南京大學
NANJING UNIVERSITY

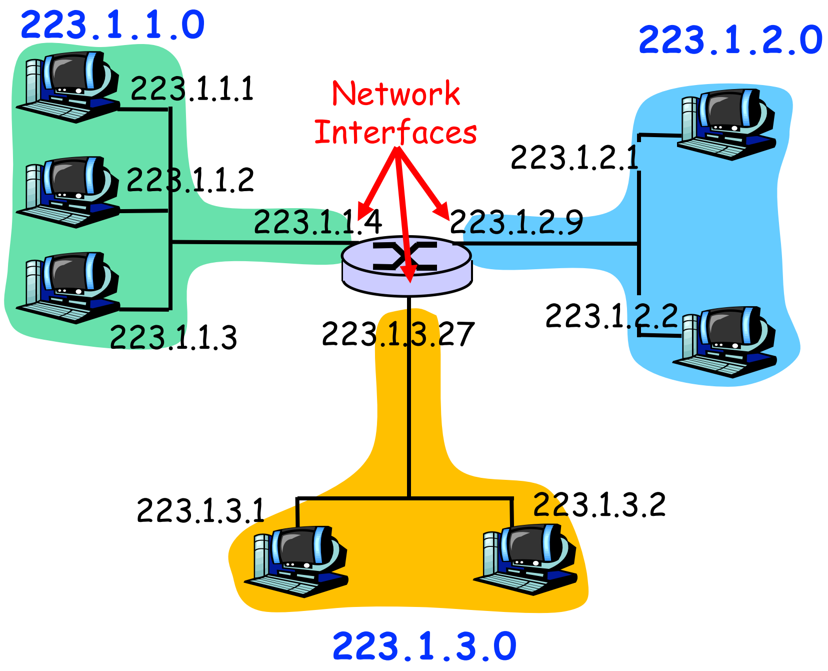
网络层: 数据平面

Outline

- IP Addressing
- Network Address Translation
- IPv6
- Generalized Forwarding and SDN
- Middleboxes

IP Addressing

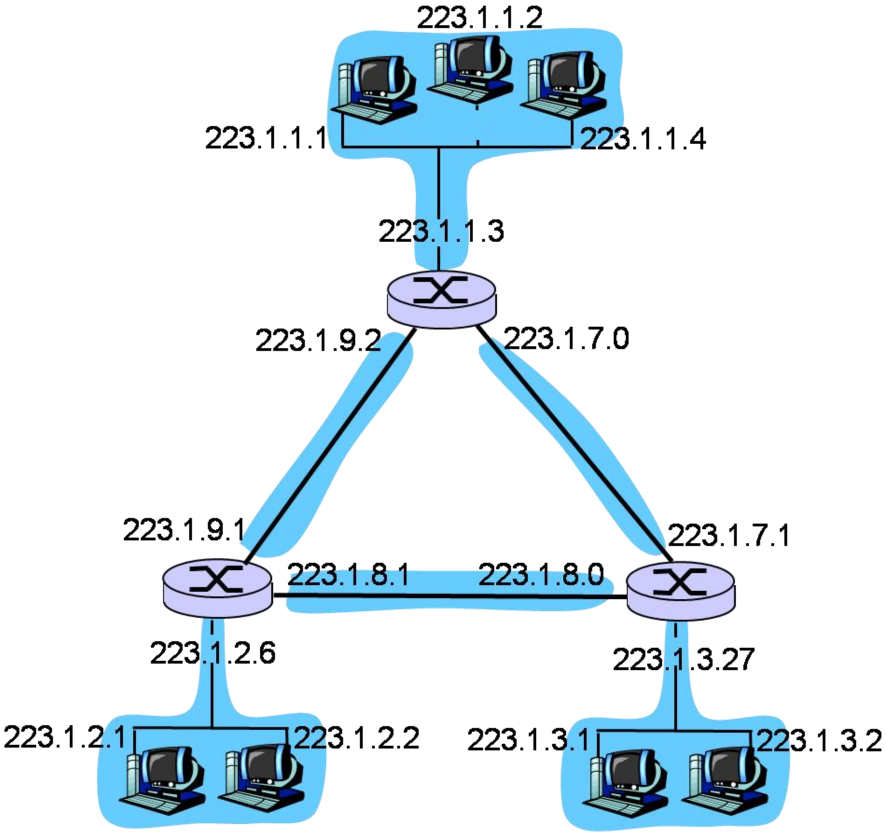
- IP address
  - 32 bit global internet address for each interface
  - Network part (high order bits)
  - Host part (low order bits)
- Physical network (from IP perspective)
  - Can reach each other without intervening router



$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

Count the Physical Networks

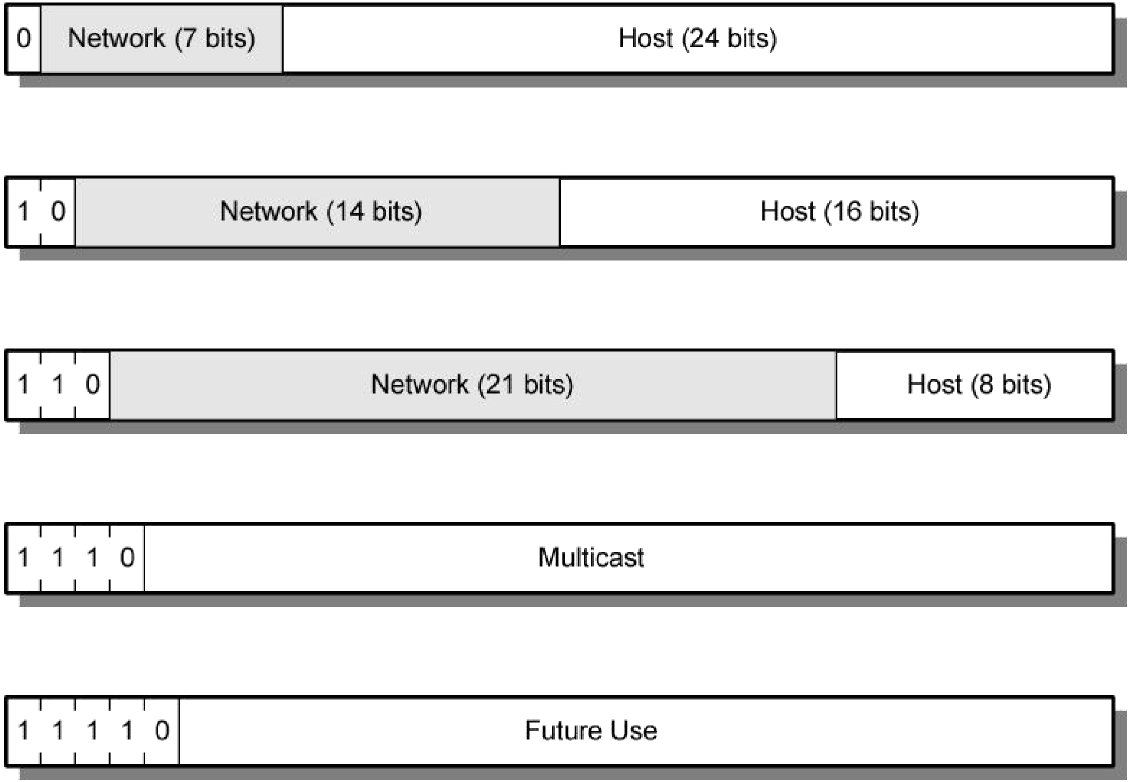
- How many ?



IP Address

- A separate address is required for each physical interface of a host/router to a network
  - Facilitates routing
- Use Dotted-Decimal Notation
- netid unique & administered by
  - American Registry for Internet Numbers (ARIN)
  - Reseaux IP Europeens (RIPE)
  - Asia Pacific Network Information Centre (APNIC)
- hostid assigned within designated organization

IPv4 Address Formats



0 | Network (7 bits) | Host (24 bits) — Class A

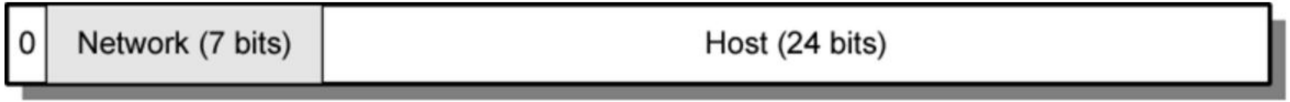
1 0 | Network (14 bits) | Host (16 bits) — Class B

1 1 0 | Network (21 bits) | Host (8 bits) — Class C

1 1 1 0 | Multicast — Class D

1 1 1 1 0 | Future Use — Class E

IP Addresses - Class A

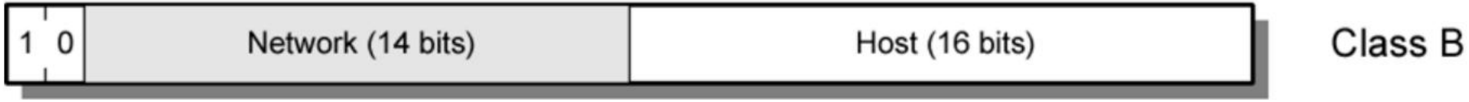


- Start with binary 0
- Reserved netid
  - All 0 reserved
  - 01111111 (127) reserved for loopback
- Range 1.x.x.x to 126.x.x.x
- Up to 16 million hosts
- All allocated

A类地址：
- 首位为0；
- 支持2^7-2=126个网段；
- 每个网段支持主机数为2^{24}-2 =16777214（全0和全1的地址要扣除，全0是网络号，全1是广播号）

- 127.*.*.*: 回环测试，用于测试本地网卡。127.0.0.1 “localhost”

IP Addresses - Class B

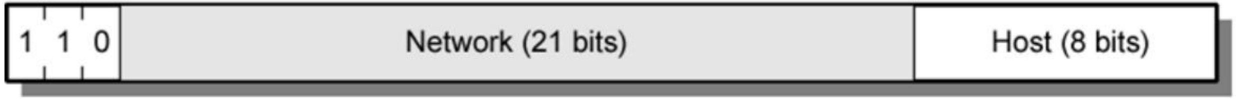


1 | 0 | Network (14 bits) | Host (16 bits)

Class B

- Start with 10
- Range 128.0.x.x to 191.255.x.x
- Second Octet also included in network address
- 2^{14} = 16,384 class B networks
- Up to 65,000 (=2^{16}-2) hosts
- *All allocated*

IP Addresses - Class C



1 1 0

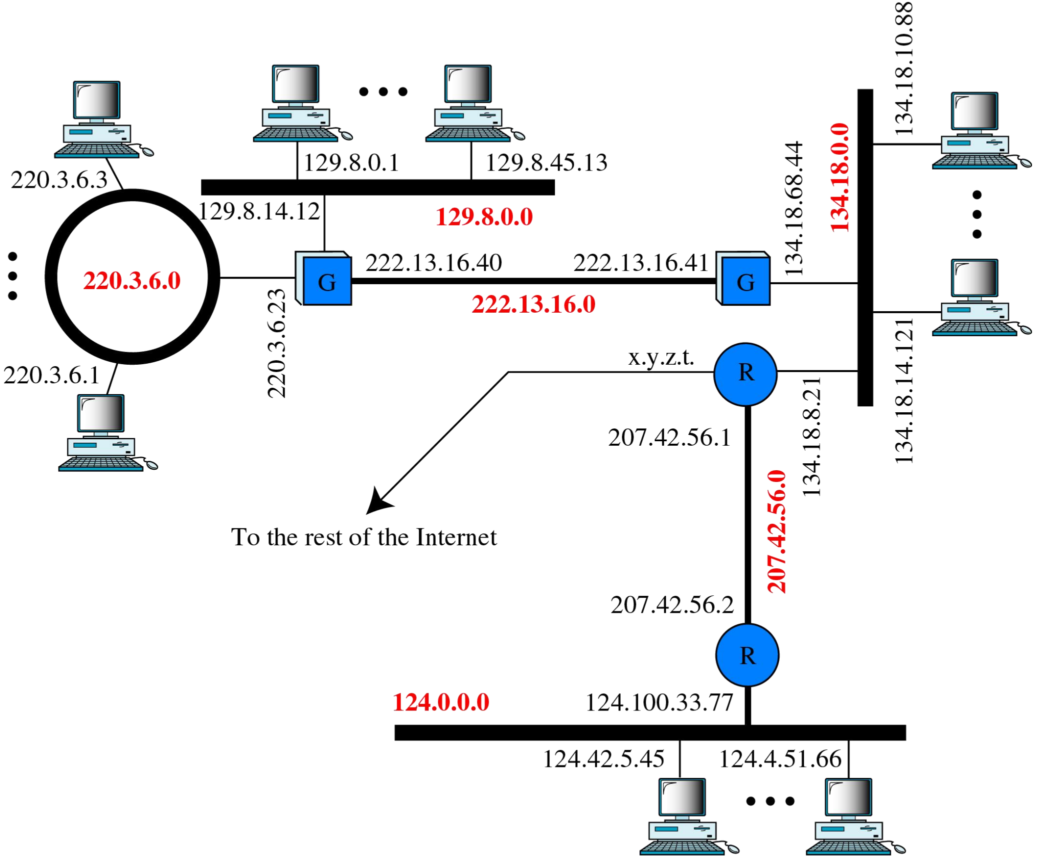
Network (21 bits)

