Chapter 4: outline

4.1 Overview of Network layer
- data plane
- control plane

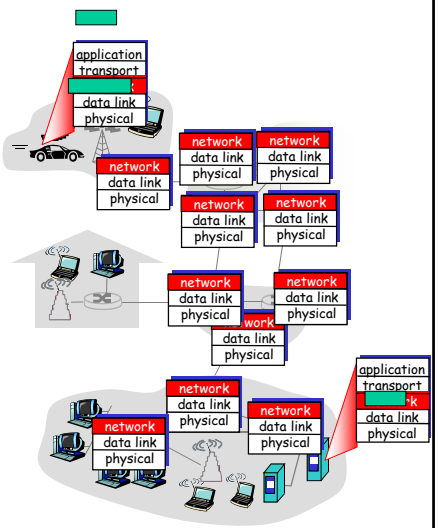
4.2 What's inside a router

4.3 IP: Internet Protocol
- IPv4 datagram format
- fragmentation
- v4 addressing
- network address translation
- IPv6

4.4 Generalized Forwarding and SDN
- match
- action
- OpenFlow examples of match-plus-action

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 about 10 years ago to provide virtual circuits supported by IP routers *(typically within the same AS); it borrows the idea of "labels" from ATM and frame relay*
- Today, such virtual circuits may serve as virtual links in Internet

Network layer service models:

Network Architecture	Service Model	Guarantees?				Congestion feedback
		Bandwidth	loss	Order	Timing	
Internet	best effort	none	no	no	no	no (TCP infers via loss)
ATM	CBR	constant rate	*yes*	*yes*	yes	no congestion
ATM	VBR	guaranteed rate	*yes*	*yes*	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	*UBR*	*none*	*no*	*yes*	*no*	*no*

Origins of datagram and VC

Internet (datagram)

- data exchange between computers
  - "elastic" service, no strict timing requirement
- many link types
  - different characteristics
  - uniform service difficult
- "smart" end systems (computers)
  - can adapt, perform control, error recovery
  - ***simplicity*** inside network, *complexity* at "edge"

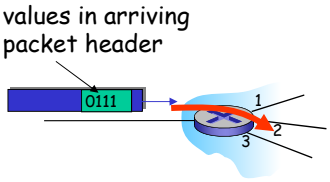
ATM (VC)

- evolved from telephony
- human conversation:
  - strict timing, loss requirements
  - need for guaranteed services
- "dumb" end systems
  - telephones
  - complexity inside network

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

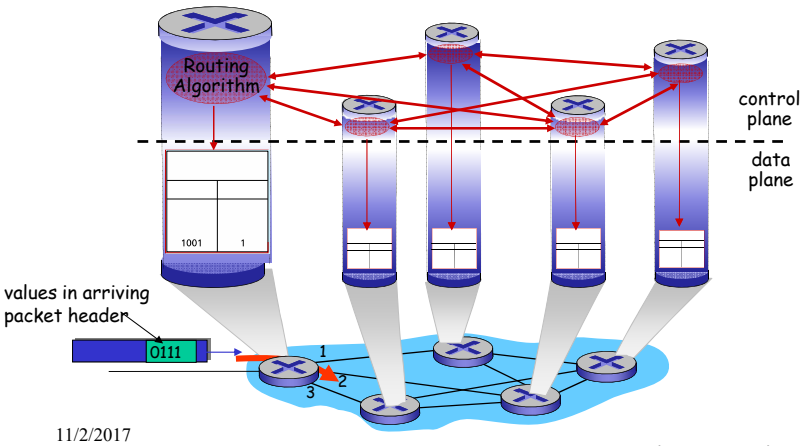


Control plane

- 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)

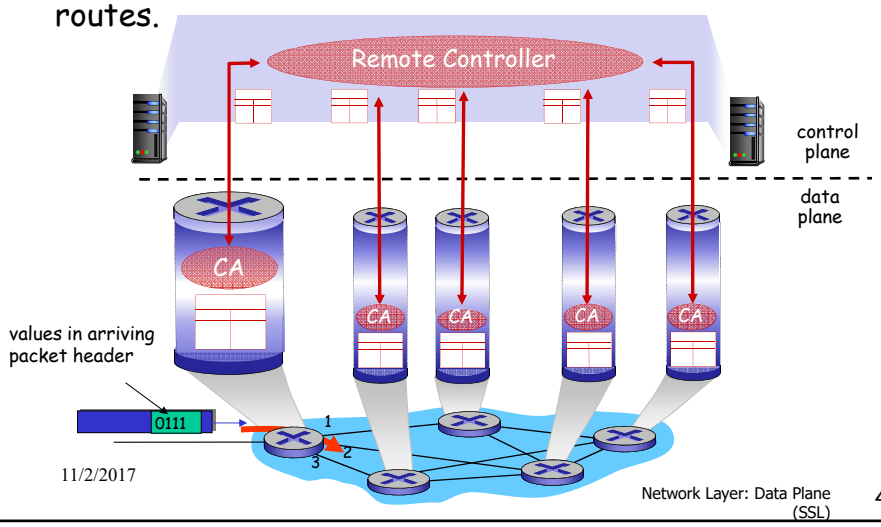
Per-router control plane (more in Chapter 5)

