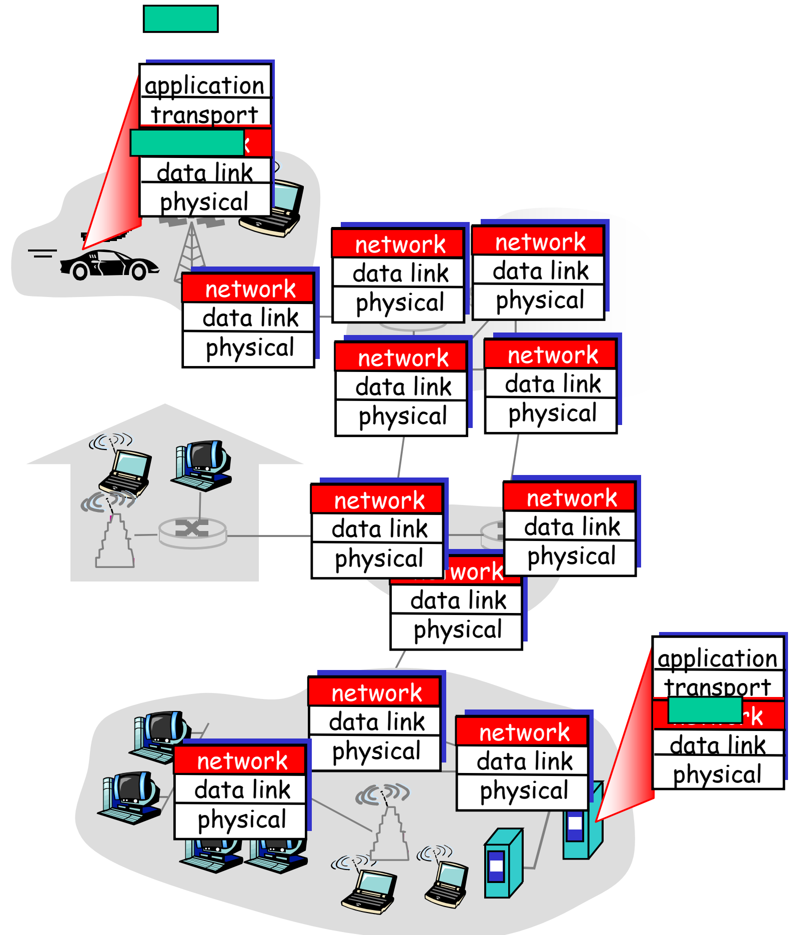
Network Data Plane

Network layer

- delivers segments from sending to receiving host
  - sender encapsulates segments into datagrams
  - Receiver de-encapsulates and delivers segments to transport layer
- network layer in ***every*** host, ***every*** router
- Router examines IP header field in every passing datagram (exception: routers running MPLS)



Key Network-Layer Functions

- *forwarding:* move a packet from router's input interface to an appropriate output interface

- *routing:* determine route taken by packets from source to destination
  - *routing protocols (**intra-AS** and **inter-AS**) where AS is acronym for "Autonomous System"*
  - *every AS runs the same inter-AS protocol*

Virtual-circuit networks need 3rd function

- Before datagrams can flow, end hosts and routers between them establish a *virtual circuit*
  - Routers maintain ***state*** info
  - Earlier networks designed initially to compete with IP:

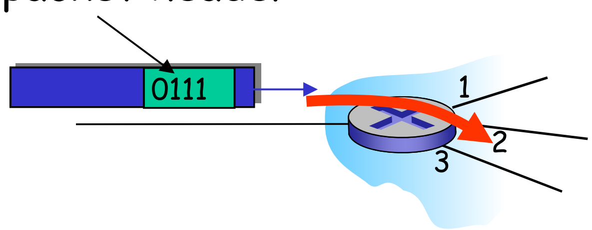
    ATM, frame relay, X.25 *(from old to very old)*
  - *MPLS* protocol designed more recently to provide virtual circuits supported by IP routers *(typically within the same AS/ISP)*
- Today, such virtual circuits serve as virtual links in Internet

Network layer: data plane, control plane

Data plane

- local, per-router function
- determines how datagram arriving on an input port is *forwarded* to an output port

values in arriving packet header

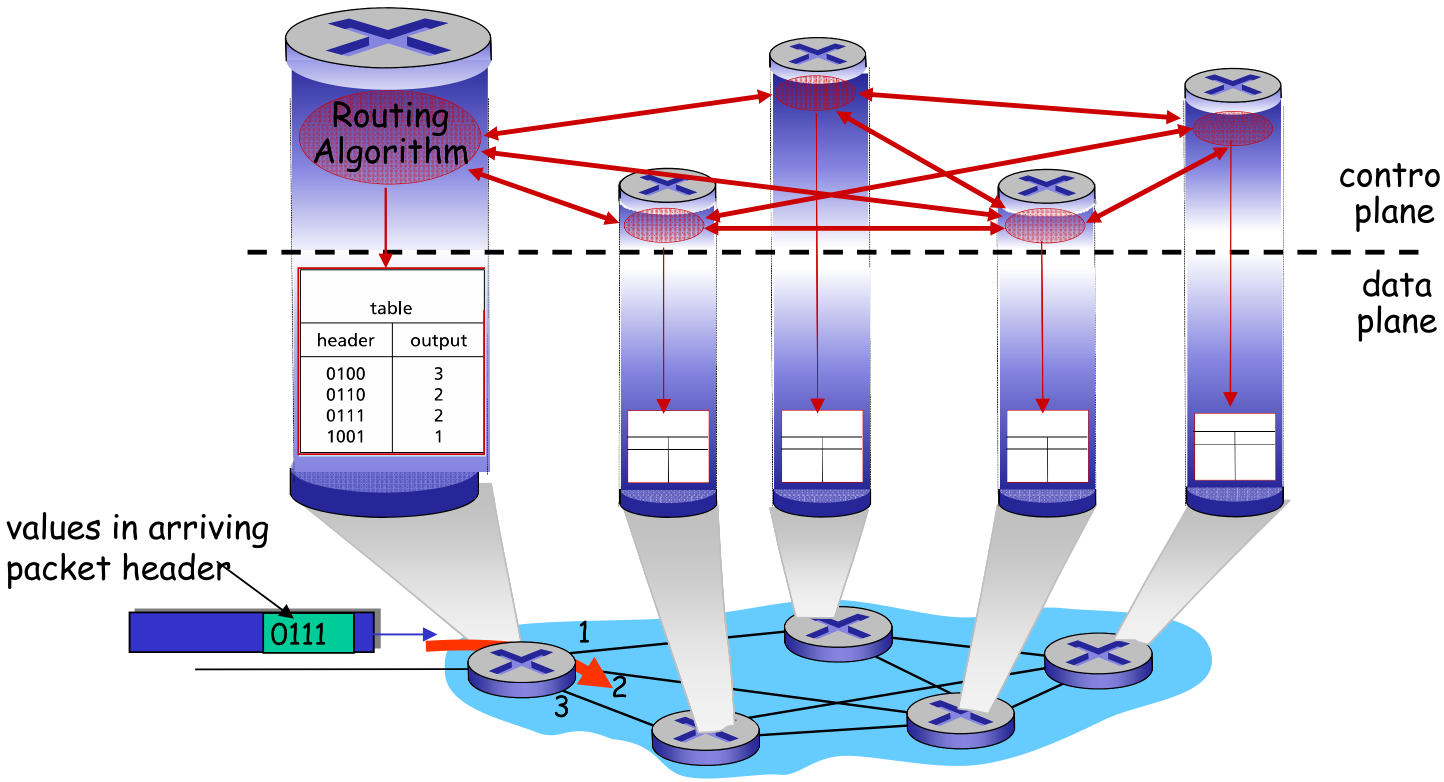


Control plane

- network-wide logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- main approach:
  - *routing protocols* implemented in routers
- new approach
  - *software-defined networking (SDN)*: implemented in logically centralized server(s)

