Computer Networks
@CS.NYCU

Lecture 4: Network Layer: Data Plane

Instructor: Kate Ching-Ju Lin (林靖茹)

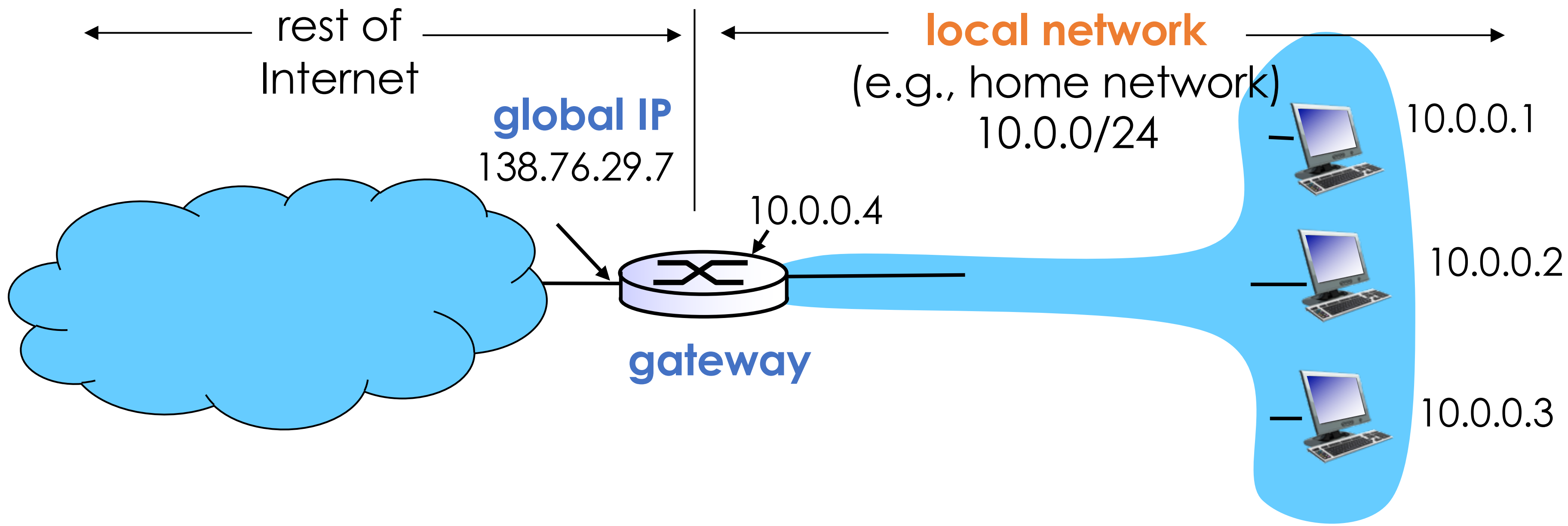
Slides modified from
"Computer Networking: A Top-Down Approach" 7th Edition

Outline

- Overview of network layer
- What's inside a router
- **IP: Internet Protocol**
  - Datagram format
  - IPv4 addressing
  - DHCP
  - **Network address translation (NAT)**
  - IPv6
- Software defined networking

NAT: Network Address Translation

- More and more devices, each needs a global unique IP address?
  - All the devices in a private net use **private IP address**
  - Only the **gateway** gets a global unique IP address
    → translator! Modify the address field of each packet

