Internet Protocols EBU5403
Live Lecture C1/C2/C3

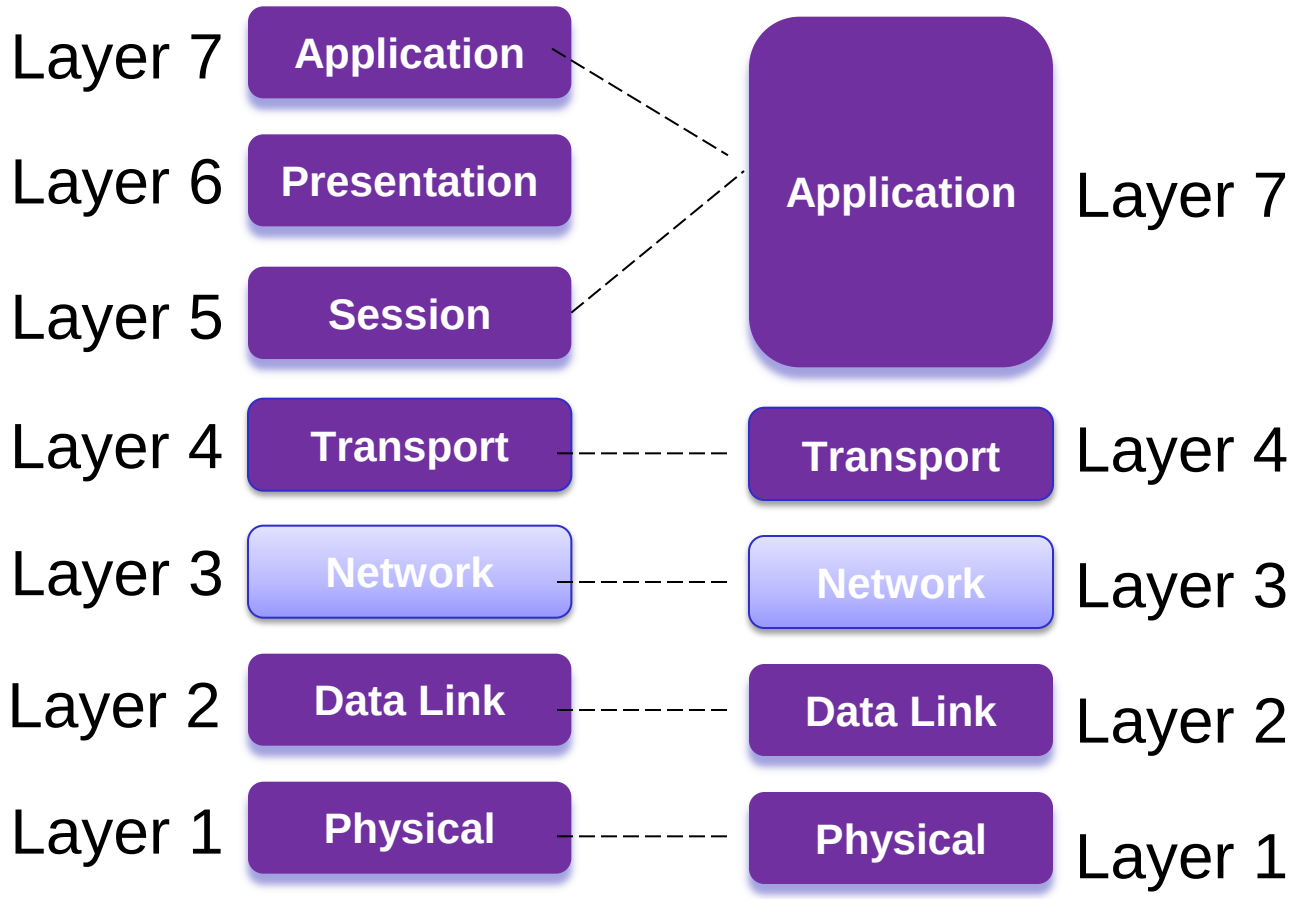
Module organiser: Richard Clegg
(r.clegg@qmul.ac.uk)
Michael Chai (michael.chai@qmul.ac.uk)
Cunhua Pan(c.pan@qmul.ac.uk)

	Part 1	Part 2	Part 3	Part 4
Ecommerce + Telecoms 1	Richard Clegg		Cunhua Pan	
Telecoms 2	Michael Chai			

Structure of course

- Part A
  - Introduction to IP Networks
  - The Transport layer (part I)
- Part B
  - The Transport layer (part II)
  - The Network layer (part I)
  - Class test
- Part C
  - The Network layer (part II)
  - The Data link layer (part I)
  - Router lab tutorial (assessed lab work after this week)
- Part D
  - The Data link layer (part II)
  - Network management and security
  - Class test