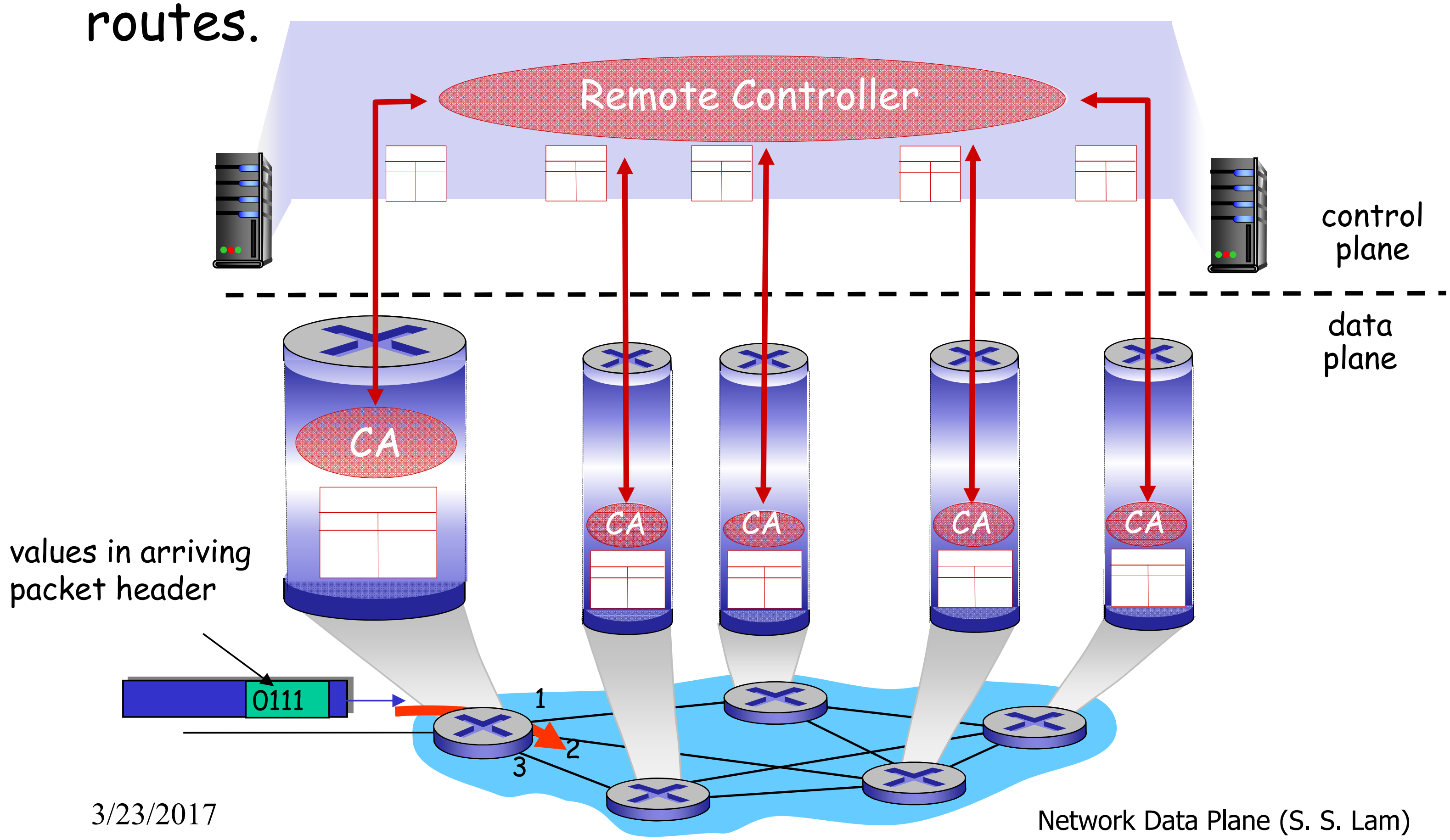
Per-router control plane

Individual routing process *in every router. They* interact by exchanging routing protocol messages



Logically centralized control plane

A distinct (typically remote) controller interacts with local control agents (CAs). The controller computes routes.

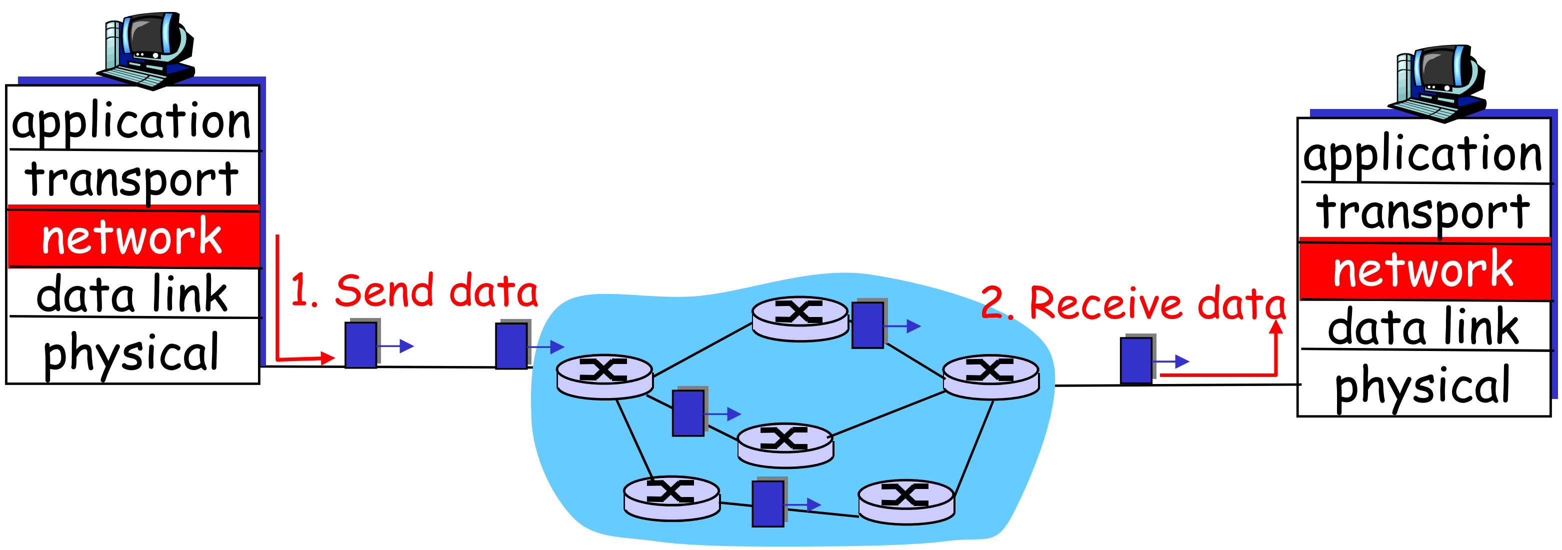


The big picture (preview)

- Data plane
  - Forwarding using network and link headers
    - Datagrams
    - VLANs
    - MPLS virtual circuits and IP tunnels (transformers)
    - NATs (transformers)
  - Filtering (access control lists, firewalls)
    - using transport, network, link headers
  - OpenFlow (SDN)
    - match+action abstraction unifies routers, switches, firewalls, and NATs (but not VCs and tunnels)
- Control plane
  - Routing protocols
    - intra-AS (OSPF, distance vector, Cisco proprietary)
    - inter-AS (eBGP, iBGP)
  - SDN
    - centralized controller

Datagram networks

- IPv4, IPv6
- no network-level concept of "connection" or "flow"
- each packet forwarded *independently* using destination host address
  - packets between same source-dest pair may take different paths



IPv4 addressing: CIDR

Classful addressing (now obsolete): fixed-length subnet portion of 8, 16 or 24 bits

CIDR: Classless InterDomain Routing

- subnet portion of address of **variable length**
- address format: a.b.c.d/x, where x is # bits in subnet portion of address

subnet part | host part

11001000 00010111 00010000 00000000

200.23.16.0/23

IPv4 Forwarding table

4 billion possible entries

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011000 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Longest prefix match

Prefix	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

Examples

DA: 11001000 00010111 00010110 10100001 Which interface?