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



Logically centralized control plane
(more in Chapter 5)

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



Chapter 4: outline

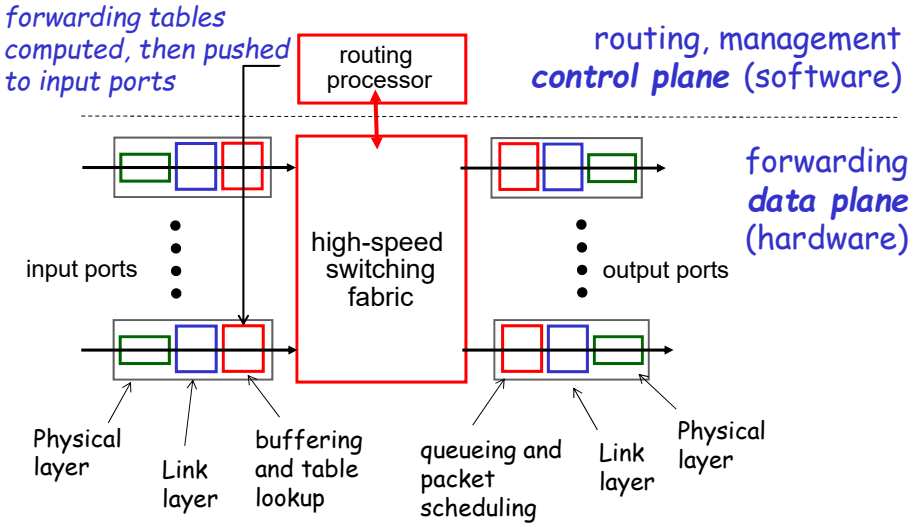
4.1 Overview of Network layer
- data plane
- control plane

4.2 What's inside a router

4.3 IP: Internet Protocol
- datagram format
- fragmentation
- v4 addressing
- network address translation
- IPv6

4.4 Generalized Forwarding and SDN
- match
- action
- OpenFlow examples of match-plus-action

Router architecture overview

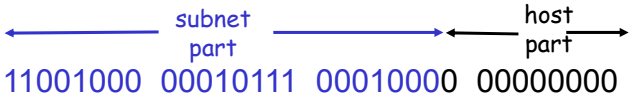


IPv4 addressing: CIDR

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

CIDR: Classless InterDomain Routing

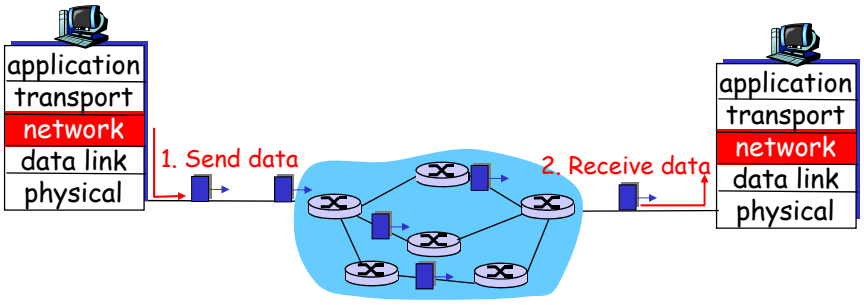
- subnet portion of address of **variable length**
- address format: a.b.c.d/x, where x is # bits in subnet portion of address



200.23.16.0/23

Datagram networks

- IPv4
- 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



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?

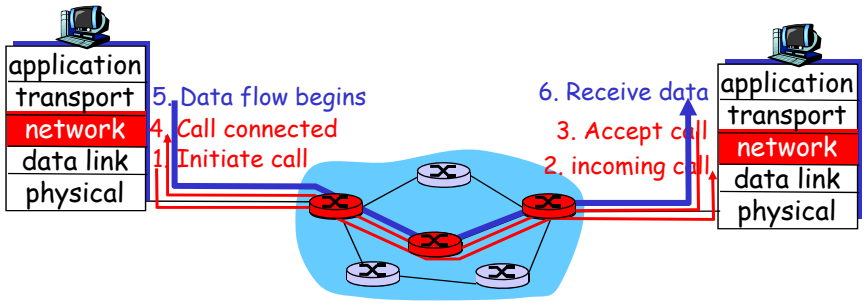
DA: 11001000 00010111 00011000 10101010 Which interface?