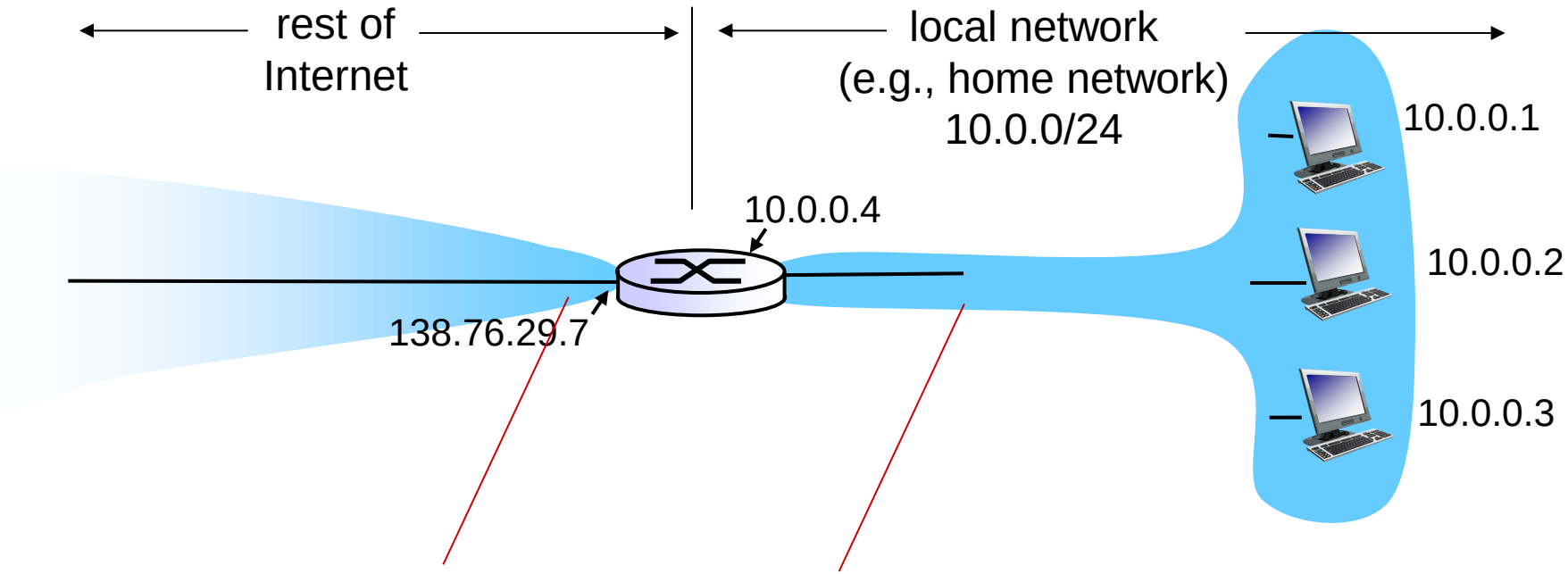
Network Layer



Reminder of lecture contents

- Lecture C1
  - Network address translation
  - IPv 6
  - SDN (e.g. OpenFlow)
- Lecture C2
  - Link State Routing
  - Distance Vector Routing
  - Routing in Autonomous systems
  - OSPF (Open Shortest Path First)
- Lecture C3
  - Routing between/within Autonomous Systems
  - Software defined networking control plane

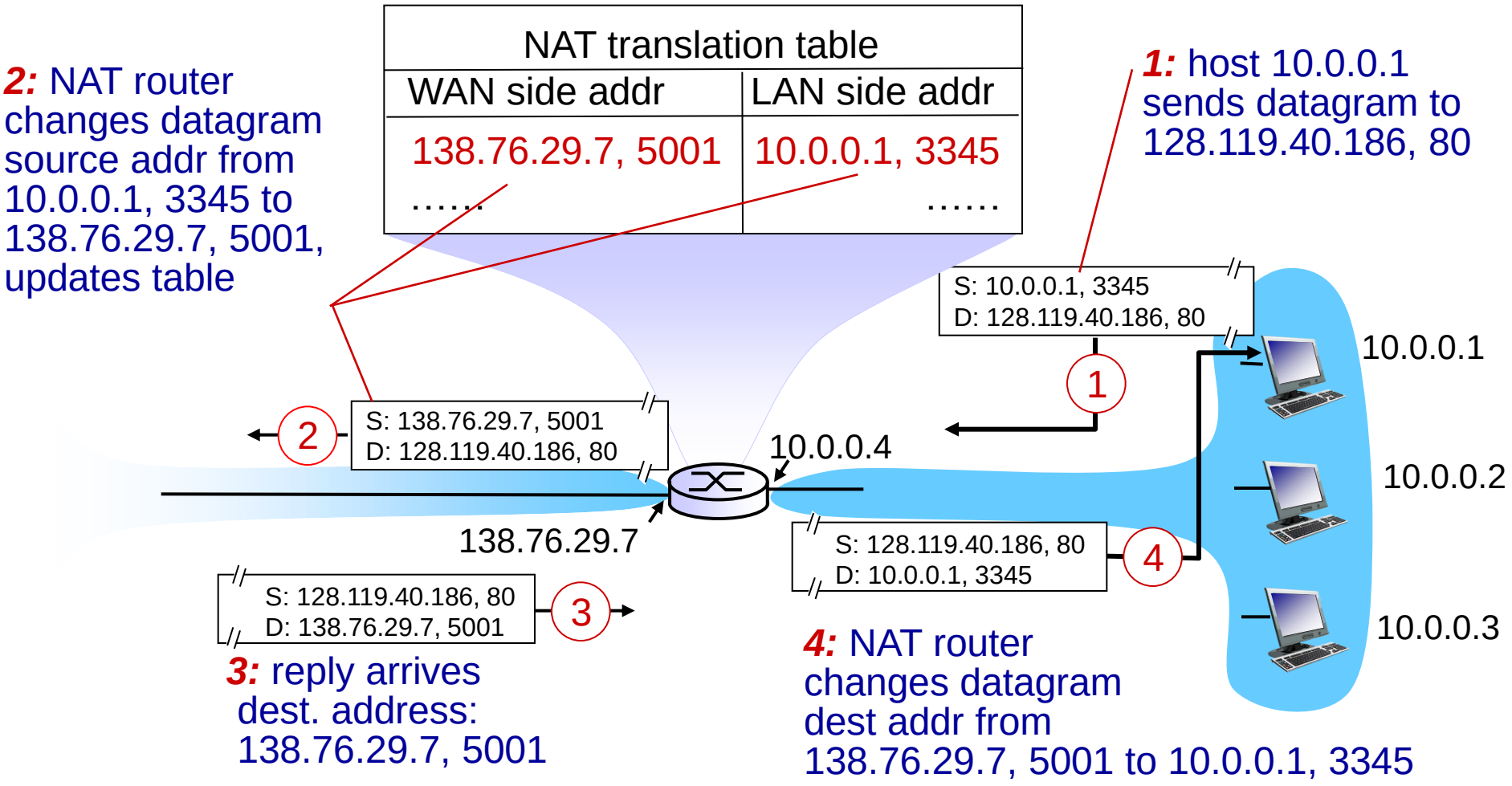
NAT: network address translation



all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7,different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: network address translation



* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

NAT: network address translation

Question: Assume that the local IP address and port number of a host in a local network are 10.0.0.4 and 99, respectively. The NAT IP address is 192.138.45.2, and the new port is 202. Pls provide the NAT translation table.

NAT translation table	
WAN side addr	LAN side addr

NAT: network address translation

Question: Assume that the local IP address and port number of a host in a local network are 10.0.0.4 and 99, respectively. The NAT IP address is 192.138.45.2, and the new port is 202. Pls provide the NAT translation table.