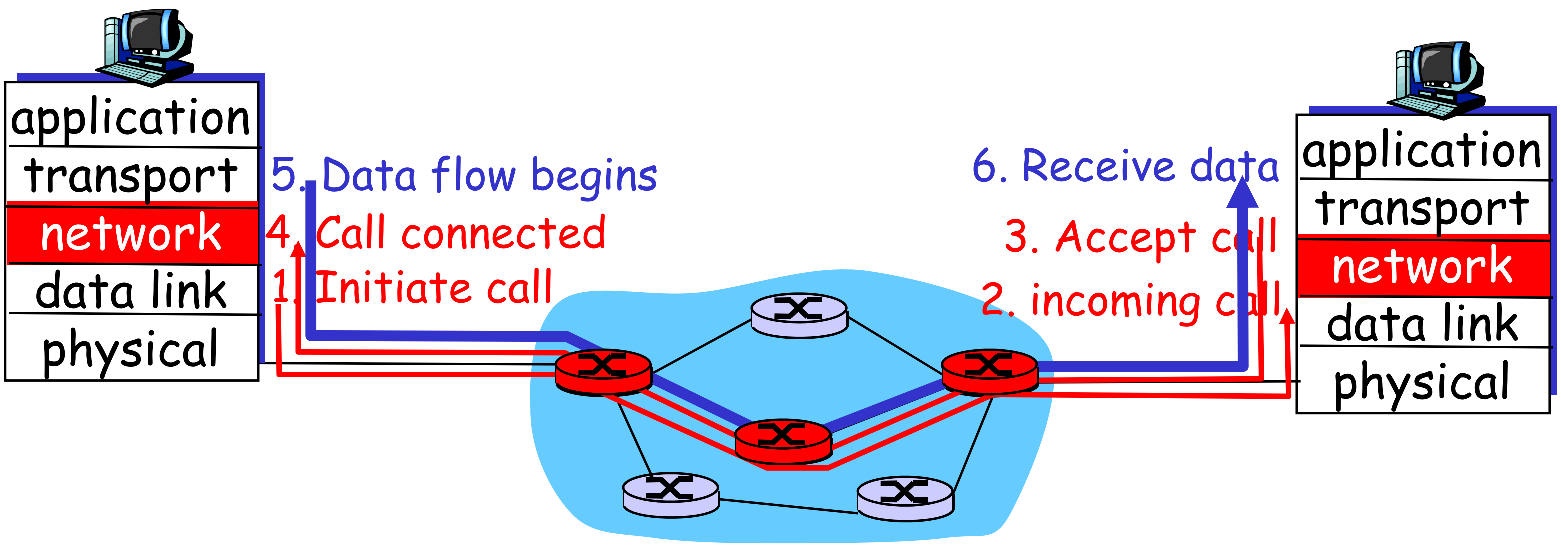
DA: 11001000 00010111 00011000 10101010 Which interface?

A forwarding table in an Internet core router has more than 500,000 IP prefixes

Fast implementation uses Ternary Content Addressable Memory (TCAM), prefixes sorted in decreasing order

Virtual circuits: signaling protocols

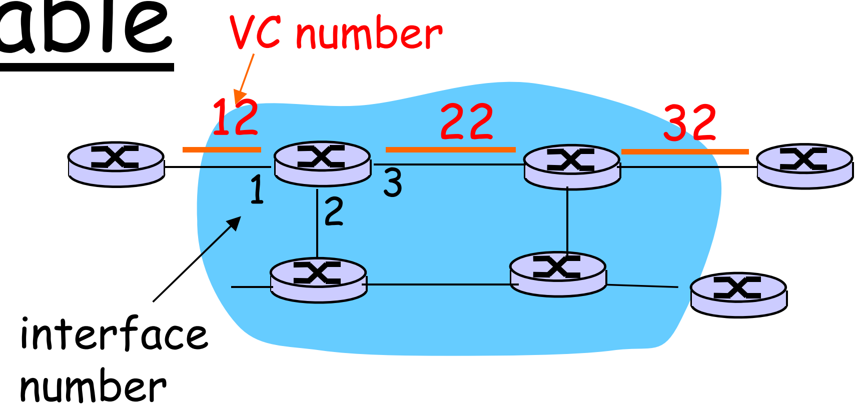
- used to set up, maintain, tear down VC
- not used in Internet's network layer, but may be used underneath the IP layer to provide a virtual link (e.g., MPLS tunnel) in an AS



Virtual circuit (VC)

- call setup, teardown for each call *before* data can flow
- each packet carries a VC identifier which
  - is fixed length and short
  - only needs to be unique for a link
  - is carried in an ***additional header*** inserted between link and network layer headers (called layer 2½)
- *every* router on source-dest path maintains state information for each passing VC
  - incoming and outgoing VC identifiers,
  - resources allocated to VC (bandwidth, buffers)

VC Forwarding table



Forwarding table in northwest router:

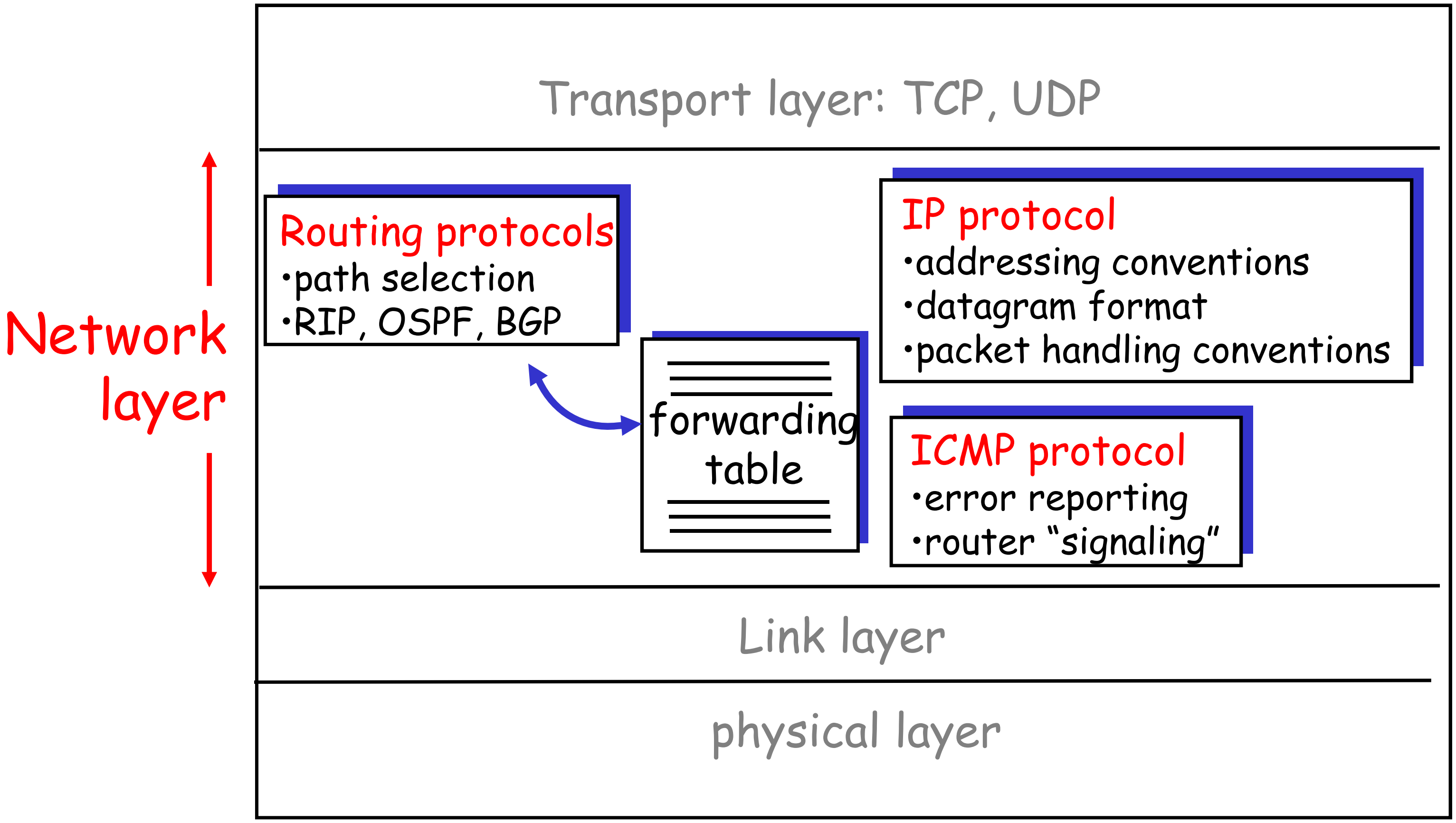
Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

