Chapter 4
Network Layer

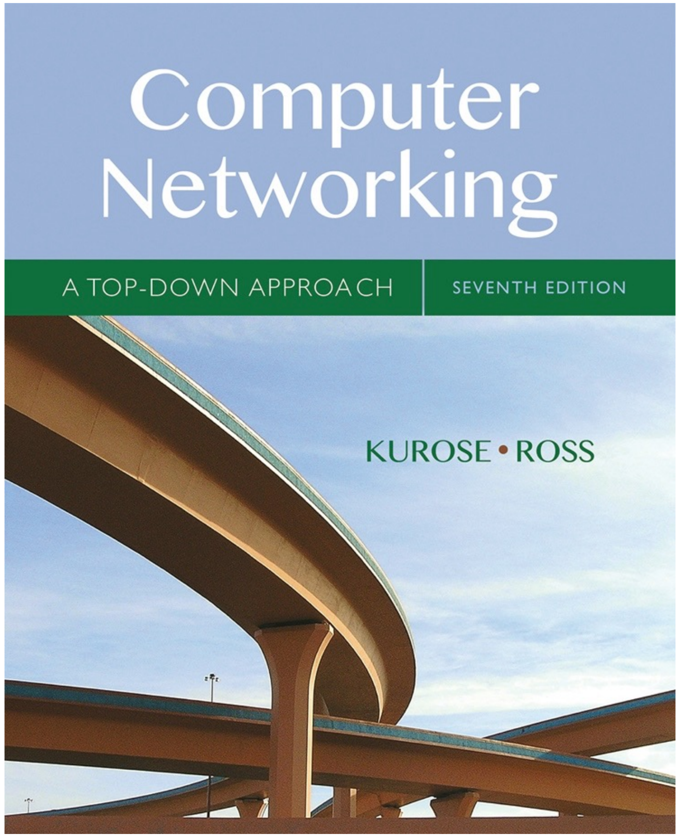
A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR, Edited 2016, UMinho.

Nota: Conteúdo atualizado, Uminho, PMC, 2022.



Computer Networking: A Top Down Approach
7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol
- intro, ICMP
- datagram format
- IPv4 addressing
- routing
- NAT
- IPv6

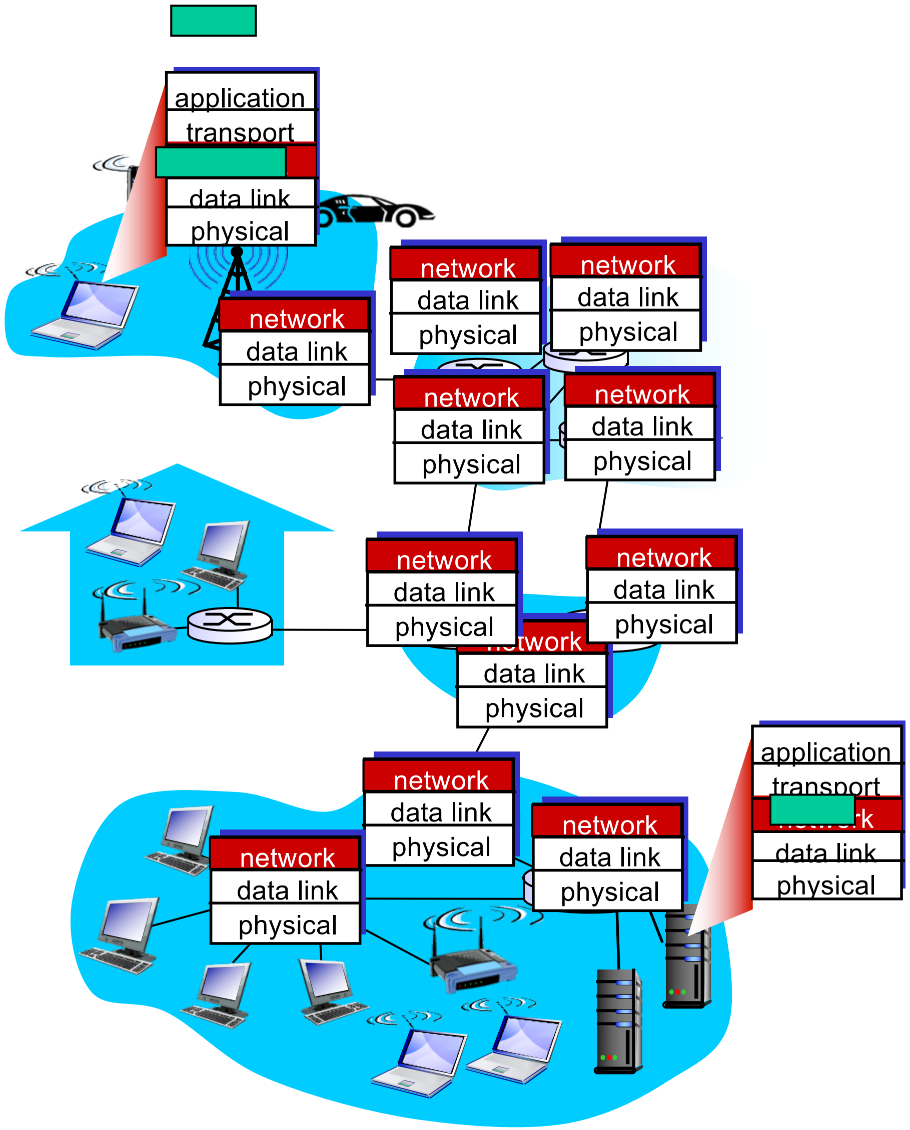
4.5 routing algorithms
- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet
- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Network layer

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- router examines header fields in all IP datagrams passing through it



Two key network-layer functions

- *forwarding:* move packets from router's input to appropriate router output

- *routing:* determine route taken by packets from source to dest.
  - *routing algorithms*

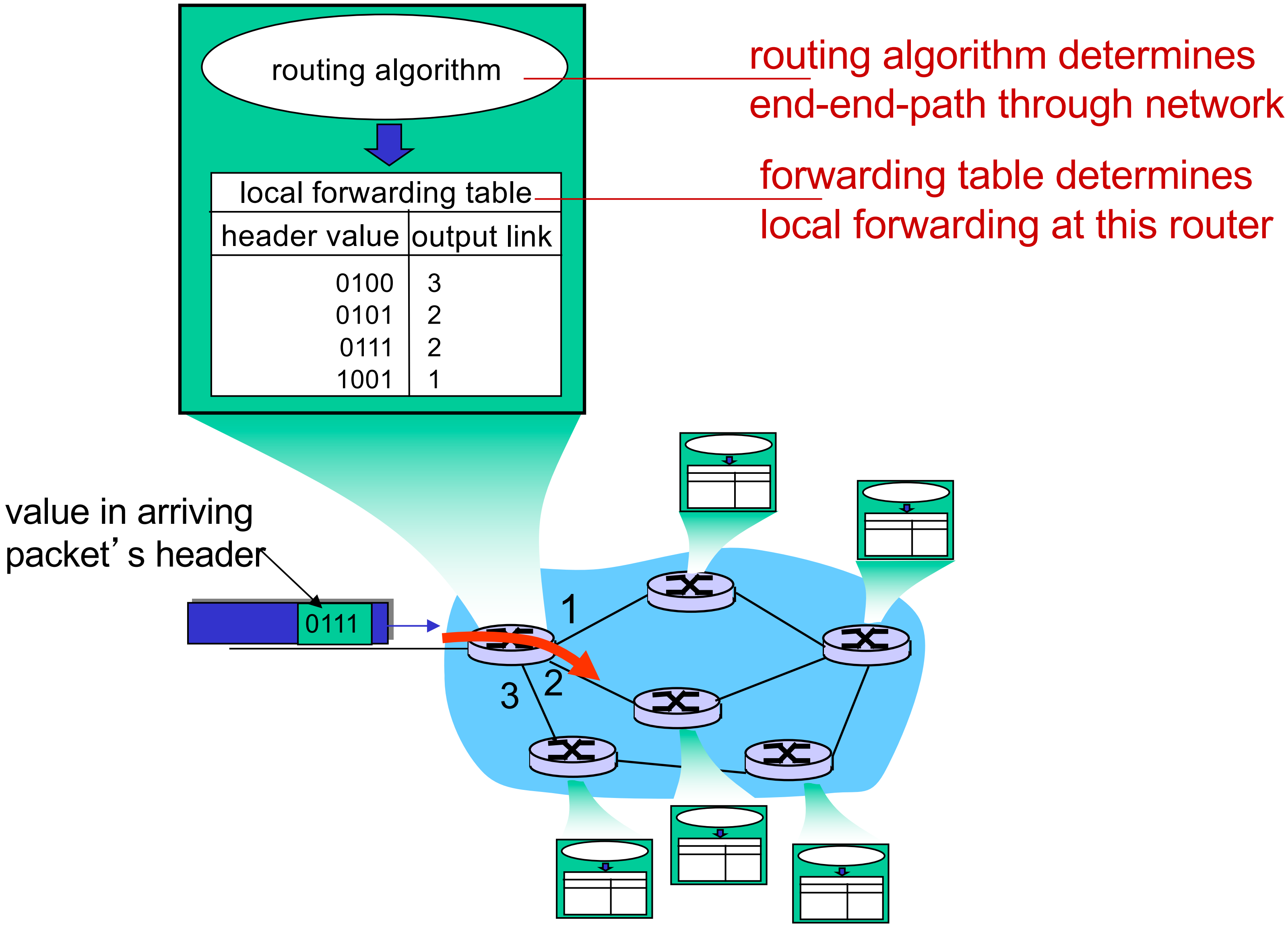
analogy:

- *forwarding:* process of getting through single interchange

- *routing:* process of planning trip from source to dest

- *other important functions:* L2 independent PDU, fragmentation, universal addressing.

Interplay between routing and forwarding



Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol
- intro, ICMP
- datagram format
- IPv4 addressing
- routing
- NAT
- IPv6

4.5 routing algorithms
- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet
- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

Connection / Connectionless Network Service

- *datagram* network provides network-layer *connectionless* service
- *virtual-circuit* network provides network-layer *connection* service
- analogous to TCP/UDP connecton-oriented / connectionless transport-layer services, but:
  - *service:* host-to-host (not end-to-end…)
  - *no choice:* network provides one or the other
  - *implementation:* in network core

Virtual circuits

"source-to-dest path behaves much like telephone circuit"

- performance-wise
- network actions along source-to-dest path

- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)

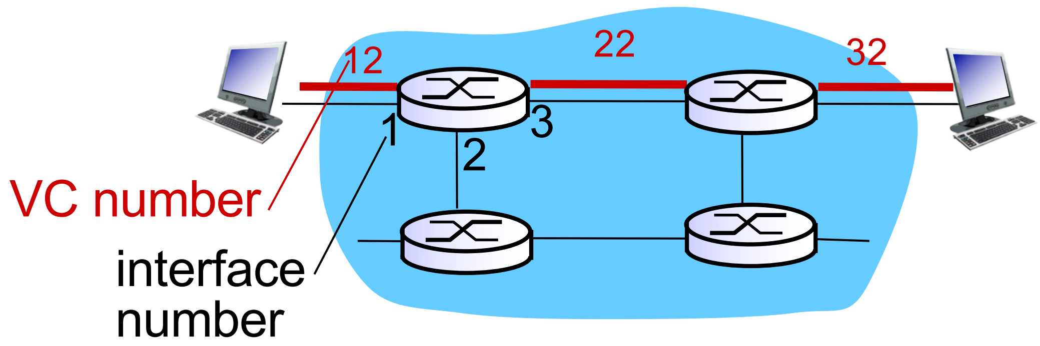
VC implementation

a VC consists of:

1. *path* from source to destination
2. *VC numbers*, one number for each link along path
3. *entries in forwarding tables* in routers along path

- packet belonging to VC carries VC number (rather than dest address)
- VC number can be changed on each link.
  - new VC number comes from forwarding table