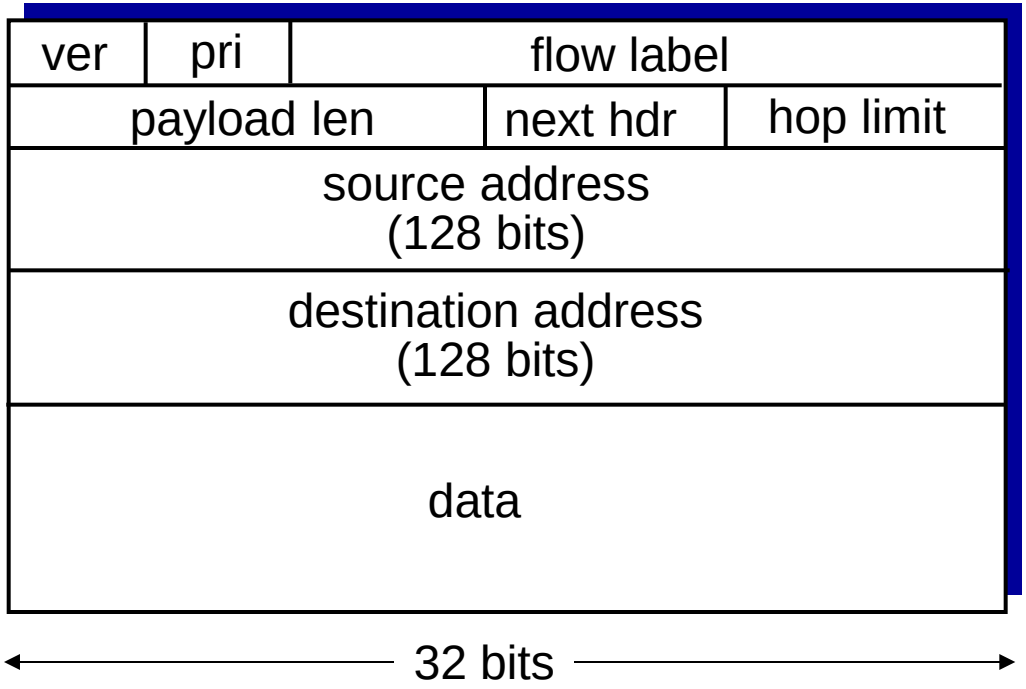
NAT translation table	
WAN side addr	LAN side addr
192.138.45.2,202	10.0.0.4, 99
……	……

IPv6 datagram format

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



IPv 6

Question: List some differences between IPv 4 and IPv 6?

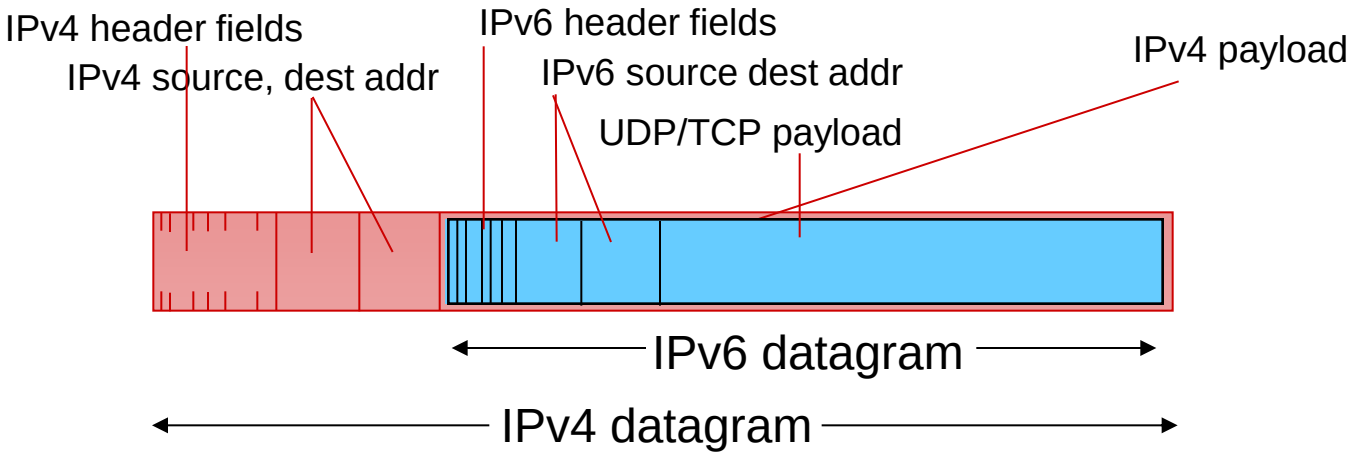
IPv 6

Question: List some differences between IPv 4 and IPv 6?

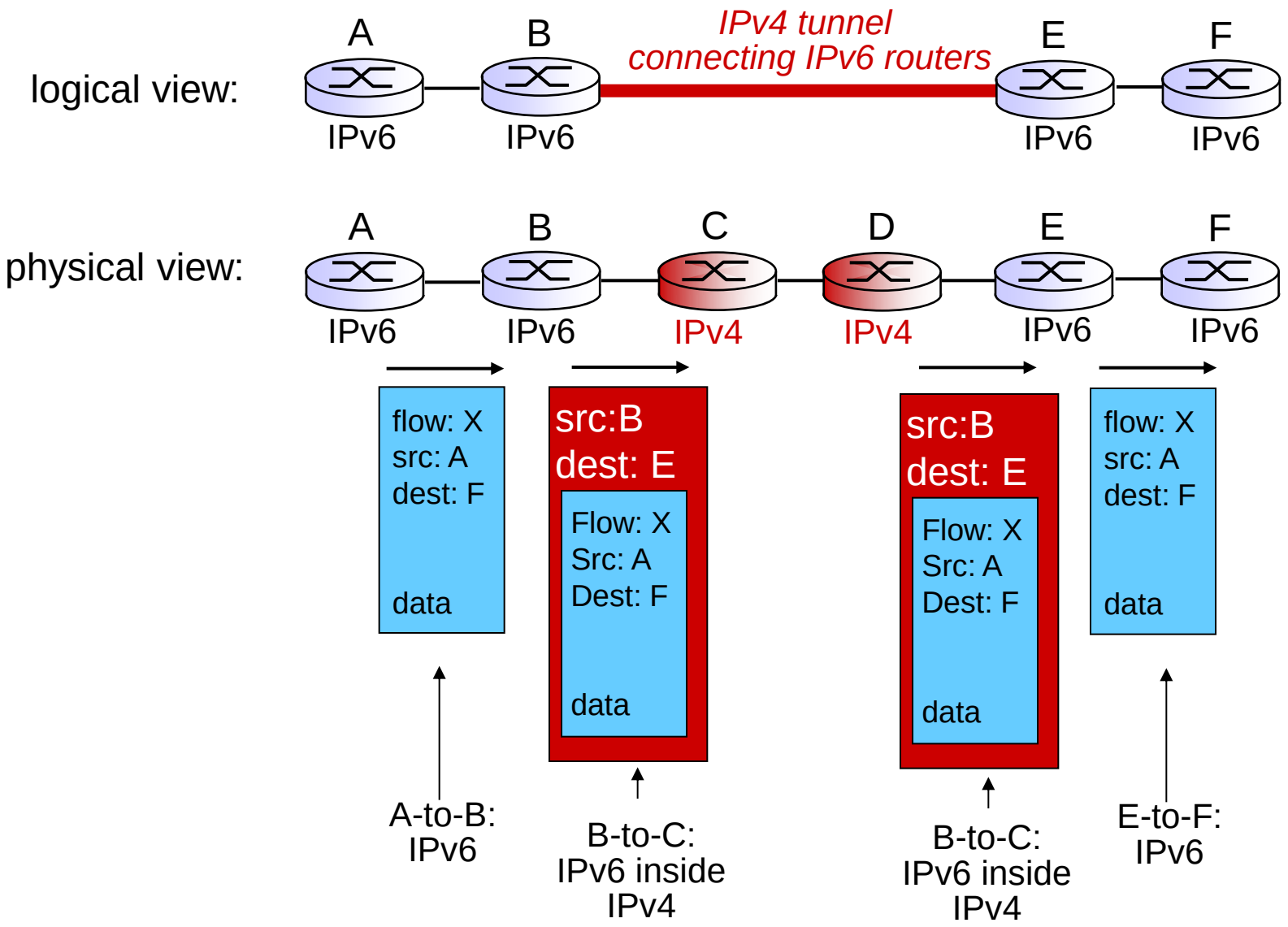
1. The number of address bits of IPv 4 is 32, while that of IPv 6 is 128
2. No fragmentation and checksum in IPv 6, which speeds the signal processing and forwarding