- **Forwarding is fast because short fixed-length VC numbers are used** vs. IP forwarding table with variable-length prefixes. (This is not forwarding in IP layer but it is considered to be in ***data plane.)***

- *May have additional state information about service guarantees*

The Internet Network layer

Host, router network layer functions:



IP datagram format

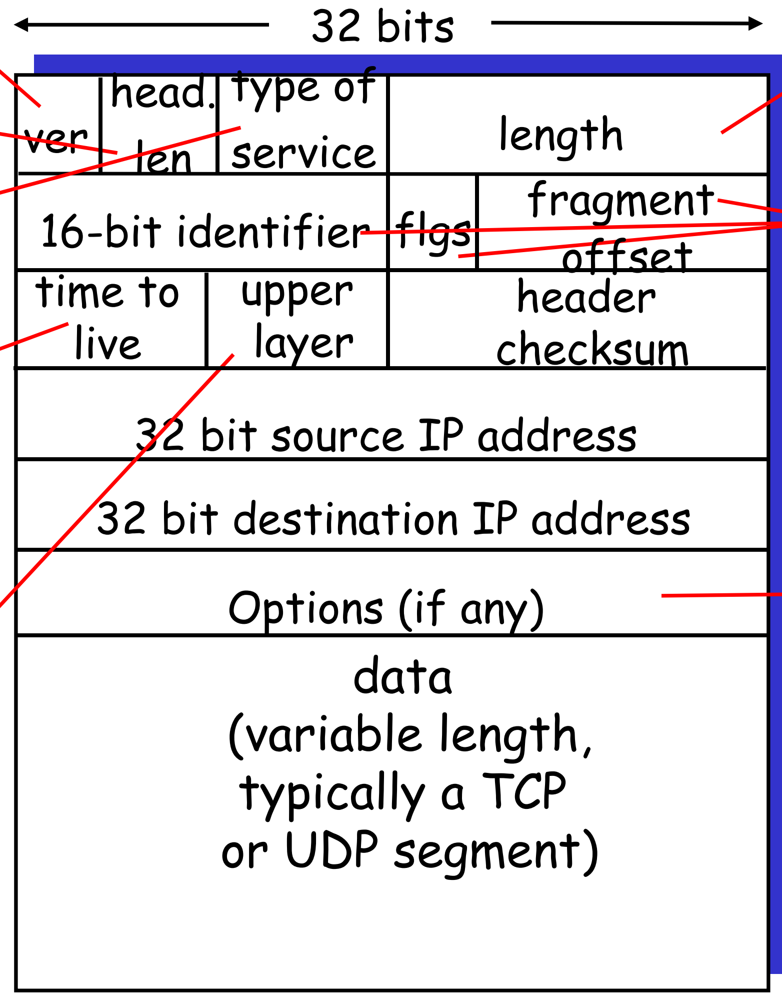
IP protocol version number

header length

"type" of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to



total datagram length (bytes)

for fragmentation/ reassembly

E.g. timestamp, record route taken, specify list of routers to visit.

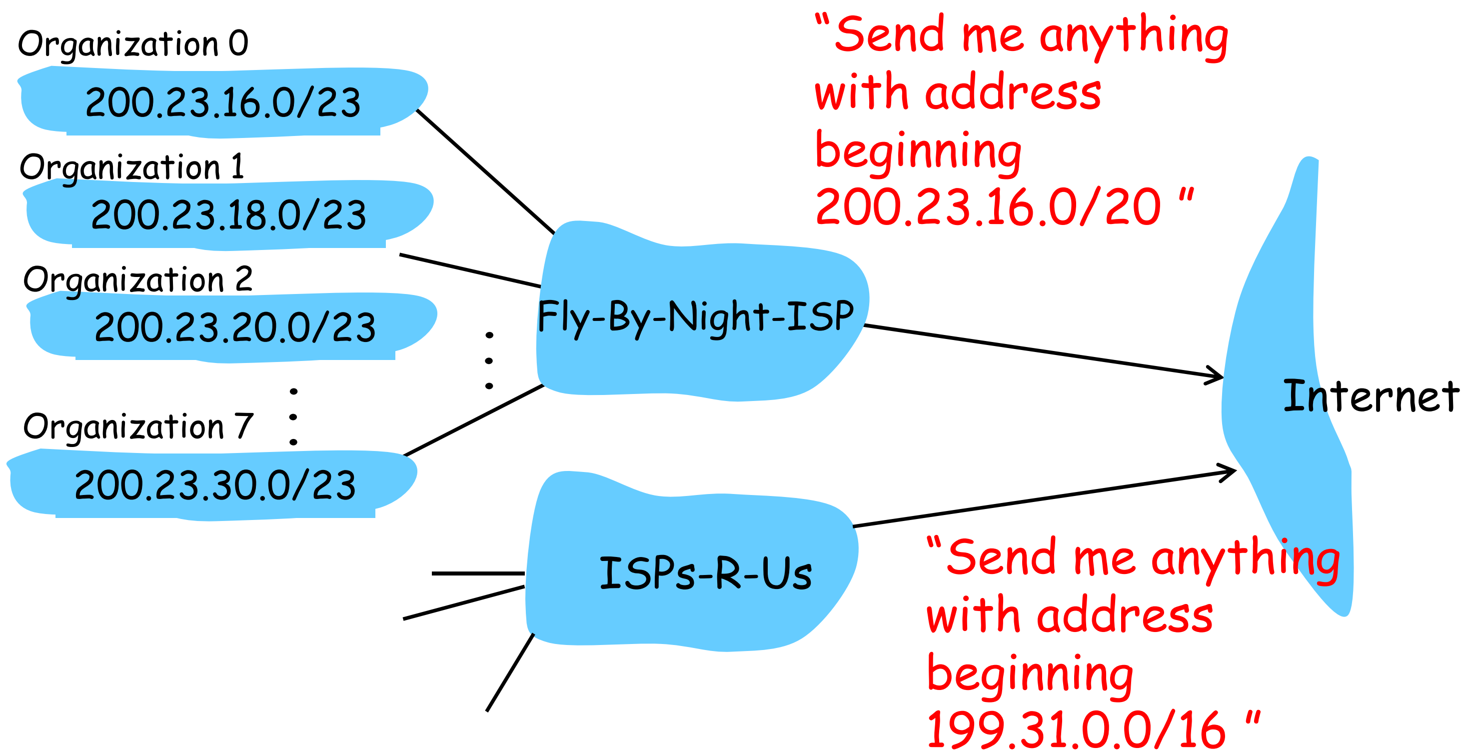
IP address prefix: how to get one?