Host (8 bits)

Class C

- Start with 110
- Range 192.0.0.x to 223.255.255.x
- Second and third octet also part of network address
- 2^{21} = 2,097,152 networks
- Up to 254 (=2^8-2) hosts
- Nearly all allocated

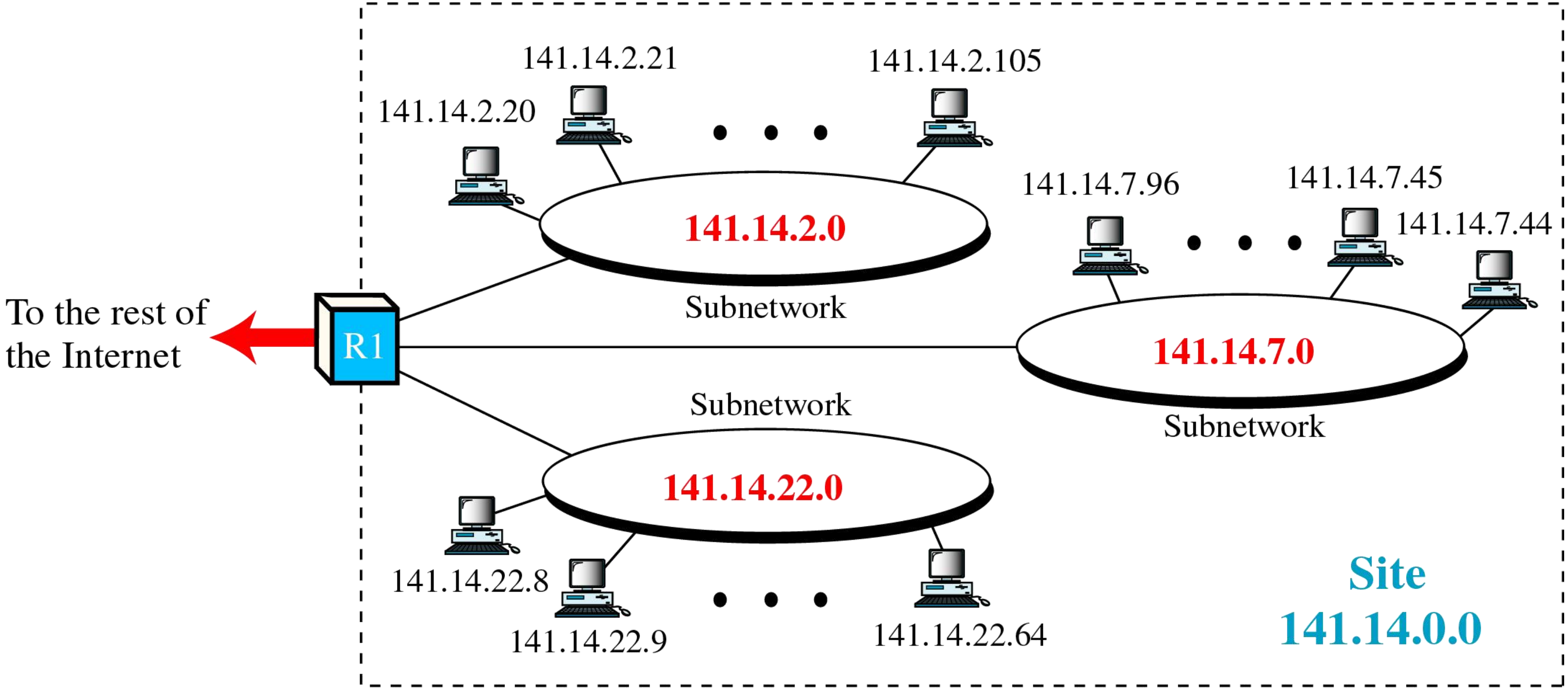
Inter-Networks with Addresses



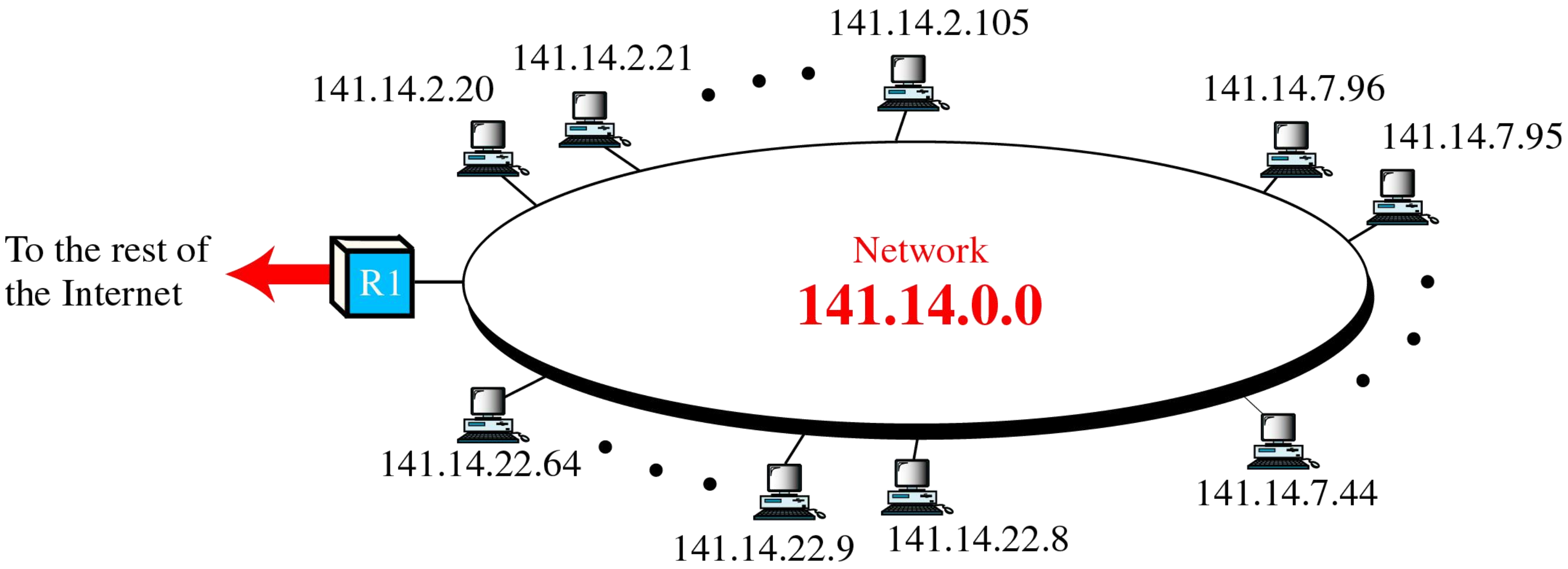
Subnets and Subnet Masks

- Handle problem of network address inadequacy
- Host portion of address partitioned into subnet number and host number
  - Subnet mask indicates which bits are subnet number and which are host number
  - Each LAN assigned a subnet number, more flexibility
  - Local routers route within subnetted network
- Subnets looks to rest of internet like a single network
  - Insulate overall Internet from growth of network numbers and routing complexity

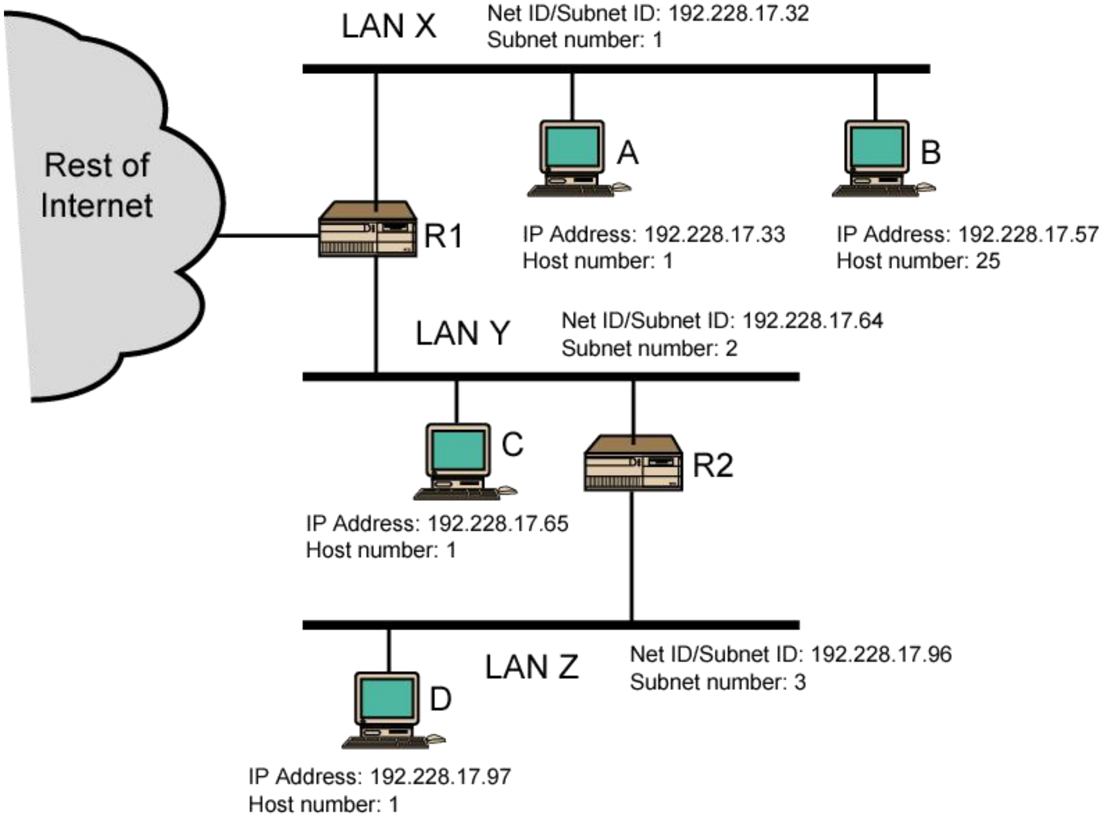
Subnets Example



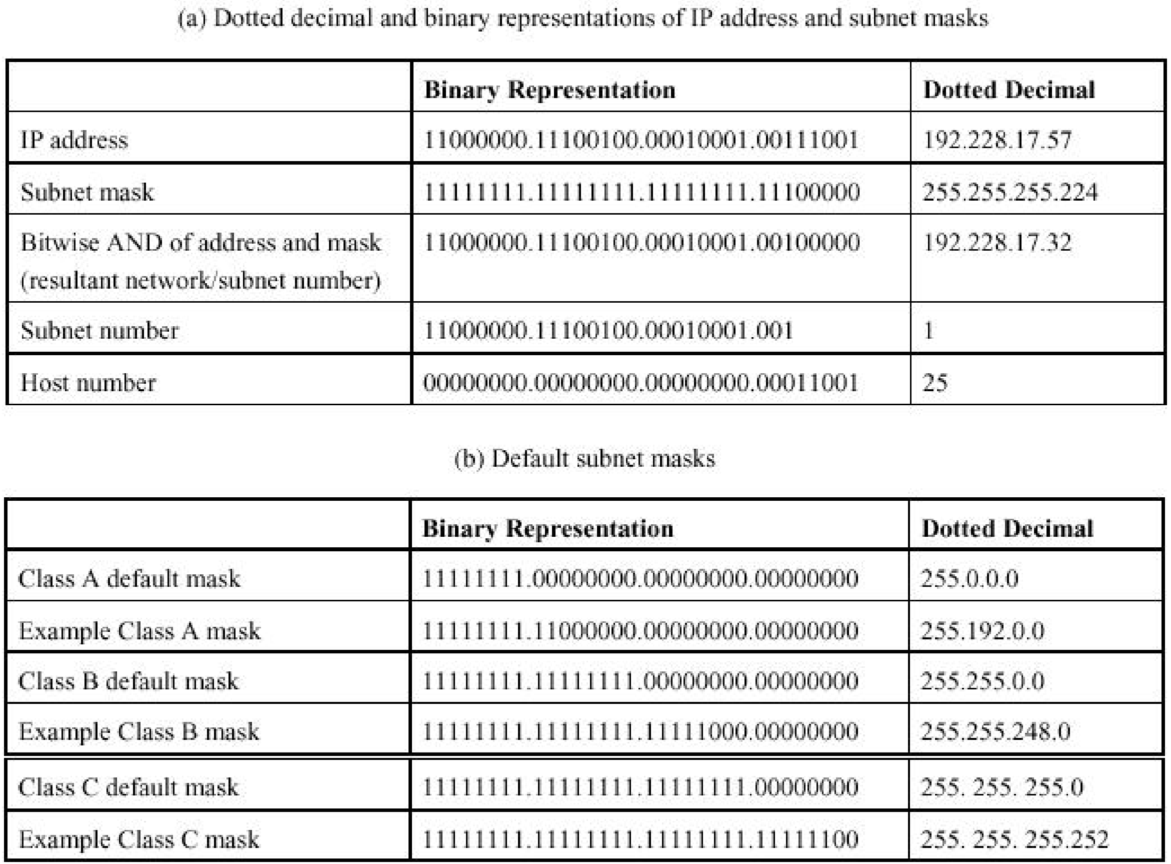
Subnets to the Rest



Routing Using Subnets (1)