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
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-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

SDN

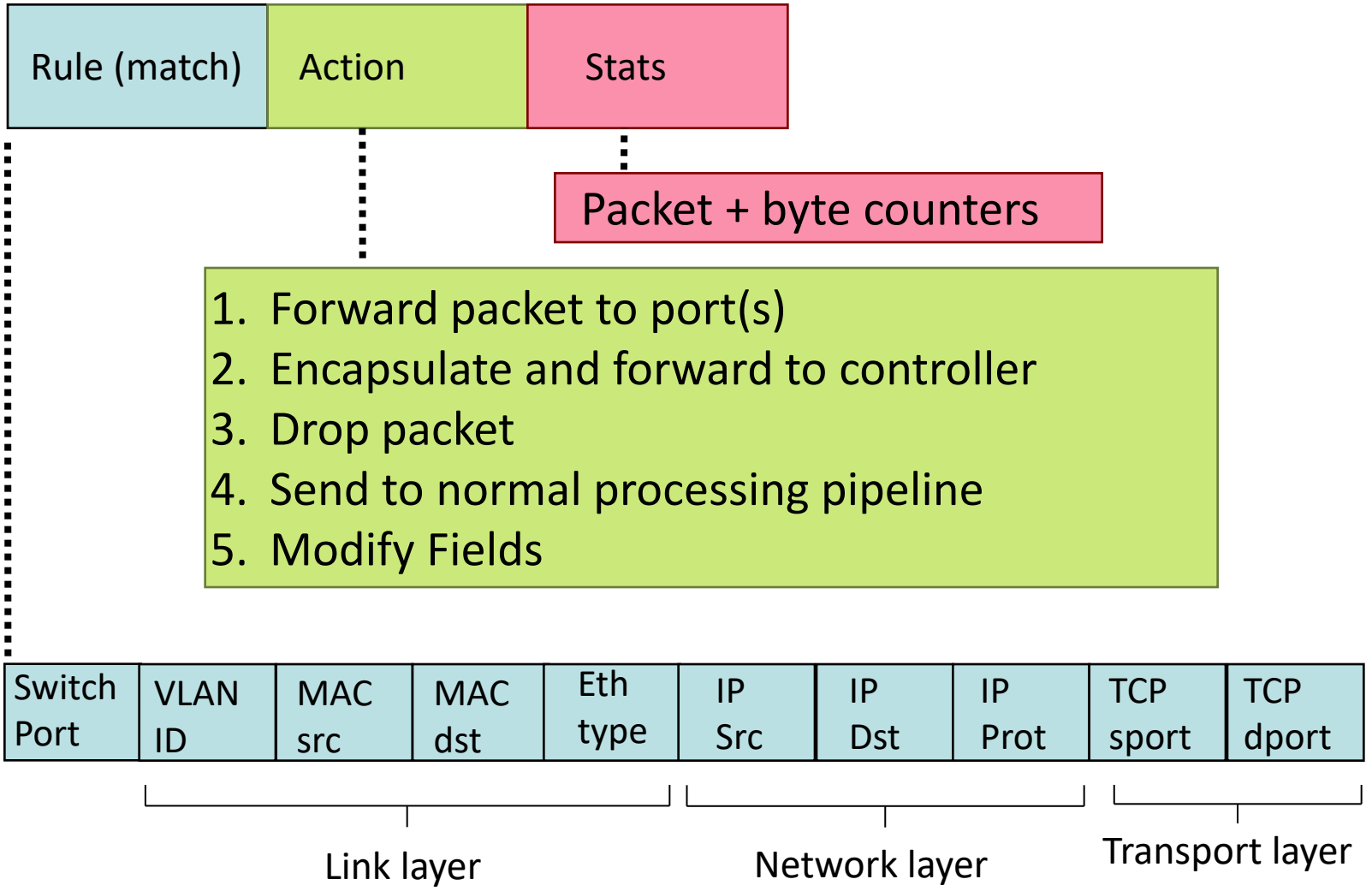
Question: Concerning the routing method, what are the advantages of using SDN compared with traditional routing methods:

SDN

Question: Concerning the routing method, what are the advantages of using SDN compared with traditional routing methods:

- Traditional routing methods:
  - Data Plane is longest-prefix match forwarding using a forwarding table.
  - The control plane calculates the forwarding table for each router.
  - The forwarding table can forward packets by their IP address and nothing else.
- SDN allows more flexibility in the control and data plane.
  - Program your own control algorithms in language you know (Java, python etc).
  - Forwarding can use any part of the packet header.