A: Typically, a customer network gets allocated a portion of its provider ISP's address space

ISP's block	11001000 00010111 00010000 00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000 00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010 00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100 00000000	200.23.20.0/23
...	....	....
Organization 7	11001000 00010111 00011110 00000000	200.23.30.0/23

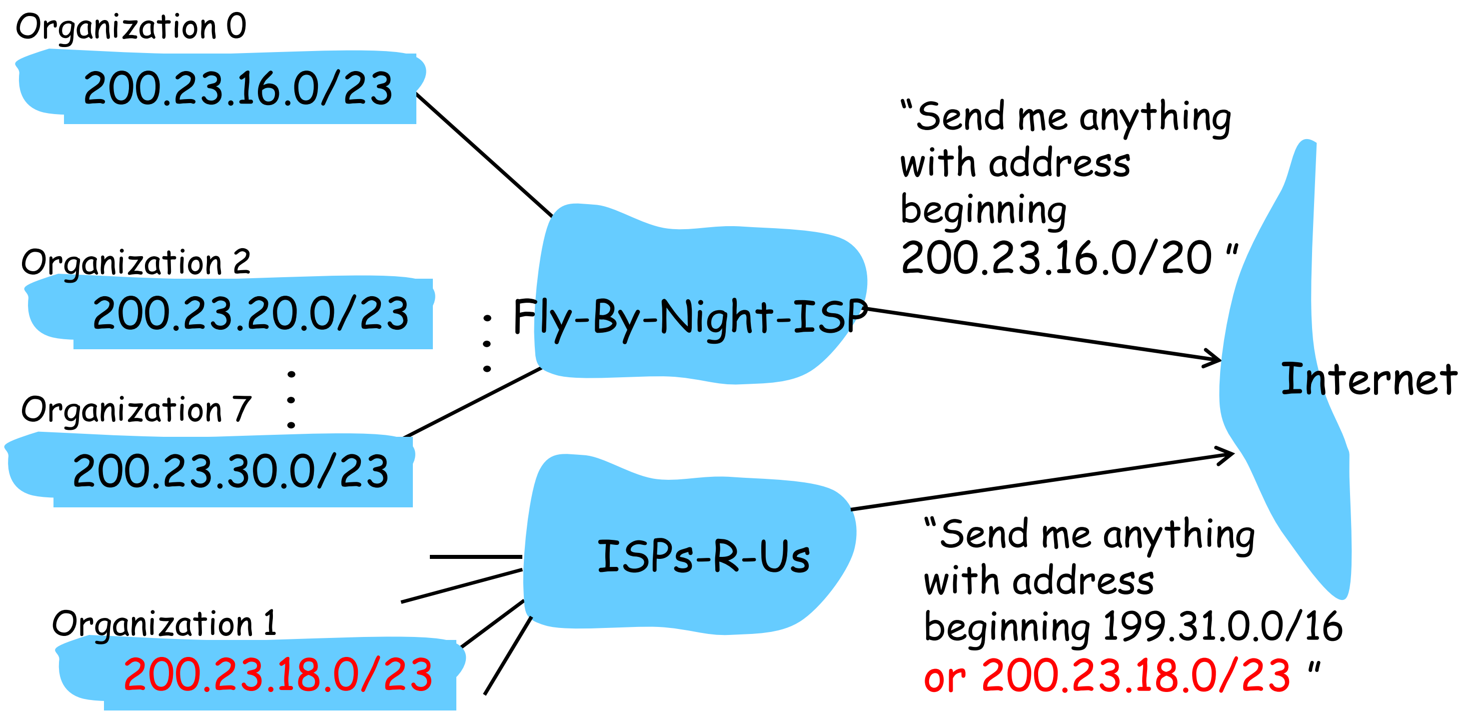
Hierarchical addressing: route aggregation

allows efficient advertisement of routing information



Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1
Hole(s) in a block of addresses <- reason for longest prefix match



Access Control List (ACL)

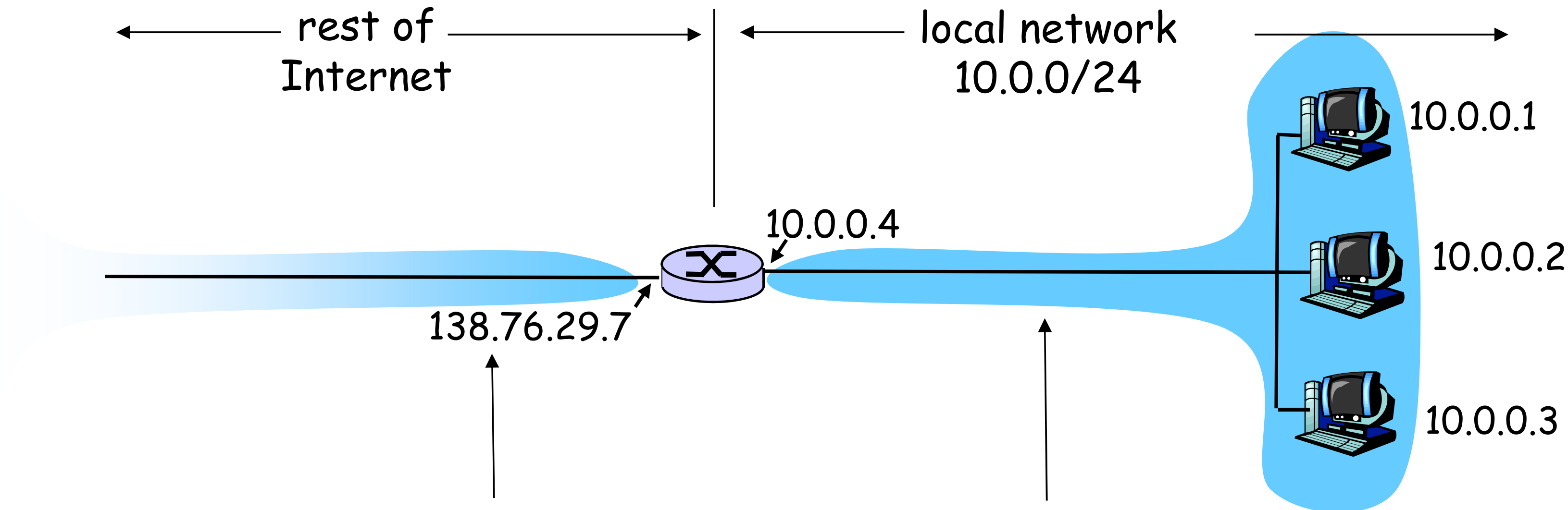
- lists of rules used in firewalls and for guarding input ports and output ports
- *first match* determines action to take on packet

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Packet filters and transformers in the data plane

- Conceptual framework for data plane verification
  - Let the packet universe be the set of all possible bit strings representing all feasible packet headers (or packets), namely: the packet space
- A packet filter allows a subset of packets to pass through, while dropping all other packets.
  - Forwarding tables and ACLs can both be modeled as packet filters
- We next consider network devices that *transform* packet headers.