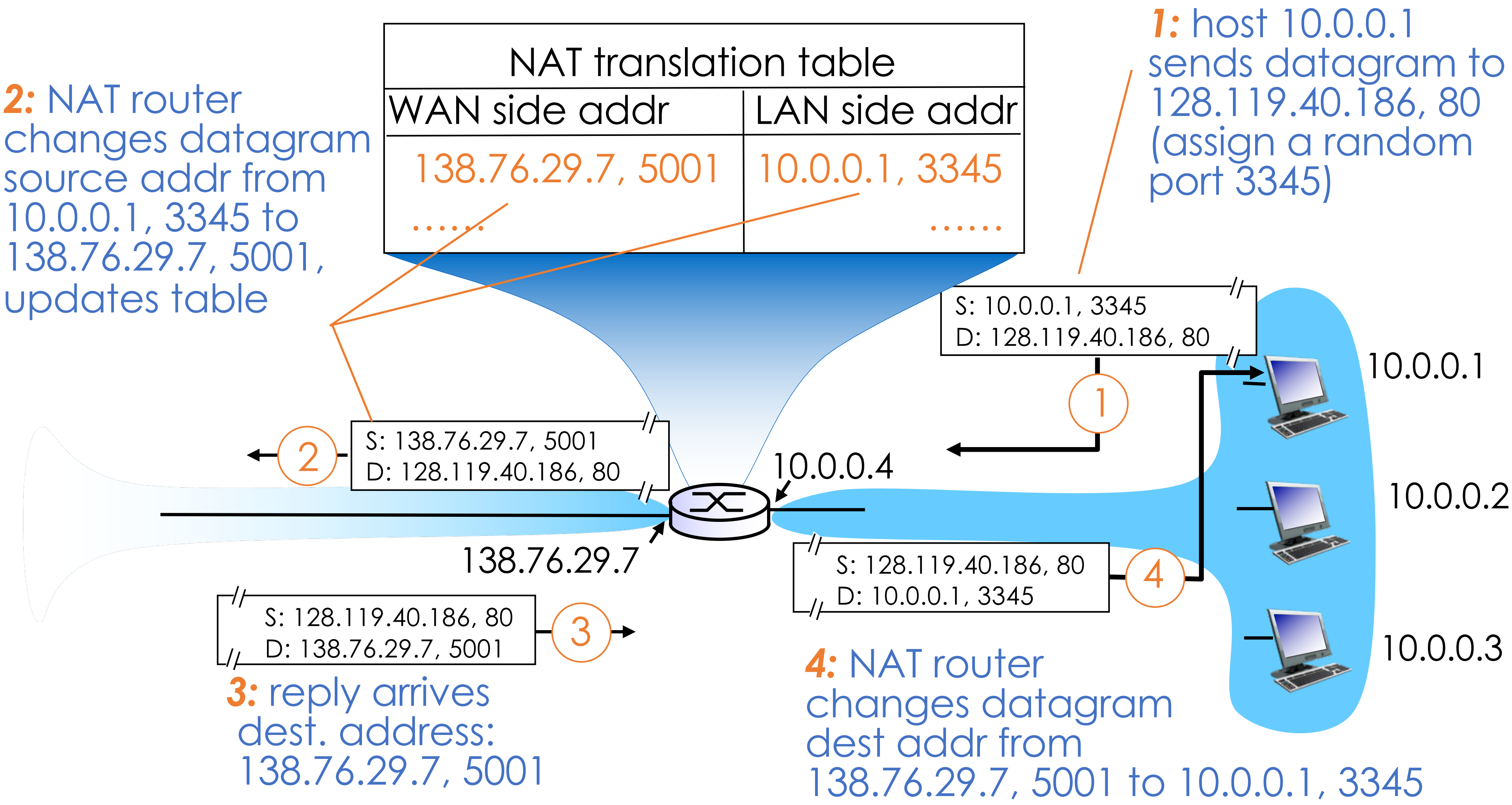


NAT: Network Address Translation

- Packets from all the devices in the private net use the same global public IP address
  - Public IP is assigned by the ISP
  - Private IP addresses are allocated by the gateway

- NAT gateway (router)
  - Translate between public and private IP
  - Modify each packet header
  - Re-route packets to/from the Internet

NAT Translation Table



NAT: Challenges

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - Routers should only process up to layer 3
  - Address shortage should be solved by IPv6
  - Violate end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - **NAT traversal**: what if client wants to connect to server behind NAT?

Outline

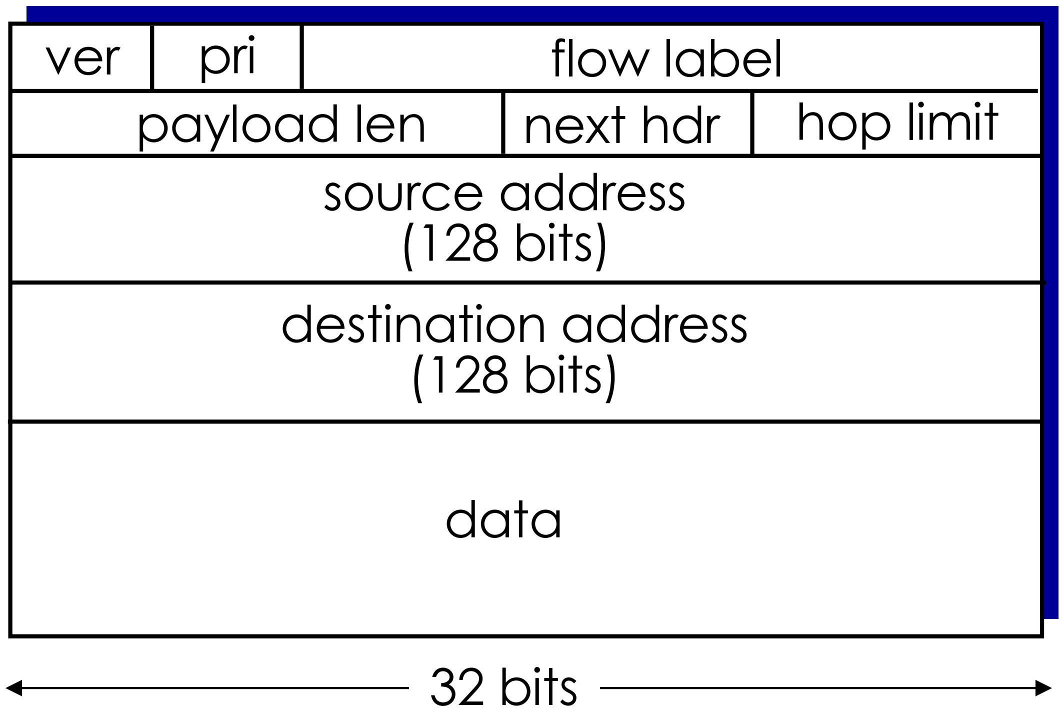
- Overview of network layer
- What's inside a router
- **IP: Internet Protocol**
  - Datagram format
  - IPv4 addressing
  - DHCP
  - Network address translation (NAT)
  - **IPv6**
- Software defined networking