VC forwarding table



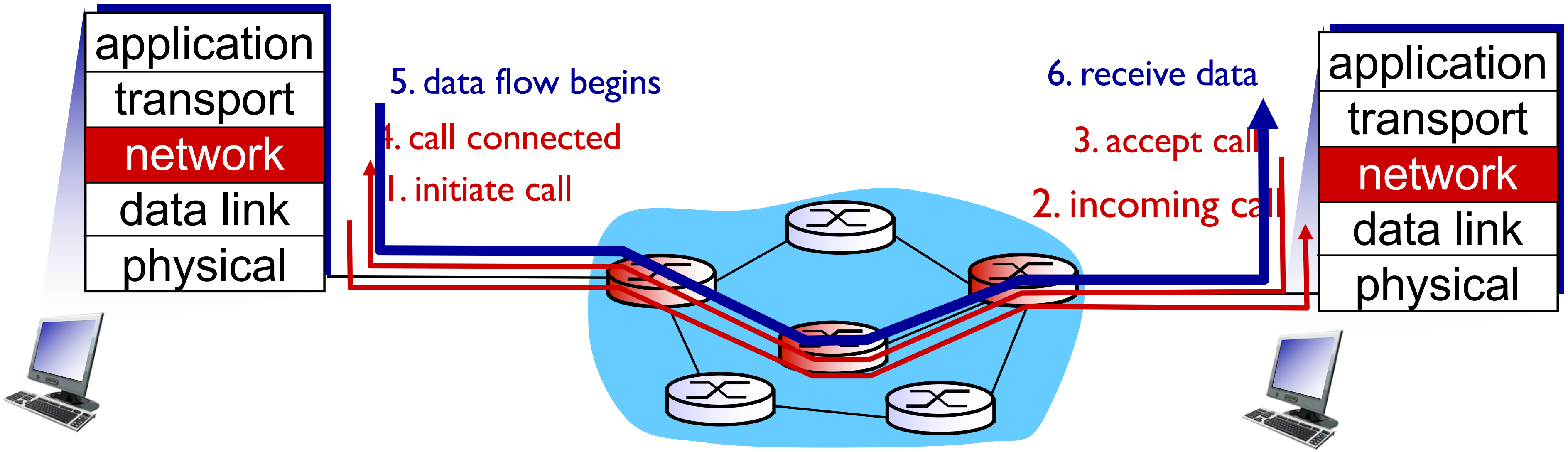
forwarding table in northwest router:

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

VC routers maintain connection state information!

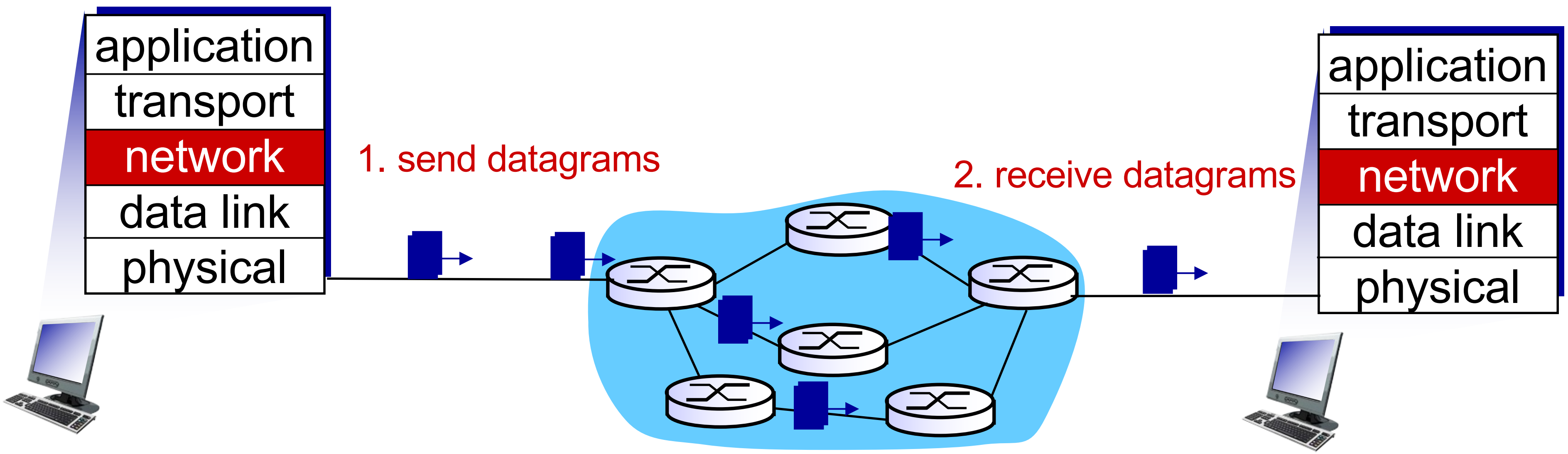
Virtual circuits: signaling protocols

- used to setup, maintain teardown VC
- used in ATM or frame-relay networks
- not used in today's Internet (network layer!)

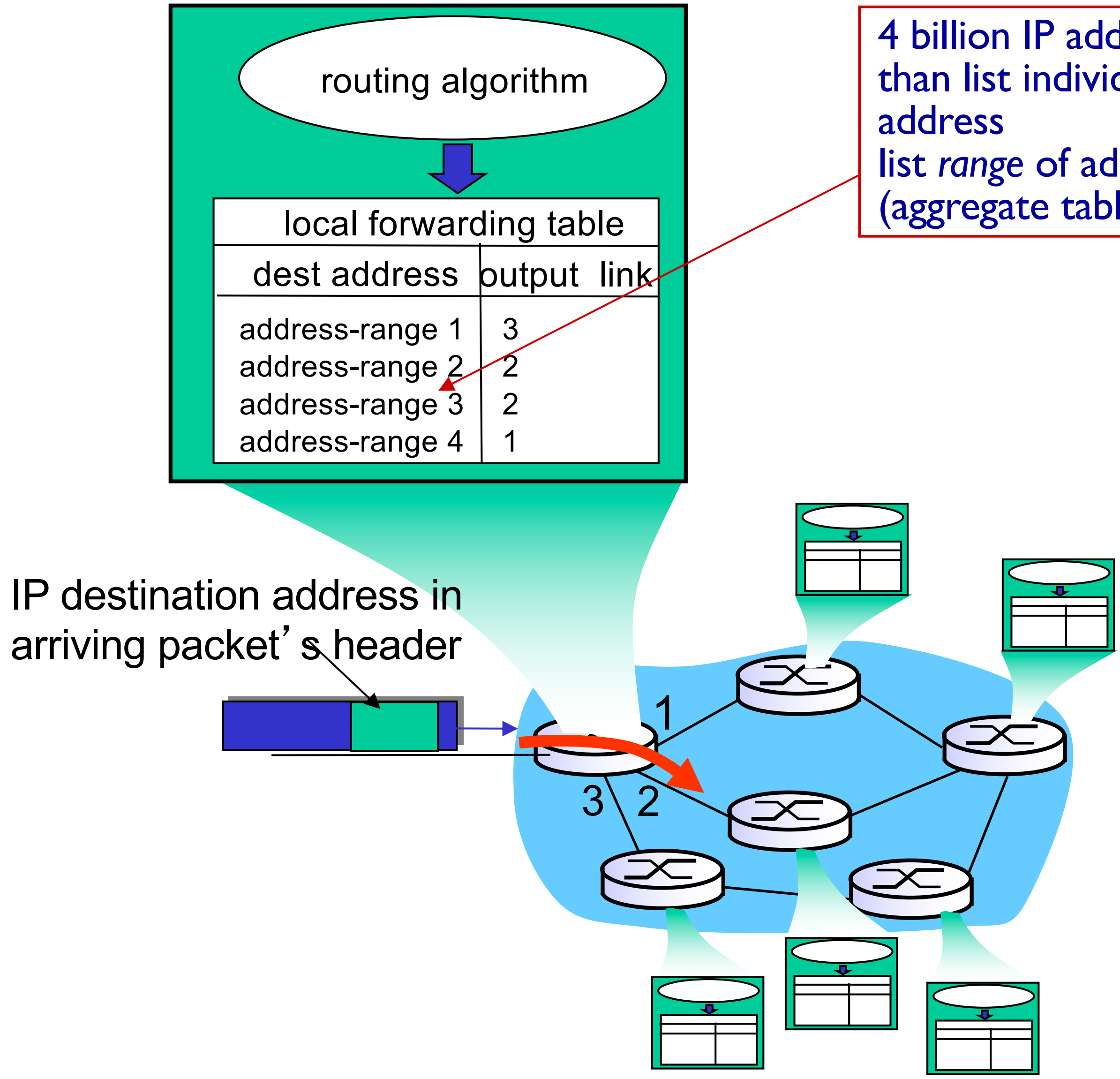


Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address



Datagram forwarding table



4 billion IP addresses, so rather than list individual destination address
list *range* of addresses (aggregate table entries)

Datagram forwarding table

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 00011001 00000000`
through
`11001000 00010111 00011111 11111111`
	2

otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Longest prefix matching

longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

DA: 11001000 00010111 00011110 10100001 which interface?

DA: 11001000 00010111 00011000 10101010 which interface?

Datagram or VC network: why?

Internet (datagram)

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

ATM (VC)

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

Datagram or VC network (revisited)

Função	Rede de Datagramas	Rede de Circuitos Virtuais (VC)
Estabelecimento prévio da conexão (ou circuito)	Não é necessário	É necessário
Endereçamento	Endereço de origem e destino em cada PDU	PDUs contêm o identificador do circuito
Routing / Forwarding	PDUs são encaminhados de forma independente entre si	A rota é estabelecida inicialmente e todos os PDUs utilizam essa rota
Informação de estado	não é necessária	necessária por VC
Falha de um elemento de rede	não é normalmente problemática	todos os VC são terminados
Controlo de tráfego e Controlo de congestão	difícil	fácil, se os recursos atribuídos são suficientes

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- intro, ICMP
- datagram format
- IPv4 addressing
- routing
- NAT
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

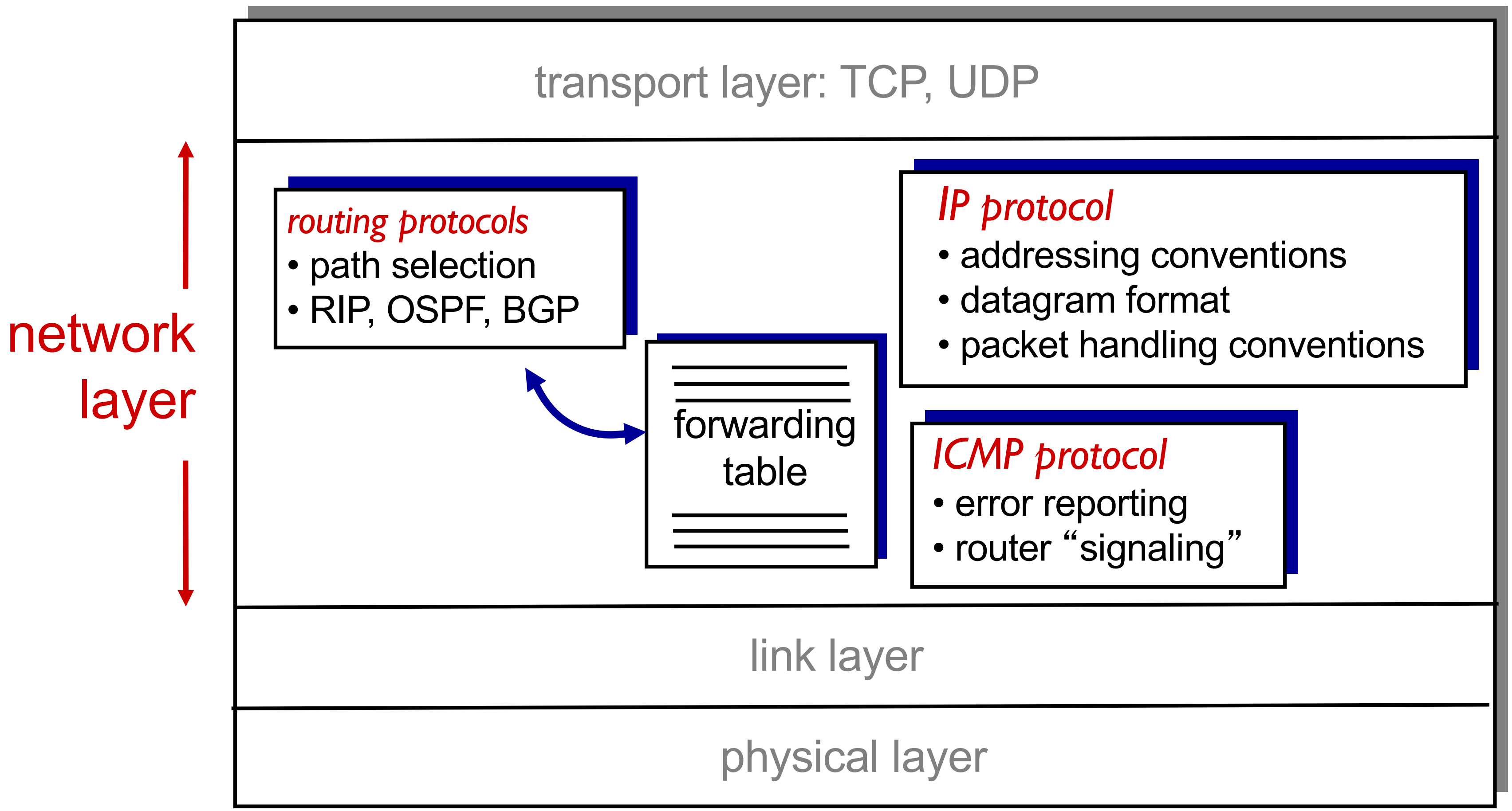
4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