NAT: Network Address Translation



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

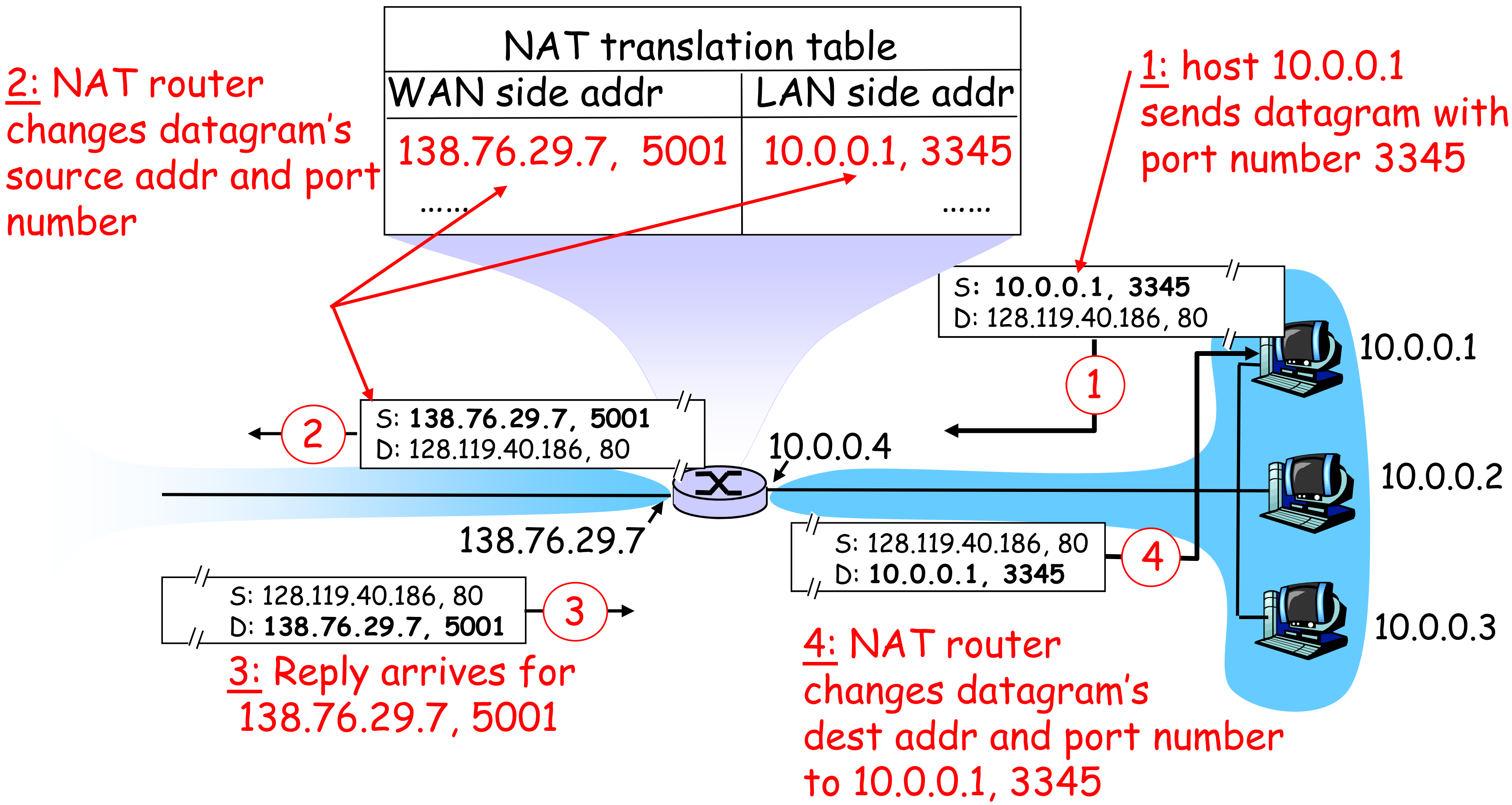
Datagrams with source or destination within network have 10.0.0/24 addresses for source, destination

NAT: Network Address Translation

Motivation: local network uses just one IP address as far as outside world is concerned

- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly addressable/visible by outside world (a security plus).

NAT: Network Address Translation



NAT: Network Address Translation

- 16-bit port-number field:
  - 60,000+ simultaneous connections with a single IP address
- NAT is controversial:
  - routers should only process up to layer 3
    - violates "end-to-end argument"
  - NAT possibility must be taken into account by app designers, e.g., IPsec, P2P applications, etc.
  - address shortage should instead be solved by IPv6

IPv6

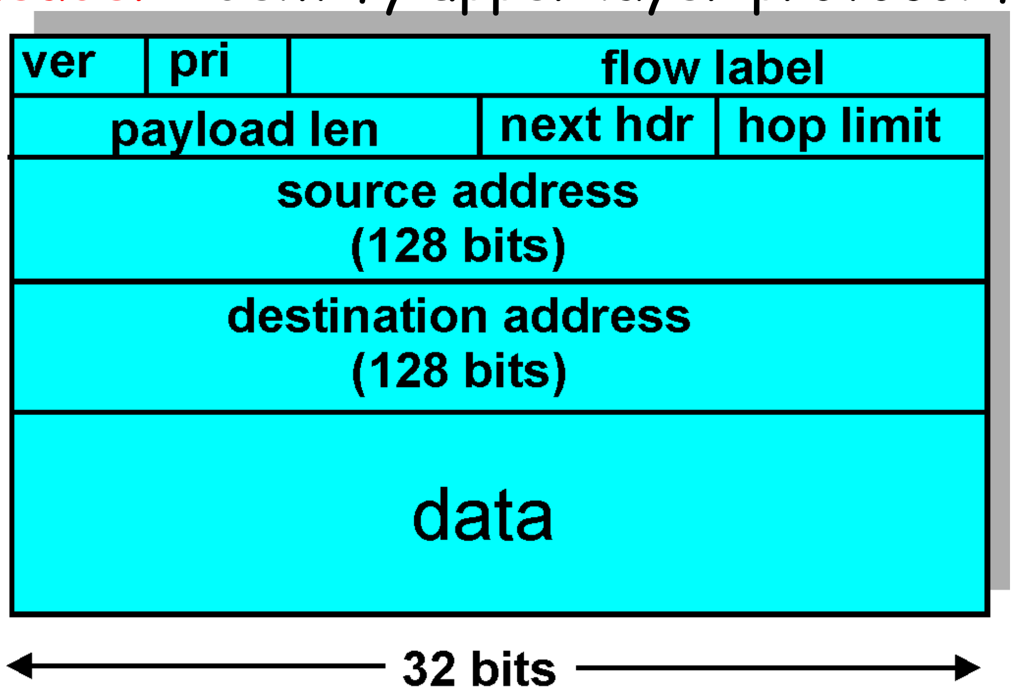
- **Initial motivation:** 32-bit address space soon to be completely allocated (mid 1990s).
- **Additional motivation:**
  - simpler header format to speed up processing/forwarding
  - header change to facilitate QoS
- **IPv6 datagram format:**
  - fixed-length 40 byte header
  - no fragmentation allowed

IPv6 Header (Cont)

Priority: identify priority of datagrams within flow or in different apps

Flow Label: identify datagrams in same "flow." (concept of "flow" not defined).

Next header: identify upper layer protocol for data



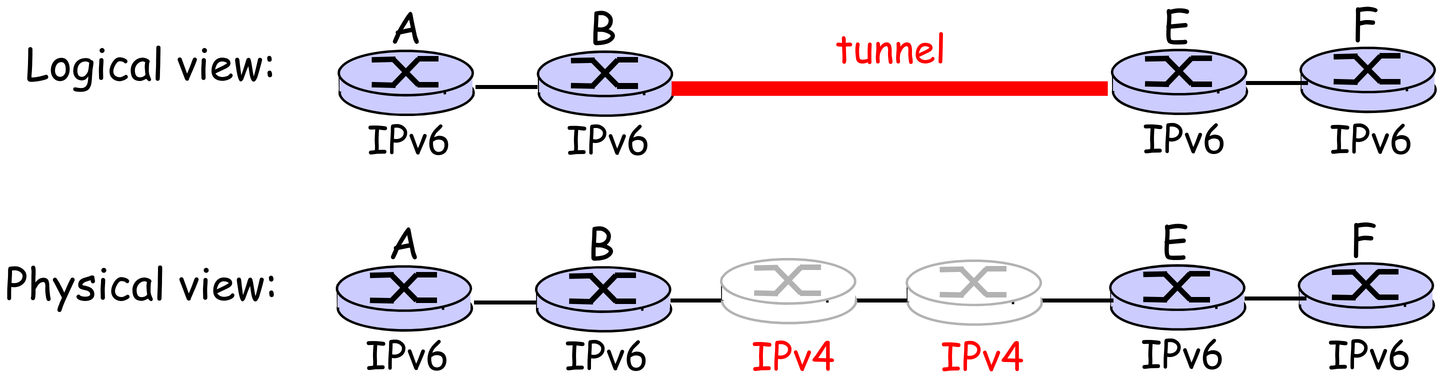
Other Changes from IPv4

- *Checksum*: removed entirely to reduce processing time at each hop
- *Options:* allowed, but outside of header, indicated by "Next Header" field
- *ICMPv6:* new version of ICMP
  - *additional message types, e.g. "Packet Too Big"*
  - *including multicast group management functions*