Why IPv6

- **Initial motivation**:
  - 32-bit address space soon to be completely allocated
  - v6: 128-bit address
- Additional motivation:
  - Header format helps speed processing/forwarding
  - Header changes to facilitate QoS
- IPv6 datagram format:
  - fixed-length 40 byte header
  - no fragmentation allowed

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

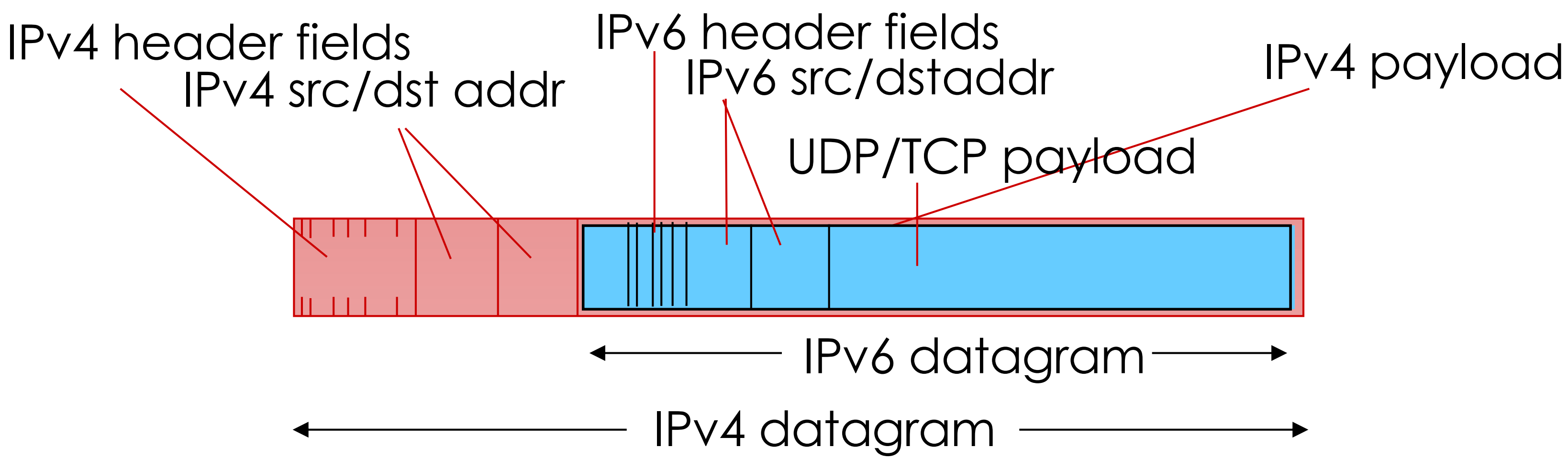


Other Changes from IPv4

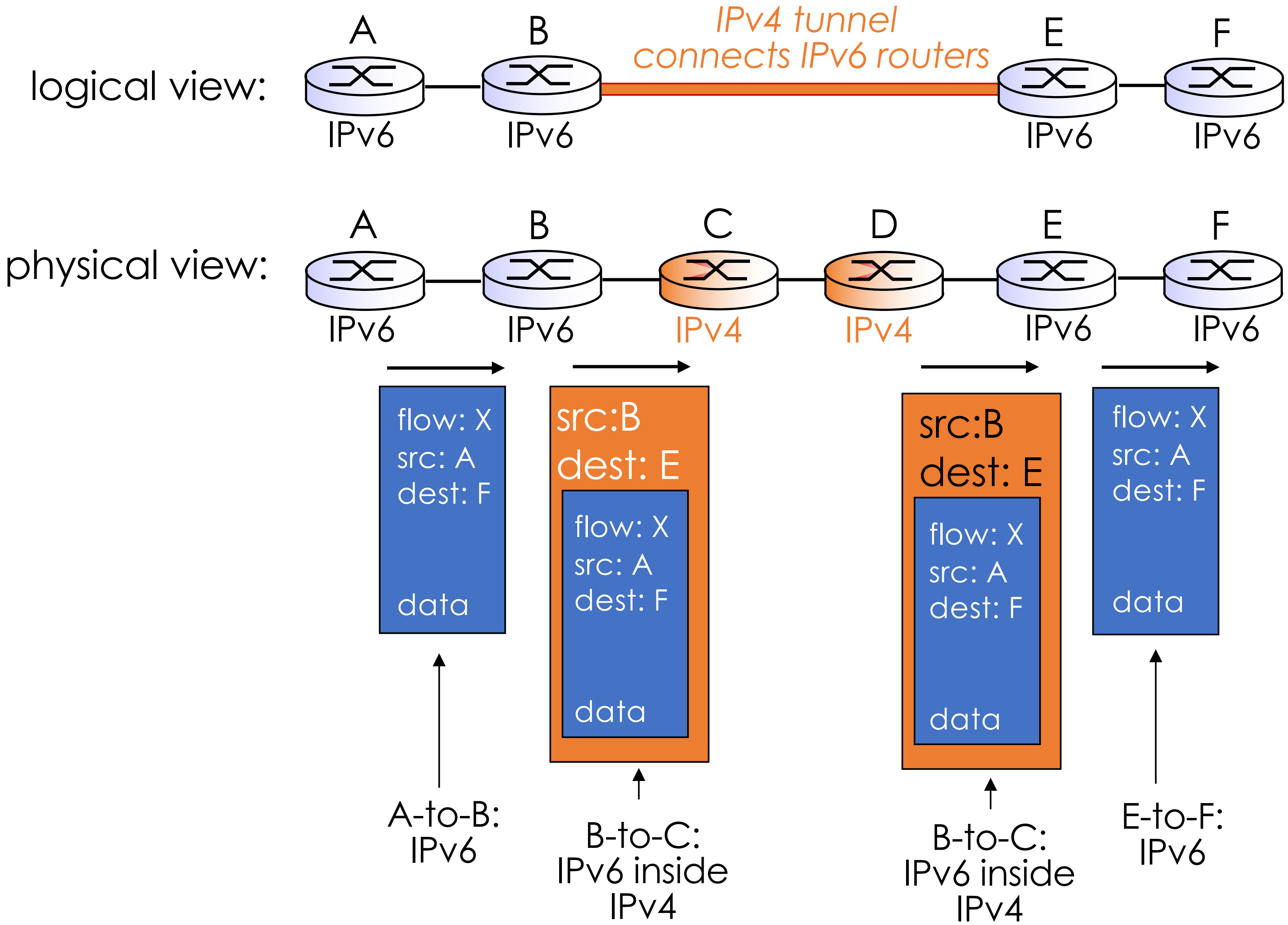
- **Fragmentation/reassembly:**
  - not allowed in routers
  - only performed by a source/destination
- **Checksum:** removed entirely to reduce processing time at each hop
- **Options:** allowed, but outside of header, indicated by "Next Header" field
  - Fix the header to 40 bytes
- **ICMPv6:** new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

Transition from IPv4 to IPv6

- All hosts upgrade simultaneously?
  - Hard in practice
- IPv4 and IPV6 coexist: **tunneling**
  - IPv6 datagram carried as payload in IPv4 datagram among IPv4 routers



Tunneling



logical view:
A
B
IPv4 tunnel
connects 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