LAN X
Net ID/Subnet ID: 192.228.17.32
Subnet number: 1
A
B
Rest of
Internet
R1
IP Address: 192.228.17.33
Host number: 1
IP Address: 192.228.17.57
Host number: 25
LAN Y
Net ID/Subnet ID: 192.228.17.64
Subnet number: 2
C
R2
IP Address: 192.228.17.65
Host number: 1
LAN Z
Net ID/Subnet ID: 192.228.17.96
Subnet number: 3
D
IP Address: 192.228.17.97
Host number: 1

Routing Using Subnets (2)



(a) Dotted decimal and binary representations of IP address and subnet masks

	Binary Representation	**Dotted Decimal**
IP address	11000000.11100100.00010001.00111001	192.228.17.57
Subnet mask	11111111.11111111.11111111.11100000	255.255.255.224
Bitwise AND of address and mask (resultant network/subnet number)	11000000.11100100.00010001.00100000	192.228.17.32
Subnet number	11000000.11100100.00010001.001	1
Host number	00000000.00000000.00000000.00011001	25

(b) Default subnet masks

	Binary Representation	**Dotted Decimal**
Class A default mask	11111111.00000000.00000000.00000000	255.0.0.0
Example Class A mask	11111111.11000000.00000000.00000000	255.192.0.0
Class B default mask	11111111.11111111.00000000.00000000	255.255.0.0
Example Class B mask	11111111.11111111.11111000.00000000	255.255.248.0
Class C default mask	11111111.11111111.11111111.00000000	255. 255. 255.0
Example Class C mask	11111111.11111111.11111111.11111100	255. 255. 255.252

CIDR Notation

- Classless Inter Domain Routing (CIDR)
  - An IP address is represented as "A.B.C.D/n", where n is called the IP (network) prefix

IP Address	10 00001010	217 11011001	123 01111011	7 00000111
Subnet	255 11111111	255 11111111	240 11110000	0 00000000
Network ID	00001010	11011001	01110000	00000000
CIDR	10.217.112.0/20			

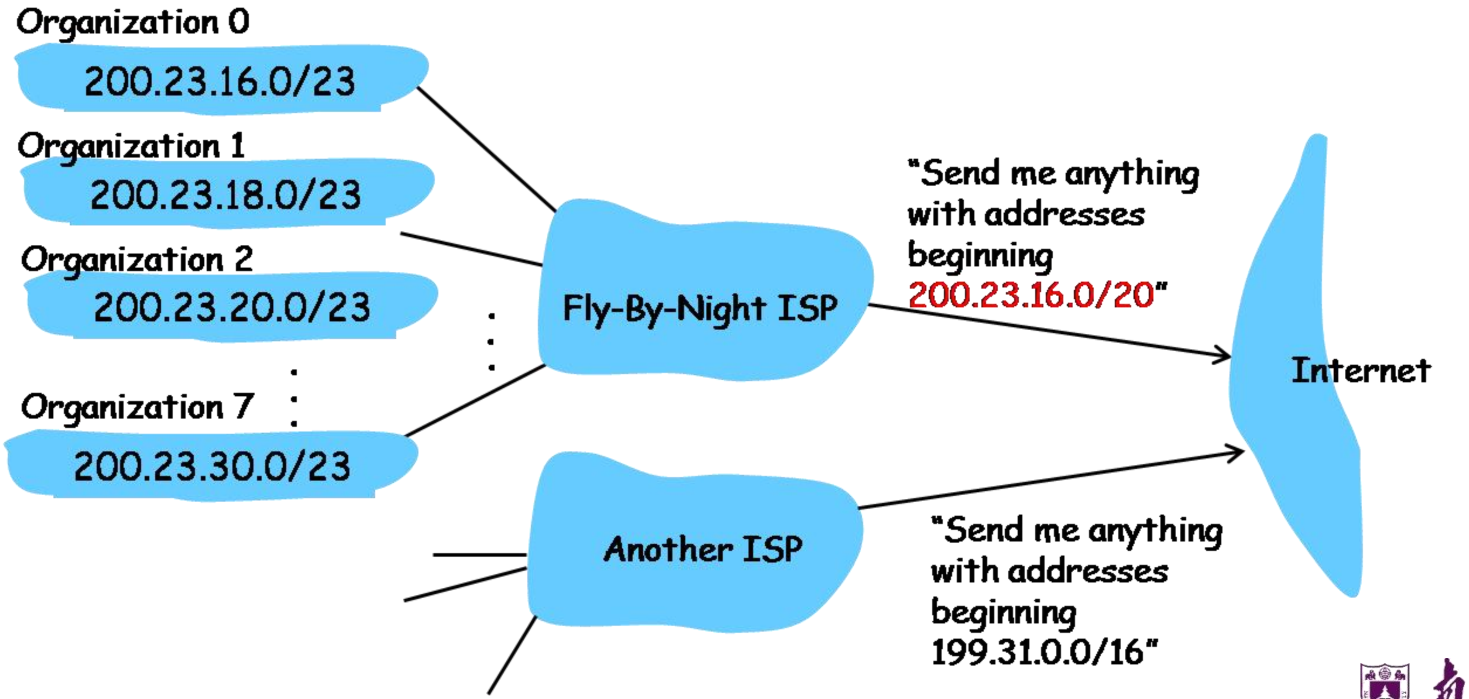
More General Case

- An ISP can be looked as a set of subnets
  - Support many organizations (Intranets)
  - Hierarchical addressing

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

Route Aggregation

- Allows efficient advertisement of routing information



"Send me anything with addresses beginning 200.23.16.0/20"

"Send me anything with addresses beginning 199.31.0.0/16"

IP addresses: how to get one?

That's actually two questions:

1. Q: How does a host get IP address within its network (host part of address)?
2. Q: How does a network get IP address for itself (network part of address)

How does host get IP address?

- hard-coded by sysadmin in config file (e.g., /etc/rc.config in UNIX)
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"

DHCP: Dynamic Host Configuration Protocol

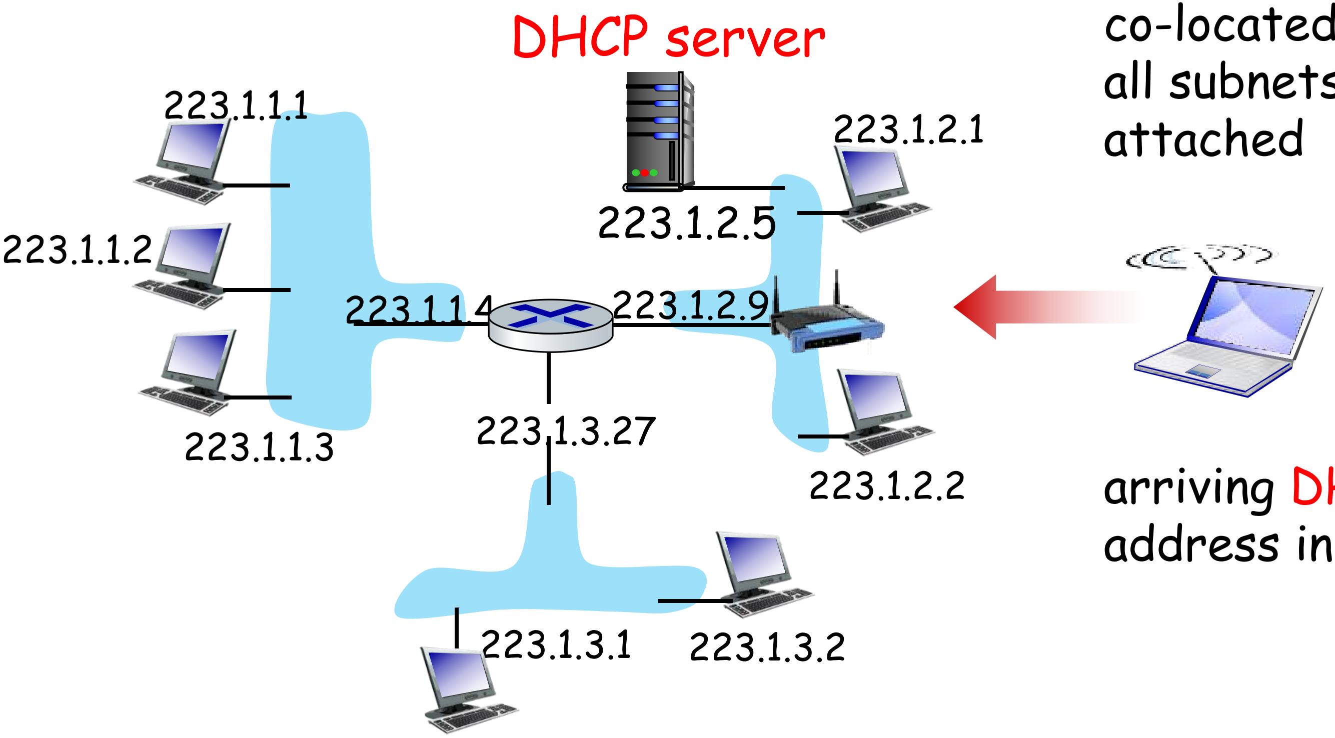
Goal: host dynamically obtains IP address from network server when it "joins" network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/on)
- support for mobile users who join/leave network

DHCP overview:

- host broadcasts DHCP discover msg [optional]
- DHCP server responds with DHCP offer msg [optional]
- host requests IP address: DHCP request msg
- DHCP server sends address: DHCP ack msg

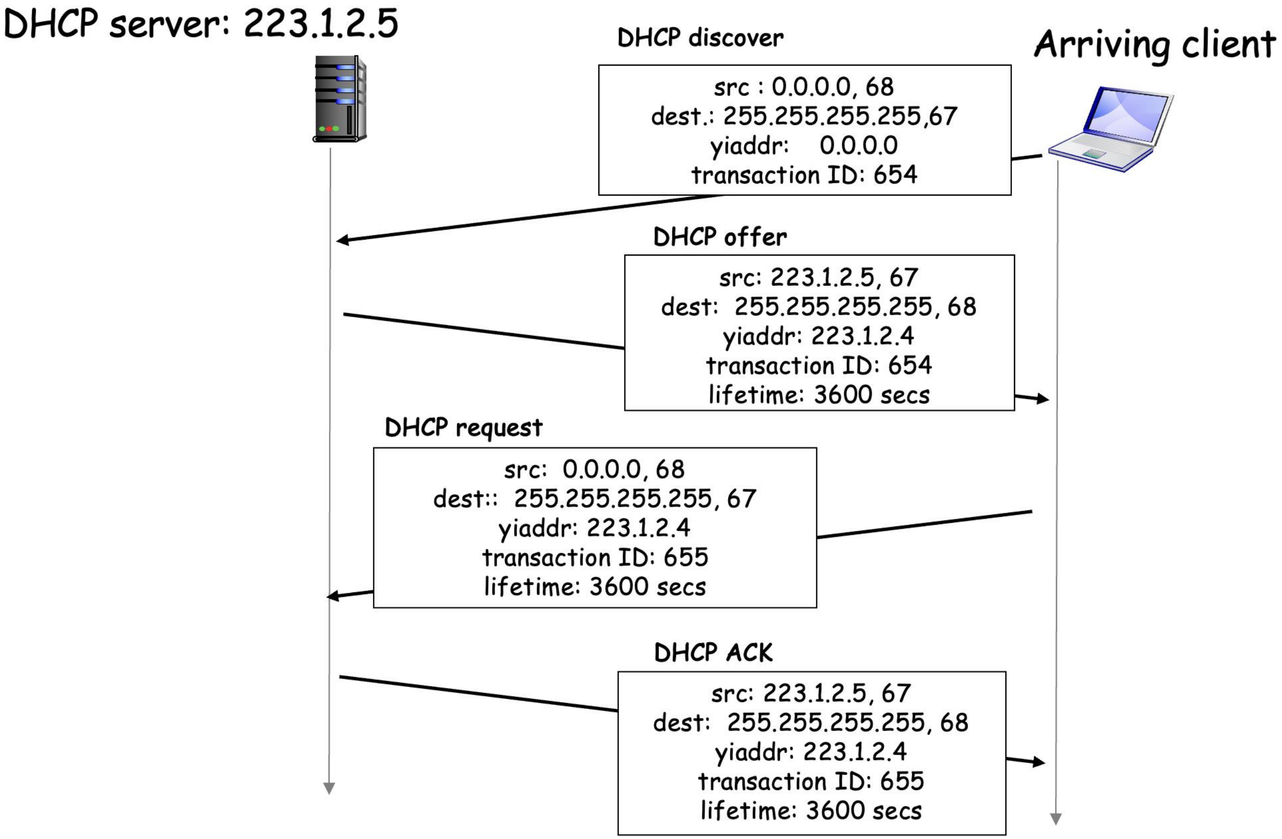
DHCP client-server scenario



Typically, DHCP server will be co-located in router, serving all subnets to which router is attached

arriving DHCP client needs address in this network

DHCP client-server scenario



DHCP server: 223.1.2.5

Arriving client

DHCP discover

src : 0.0.0.0, 68
dest.: 255.255.255.255,67
yiaddr: 0.0.0.0
transaction ID: 654

DHCP offer

src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 654
lifetime: 3600 secs

DHCP request

src: 0.0.0.0, 68
dest:: 255.255.255.255, 67
yiaddr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs

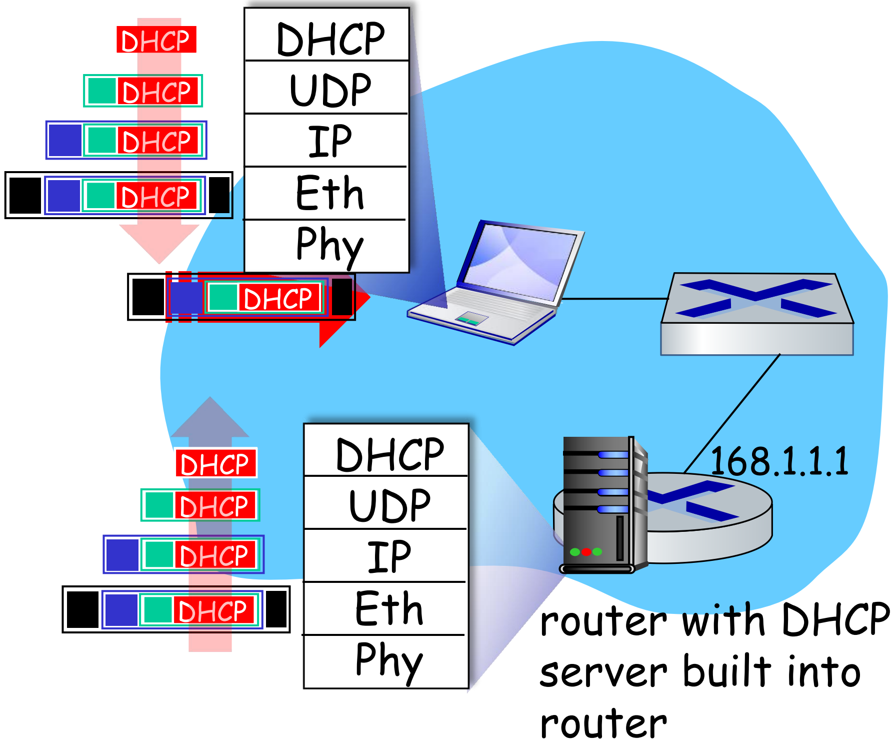
DHCP ACK

src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs

DHCP: more than IP addresses

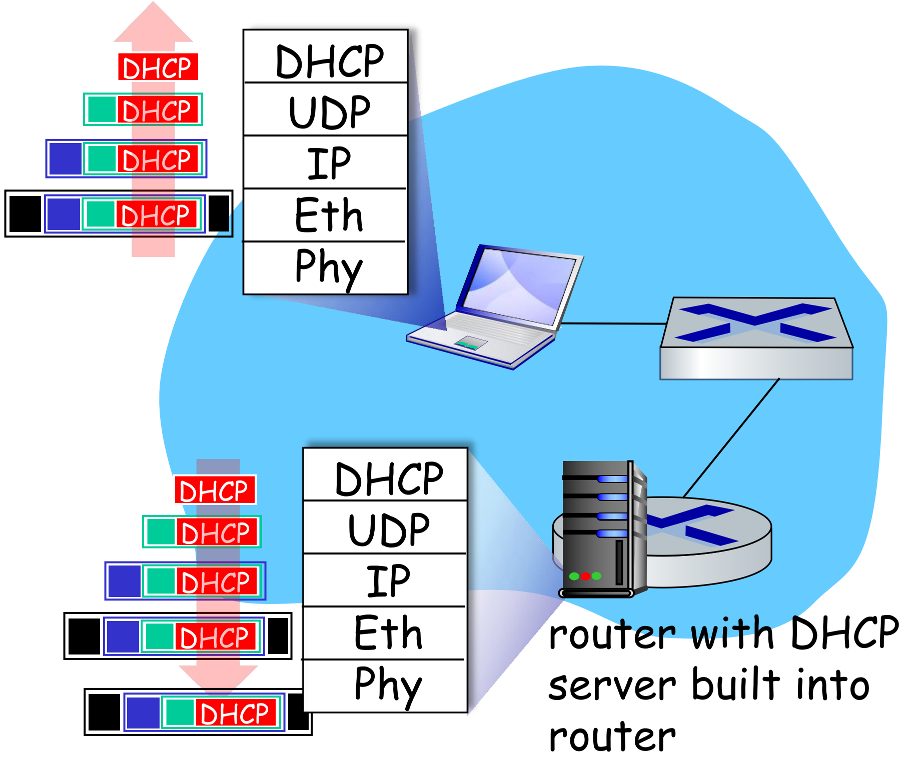
- DHCP can return more than just allocated IP address on subnet:
  - address of first-hop router for client
  - name and IP address of DNS sever
  - network mask (indicating network versus host portion of address)

DHCP: example



- Connecting laptop will use DHCP to get IP address, address of first-hop router, address of DNS server.
- DHCP REQUEST message encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet de-mux'ed to IP de-mux'ed, UDP de-mux'ed to DHCP

DHCP: example



router with DHCP server built into router

- DHCP server formulates DHCP ACK containing client’s IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulated DHCP server reply forwarded to client, de-muxing up to DHCP at client
- client now knows its IP address, name and IP address of DNS server, IP address of its first-hop router

Outline

- IP Addressing
- Network Address Translation
- IPv6
- Generalized Forwarding and SDN
- Middleboxes

Network Address Translation

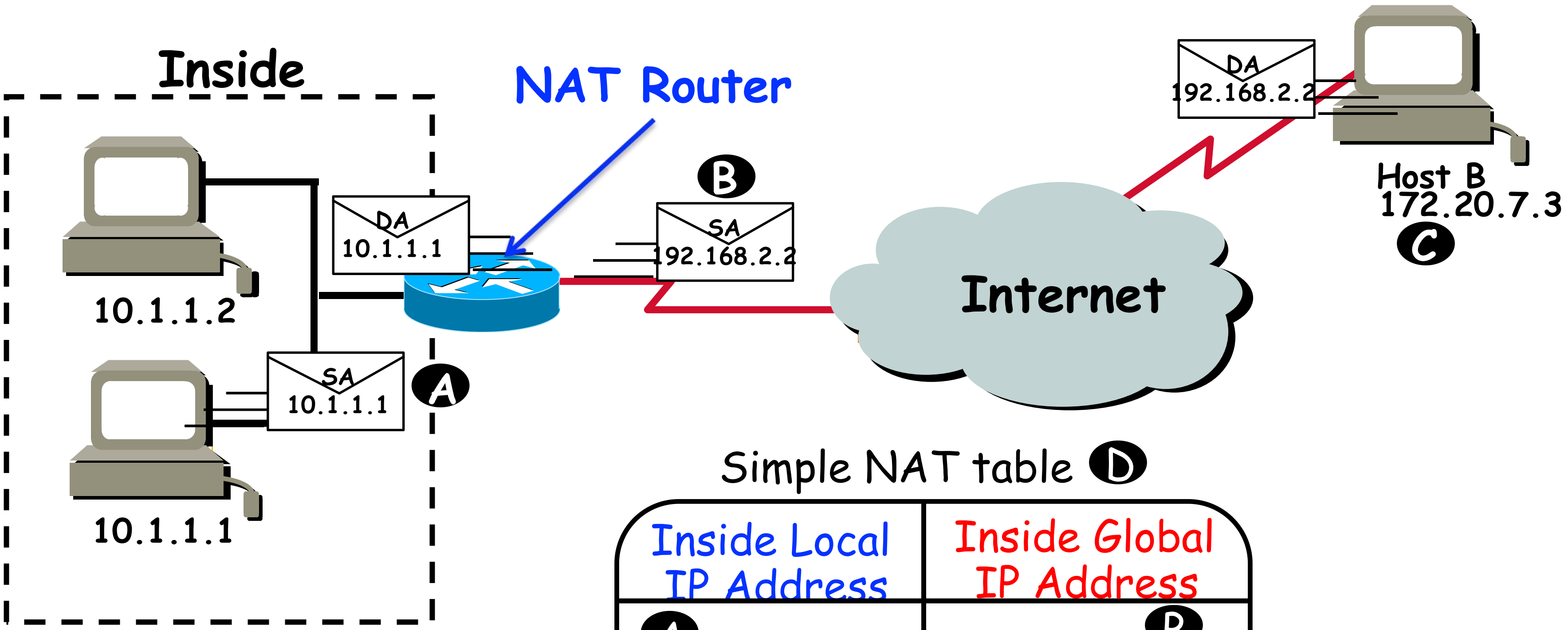
- NAT
  - Enables different sets of IP addresses for internal and external traffic
  - The IP address translations occur where the Intranet interfaces with the broader Internet

- Purposes
  - Acts as a firewall by hiding internal IP addresses
  - Enables an enterprise (organization) to use more internal IP addresses
  - Isolate the (organization / ISP) changes

3 Types of NAT

- Static NAT
  - A private IP address is mapped to one reserved public IP address
  - Usually for server hosts in Intranet
- Dynamic NAT
  - The NAT router keeps a pool of registered IP addresses, and assign to private IP addresses on demand
  - Usually for client PCs in Intranet
- Single-Address NAT/Overloading/Masquerading/Network Address Port Translation (NAPT)

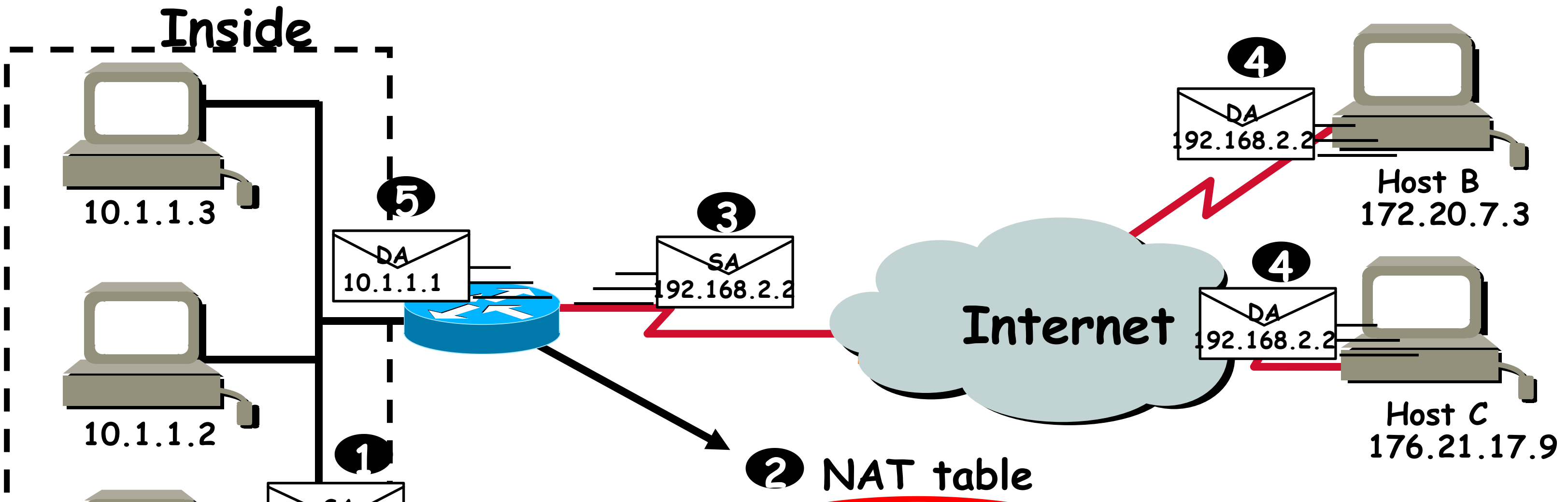
Illustration of NAT



Simple NAT table D

Inside Local IP Address	Inside Global IP Address
A 10.1.1.2	B 192.168.2.3
10.1.1.1	192.168.2.2