Transition From IPv4 To IPv6

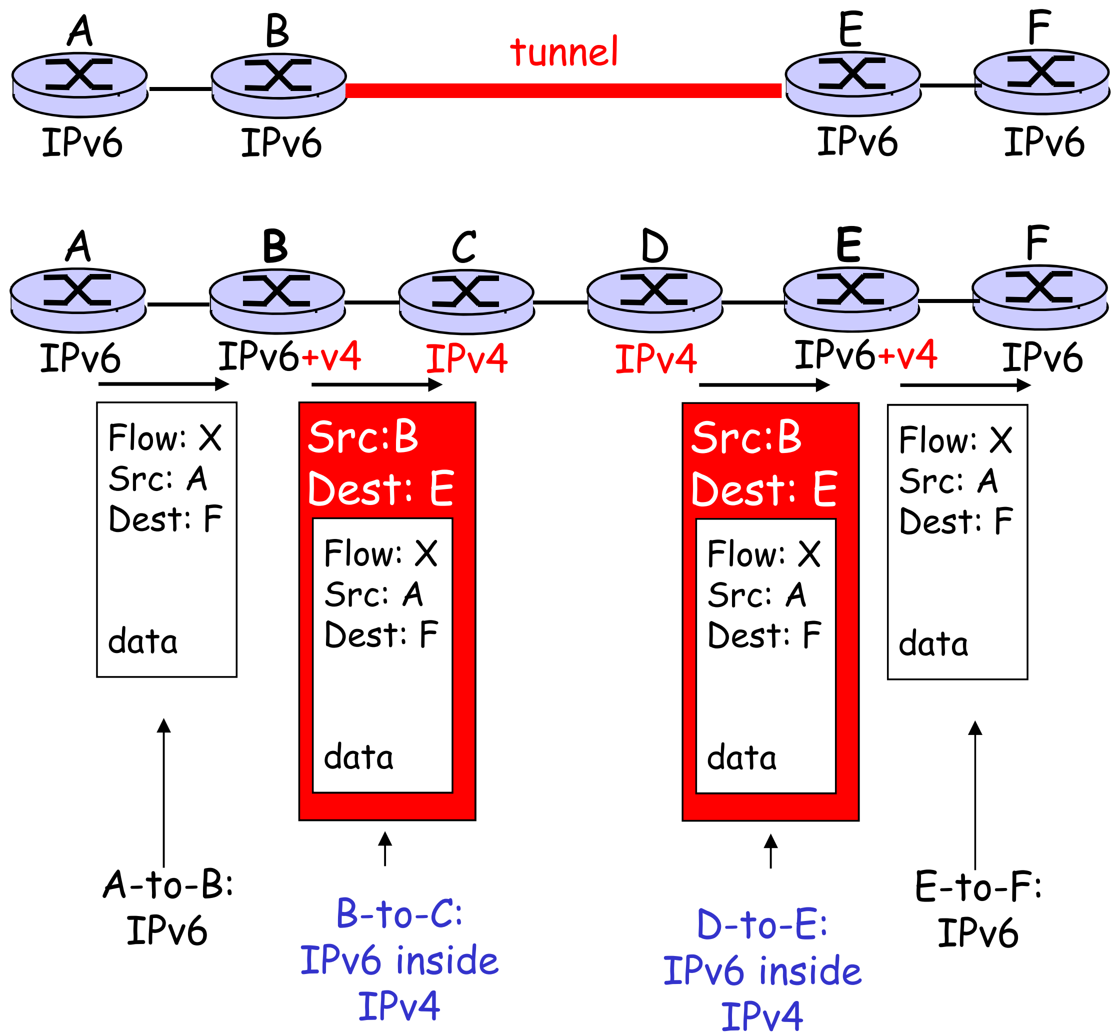
- Not all routers can be upgraded simultaneously
  - no "flag day"
  - How will the network operate with mixed IPv4 and IPv6 routers?
- *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers (also vice versa)

Tunneling



Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

Tunneling



Routers **B** and **E** have **dual stacks**.

In this example, **B** encapsulates v6 packet in v4 packet.

***E** extracts v6 packet from v4 packet.*

Concept - Tunnel as a virtual link

Many possibilities:

- IPv6 in IPv4 tunnel (previous example)
- IPv4 in IPv6 tunnel
- IPv4 in IPv4 tunnel
  - new routing path
- IPv4 in MPLS tunnel
  - virtual circuit
- ...

Link Virtualization: A Network as a Link

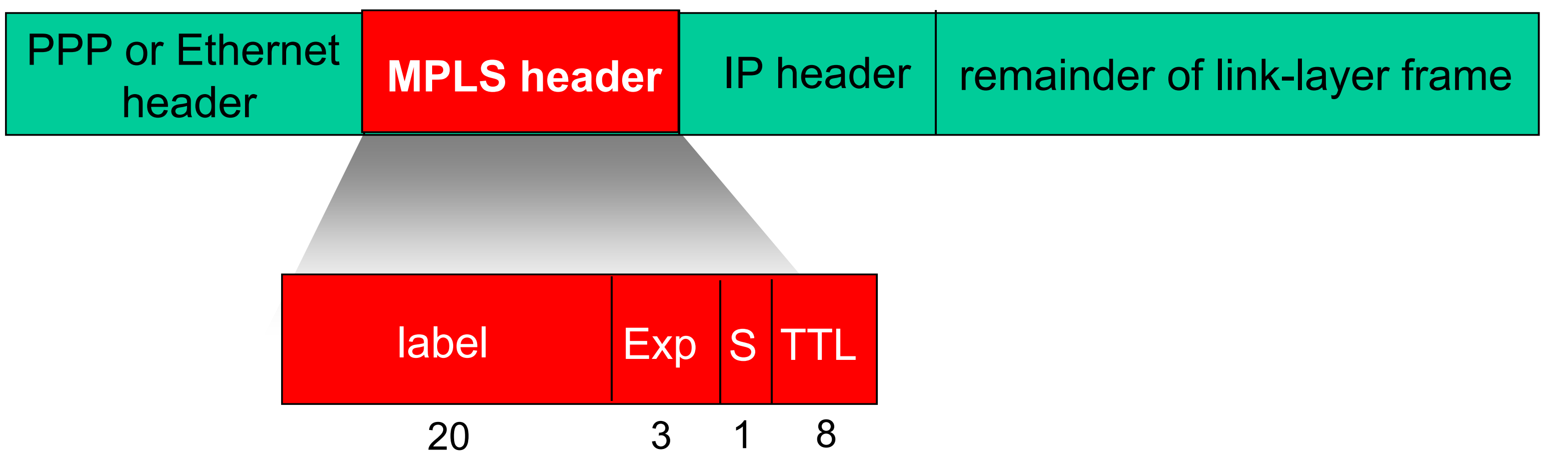
Virtual circuits provided by

- ATM, frame relay, which are packet-switching networks in their own right *(obsolete)*
  - with service models, addressing, routing different from Internet
- A subnet of MPLS capable routers

Each is viewed as a link connecting two IP nodes

Multiprotocol label switching (MPLS)

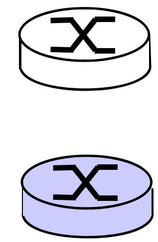
- *initial goal*: speed up IP forwarding by using fixed-length label (instead of variable-length IP prefix) to do forwarding
  - borrowing ideas from Virtual Circuit (VC) approach
  - MPLS routers insert and remove MPLS header but IP datagram still keeps IP address