A forwarding table in an Internet core router has more than 400,000 IP prefixes *(from 2014 data)*

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

Virtual circuits: signaling protocols

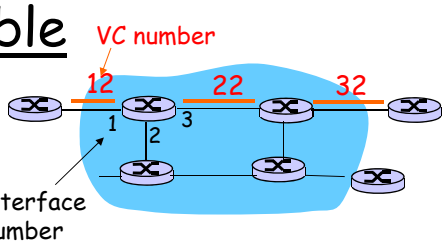
- used to set up, maintain, tear down VC
- not used in Internet's network layer, but may be used underneath the IP layer to provide a virtual link (e.g., MPLS tunnel)



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 $2\frac{1}{2}$)
- *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

Chapter 4: outline

4.1 Overview of Network layer
- data plane
- control plane

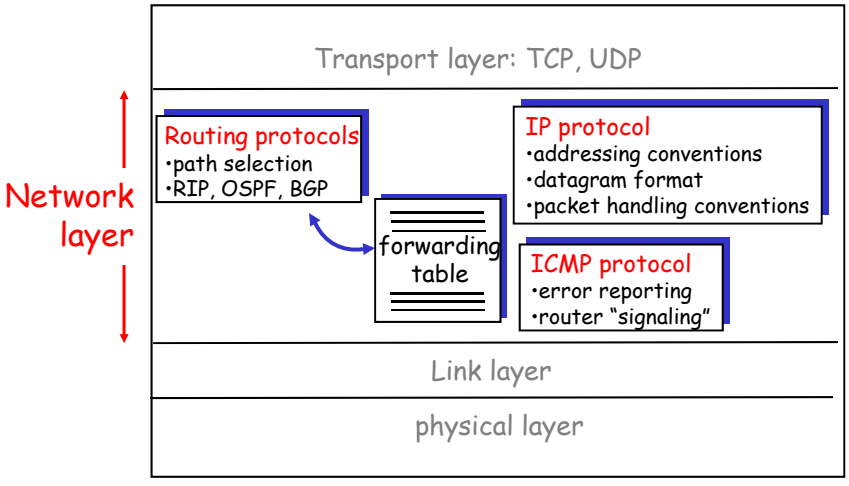
4.2 What's inside a router

4.3 IP: Internet Protocol
- datagram format
- fragmentation
- v4 addressing
- network address translation
- IPv6

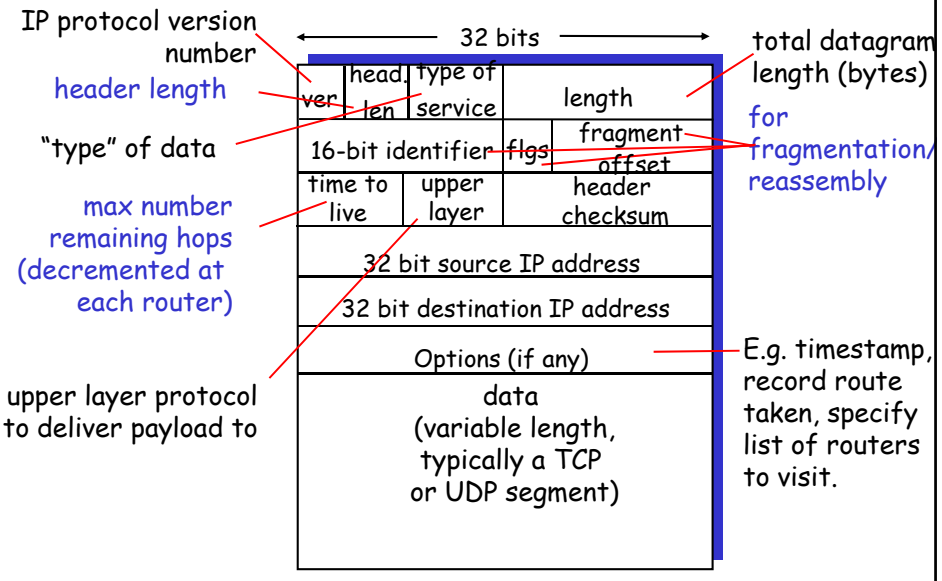
4.4 Generalized Forwarding and SDN
- match
- action
- OpenFlow examples of match-plus-action

The Internet Network layer

Host, router network layer functions:

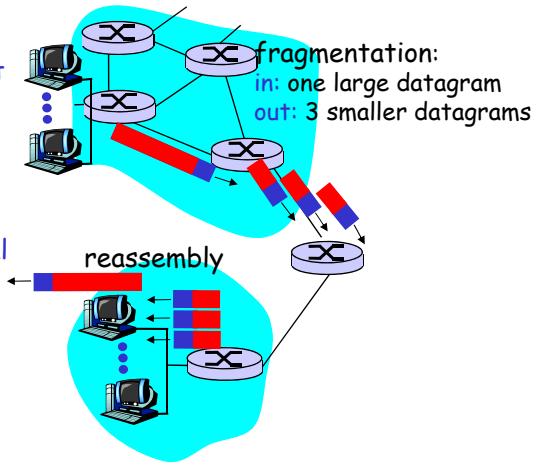


IP datagram format

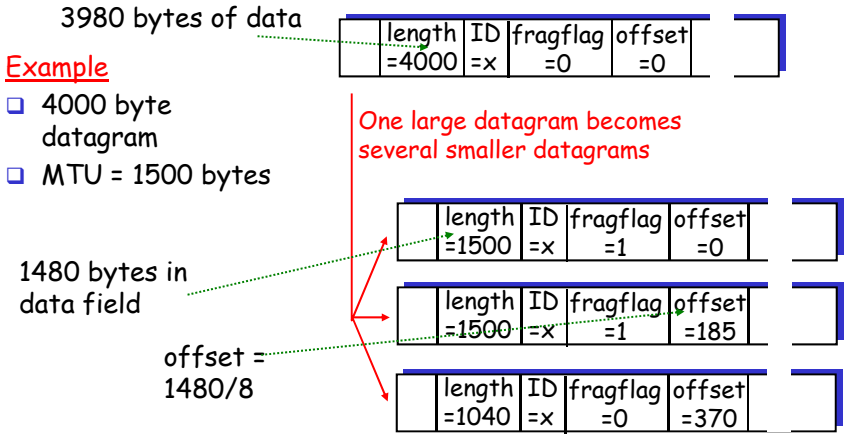


IP Fragmentation & Reassembly

- MTU (max.transfer size)
  - different link types, different MTUs
  - Support MTU of at least 576 bytes
- too large IP datagram "fragmented" within net
  - reassembled only at final destination host
  - IP header bits used to identify, order related fragments



IP Fragmentation and Reassembly



Chapter 4: outline

4.1 Overview of Network layer
- data plane
- control plane

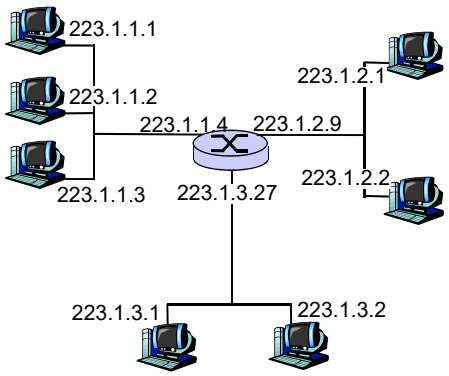
4.2 What's inside a router

4.3 IP: Internet Protocol
- datagram format
- fragmentation
- v4 addressing
- network address translation
- IPv6

4.4 Generalized Forwarding and SDN
- match
- action
- OpenFlow examples of match-plus-action

IP Addressing: introduction

- IP address: 32-bit identifier for an *interface*
- *interface:* connection between host/router and physical link (*wired or wireless*)
  - a router typically has *multiple* interfaces
  - a host typically has one interface



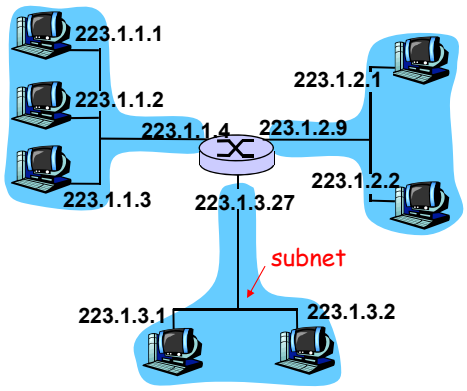
Dotted decimal notation

223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Subnets

- IP address:
  - **subnet** part (high order bits)
  - **host** part (low order bits)
- *What's a subnet ?*
  - device interfaces with same subnet part of IP address
  - **can physically reach each other without a router**



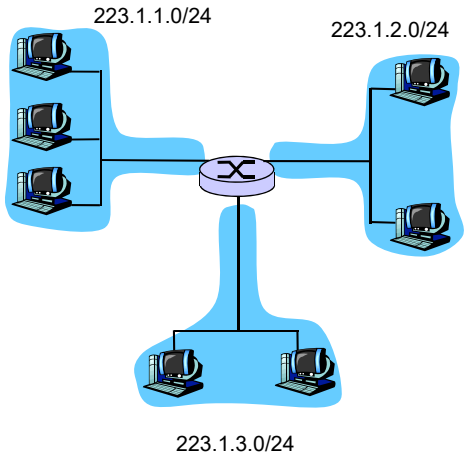
network consisting of 3 subnets

A layer-2 switch is considered part of a link

Subnets

Recipe

- To determine the subnets, detach each interface from its host or router, creating islands of isolated networks. Each isolated network is a subnet.

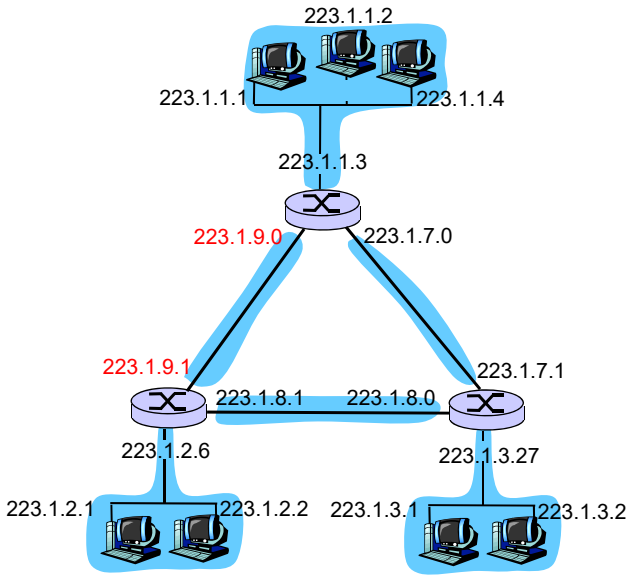


Subnet mask: /24

Note: There are also virtual LANs (VLANs) – see Chapter 6

Subnets

How many?