OpenFlow: Flow Table Entries



Examples

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

do not forward (block) all datagrams destined to TCP port 22

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1

OpenFlow abstraction

- *match+action:* different kinds of devices become one

- Router
  - *match:* longest destination IP prefix
  - *action:* forward out a link
- Switch
  - *match:* destination MAC address
  - *action:* forward or flood
- Firewall
  - *match*: IP addresses and TCP/UDP port numbers
  - *action:* permit or deny
- NAT
  - *match:* IP address and port
  - *action:* rewrite address and port

Routing algorithm classification

Q: global or decentralized information?

global:

- all routers have complete topology, link cost info
- "link state" algorithms such as *Dijkstra's algorithm*

decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" algorithms

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 dest.'s

notation:

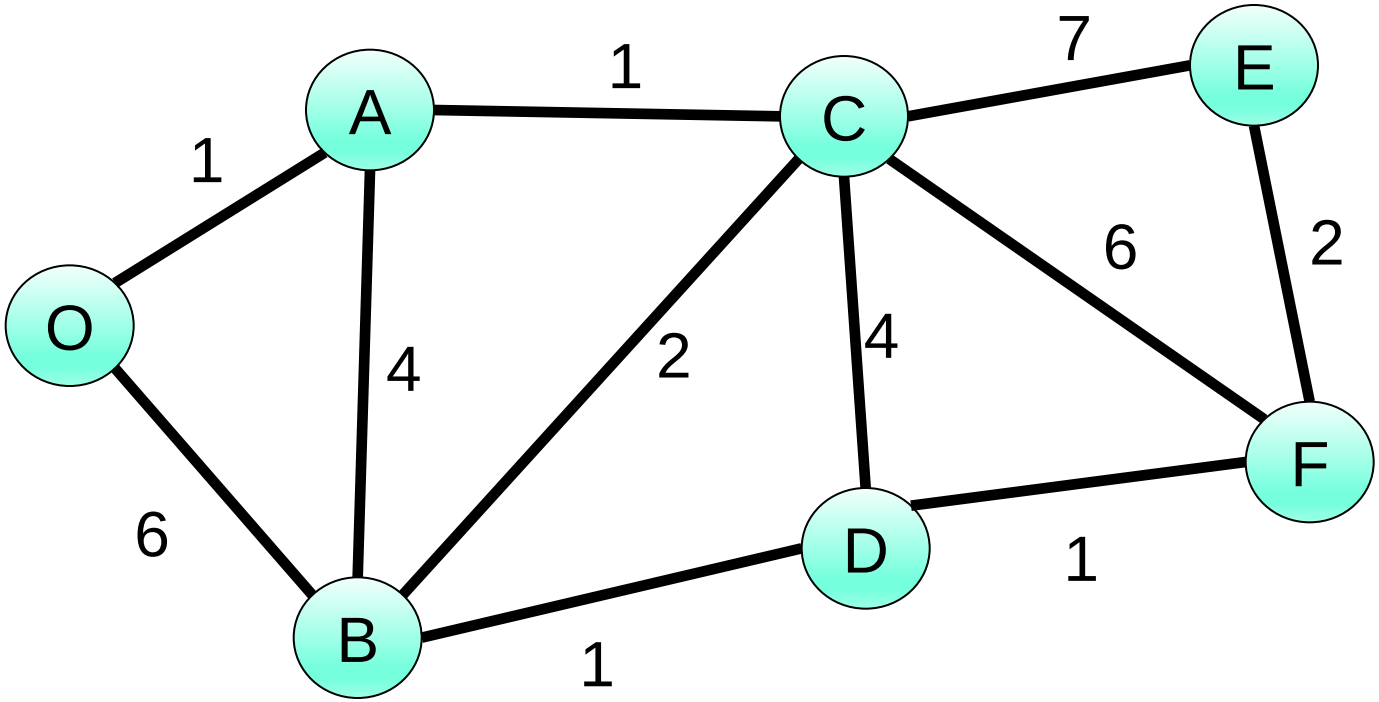
- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current value of cost of path from source to dest. v
- p(v): predecessor node along path from source to v
- N': set of nodes whose least cost path definitively known

Student Questions: Dijkstra

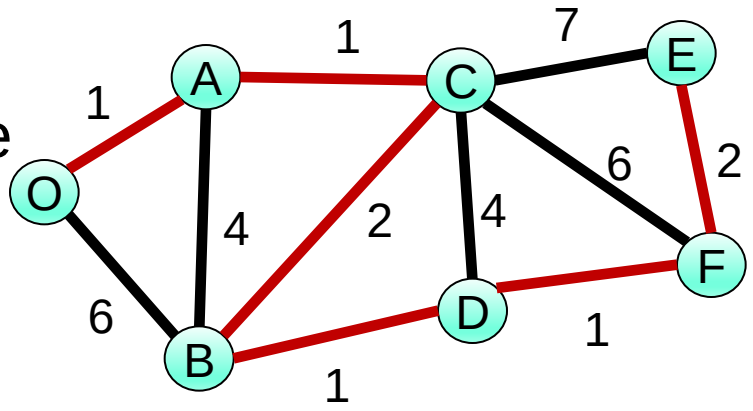
- Key things to remember about Dijkstra
- Need to keep track of:
  - Permanent set of nodes N'= {…}
  - Estimated cost to all nodes D(v)
  - Predecessor node p(v)
- Initialise (imagine we start from node A)
  - N'={A}
  - If node connects to A D(v) = c(A,v)
  - If node connects to A p(v) = A
- At every step – find node not in N' that has smallest D(v) --- call this node B for now:
  - N':= N'UB (add B into N')
  - Update distance to any node that has a connection to B
  - For all v not in N' D(v) = min(D(v), D(B) + c(B,v))

Dijkstra Example network

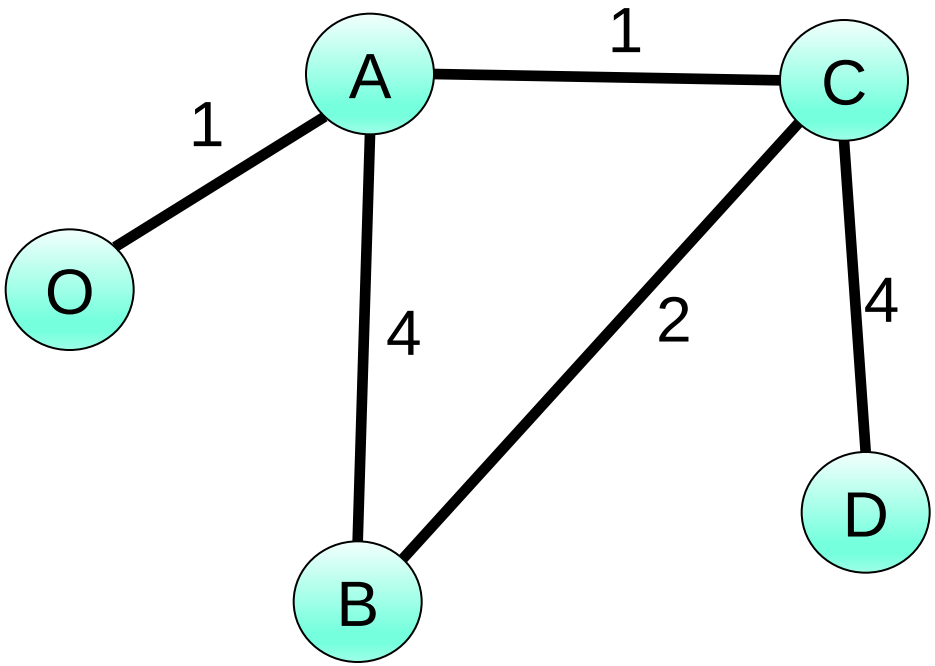
- Find shortest path to all dest. starting from O.