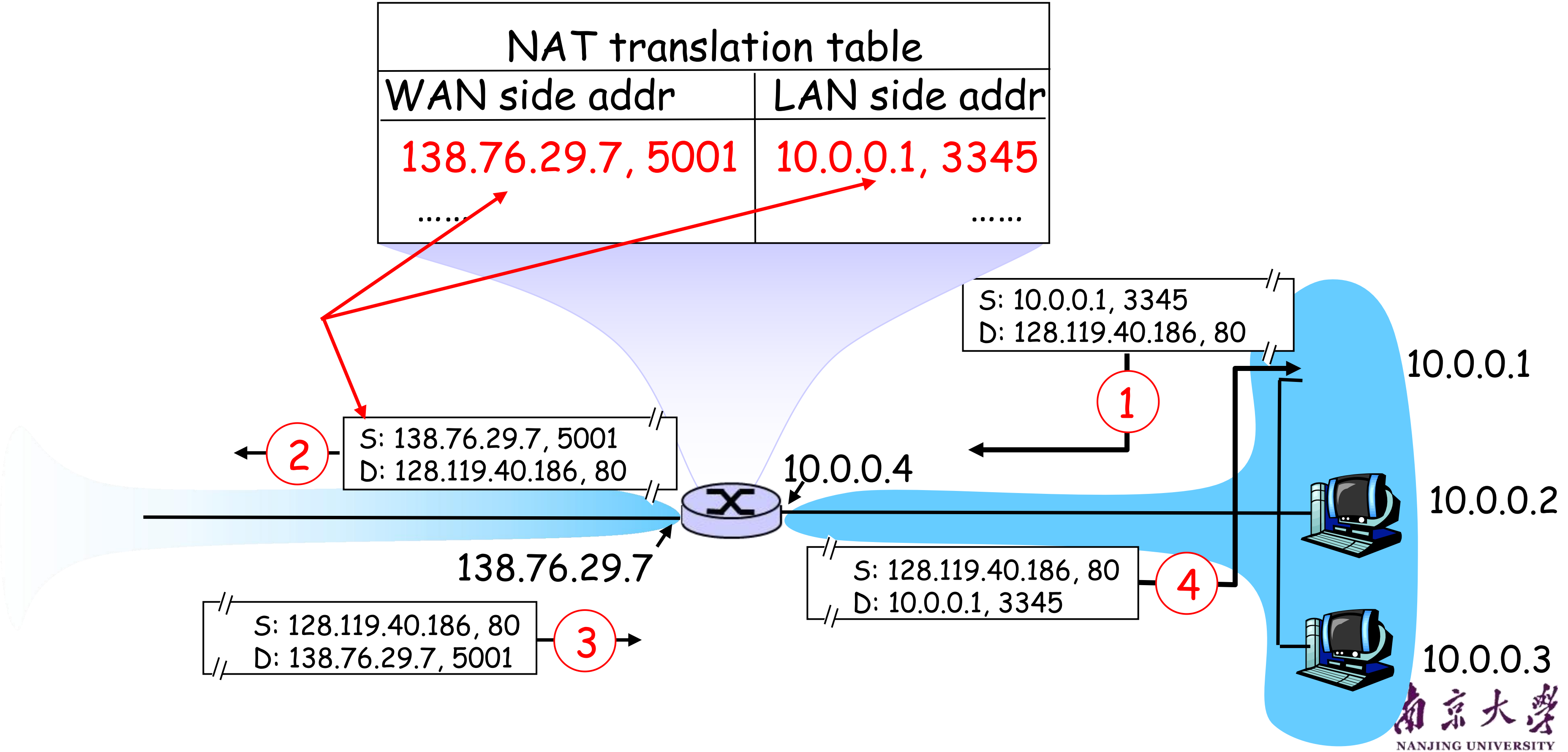
Overloading Global Address



NAT table

Protocol 10.1.1.1	Inside Local IP Address: Port	Inside Global IP Address: Port	Outside Global IP Address: Port
TCP	10.1.1.3:1723	192.168.2.2:1492	176.21.17.9:23
TCP	10.1.1.2:1723	192.168.2.2:1723	176.21.17.9:23
TCP	10.1.1.1:1024	192.168.2.2:1024	172.20.7.3:23

Network Address Translation



NAT translation table

WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

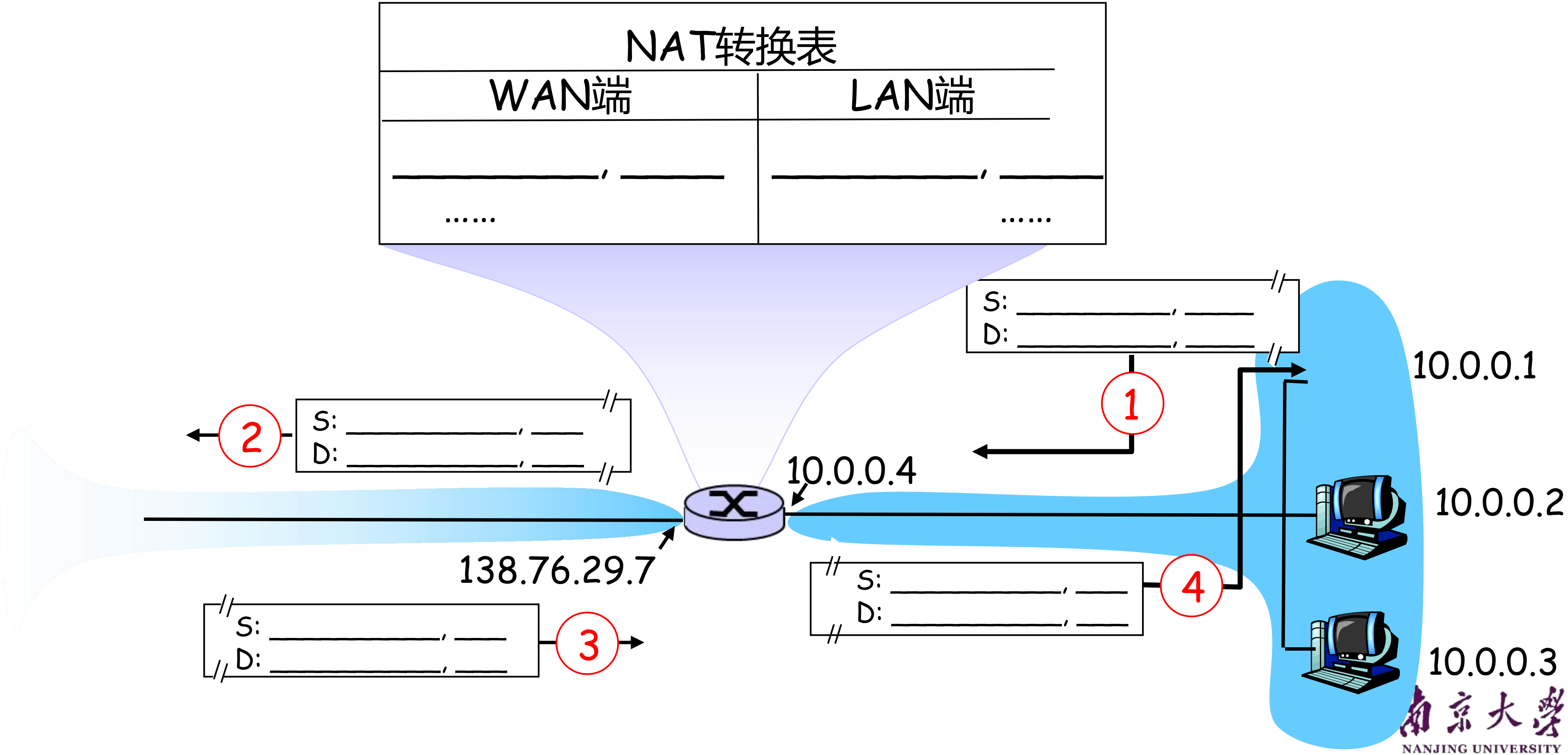
1
S: 10.0.0.1, 3345
D: 128.119.40.186, 80

2
S: 138.76.29.7, 5001
D: 128.119.40.186, 80

3
S: 128.119.40.186, 80
D: 138.76.29.7, 5001

4
S: 128.119.40.186, 80
D: 10.0.0.1, 3345

Network Address Translation

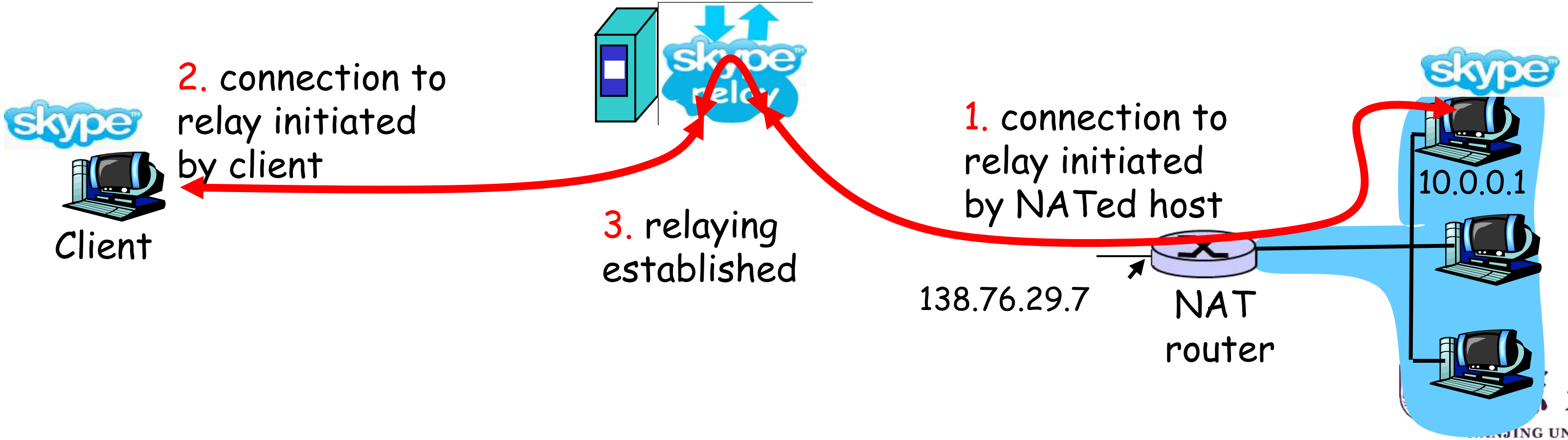


NAT转换表

WAN端	LAN端
____________, _____	____________, _____
……	……

NAT is Controversial

- Addresses changes from time to time
  - E.g. must be taken into account by P2P applications
- Relaying in Skype
  - NATed supernodes establishes connection to relay
  - External client connects to relay
  - Relay bridges packets between 2 connections



skype
relay
skype
Client
2. connection to relay initiated by client
3. relaying established
1. connection to relay initiated by NATed host
138.76.29.7
NAT router
skype
10.0.0.1