The Internet network layer

host, router network layer functions:



ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

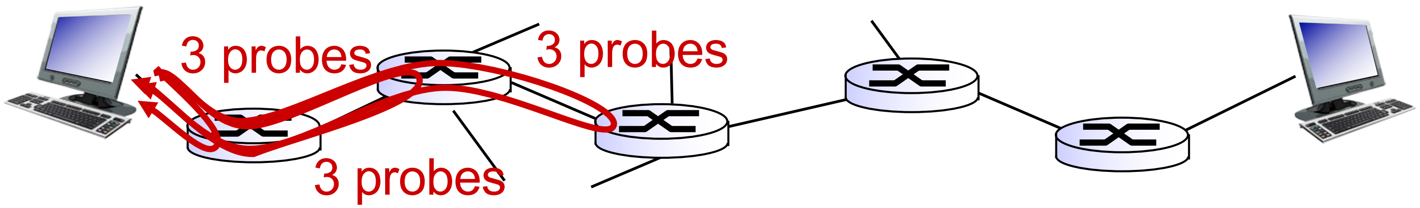
Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Traceroute and ICMP

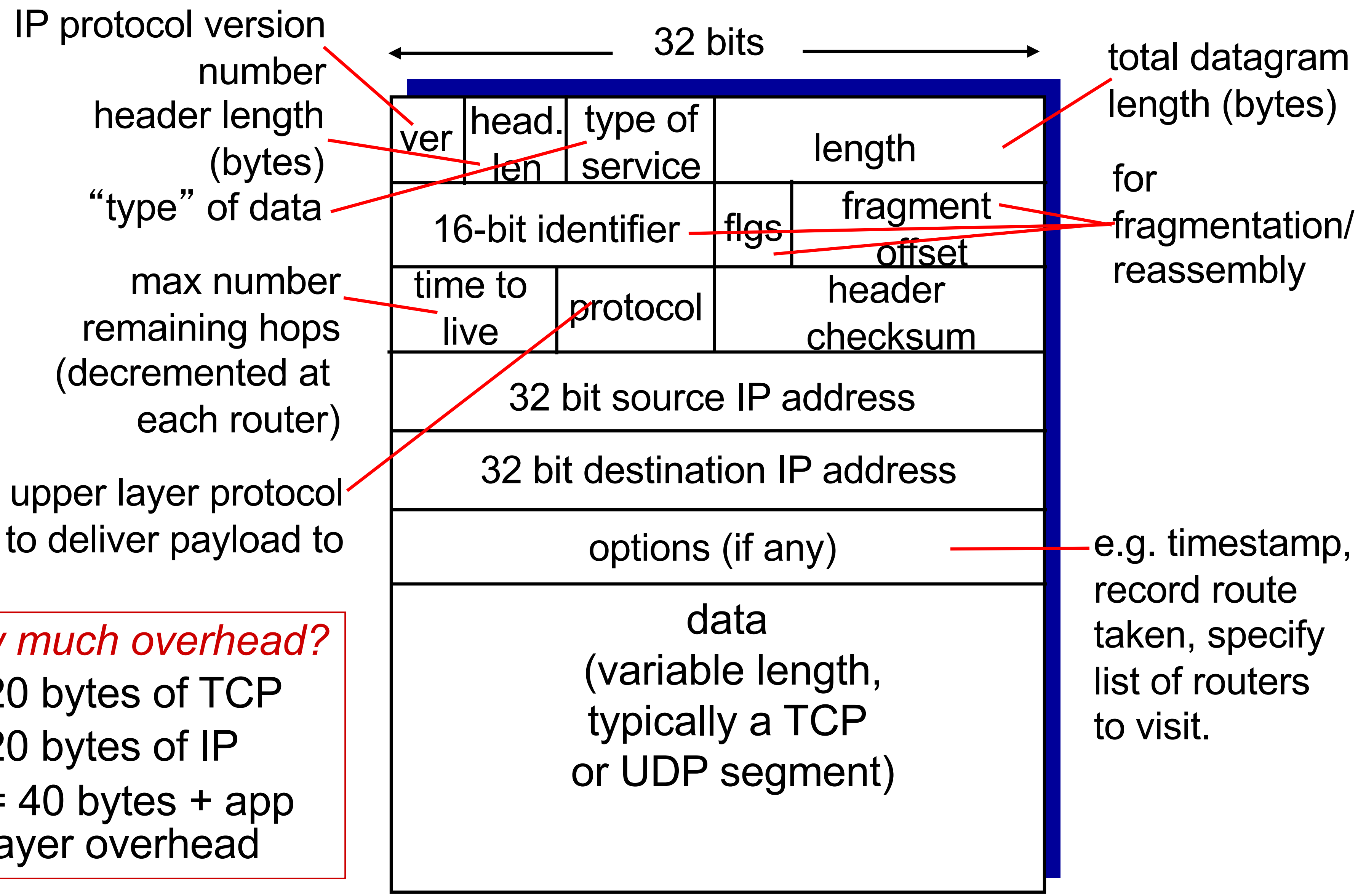
- source sends series of UDP segments (or ICMP with flag -I) to dest
  - first set has TTL =1
  - second set has TTL=2, etc.
  - unlikely port number
- when n^{th} set of datagrams arrives to n^{th} router:
  - router discards datagrams
  - and sends to source ICMP messages (type 11,code 0)
  - ICMP messages includes name of router & IP address
- when ICMP messages arrives, source records RTTs

stopping criteria:

- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops



IP datagram format

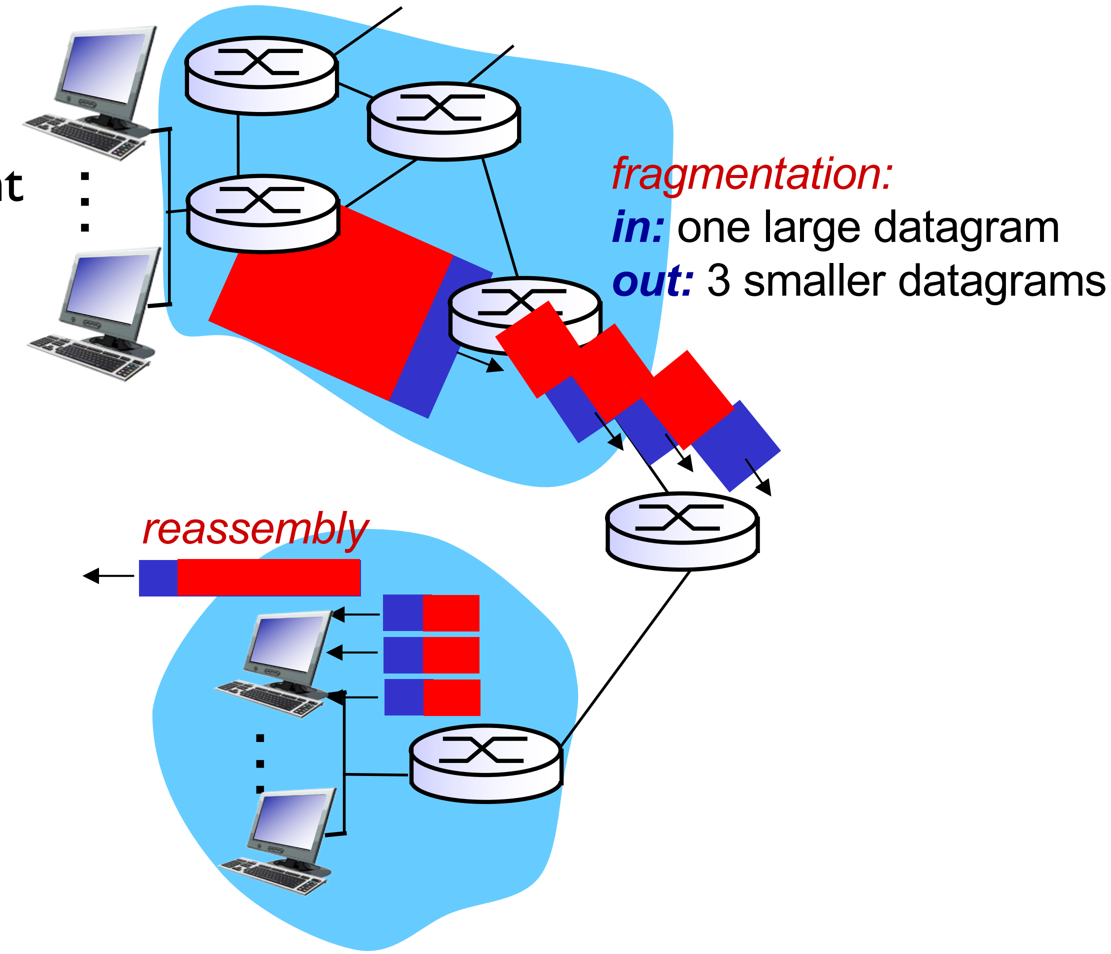


how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

IP fragmentation, reassembly

- network links have MTU (max transfer unit) size – the largest possible link-level frame
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments



IP fragmentation, reassembly

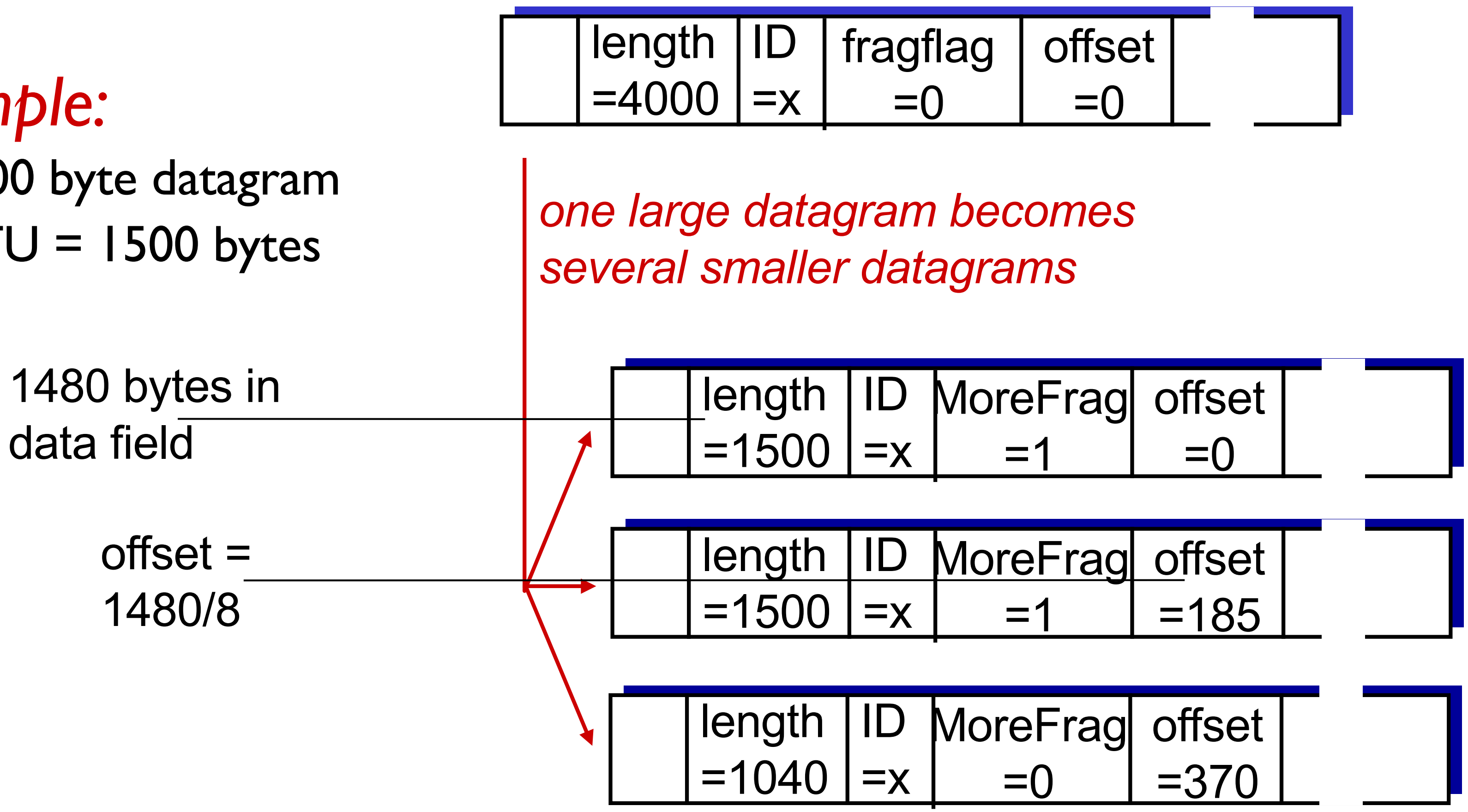
Campos manipulados na fragmentação IPv4:

- *identification* - identifica fragmentos pertencentes ao mesmo datagrama original
- *more fragments* - *flag* que determina se há mais fragmentos e também saber se o fragmento é o último
- *may fragment* - identificação da possibilidade ou não do datagrama ser fragmentado pela rede
- *fragment offset* - *offset* dos dados do fragmento relativamente ao datagrama original

Em IPv6, por defeito, não está prevista fragmentação!

IP fragmentation, reassembly

example:
- 4000 byte datagram
- MTU = 1500 bytes



(offsets in bits: 0, 1480, 2960)

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- intro, ICMP
- datagram format
- IPv4 addressing
- routing
- NAT
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

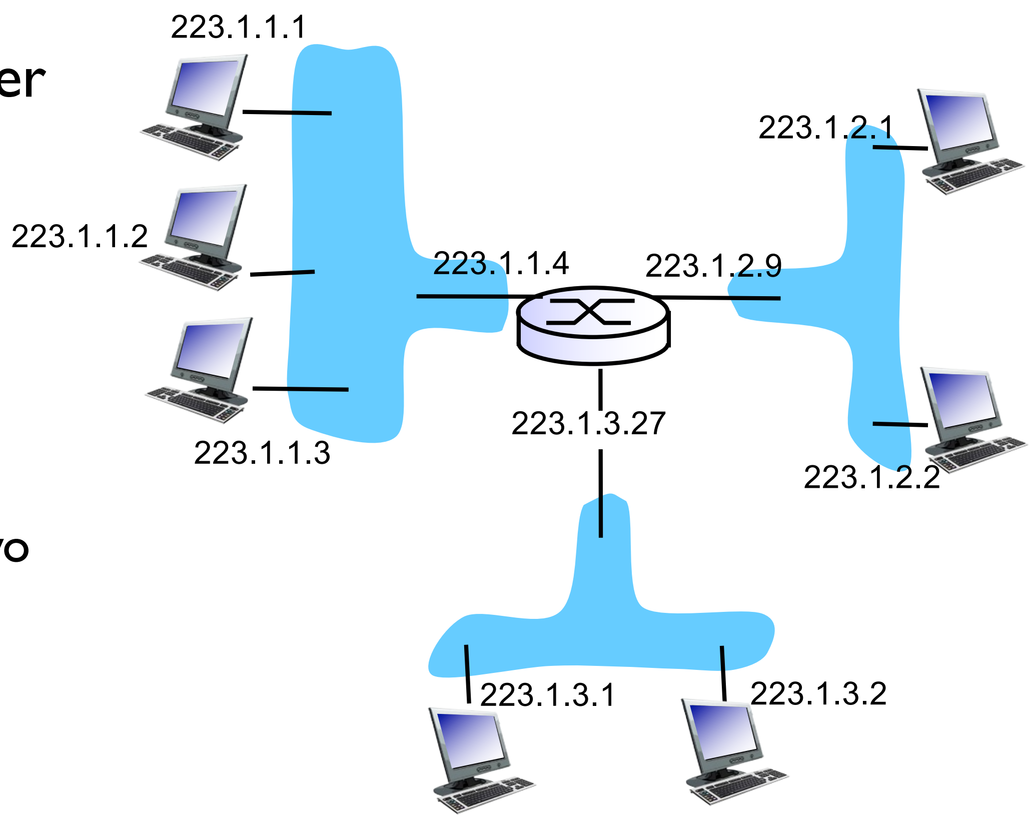
4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

IP addressing: introduction

- *IP address:* 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- *IP addresses associated with each interface*

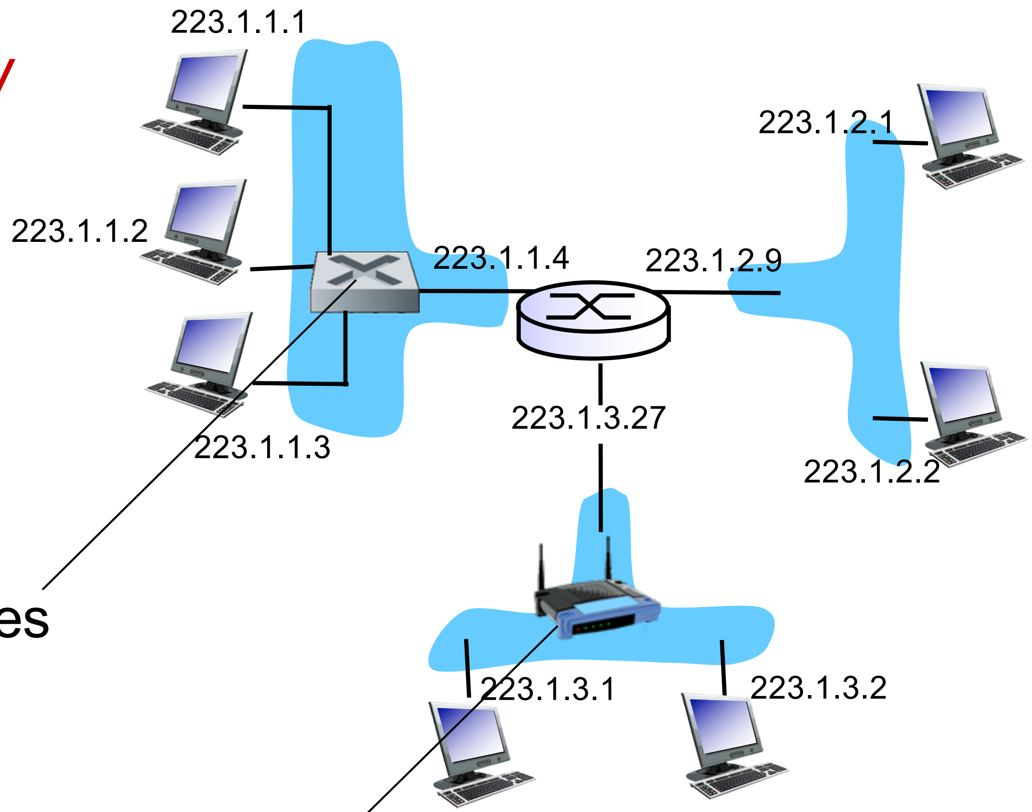


$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

IP addressing: introduction

Q: how are interfaces actually connected?

A: we'll learn about that in chapter 5, 6.



A: wired Ethernet interfaces connected by Ethernet switches

A: wireless WiFi interfaces connected by WiFi base station

For now: don't need to worry about how one interface is connected to another (with no intervening router)

IP addressing: introduction

IPv4: 32-bit *unsigned binary value*

xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

(em notação decimal - *dot decimal notation)*

- uma parte identifica a rede (ou subrede) e a outra identifica a interface da estação *(host)* nessa rede

  <rede id><host id>

- na *Internet*, cada *endereço de rede* tem de ser único
- distribuídos originalmente por 5 classes (A a E)
- atribuídos pela IANA *(Internet Assigned Number Authority)*

IP addressing: original scheme

Identificador da classe	Parte do Endereço de Rede	Parte do Endereço de Estação

Classe A

0	7 bits de end. de rede	24 bits de endereço de estação

Classe B

10	14 bits de endereço de rede	16 bits de endereço de estação

Classe C

110	21 bits de endereço de rede	8 bits end. de estação

Classe D

1110	Endereços Multicast no intervalo 224.0.0.0 - 239.255.255.255

Classe E

11110	Classe E – Reservado para utilização futura

IP addressing: classful vs. classless

Endereçamento por classes **(ou *Classful*)**

- esquema original, baseado na RFC 791
- usa os primeiros bits como identificadores de classe

Endereçamento sem classes **(ou *Classless*)**

- não considera os bits de classe; é utilizada uma máscara de 32 bits para determinar o endereço de rede
- permite routing mais eficiente por agregação de rotas, designado **CIDR** (*Classless Internet Domain Routing*)
- tabelas de encaminhamento mais pequenas: as rotas são agregadas por grupos de endereços adjacentes
- usado pelas tabelas de routing de ISPs

IP addressing: CIDR

CIDR: Classless InterDomain Routing

- subnet portion of address of arbitrary 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

IP addressing: CIDR

Máscara de endereço

Padrão que conjugado com o endereço IP devolve a parte do endereço de rede (ou sub-rede)

No endereçamento, por defeito, as máscaras usadas são:

- (Classe A) 11111111.00000000.00000000.00000000
  notação decimal: 255.0.0.0 notação CIDR: /8
- (Classe B) 11111111.11111111.00000000.00000000
  notação decimal: 255.255.0.0 notação CIDR: /16
- (Classe C) 11111111.11111111.11111111.00000000 notação decimal: 255.255.255.0 notação CIDR: /24

No endereçamento **sem classes** as máscaras podem ter qualquer outro valor, permitindo a criação de *subnets* (subredes) da classe original, ou *supernets* (agregação de endereços)

IP addressing: CIDR

Endereçamento sem classes e *subnetting*

Considere-se o endereço IP 130.1.5.1

- é o endereço da estação **5.1** da rede **130.1.0.0** (classe B)
  considerando máscara por defeito (default mask): 255.255.0.0 ou /16

Considere-se o endereço IP **130.1.5.1/24**

- é o endereço da estação **1** da sub-rede **130.1.5.0**
- o subnetting é definido no espaço host ID inicial
- <rede id><subrede id><host id>

8 bits para subnetting:
Nº subredes: 2^8(-2), Nº hosts: 2^8-2

Rede	Estação
130.1	5.1

interpretação original por classe

Máscara de subrede	Rede	Subrede	Estação
255.255.255.0	130.1	5	1

interpretação sem classe (CIDR)