Greedily pick the closest node at every stage



N'	D(A)p(A)	D(B)p(B)	D(C)p(C)	D(D)p(D)	D(E)p(E)	D(F)p(F)
O	1,O	6,O	∞	∞	∞	∞
O,A		5,A	2,A	∞	∞	∞
O,A,C		4,C		6,C	9,C	8,C
OACB				5,B	9,C	8,C
OACBD					9,C	6,D
OACBDF					8,F	
OACBDFE						


A
C
O
B
D
1
1
4
2
4

Distance vector algorithm

Bellman-Ford equation (dynamic programming)

let

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

then

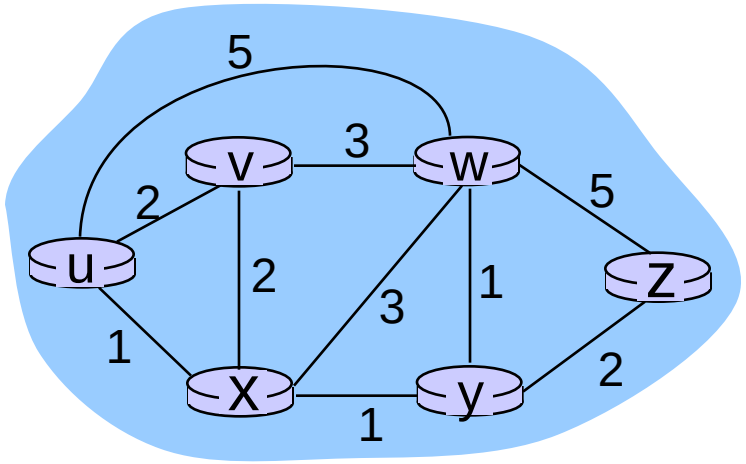
$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

$d_v(y)$: least-cost from neighbor v to destination y

$c(x,v)$: cost to neighbor v

$\min_v$: *min* 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:

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

node achieving minimum is next hop in shortest path, used in forwarding table

Distance vector algorithm

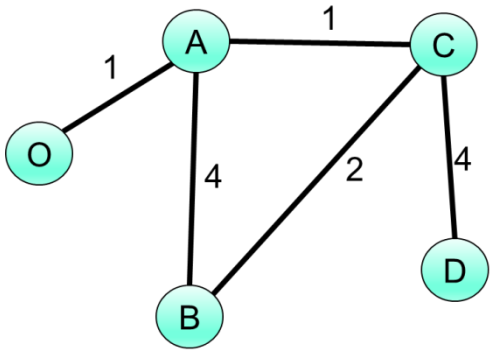
key idea:

- from time-to-time, each node sends its own distance vector estimate to neighbors
- when x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

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

- under minor, natural conditions, the estimate $D_x(y)$ *converge to the actual least cost* $d_x(y)$

Bellman-Ford algorithm



- Consider Bellman-Ford algorithm as executed on Nodes A and C. Each node knows the distance to its neighbours. They will have initial distance vector tables as shown.

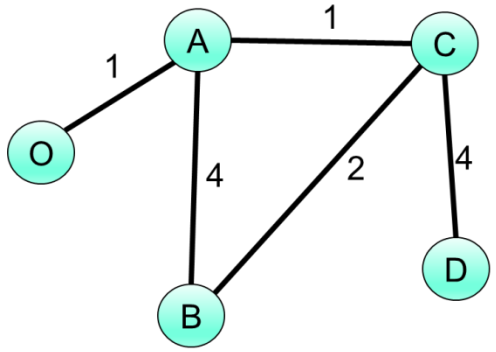
NODE A

Dest	Cost	(route)
O	1	O
C	1	C
B	4	B

NODE C

Dest	Cost	(route)
B	2	B
D	4	D
A	1	A

Bellman-Ford algorithm



- Node A sends its table to node C

Dest	Cost
O	1
C	1
B	4

NODE A table

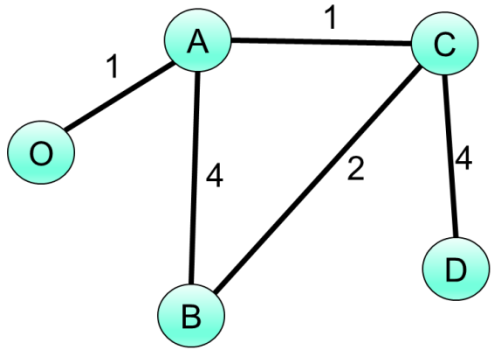
NODE C old table

Dest	Cost	(route)
B	2	B
D	4	D
A	1	A

NODE C new table

Dest	Cost	(route)
B	2	B
D	4	D
A	1	A
O	2	A

Bellman-Ford algorithm



- Node C sends its table to node A (because it made update)

NODE C table

Dest	Cost
B	2
D	4
A	1
O	2

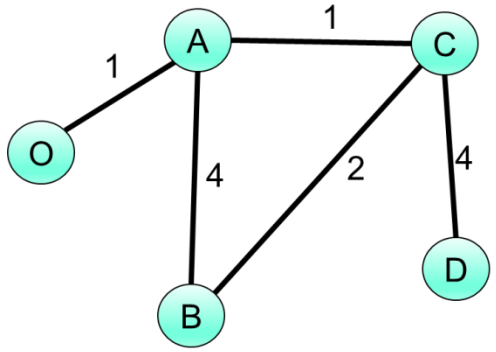
NODE A old table

Dest	Cost	(route)
O	1	O
C	1	C
B	4	B

NODE A new table

Dest	Cost	(route)
O	1	O
C	1	C
B	3	C
D	5	C

Bellman-Ford algorithm



- Node A made an update so sends its table again but node C does not need more updates. All nodes will keep their tables until the network changes.