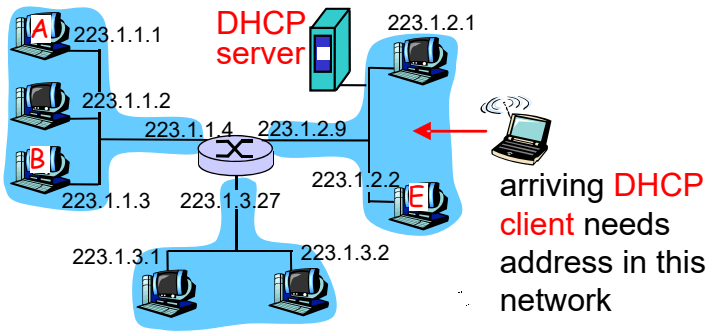


IP addresses: how to get one?

Q: How does *host* get IP address?

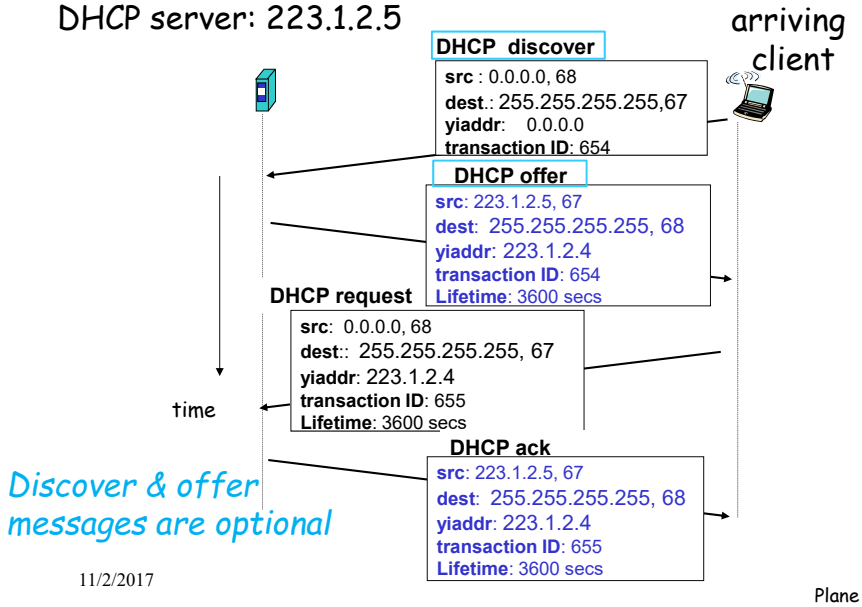
- hard-coded by system admin in a file
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from a server
  - "plug-and-play"

DHCP client-server scenario



A router may act as a relay agent

DHCP client-server scenario



DHCP: more than IP address

DHCP can return more than just an allocated IP address on subnet:

- address of first-hop router for client
- name and IP address of DNS server
- network mask (indicating subnet portion of address)

IP addresses: how to get them?

- ICANN (Internet Corporation for Assigned Names and Numbers)/IANA (Internet Assigned Numbers Authority)
  - allocates IP address blocks (IPv4 address exhaustion on 1/31/2011, IPv6)
  - oversees DNS
    - root name servers, top level domain name servers (domain name *registries*)
    - domain name *registrars*, resolves disputes
- Regional, national, and local Internet registries, and ISPs
  - End-user organization can be assigned IP address space from one of the above

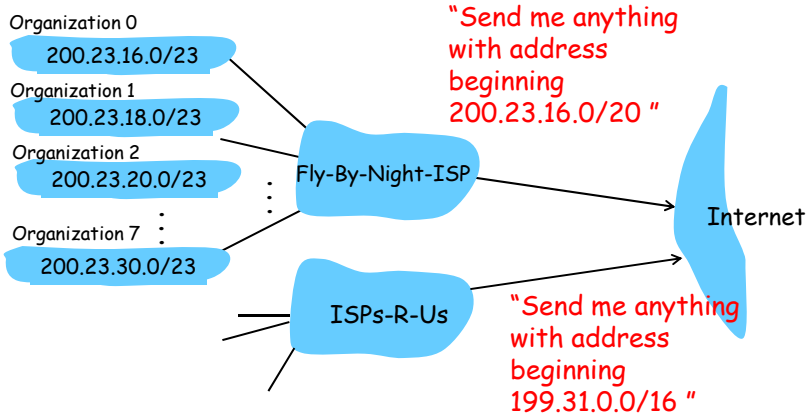
IP address prefix: how to get one?