IPv6 Adoption

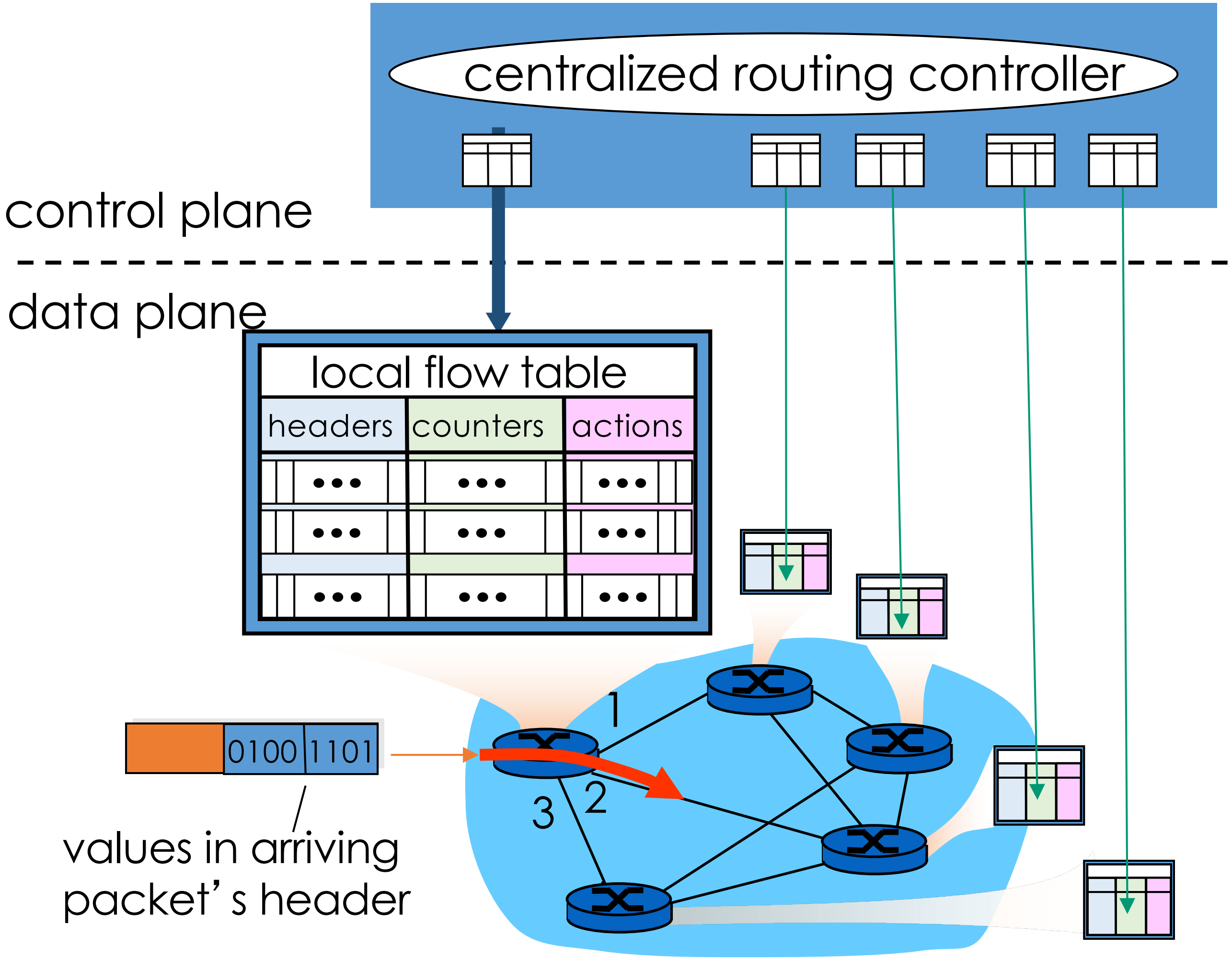
- Google: In 2015. Google reports that only ~8% of clients accessing Google via IPv6
- NIST: in 2015, NIST reports that <1/3 of US governments are IPv6-enabled
- Long (long!) time for deployment, use
  - 20 years and counting!
  - think of application-level changes in last 20 years: WWW, Facebook, streaming media, Skype, …
  - **Why?**

Outline

- Overview of network layer
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - DHCP
  - Network address translation (NAT)
  - IPv6
- **Software defined networking**

Software Defined Networks (SDN)

- Each router has a **flow table** that is computed and installed by a centralized controller