IP addressing: CIDR

Exemplo de máscaras de rede + subrede em endereços de Classe B

Endereço de Host

128 64 32 16 8 4 2 1

11111111 11111111 0 0 0 0 0 0 0 0 0 0 0 0 0 0 00

255 255

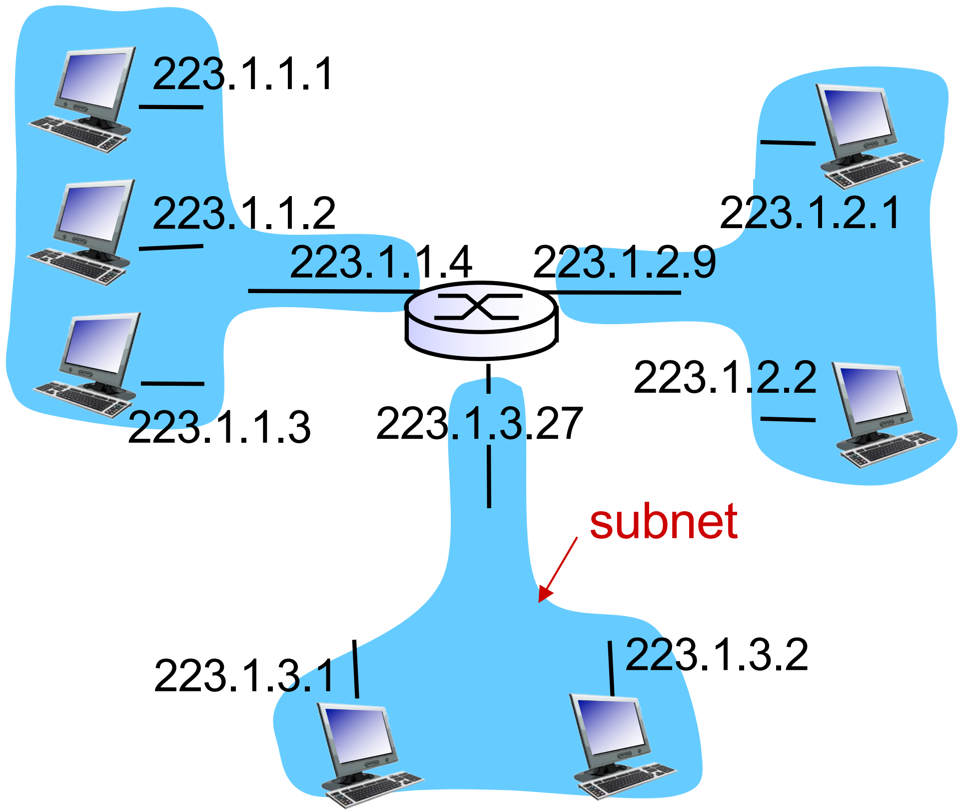
128
192
224
240
248
252
254
255

128
192
224
240
248
252

Valores para a máscara

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 intervening router*

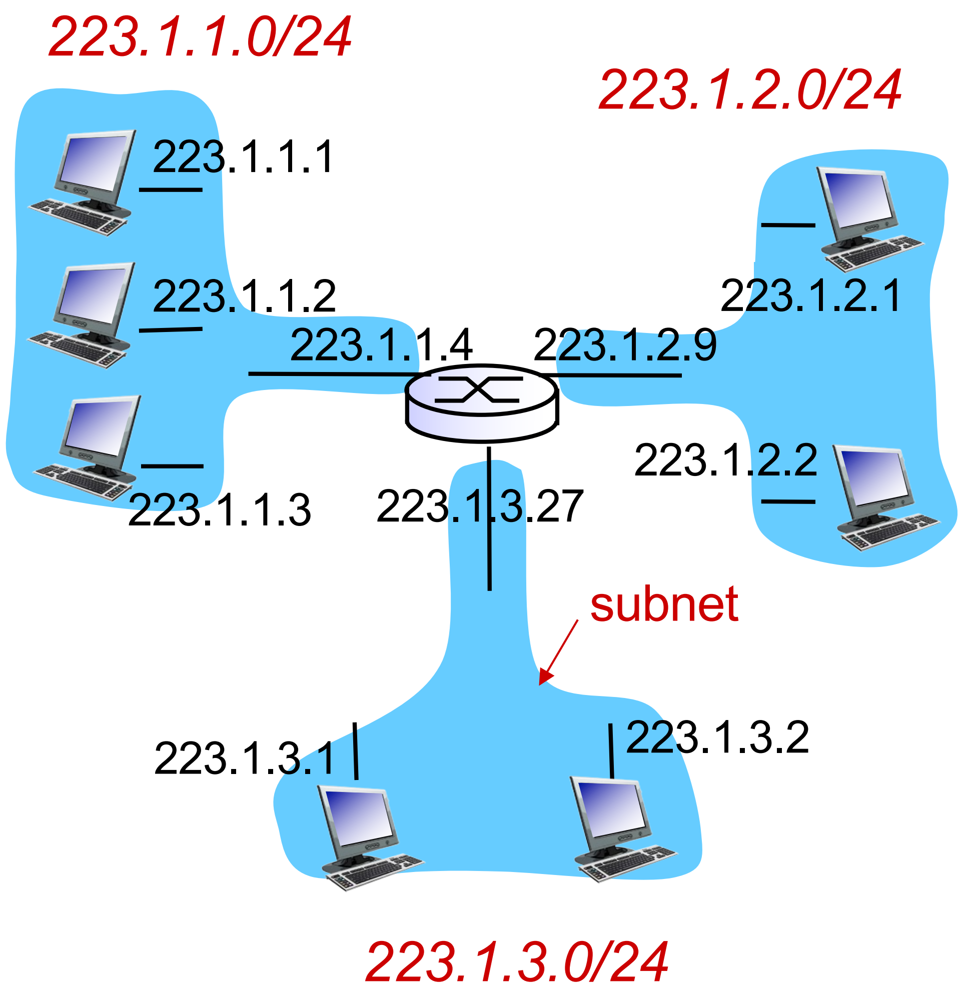


network consisting of 3 subnets (/24)

Subnets

recipe

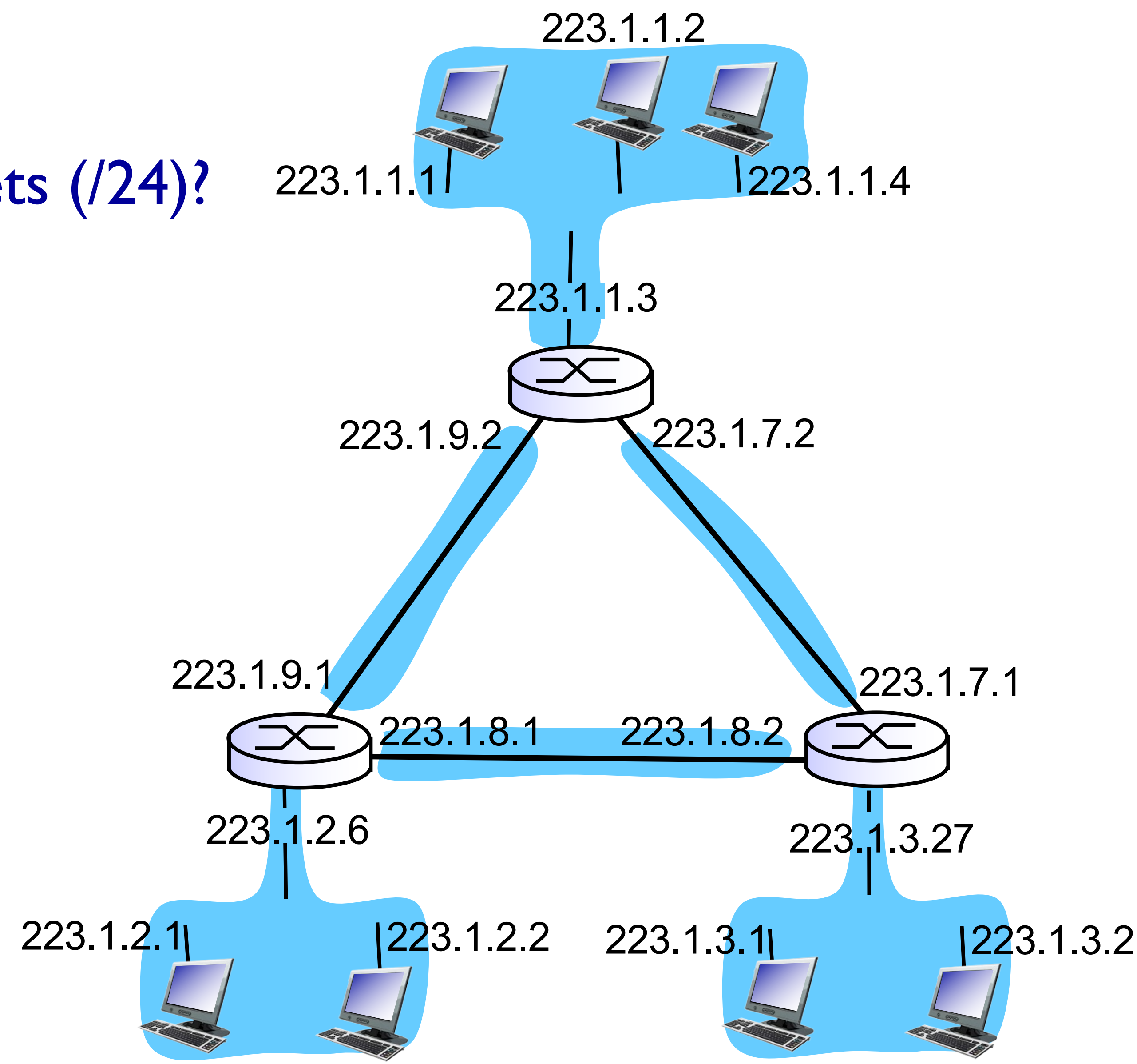
- to determine the subnets, detach each interface from its host or router, creating islands of isolated networks
- each isolated network is called a *subnet*



subnet mask: /24

Subnets

how many subnets (/24)?



Subnets

vantagens vs. custo

- permite melhor organização e gestão dos endereços
- permite introduzir mais níveis hierárquicos para routing

- contudo reduz espaço de endereçamento (vários endereços passam a não utilizáveis)
- gestão mais trabalhosa

IP addressing: reserved/private addr

Endereços reservados:

- os primeiros 4 bits não podem ser 1 (classe E)
- 127.x.x.x é o endereço reservado para *loopback*
- bits de host a 0s ou 1s (qualquer host, todos os hosts)
- bits de rede / subrede a 0s ou 1s (qualquer rede, todas as redes)

Endereços privados: atribuídos para internets privadas (sem conectividade IP global, não devem ser visíveis, nem são encaminhados na Internet) (ver RFC1918):

- bloco 192.168.0.0 - 192.168.255.255 / 16
- bloco 172.16.0.0 - 172.31.255.255 /12
- bloco 10.0.0.0 - 10.255.255.255 /8

Host com várias interfaces é designado de *multihomed*

IP addressing: reserved/private addr

Endereços para configuração dinâmica do Link-Local:

- O bloco 169.254.0.0 /16 está reservado para comunicação entre estações ligadas ao mesmo meio físico nas seguintes condições:
- Quando um interface não foi configurado com um endereço IP, nem manualmente nem por uma fonte na rede (ex: DHCP) a estação pode configurar automaticamente o interface com um endereço IPv4 de prefixo 169.254.0.0/16 (RFC 3927)
- Algoritmo:
  1. Gera um endereço aleatório uniformemente distribuído no intervalo [169.254.1.0 , 169.254.254.255]
  2. Envia ARP-request com endereço de destino igual ao gerado (probe)
  3. Se houver ARP-reply então repete 1. porque há colisão de endereço
  4. Senão anuncia endereço gerado através de um ARP-announcement

IP addresses: how to get one?

Q: How does a *host* get IP address?

- hard-coded by system admin in a file
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- intro, ICMP
- datagram format
- IPv4 addressing
- routing
- NAT
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

IP routing: introduction

- Tanto os routers como as estações, possuem uma tabela de encaminhamento
- As entradas na tabela incluem:
  - 1ª coluna: Endereço da Rede de destino (mais máscara)
  - 2ª coluna: Endereço IP da interface de entrega (*next hop*)
  - N coluna: Identificador da interface de saída da máquina local
  - colunas opcionais: *flags*, tráfego no interface, custo, etc.
- A entrega (forwarding), ou salto (hop) seguinte de um datagrama IP, é decidida em função do endereço IP destino do datagrama

IP routing: introduction

Exemplo: tabela de encaminhamento host c/ IP 192.110.1.240

\> netstat -nr

destination	next_hop	netmask	flags	use	interface
default	192.110.1.254	0.0.0.0	UG	102410	tu0
192.110.1.0	192.110.1.240	255.255.255.0	UH	234576	tu0
....	......	…….	....	......	....
192.168.1.0	192.110.1.253	255.255.255.0	UG	124586	tu0

Leitura da última linha:

Um datagrama destinado à rede 192.168.1.0 será entregue na interface de endereço 192.110.1.253 saindo pela interface local tu0

Qual a topologia de rede que se pode inferir da tabela?