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

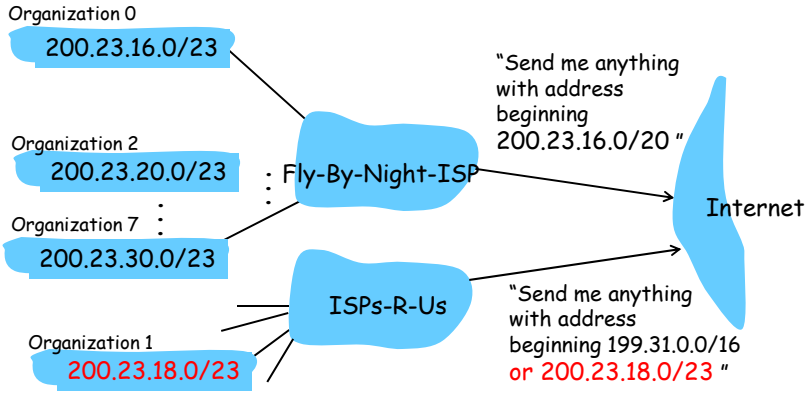
allows efficient advertisement of routing information



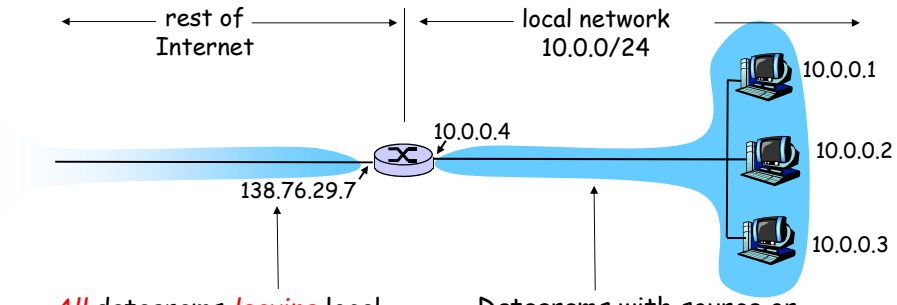
Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1

Hole(s) in a block of addresses <- reason for longest prefix match



NAT: Network Address Translation



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

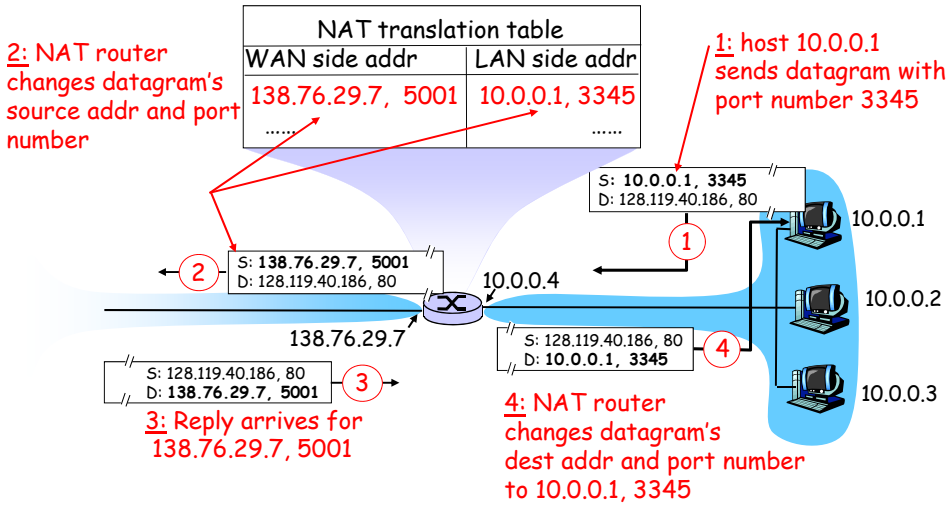
Datagrams with source or destination within network have 10.0.0/24 addresses for source, destination

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

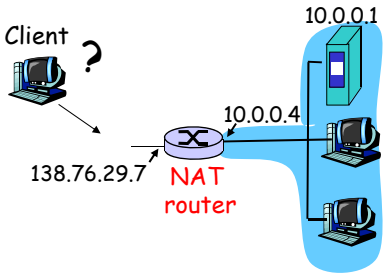


NAT: Network Address Translation

- 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., P2P, IPsec applications, etc.
  - address shortage should instead be solved by IPv6

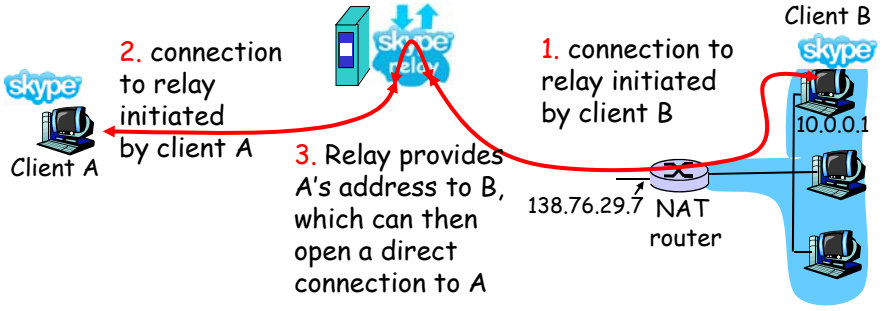
NAT traversal problem

- client wants to connect to server with address 10.0.0.1
  - only one externally visible IP address: 138.76.29.7
- configure NAT to forward incoming connection requests at given port to server
  - e.g., (123.76.29.7, port 2500) always forwarded to 10.0.0.1 port 2500