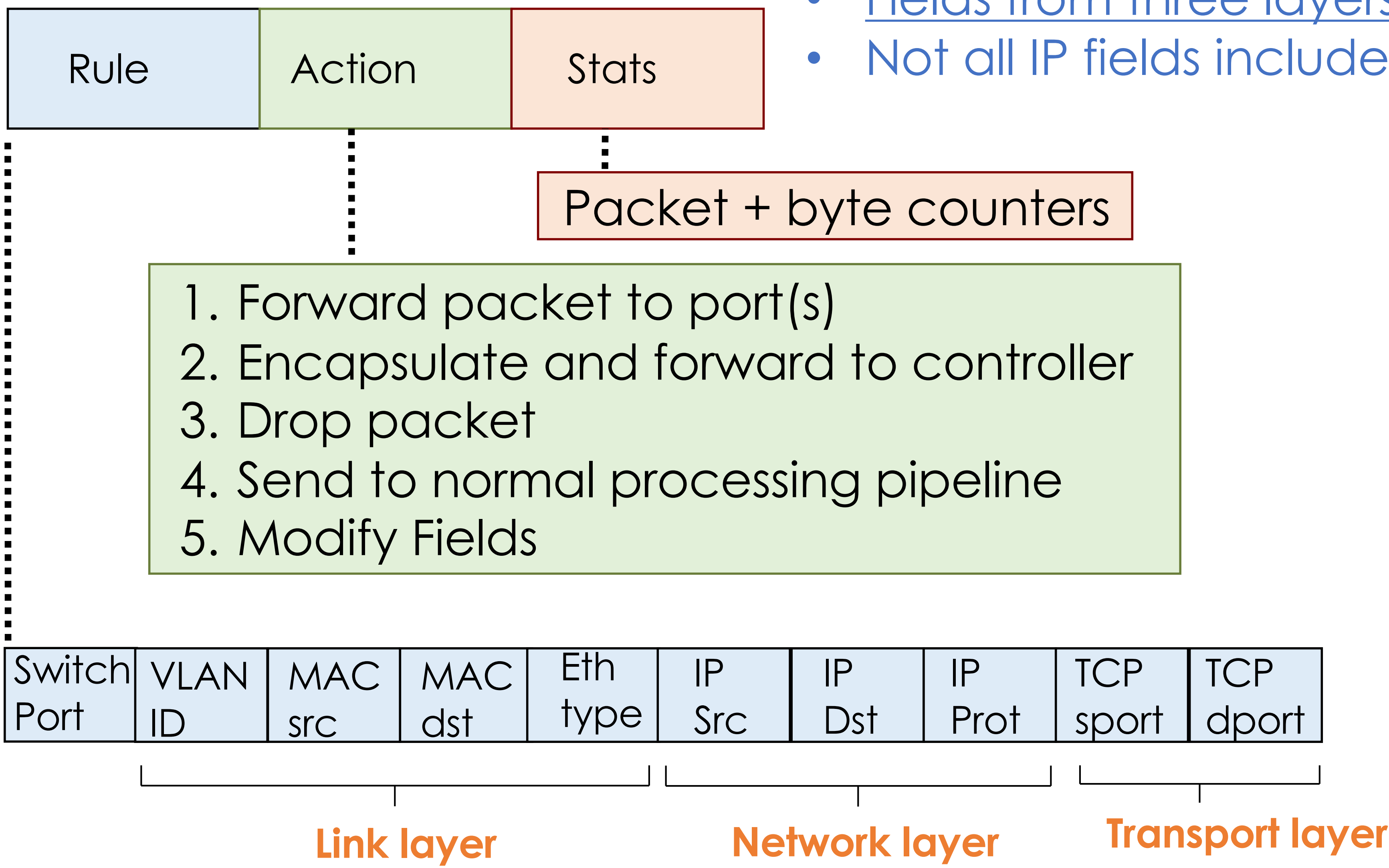


OpenFlow

- Standard for SDN data plane and controllers
  - Currently, version 1.5 (v1.6 only for ONF)
- Match-plus-Action
  - **Match**
    - Look up the felids in each packet header
    - Hardware-based matching: performed in TCAM (fast, but expensive, power consuming)
  - **Action**
    - Forwarding: to one ore more output port
    - Load balancing
    - Rewrite: rewrite header values (e.g., NAT)
    - Blocking/dropping
    - Further processing: send to the controller
  - **Counter**
    - Keep statistics (# bytes or # packets)

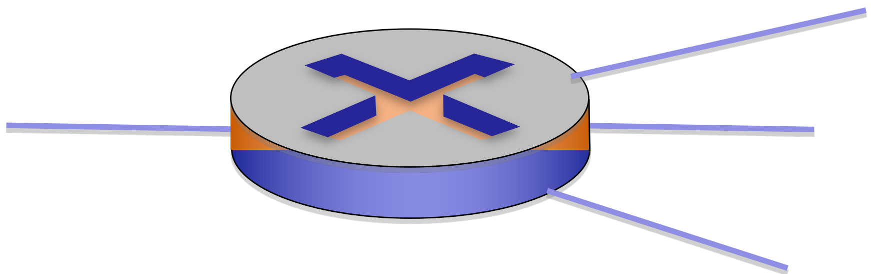
Packet Header Field

- Fields from three layers!
- Not all IP fields included



Match-plus-Action

- Functionality: limited by available fields and actions
- * means wildcard
- Each flowtable entry has a priority