MPLS capable routers

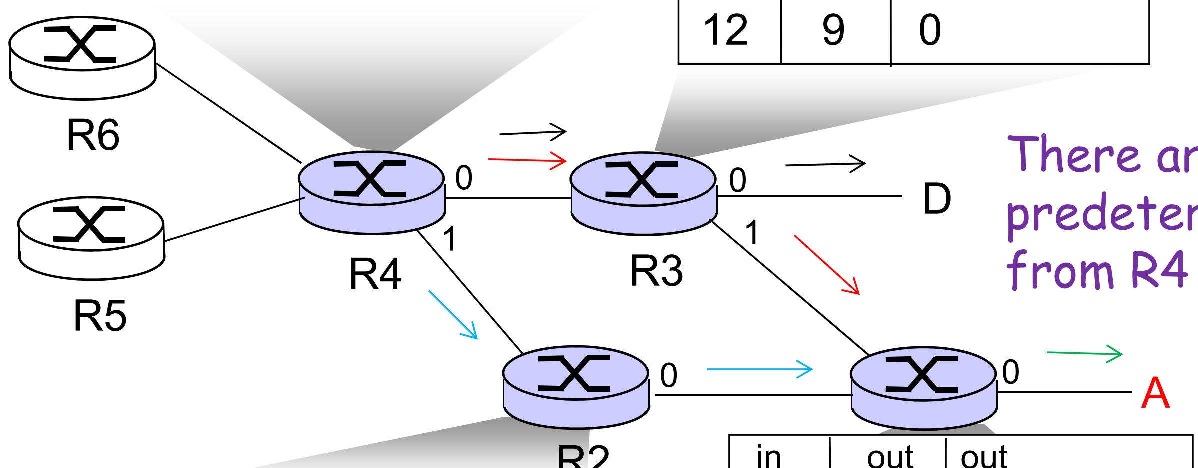
- a.k.a. label-switched router
- forward packets in a "forward equivalence class" to outgoing interface based only on label value (*does not inspect IP address*)
  - Much faster than longest prefix match
  - MPLS forwarding table distinct from IP forwarding tables
- ***flexibility***: MPLS forwarding decisions can *differ* from those of IP

MPLS forwarding



in label	out label	out interface
	10	0
	12	0
	8	1

in label	out label	out interface
10	6	1
12	9	0



There are two predetermined routes from R4 to A

in label	out label	out interface
8	7	0

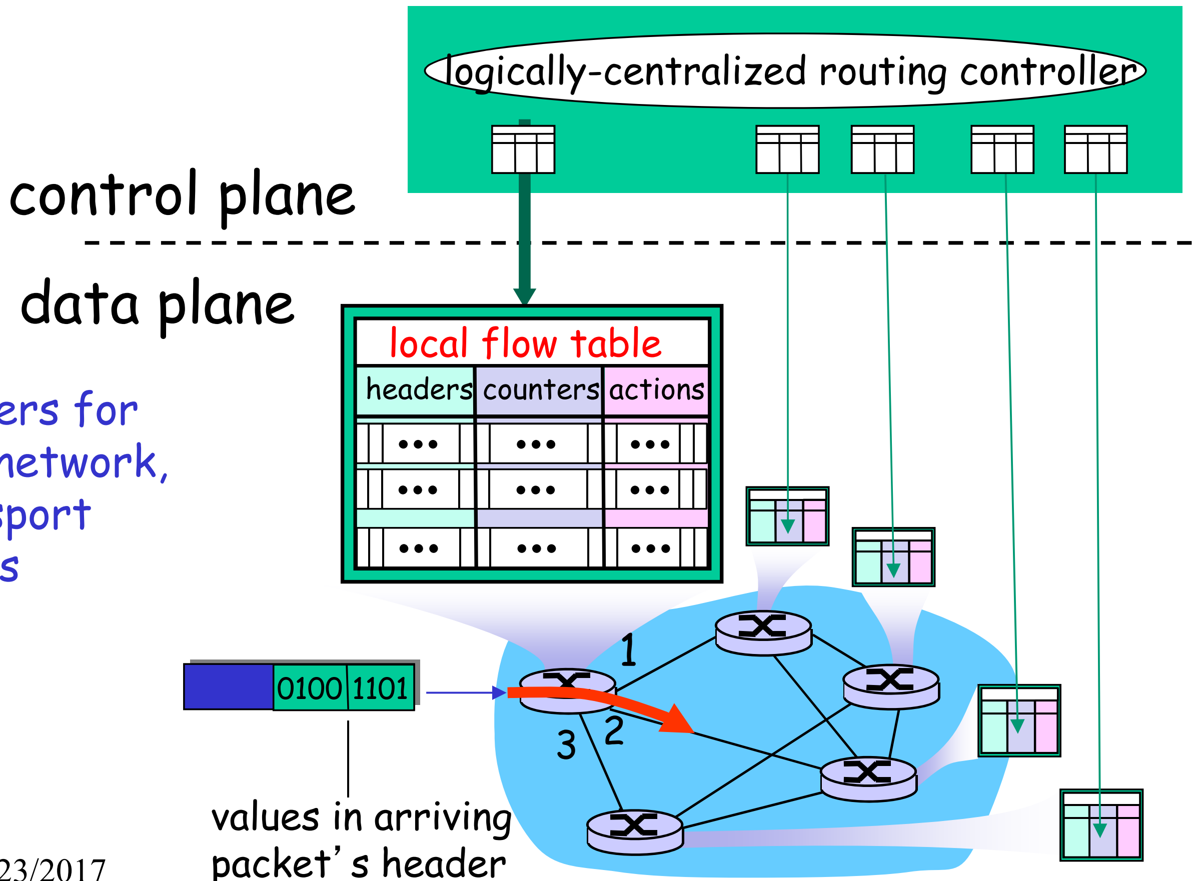
in label	out label	out interface
6	-	0
7	-	0

MPLS applications

- Fast failure recovery - rerouting flows quickly to pre-computed backup paths (useful for VoIP)
- Traffic engineering – network operator can override IP routing and allocate traffic toward the same destination to multiple paths
- Resource provision for virtual private networks

Generalized Forwarding in Software Defined Networking (SDN)

Each router contains a *flow table* that is computed and distributed by a *logically centralized* routing controller



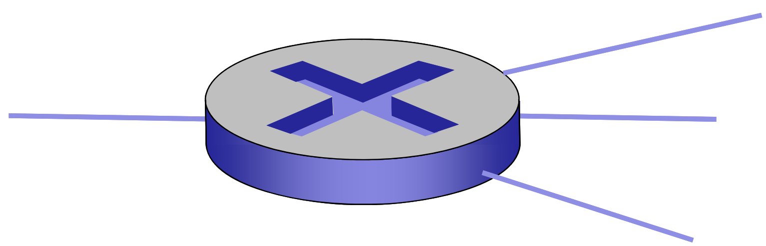
OpenFlow abstraction

- *match+action:* unifies different kinds of devices

- Router (layer3)
  - *match:* longest destination IP prefix
  - *action:* forward to a port
- Switch (layer 2)
  - *match:* destination MAC/VLAN address
  - *action:* forward to port or flood
- Firewall
  - *match*: IP addresses and protocol field, TCP/UDP port numbers
  - *action:* permit or deny
- NAT
  - *match:* IP address and port
  - *action:* rewrite address and port

OpenFlow data plane abstraction

- *flow*: defined by header fields (for link, network, transport layers)
- generalized forwarding
  - *Flow entry*: match fields, priority, counters, instructions
  - *Actions: for matched packet* - drop, forward, modify the packet, or send it to controller



Flow table in a router/switch (computed and distributed by controller) defines router's match+action rules

OpenFlow: Flow Table Entries

Rule	Action	Stats

Packet + byte counters

1. Drop packet
2. Forward packet to port(s)
3. Modify Fields
4. Encapsulate and send to controller

Switch Port	VLAN ID	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP Prot	TCP Src_port	TCP Dst_port

$\underbrace{\text{VLAN ID} \quad \text{MAC src} \quad \text{MAC dst} \quad \text{Eth type}}_{\text{Link layer}}$ $\underbrace{\text{IP Src} \quad \text{IP Dst} \quad \text{IP Prot}}_{\text{Network layer}}$ $\underbrace{\text{TCP Src\_port} \quad \text{TCP Dst\_port}}_{\text{Transport layer}}$

The big picture (review)

- Data plane
  - Forwarding using network and link headers
    - Datagrams
    - VLANs
    - MPLS virtual circuits and IP tunnels (transformers)
    - NATs (transformers)
  - Filtering (access control lists, firewalls)
    - using transport, network, link headers
  - OpenFlow (SDN)
    - match+action abstraction unifies routers, switches, firewalls, and NATs (but not VCs and tunnels)
- Control plane
  - Routing protocols
    - intra-AS (OSPF, distance vector, Cisco proprietary)
    - inter-AS (eBGP, iBGP)
  - SDN
    - centralized controller

The End