NODE A final table

Dest	Cost	(route)
O	1	O
C	1	C
B	3	C
D	5	C

NODE C final table

Dest	Cost	(route)
B	2	B
D	4	D
A	1	A
O	2	A

Internet approach to scalable routing

aggregate routers into regions known as "autonomous systems" (AS) (a.k.a. "domains") – big areas of internet

think of a large university or company or ISP.

intra-AS routing

- routing among hosts, routers in same AS ("network")
- all routers in AS must run *same* intra-domain protocol
- routers in *different* AS can run *different* intra-domain routing protocol
- gateway router: at "edge" of its own AS, has link(s) to router(s) in other AS'es

inter-AS routing

- routing among AS'es
- gateways perform inter-domain routing (as well as intra-domain routing)

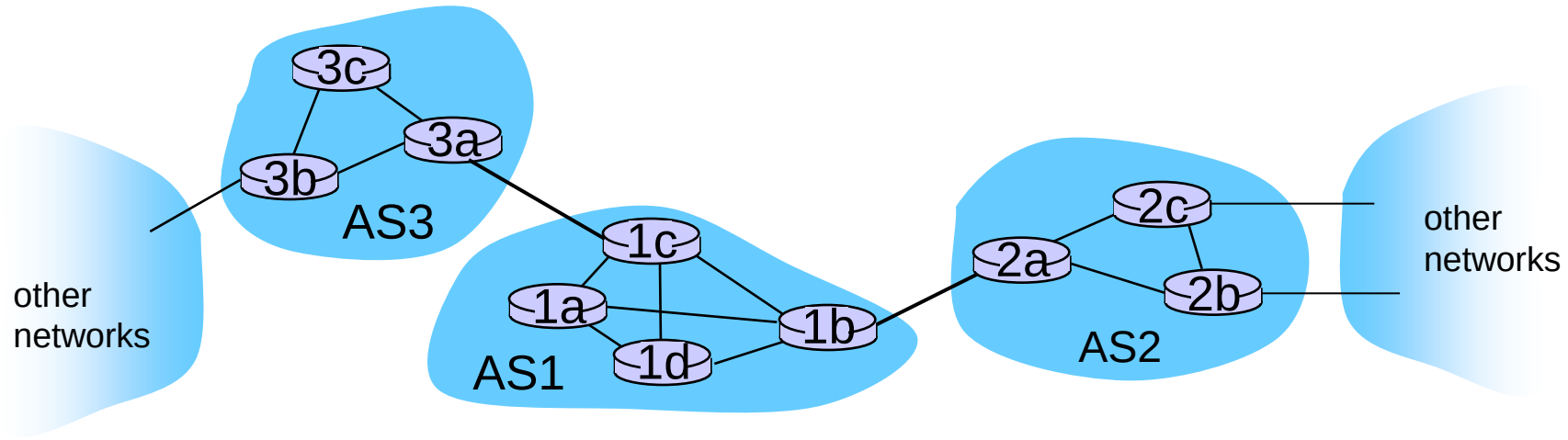
Inter-AS tasks

- 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!



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 (IS-IS protocol essentially same as OSPF)
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary for decades, until 2016)

OSPF (Open Shortest Path First)

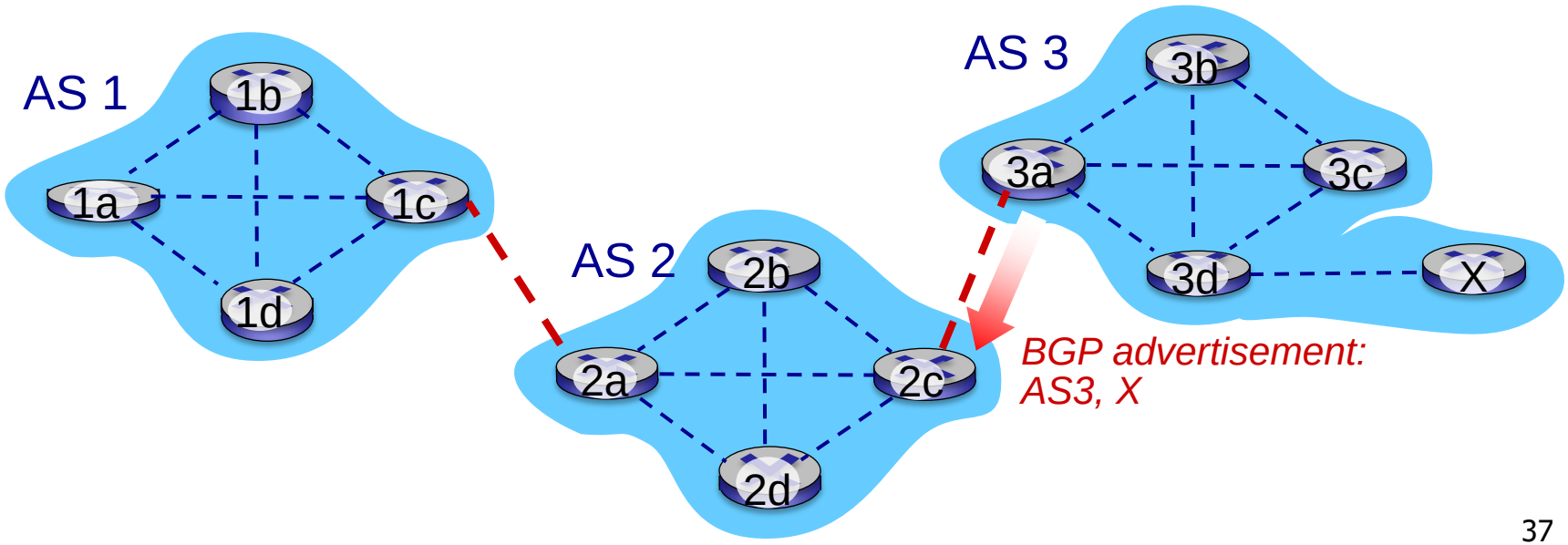
- "open": publicly available
- uses link-state algorithm
  - link state packet dissemination
  - topology map at each node
  - route computation using Dijkstra's algorithm
- router floods OSPF link-state advertisements to all other routers in *entire* AS
  - carried in OSPF messages directly over IP (rather than TCP or UDP)
  - link state: for each attached link
- *IS-IS routing* protocol: nearly identical to OSPF