NAT traversal problem

- solution used in *original* Skype (*note:current Skype is no longer P2P*)
  - with help of a super peer (server) as relay



  - Both hosts may be behind NATs

Chapter 4: outline

4.1 Overview of Network layer
- data plane
- control plane

4.2 What's inside a router

4.3 IP: Internet Protocol
- datagram format
- fragmentation
- v4 addressing
- network address translation
- IPv6

4.4 Generalized Forwarding and SDN
- match
- action
- OpenFlow examples of match-plus-action

IPv6

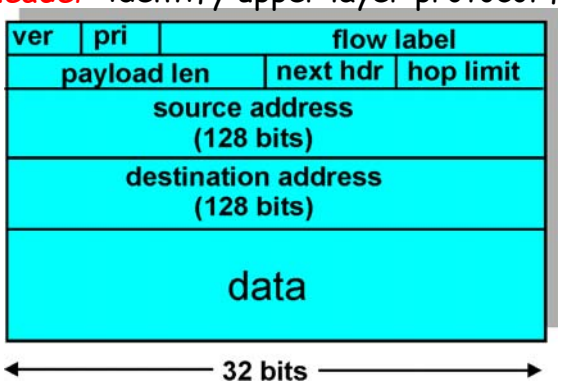
- Initial motivation: 32-bit address space soon to be completely allocated.
- Additional motivation:
  - simpler header format to speed up processing/forwarding
  - header change to facilitate QoS
- IPv6 datagram format:
  - fixed-length 40 byte header
  - no fragmentation allowed

IPv6 Header (Cont)

Priority (8 bits): identify priority of datagrams within flow or in different apps

Flow Label (20 bits): identify datagrams in same "flow." (concept of "flow" not defined).

Next header: identify upper layer protocol for data



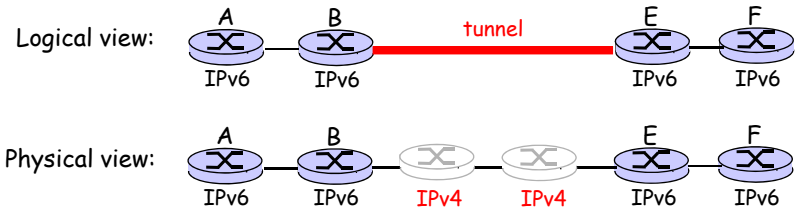
Other Changes from IPv4

- *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

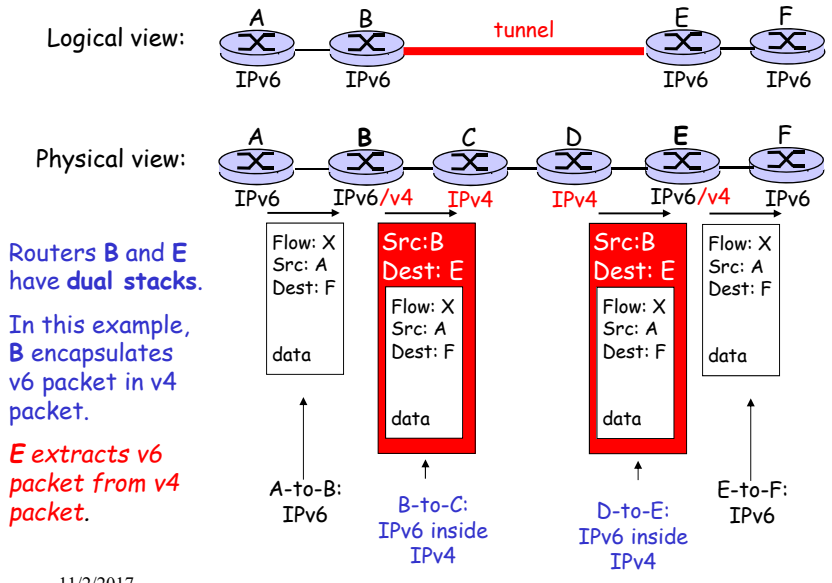
Transition From IPv4 To IPv6

- 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



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)

...