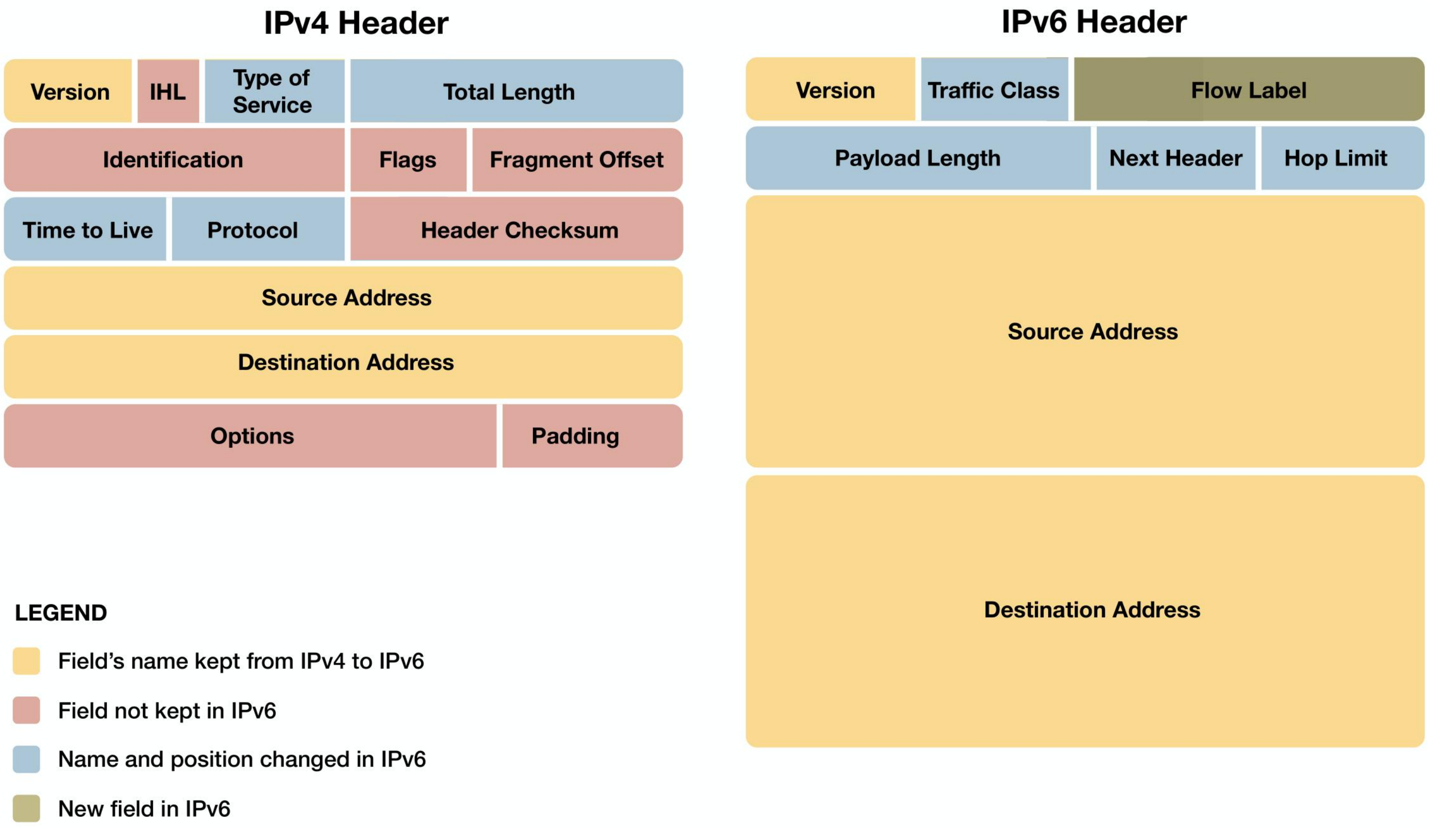
Outline

- IP Addressing
- Network Address Translation
- IPv6
- Generalized Forwarding and SDN
- Middleboxes

IPv6

- Initial motivation: address space exhaustion
  - Rapid growth of networks and the Internet
  - 32-bit address space (esp. net address) soon to be completely allocated
- Additional motivation
  - New header format helps speed processing and forwarding
  - Header changes to facilitate QOS
  - No fragmentation at router
  - New address mode: route to “best” of several replicated servers

IPv6 Header VS IPv4 Header



IPv4 Header

Version	IHL	Type of Service	Total Length
Identification		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source Address			
Destination Address			
Options			Padding

IPv6 Header

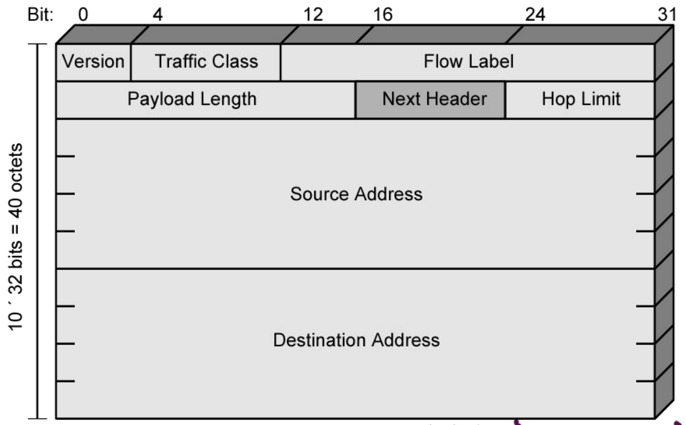
Version	Traffic Class	Flow Label
Payload Length	Next Header	Hop Limit
Source Address		
Destination Address		

LEGEND

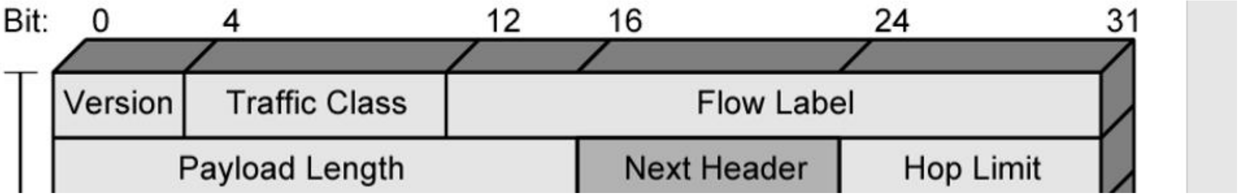
- 🟨 Field's name kept from IPv4 to IPv6
- 🟥 Field not kept in IPv6
- 🟦 Name and position changed in IPv6
- 🟫 New field in IPv6

IPv6 Header Fields

- Version (4 bits): 6
- Traffic Class (8 bits)
  - Classes or priorities of packet, identify QoS
- Flow Label (20 bits)
  - Identify datagrams in the same "flow"
- Payload length (16 bits)
  - Includes all extension headers plus user data
- Next Header (8 bits)
  - Identifies type of the next header
  - Extension or next layer up
- Source / Destination Address (128 bits)



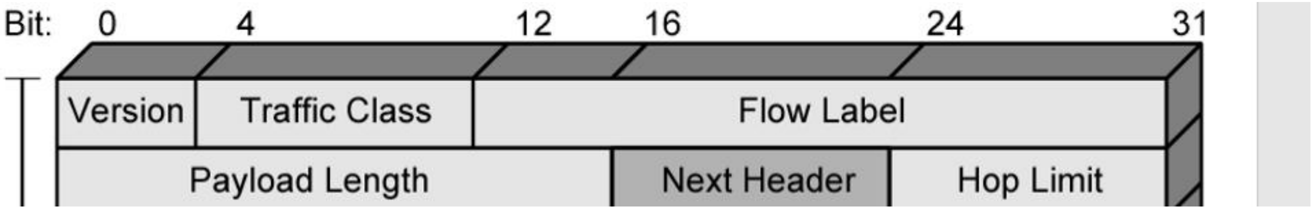
Traffic Class



- The 8-bit field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets.
  - E.g., used as the codepoint in DiffServ

- General requirements
  - Service interface must provide means for upper-layer protocol to supply the value of traffic class
  - Value of traffic class can be changed by source, forwarder, receiver
  - An upper-layer protocol should not assume the value of traffic class in a packet has not been changed.

IPv6 Flow



- A sequence of packets sent from a particular source to a particular destination
- From hosts point of view
  - Generated from one application and have the same transfer service requirements
  - May comprise a single or multiple TCP connections
  - One application may generate a single flow or multiple flows
- From routers point of view
  - Share attributes that affect how these packets are handled by the router
  - e.g. routing, resource allocation, discard requirements, accounting, and security

Flow Label

- A flow is uniquely identified by the combination of
  - Source and destination address
  - A non-zero 20-bit Flow Label
- Flow requirements are defined prior to flow commencement
  - Then a unique Flow Label is assigned to the flow
- Router decide how to route and process the packet by
  - Simply looking up the Flow Label in a table and without examining the rest of the header

Advantages of IPv6 over IPv4

- Expanded addressing capabilities
  - 128 bit
  - Scalability of multicast addresses
  - Anycast – delivered to one of a set of nodes
  - Address auto-configuration

- Improved option mechanism
  - Separate optional headers between IPv6 header and transport layer header
  - Most are not examined by intermediate routers
  - Easier to extend options
  - Checksum removed to further reduce processing time at each router

Advantages of IPv6 over IPv4

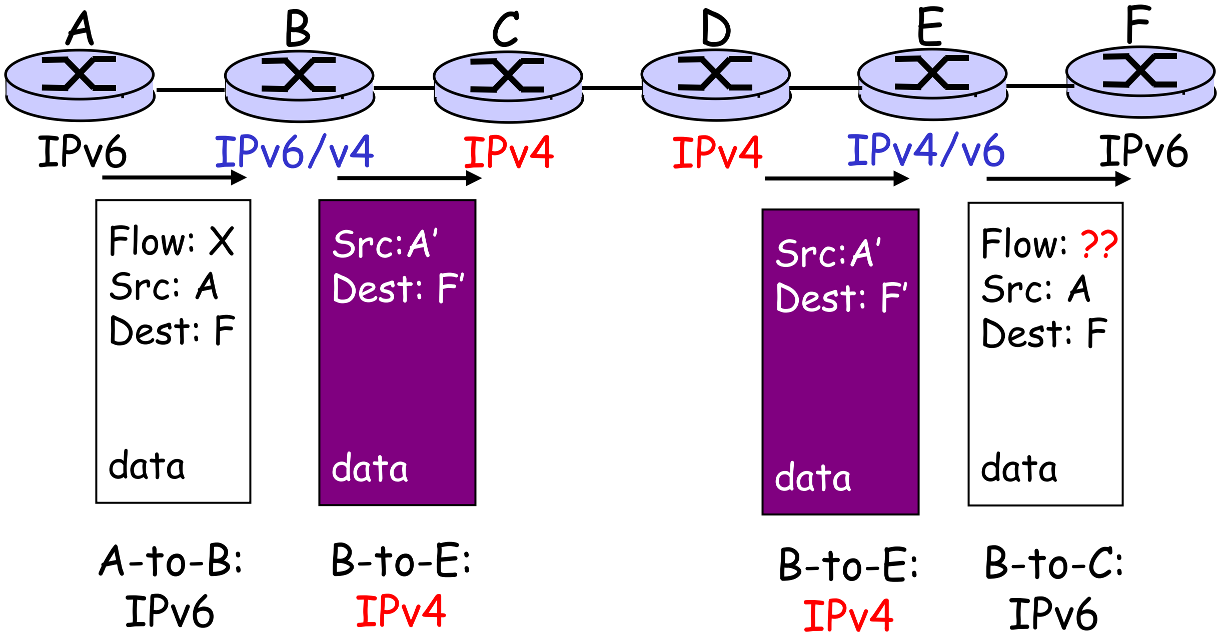
- Support for resource allocation
  - Uses traffic class
  - Grouping packets to particular traffic flow
  - Allows QoS handling other than best-effort, e.g. real-time video
- More efficient and robust mobility mechanism
- More security: Built-in, strong IP-layer encryption and authentication

Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneously
  - How will the network operate with mixed IPv4 and IPv6 routers