IP routing: forwarding algorithm

Entrega (forwarding):

- É facilitada pelo endereçamento hierárquico
- O endereço IP é: **a.b.c.d/m = X.Y** (rede.estação)

1) <u>usar</u> **máscara** para extrair o endereço de rede **X**
2) procurar entrada que melhor se ajuste a **X**

   <u>se</u> **X** é local, entregar no interface **X.Y** (entrega directa)

   <u>senão</u> usar **X** para determinar o próximo salto (*next hop*);
3) A entrada por defeito (**0.0.0.0/0)** ajusta-se a todos os **X**

IP routing: supernetting

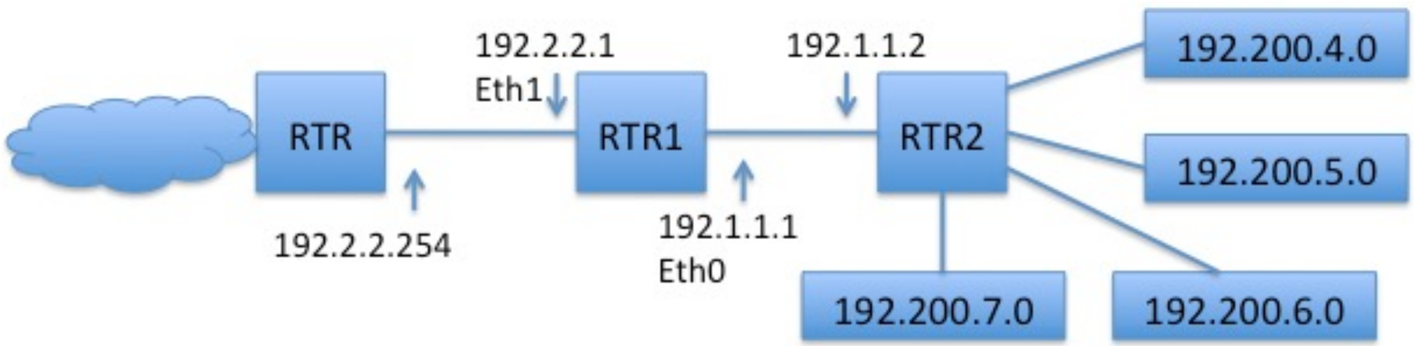


Tabela de encaminhamento de RTR1 - sem Supernetting

Destino	Próximo Nó	Máscara	Interface
192.2.2.0	192.2.2.1	255.255.255.0	Eth1
192.1.1.0	192.1.1.1	255.255.255.0	Eth0
192.200.4 (0000 0100).0	**192.1.1.2**	**255.255.255.0**	**Eth0**
192.200.5 (0000 0101).0	**192.1.1.2**	**255.255.255.0**	**Eth0**
192.200.6 (0000 0110).0	**192.1.1.2**	**255.255.255.0**	**Eth0**
192.200.7 (0000 0111).0	**192.1.1.2**	**255.255.255.0**	**Eth0**
Default	192.2.2.254	0.0.0.0	Eth1

192.200.4(0000 0100).0	**192.1.1.2**	**255.255.252 (11111100).0**	**Eth0**

IP routing: static vs. dynamic

Encaminhamento (routing):

a) Estático - baseado em rotas pré-definidas

- as rotas permanecem fixas
- reduz o tráfego na rede
- esquema simples mas pouco flexível

b) Dinâmico - rotas atualizadas ao longo do tempo

- os routers trocam informação de routing entre si
- esta actualização dinâmica de rotas é obtida através de protocolos específicos de encaminhamento (routing):
  - » RIP, OSPF, BGP, etc.
- grande flexibilidade e adaptação (automática) a falhas ou mudanças na configuração de rede
- o tráfego de actualização pode causar sobrecarga na rede

IP routing: default route

- **Caminho por defeito** é a rota a seguir caso não exista uma entrada específica na tabela para a rede de destino
  - é um caso particular de encaminhamento estático
  - a rota por defeito tem prioridade inferior à das outras rotas
  - é identificado pelo termo **default** ou pela rede **0.0.0.0**
  - permite reduzir a tabela de encaminhamento

- Os protocolos de encaminhamento modelam a rede como um <u>gráfo</u> e calculam o melhor caminho para um dado destino

IP routing: route computation

- Computação dinâmica das rotas:
  - centralizada - cada router, conhecendo a topologia da área, determina o melhor caminho para os possíveis destinos dessa área
  - distribuída - cada router envia informação de encaminhamento que conhece aos routers seus vizinhos (redes a que dá acesso)

- Princípio utilizado
  - Vector Distância (*Vector Distance*)
    - e.g. Routing Information Protocol (RIP), IGRP
  - Estado das ligações (*Link State*)
    - e.g. Open Shortest Path First (OSPF)

IP routing: route computation

- Um router pode conhecer rotas estáticas e/ou dinâmicas para um mesmo destino, aprendidas por protocolos distintos.
- Como é seleccionada a "melhor" rota?
  - distância - indicador administrativo que permite estabelecer uma relação de preferência entre rotas aprendidas por protocolos de routing distintos.
  - métrica - indicador que traduz o custo de fazer forwarding por uma determinada interface, permitindo estabelecer uma relação de preferência entre rotas aprendidas pelo mesmo protocolo de routing.

IP addresses: how to get one?

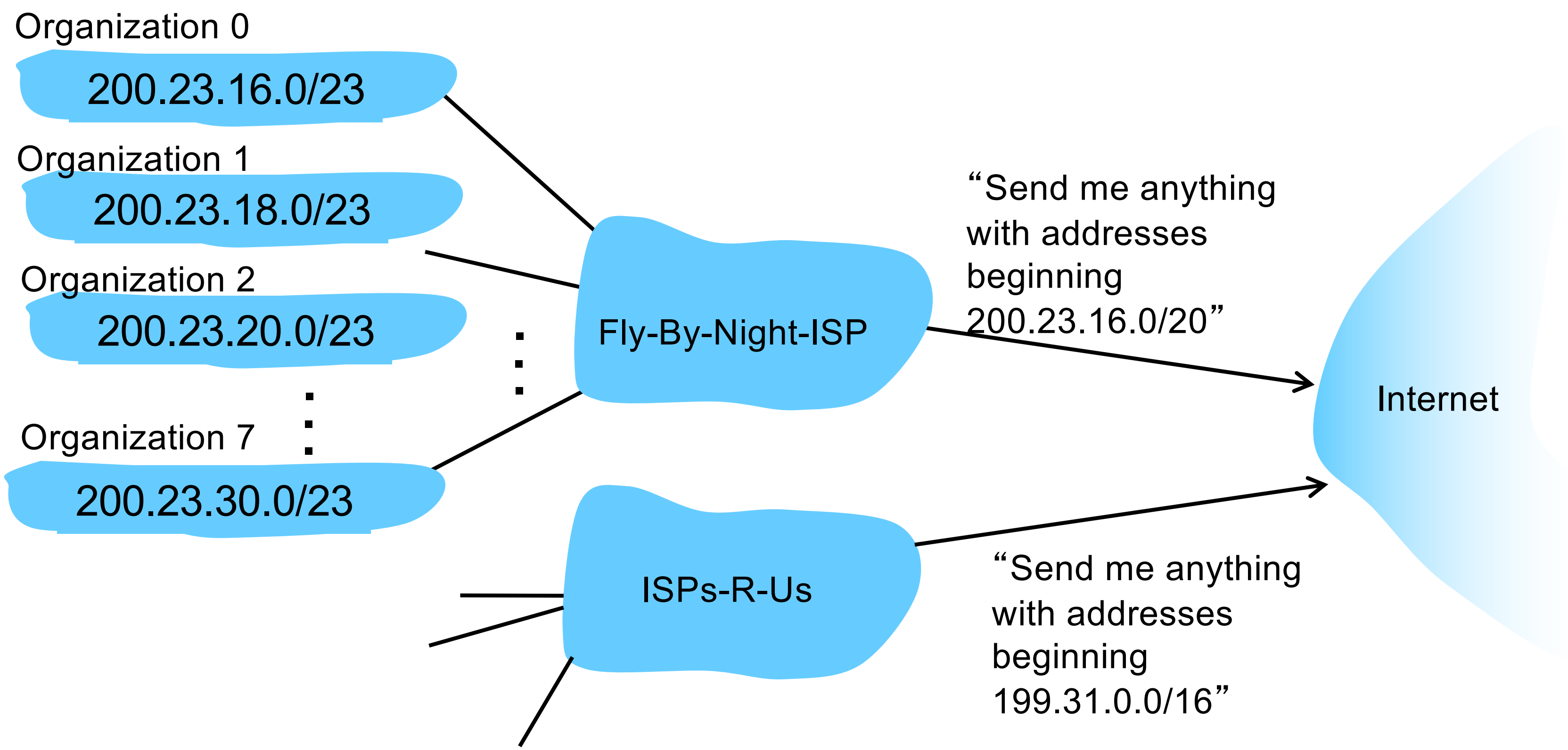
Q: how does *network* get subnet part of IP addr?

A: gets allocated 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

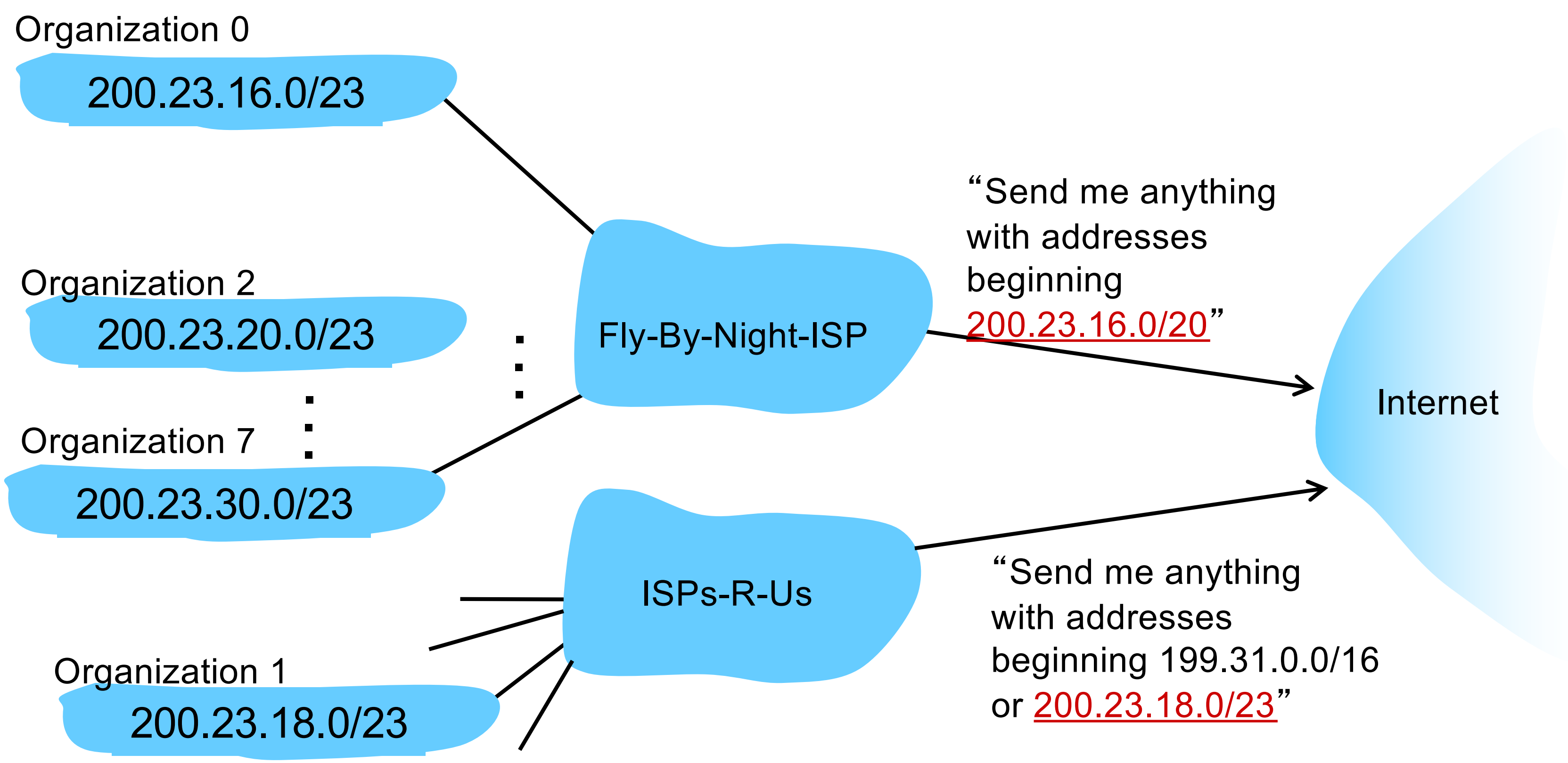
Hierarchical addressing: route aggregation

hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

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



IP addressing: the last word...