Chapter 4: outline

4.1 Overview of Network layer
- data plane
- control plane

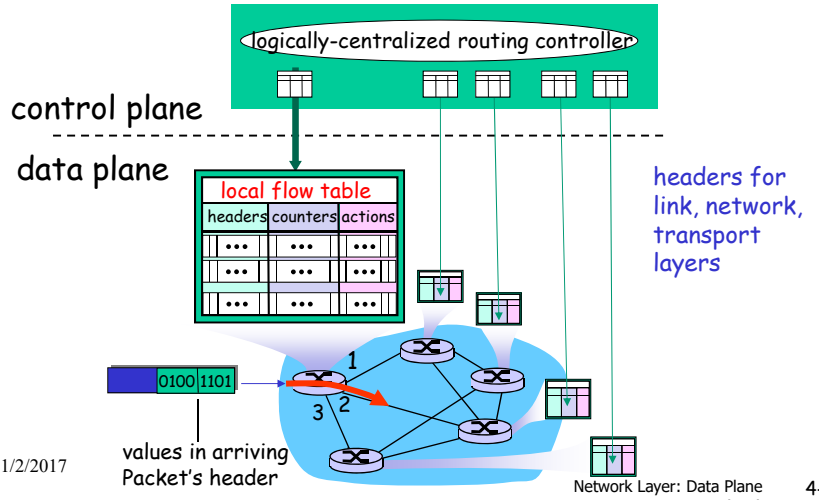
4.2 What's inside a router

4.3 IP: Internet Protocol
- datagram format
- fragmentation
- v4 addressing
- network address translation
- IPv6

4.4 Generalized Forwarding and SDN
- match
- action
- OpenFlow examples of match-plus-action

Generalized Forwarding and SDN

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



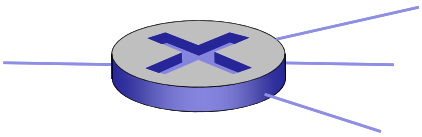
OpenFlow abstraction

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

- Router (layer3)
  - *match:* longest destination IP prefix
  - *action:* forward to one or more ports
- Switch (layer 2)
  - *match:* destination MAC 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 - with wildcards)
- generalized forwarding
  - *Flow entry:* match fields, priority, counters
  - *Actions: for matched packet* - forward (to 1 or more ports), drop, modify the packet, or send it to controller

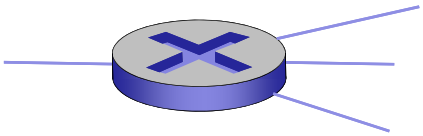


Flow table in a router/packet switch (computed and distributed by controller) defines router's match+action rules

OpenFlow data plane abstraction

- *flow*: defined by header fields
- generalized forwarding
  - *Flow entry*: match fields, priority, counters
  - *Actions*: *for matched packet* – forward (to 1 or more ports), drop, modify the packet, or send it to controller

Flow entry examples:



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

* : wildcard

Examples

Destination-based layer-3 forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	S port	D port	Action
*	*	*	*	*	*	51.6.0.*	*	*	*	port6

forward IP datagrams with destination IP address 51.6.0. to router output port 6*

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	S port	D port	Action
*	*	*	*	*	*	*	17	*	*	drop

drop all IP datagrams containing UDP segments

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	S port	D port	Action
*	*	*	*	*	128.119.1.*	*	*	*	35601	drop

drop all IP datagrams that match src 128.119.1. and D port 35601*

Examples (cont.)

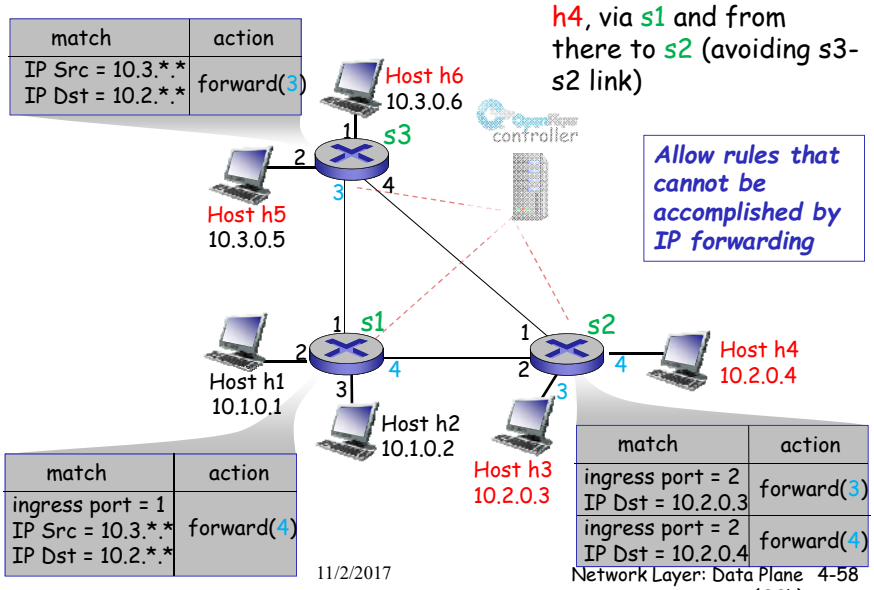
Destination-based layer 2 (switch) forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	S port	D port	Action
*	*	22:A7:23:11:E1:02	*	*	*	*	*	*	*	port3

forward layer 2 frames destined for MAC address 22:A7:23:11:E1:02 to output port 3

OpenFlow example

Example: datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2 (avoiding s3-s2 link)



Chapter 4

4.1 Overview of Network layer: data plane and control plane

4.2 What's inside a router

4.3 IP: Internet Protocol
- datagram format
- fragmentation
- v4 addressing
- NAT
- IPv6

4.4 Generalized Forward and SDN
- match plus action
- OpenFlow examples

Question: how are forwarding tables (destination-based forwarding) or flow tables (generalized forwarding) computed?

Answer: by the control plane (next chapter)

End of Chapter 4