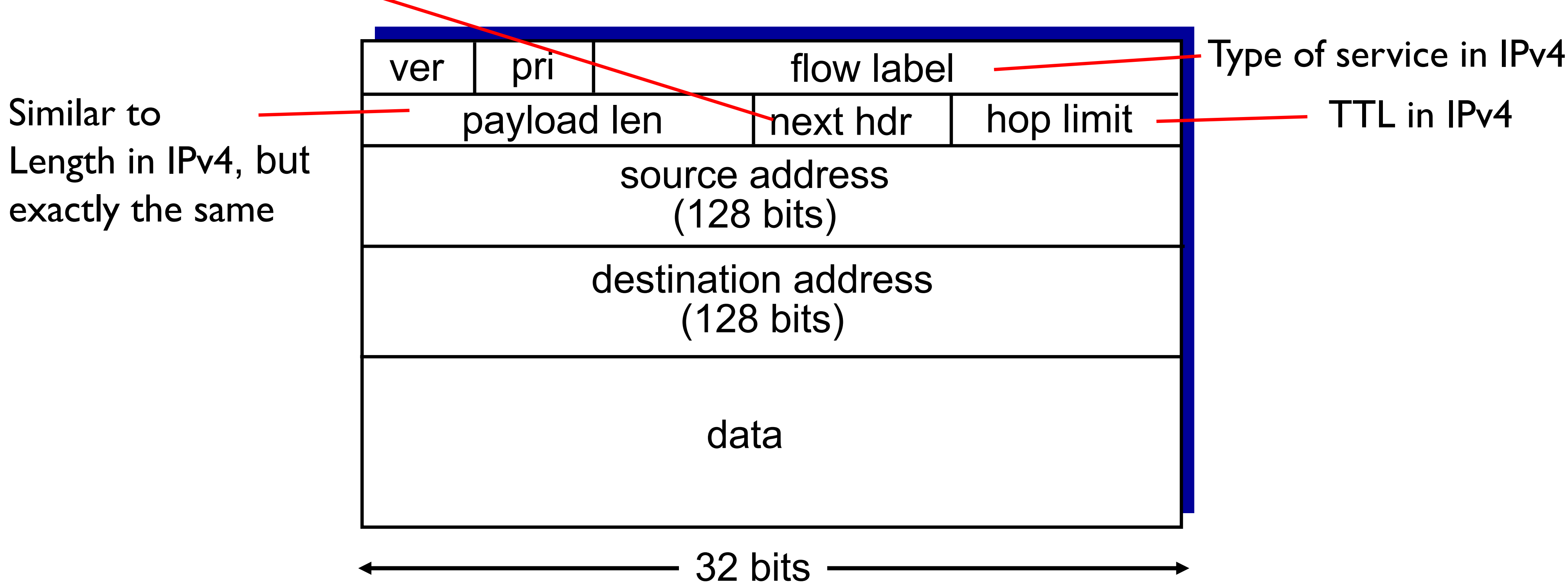
  - fixed-length 40 byte header

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.

Questions?