Q: how does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/

- allocates IP addresses, through 5 regional registries (RRs) (who may then allocate to local registries)
- manages DNS root zone, including delegation of individual TLD (.com, .edu , …) management

Q: are there enough 32-bit IP addresses?

- ICANN allocated last chunk of IPv4 addresses to RRs in 2011
- NAT (next) helps IPv4 address space exhaustion
- IPv6 has 128-bit address space

"Who the hell knew how much address space we needed?" Vint Cerf (reflecting on decision to make IPv4 address 32 bits long)

Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

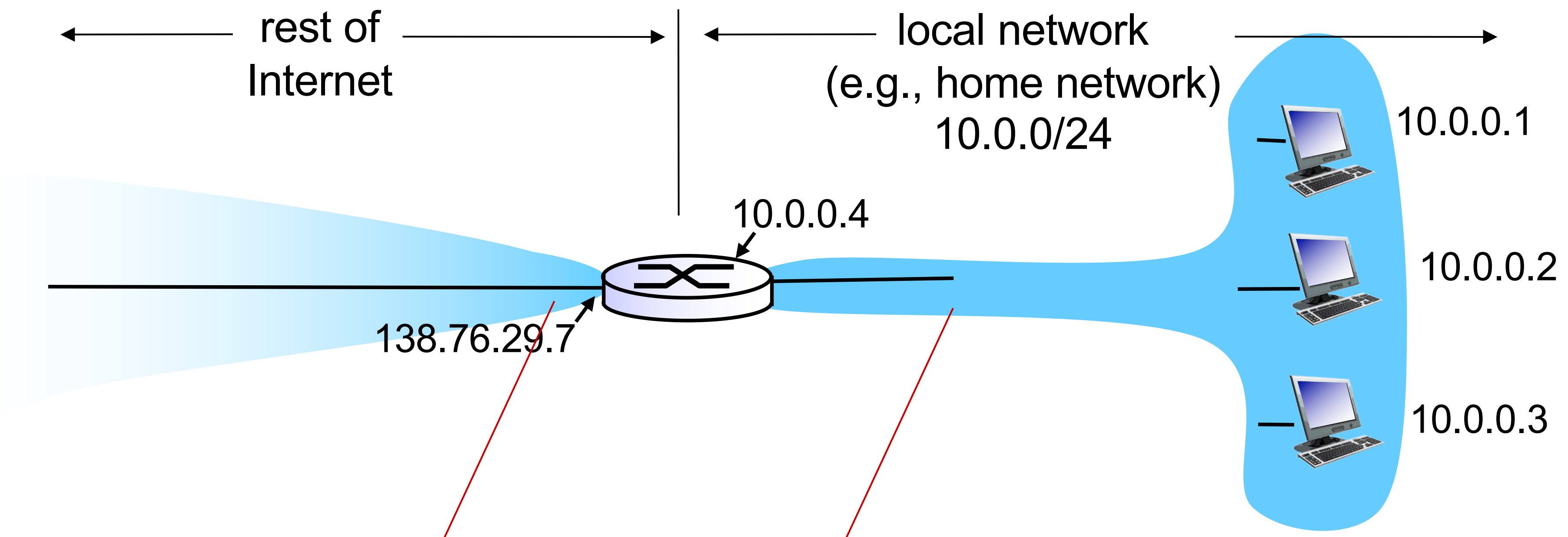
4.4 IP: Internet Protocol
- intro, ICMP
- datagram format
- IPv4 addressing
- routing
- NAT
- IPv6

4.5 routing algorithms
- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet
- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

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

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

- range of addresses not needed from ISP: just one IP address for all devices (then private addresses)
- 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

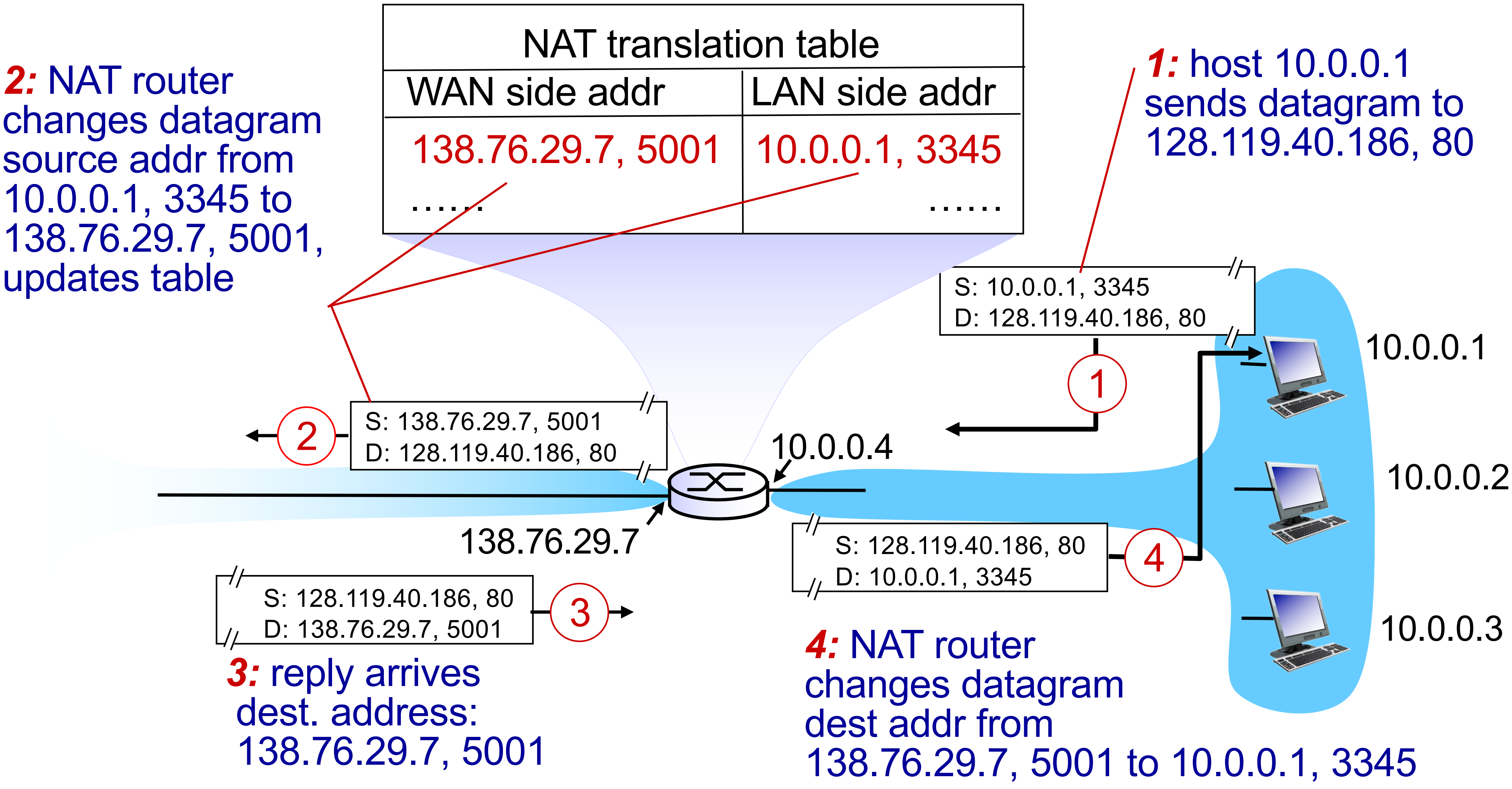
implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr

- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair

- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: network address translation

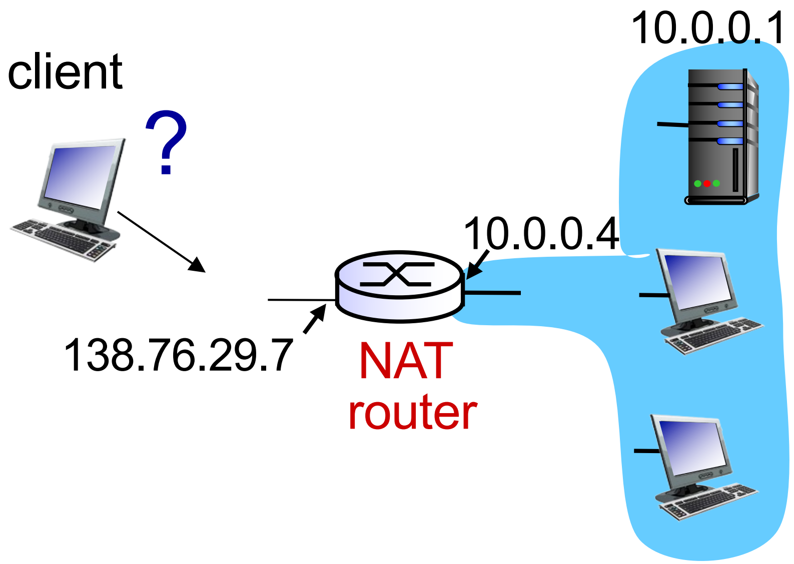


NAT: network address translation

- 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
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - address shortage should instead be solved by IPv6

NAT traversal problem

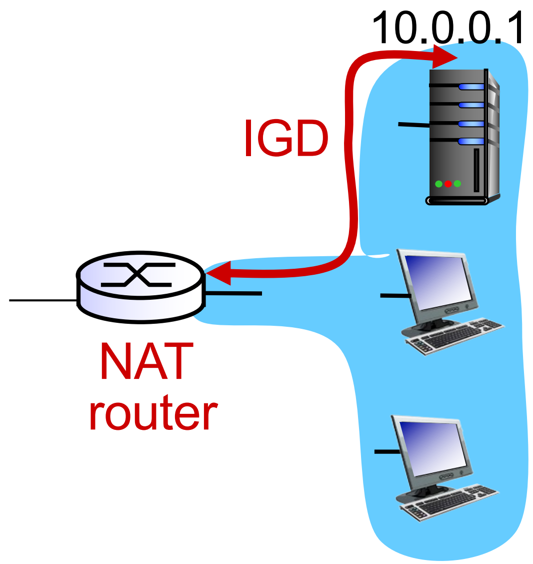
- client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
  - only one externally visible NATed address: 138.76.29.7
- *solution1:* statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (123.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000