1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

Match-plus-Action

- Offer different kinds of service

1. Router
   - Match: longest dst IP prefix
   - Action: forward to an output port
2. Switch
   - Match: destination MAC address (layer-2 addr)
   - Action: forward or flood
3. Firewall
   - Match: IP address and TCP/UDP port
   - Action: permit or deny
4. NAT
   - Match: IP address and port
   - Action: rewrite address and port

Examples of Match-plus-Action

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

Examples of Match-plus-Action

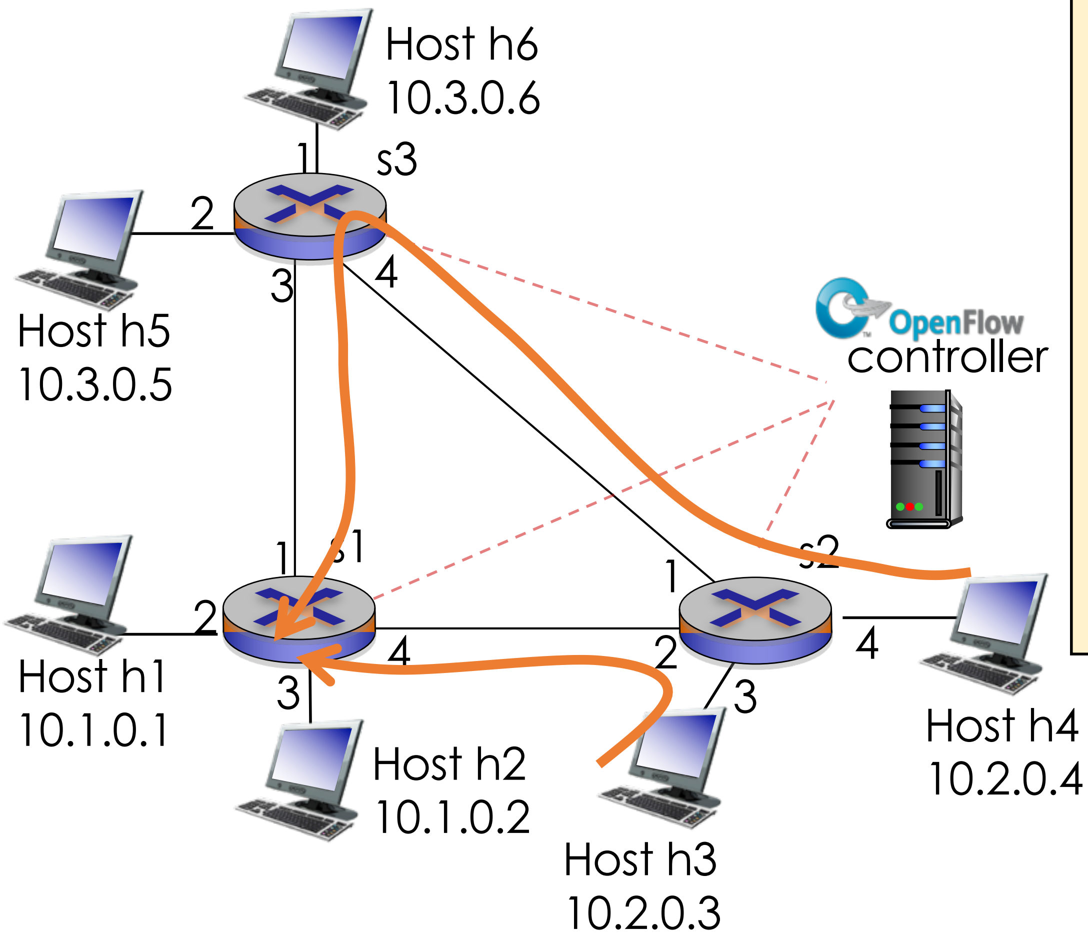
Destination-based layer 2 (switch) forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	port3

layer 2 frames from MAC address 22:A7:23:11:E1:02 should be forwarded to output port 3

OpenFlow Examples

- **Load balancing**:
  - Controller find the specific routing path



For each flow?

No! too many

- **Elephant flows**: long and huge flows (<5% flow, but occupy half bandwidth)
  → specific routing paths
- **Mice flows**: short and small flows
  → traditional shortest path routing

List of SDN Controller Software

- OpenDaylight (part of the Linux Foundation)
- ONOS (distributed via Apache 2.0 license)
  - Supported by ONF
- NOX/POX (first SDN Controller)
- Open vSwitch
- Foodlight (under an Apache 2.0 license)
- Ryu (supported by NTT Labs)
  - Easy prototyping