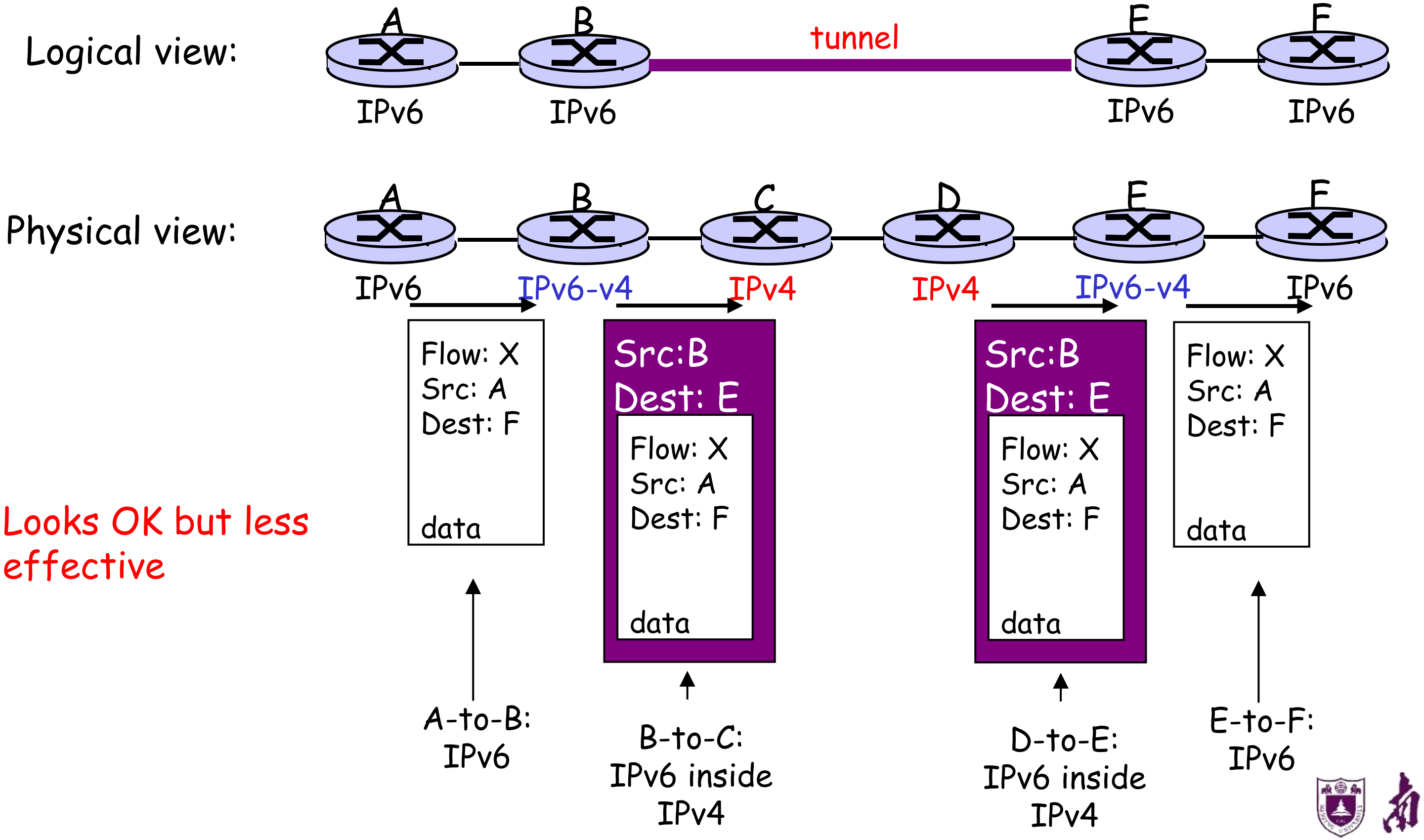
- Two proposed approaches
  - Dual Stack – some routers with dual stack (IPv6, IPv4) can translate between formats
  - Tunneling – IPv6 carried as payload in IPv4 datagram among IPv4 routers

Dual Stack Approach



- Address translation between IPv4 and IPv6 is needed
- Some IPv6 features is lost

Tunneling



Looks OK but less effective

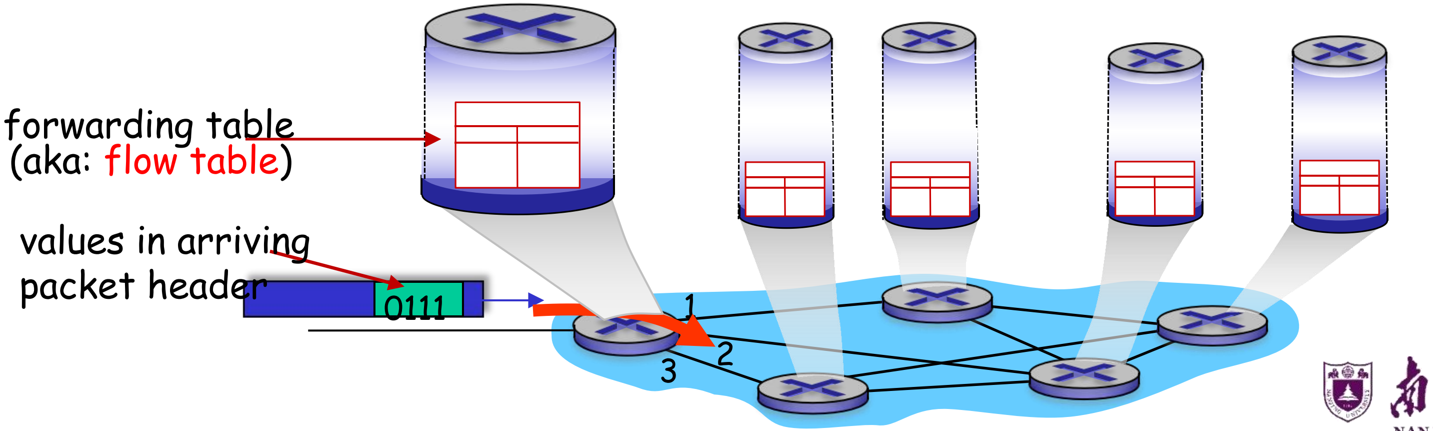
Outline

- IP Addressing
- Network Address Translation
- IPv6
- Generalized Forwarding and SDN
- Middleboxes

Generalized forwarding: match plus action

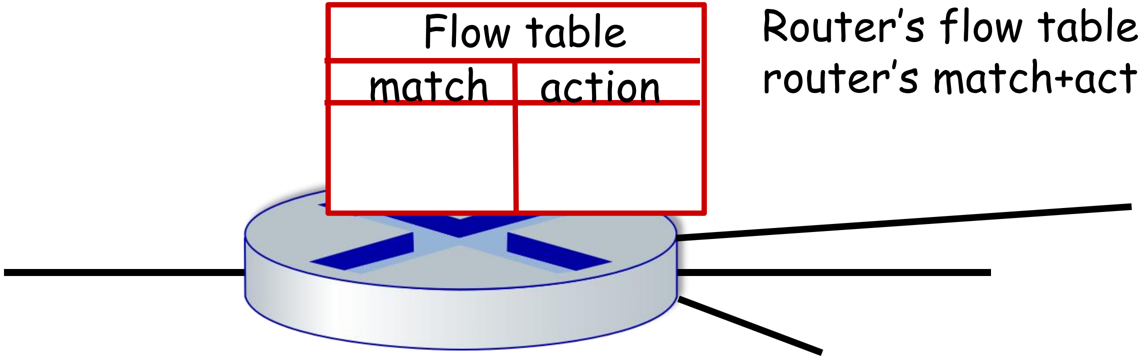
Review: each router contains a forwarding table (aka: flow table)

- "match plus action" abstraction: match bits in arriving packet, take action
- destination-based forwarding: forward based on dest. IP address
- generalized forwarding:
  - many header fields can determine action
  - many action possible: drop/copy/modify/log packet



Flow table abstraction

- flow: defined by header field values (in link-, network-, transport-layer fields)
- generalized forwarding: simple packet-handling rules
  - match: pattern values in packet header fields
  - actions: for matched packet: drop, forward, modify, matched packet or send matched packet to controller
  - priority: disambiguate overlapping patterns
  - counters: #bytes and #packets



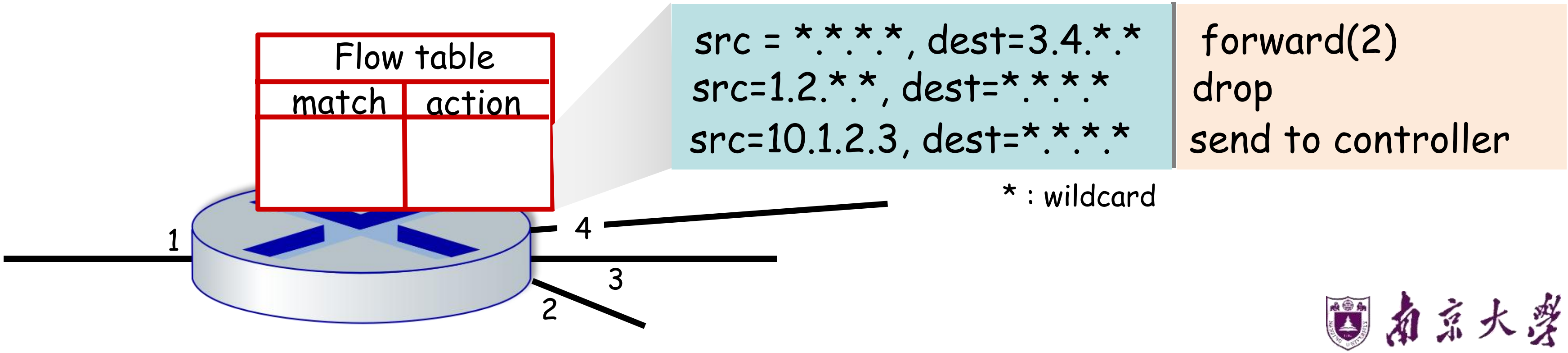
Flow table

match	action

Router’s flow table define router’s match+action rules

Flow table abstraction

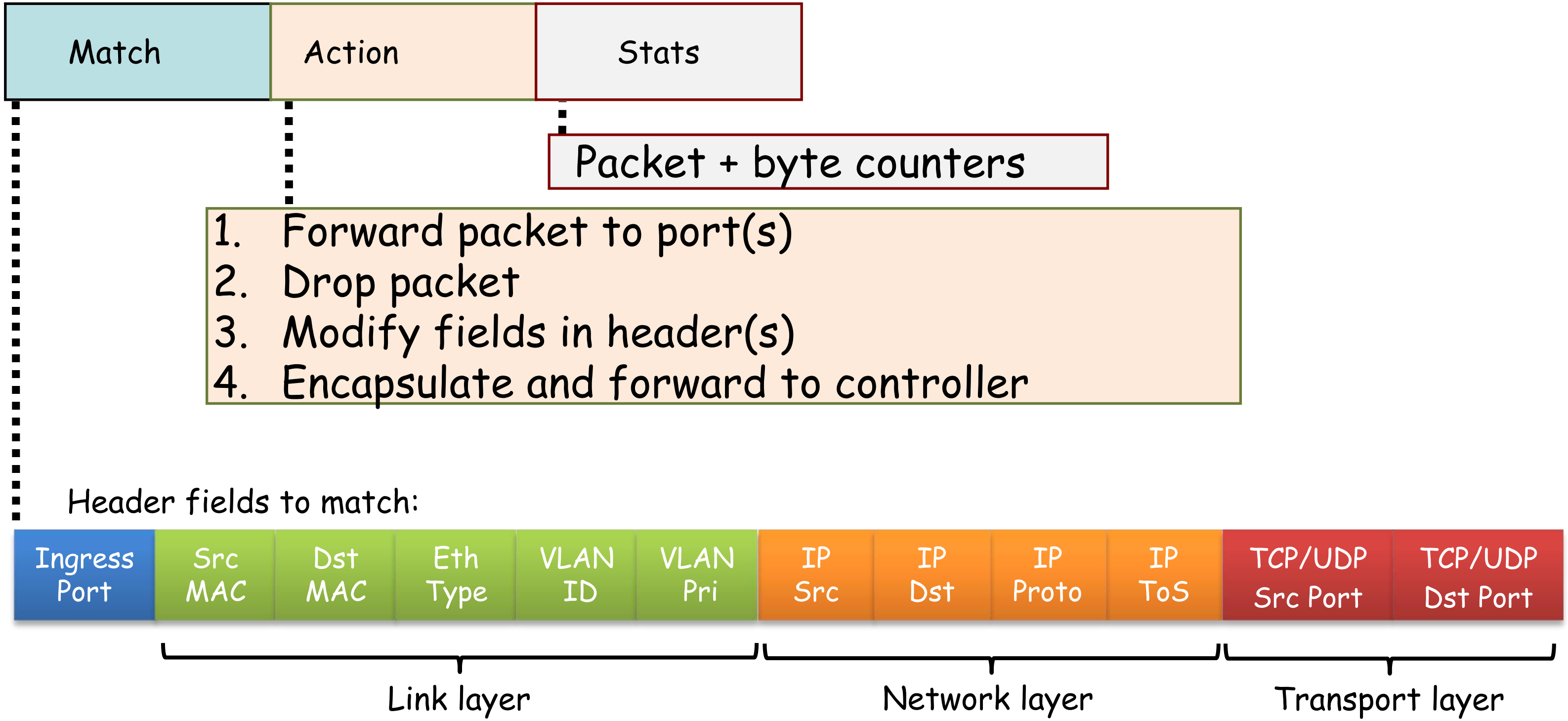
- flow: defined by header fields
- generalized forwarding: simple packet-handling rules
  - match: pattern values in packet header fields
  - actions: for matched packet: drop, forward, modify, matched packet or send matched packet to controller
  - priority: disambiguate overlapping patterns
  - counters: #bytes and #packets



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

* : wildcard

OpenFlow: flow table entries



Match	Action	Stats

Packet + byte counters

1. Forward packet to port(s)
2. Drop packet
3. Modify fields in header(s)
4. Encapsulate and forward to controller

Header fields to match:

Ingress Port	Src MAC	Dst MAC	Eth Type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Proto	IP ToS	TCP/UDP Src Port	TCP/UDP Dst Port
	Link layer					Network layer				Transport layer	