Internet inter-AS routing: BGP

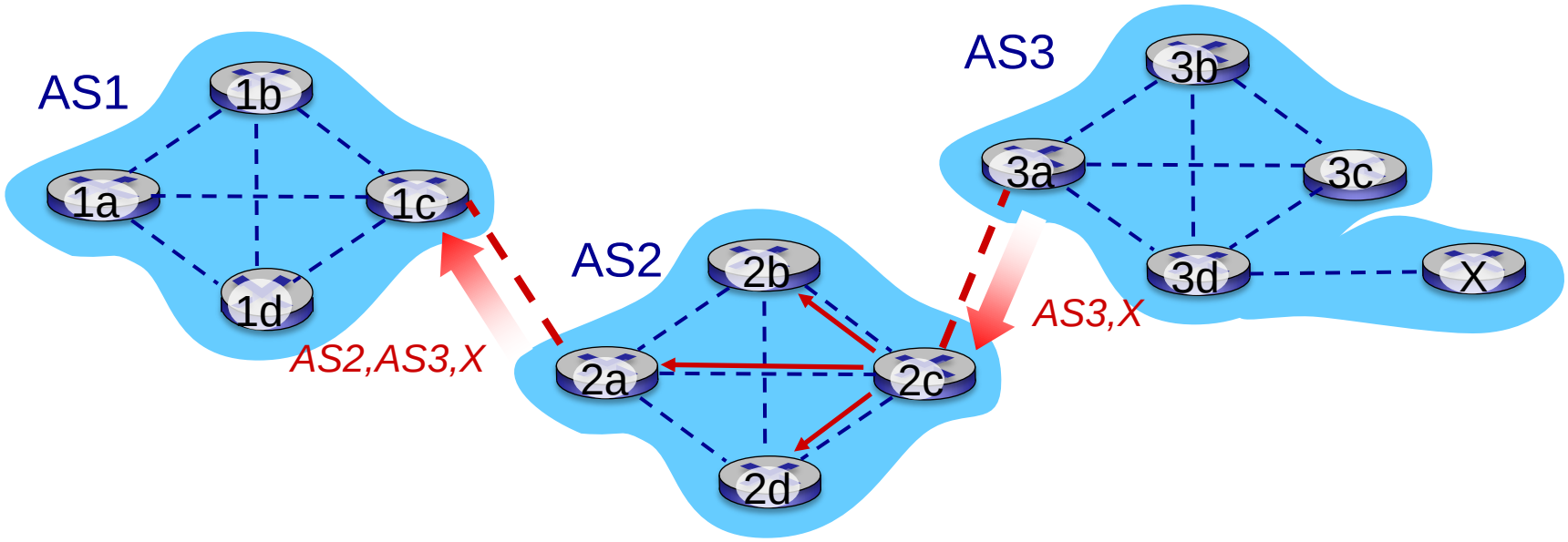
- BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- BGP provides each AS a means to:
  - eBGP: obtain subnet reachability information from neighboring ASes
  - iBGP: propagate reachability information to all AS-internal routers.
  - 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 over semi-permanent TCP connection:
  - advertising *paths* to different destination network prefixes (BGP is a "path vector" protocol)
- when AS3 gateway router 3a advertises path AS3,X to AS2 gateway router 2c:
  - AS3 *promises* to AS2 it will forward datagrams towards X

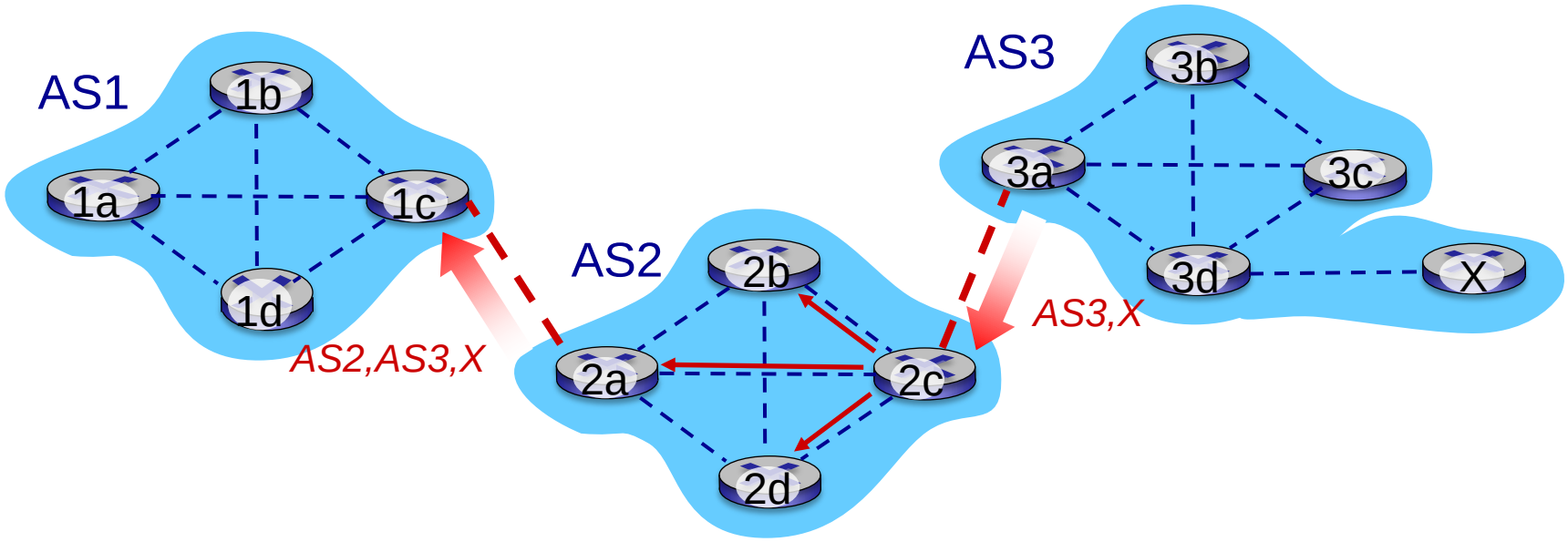


BGP path advertisement



- AS2 router 2c receives path advertisement AS3,X (via what?) from AS3 router 3a
- Based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via what?) to all AS2 routers
- Based on AS2 policy, AS2 router 2a advertises (via what) path AS2, AS3, X to AS1 router 1c

BGP path advertisement



- AS2 router 2c receives path advertisement AS3,X (via eBGP) from AS3 router 3a
- Based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- Based on AS2 policy, AS2 router 2a advertises (via what) path AS2, AS3, X to AS1 router 1c

What have we learned

- Network address translation
- IPv 6
- SDN (e.g. OpenFlow)
- Link State Routing
- Distance Vector Routing
- Routing in Autonomous systems
- OSPF (Open Shortest Path First)
- Routing between/within Autonomous Systems