P4

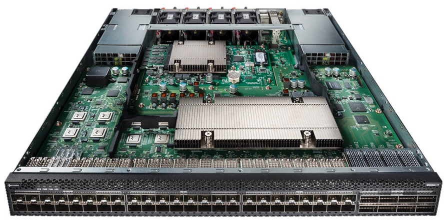
- **Programming language** for controlling packet forwarding planes
- Published in ACM SIGCOMM “**P**rogramming **P**rotocol-Independent **P**acket **P**rocessors”
- Properties
  - **Target independence** (can be compiled in any machine)
  - **Protocol independence** (no native support for any protocol, e.g., IP, TCP or Ethernet)
  - **Reconfigurability** (able to change the way they process packets)

P4 Application

Source: https://www.stordis.com/p4-programming-language/?gclid=Cj0KCQiAiZPvBRDZARIsAORkq7df755k7Fhipdkx-fso5WCiu9iFPXmEts81R0FeLj4vppcFQo2ETIEaApfbEALw_wcB

P4 Programmable Switch

- Barefoot Tofino chipset



Networks Ports

48x25G + 8x100G in 1RU Chassis
Port 1 – Port 16: Support 1/10/25GbE
Port 17 – Port 48: Support 10/25GbE

ASIC

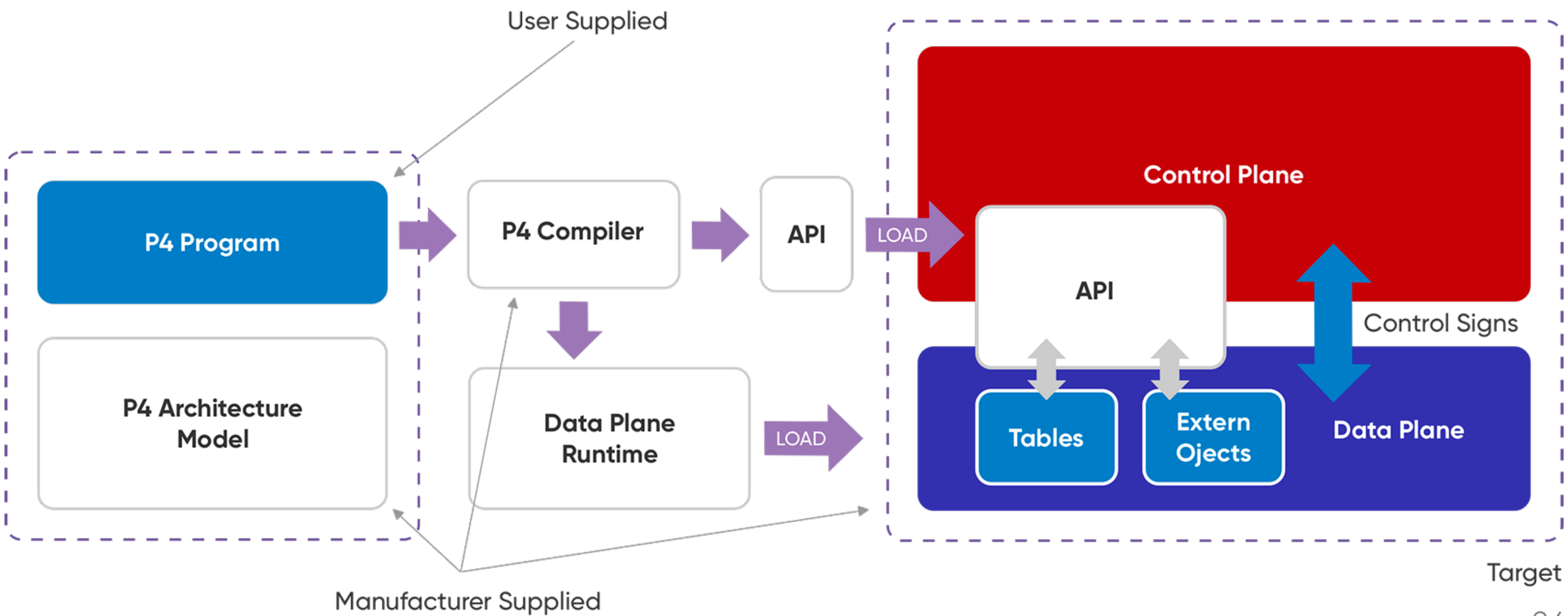
Barefoot Tofino 2.0Tbit

CPU & Core

Broadwell-DE 8-core @2.0GHz
32G DDR4
128G SSD

Timesync option

1588v2 PTP Time Synchronization



Quiz

- What does ”IPV6 Tunneling“ mean?
  - Encrypt IPv6 packets with IPv4 headers such that IPv4 routers can forward the packets

Quiz

- Explain why it is difficult to build a server within a private network
  - Hosts in the Internet do not know what is the global IP of a server behind a private network