OpenFlow abstraction

- match+action: abstraction unifies different kinds of devices

Router

- match: longest destination IP prefix
- action: forward out a link

Switch

- match: destination MAC address
- action: forward or flood

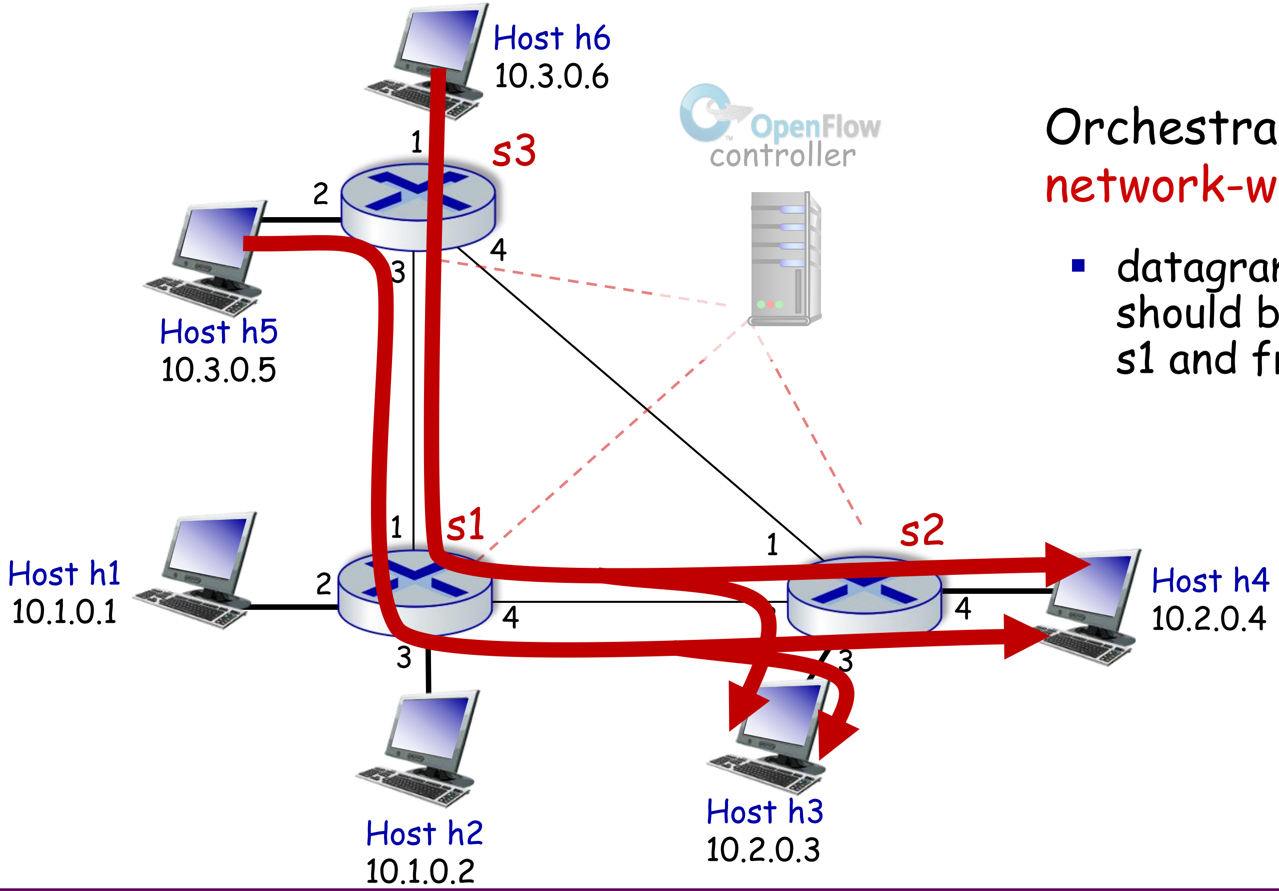
Firewall

- match: IP addresses and TCP/UDP port numbers
- action: permit or deny

NAT

- match: IP address and port
- action: rewrite address and port

OpenFlow example



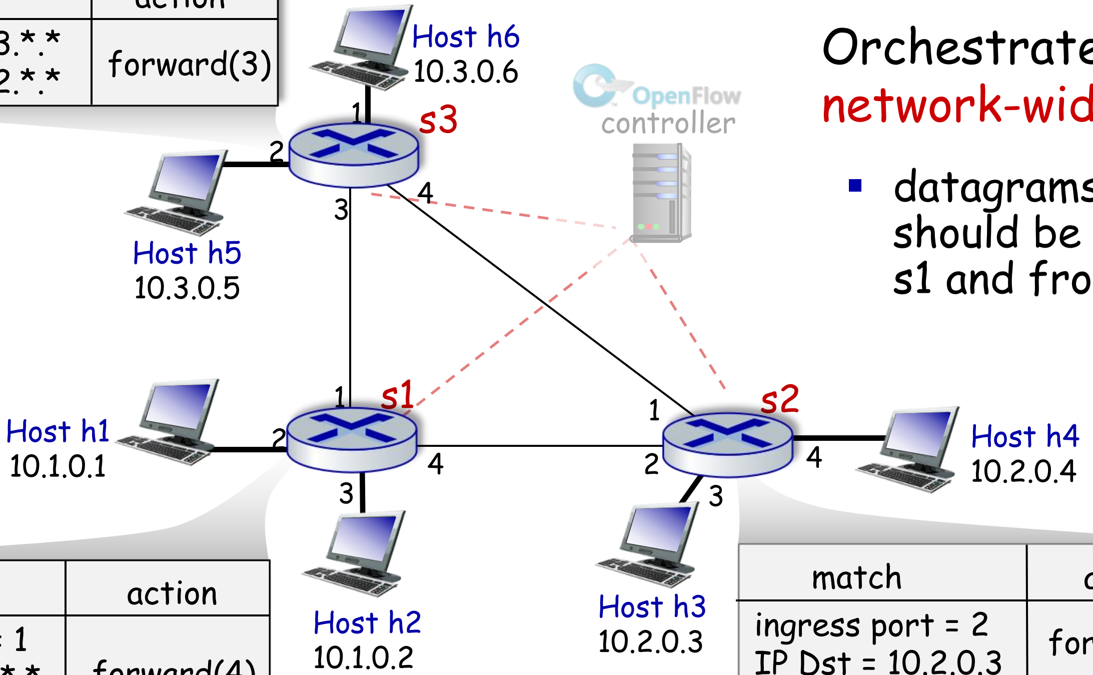
Orchestrated tables can create network-wide behavior, e.g.,:

- datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

OpenFlow example

Orchestrated tables can create network-wide behavior, e.g.,:

- datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2



match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)

match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)

Outline

- IP Addressing
- Network Address Translation
- IPv6
- Generalized Forwarding and SDN
- Middleboxes

Middleboxes

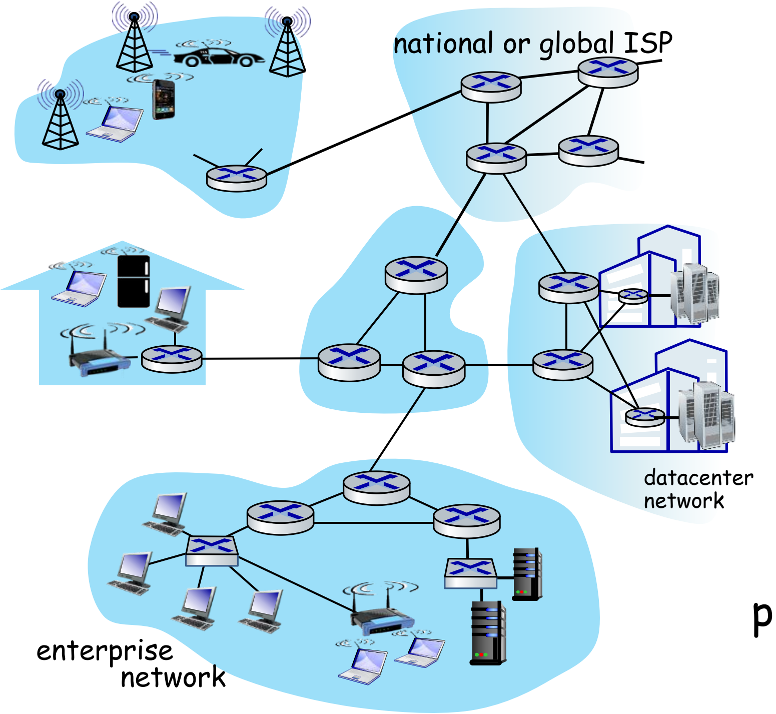
Middlebox (RFC 3234)

"any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host"

Middleboxes everywhere!

NAT: home, cellular, institutional

Application-specific: service providers, institutional, CDN



Firewalls, IDS: corporate, institutional, service providers, ISPs

Load balancers: corporate, service provider, data center, mobile nets

Caches: service provider, mobile, CDNs

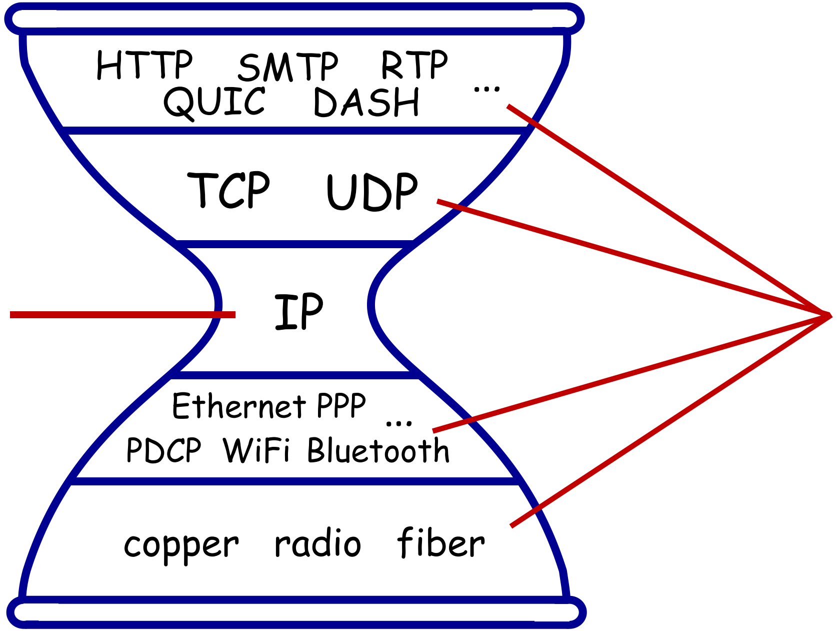
Middleboxes

- initially: proprietary (closed) hardware solutions
- move towards "whitebox" hardware implementing open API
  - move away from proprietary hardware solutions
  - programmable local actions via match+action
  - move towards innovation/differentiation in software
- SDN: (logically) centralized control and configuration management often in private/public cloud
- network functions virtualization (NFV): programmable services over white box networking, computation, storage

The IP hourglass

Internet's "thin waist":

- one network layer protocol: IP
- must be implemented by every (billions) of Internet-connected devices

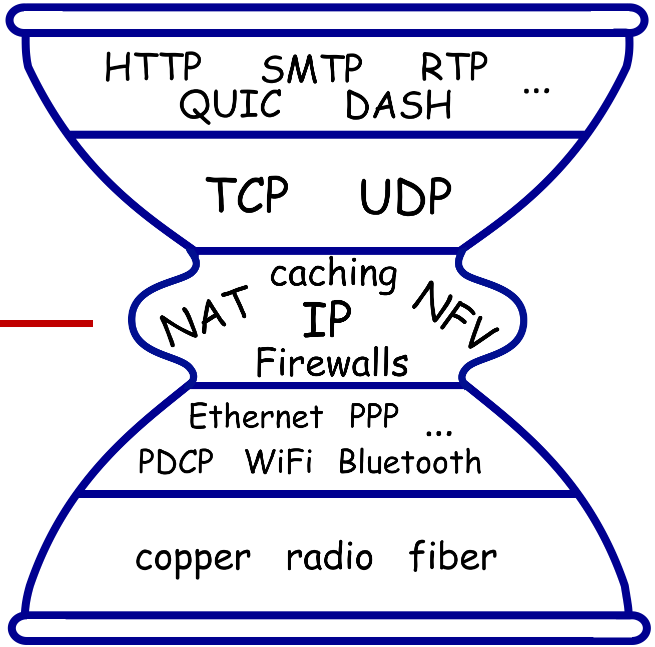


many protocols in physical, link, transport, and application layers

The IP hourglass, at middle age

Internet's middle age "love handles"?

- middleboxes, operating inside the network



提问

Q & A