NAT traversal problem

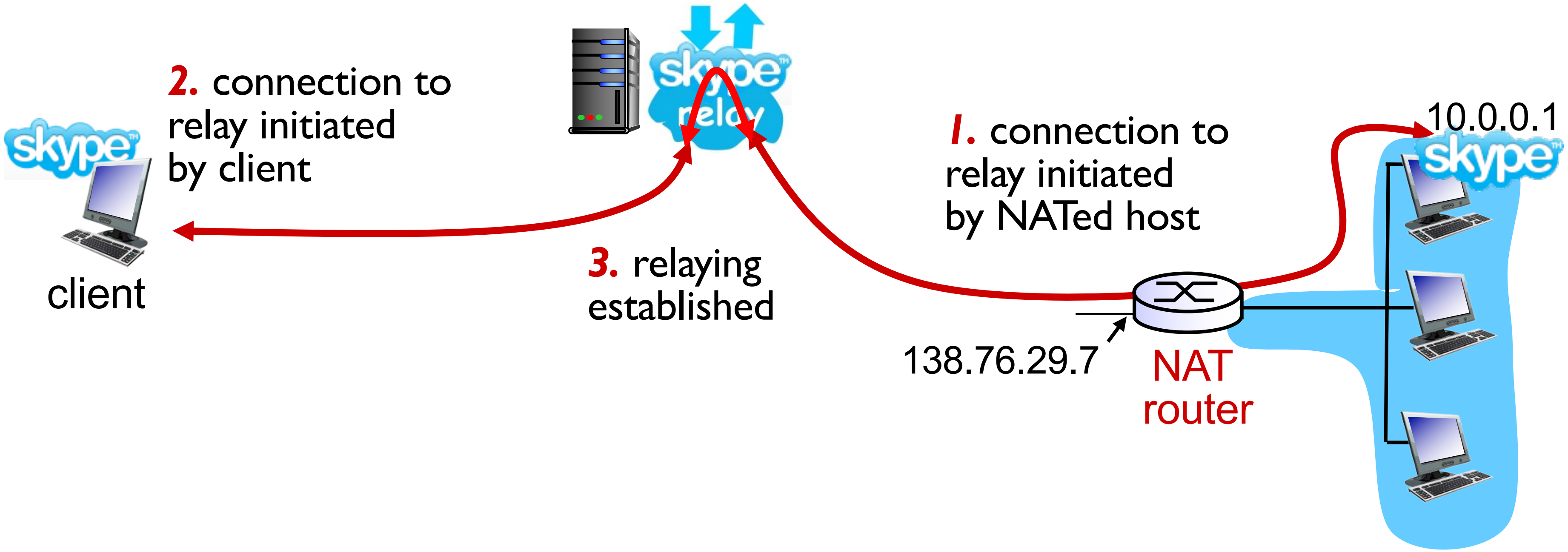
- *solution 2:* Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATed host to:
  - learn public IP address (138.76.29.7)
  - add/remove port mappings (with lease times)

  i.e., automate static NAT port map configuration



NAT traversal problem

- *solution 3:* relaying (used in Skype)
  - NATed client establishes connection to relay
  - external client connects to relay
  - relay bridges packets between to connections



Chapter 4: outline

4.1 introduction

4.2 virtual circuit and datagram networks

4.3 what's inside a router

4.4 IP: Internet Protocol

- intro, ICMP
- datagram format
- IPv4 addressing
- routing
- NAT
- IPv6

4.5 routing algorithms

- link state
- distance vector
- hierarchical routing

4.6 routing in the Internet

- RIP
- OSPF
- BGP

4.7 broadcast and multicast routing

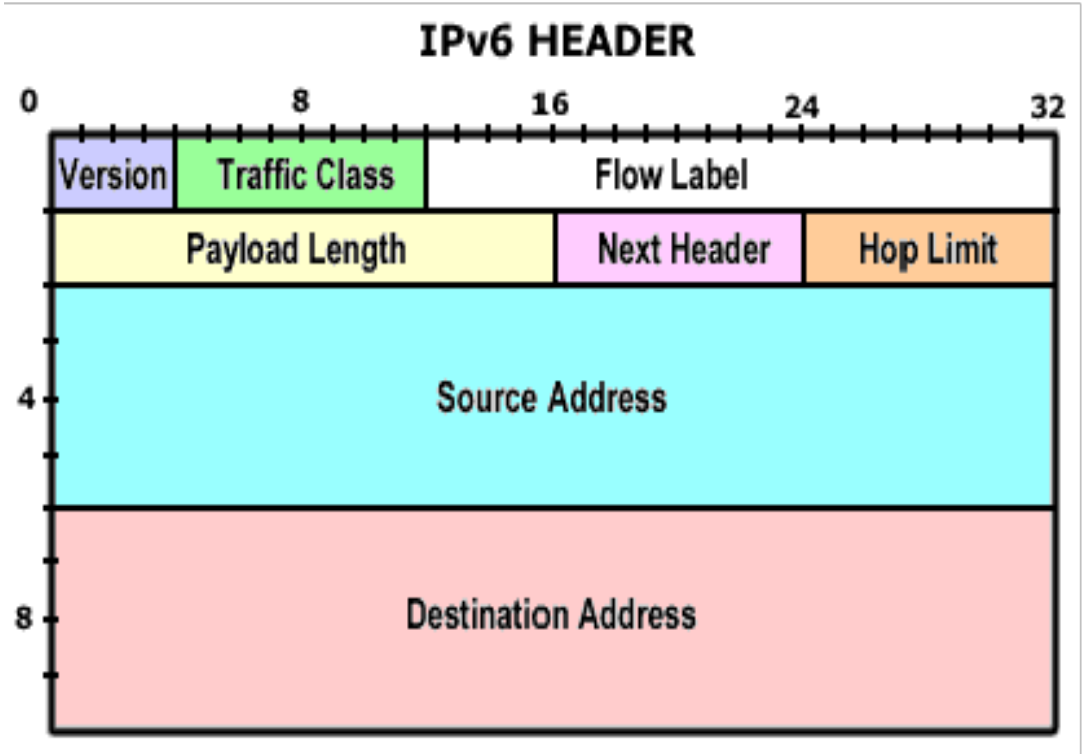
IPv6: motivation

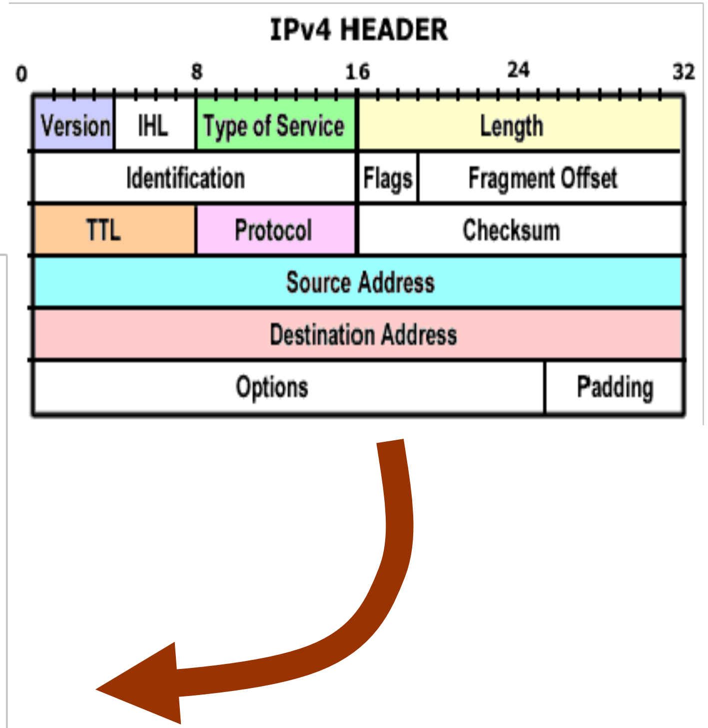
- *initial motivation:* 32-bit address space soon to be completely allocated.
- additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS (Quality of Service)

IPv6 datagram format:

  - fixed-length 40 byte header
  - no fragmentation allowed, by default

IPv6 datagram format

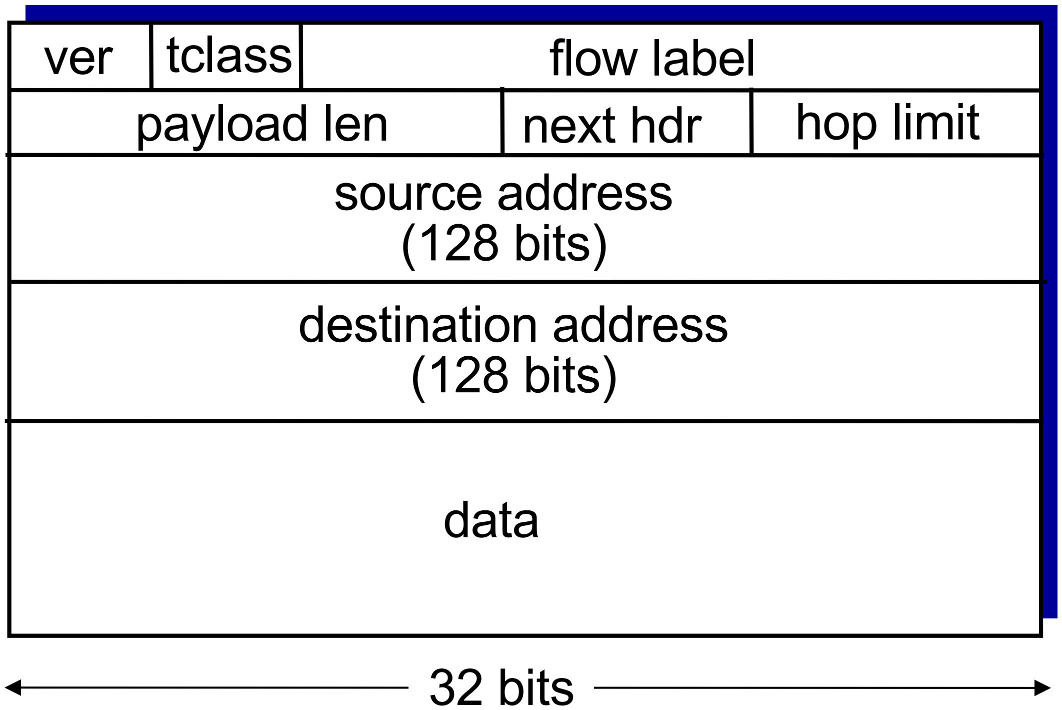




Moving to a simpler header...

IPv6 datagram format

traffic class: set 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



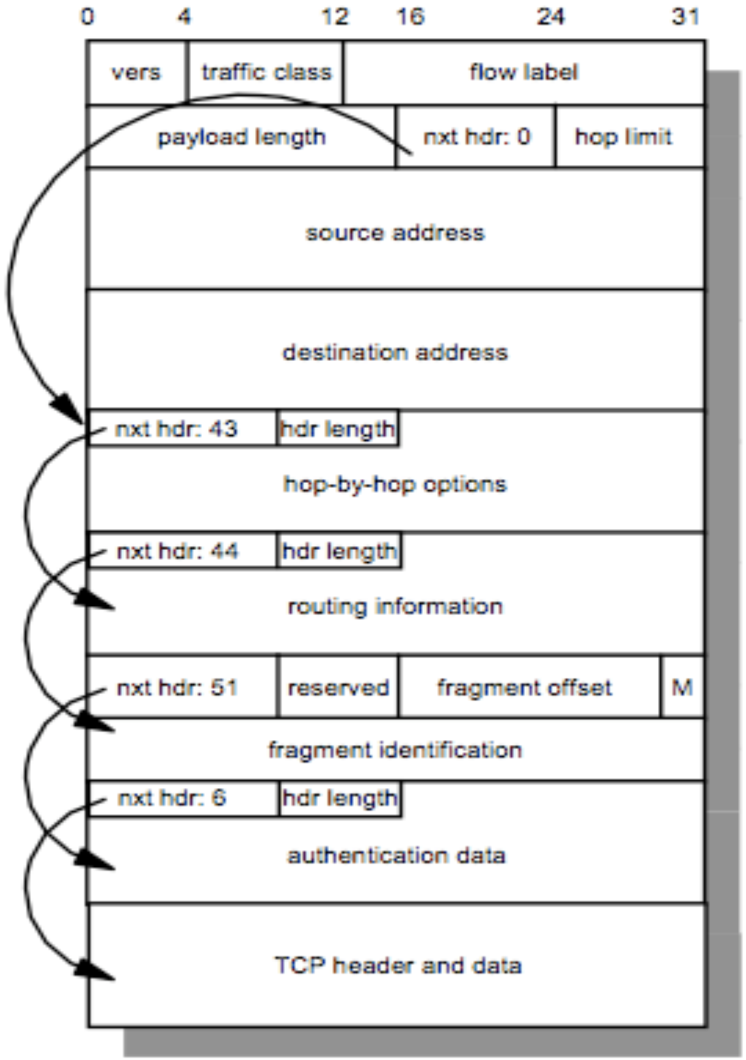
IPv6: other changes from IPv4

- *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. "ICMP Packet Too Big"
  - other control messages to support multicast, mobile IP, etc.

IPv6: other changes from IPv4

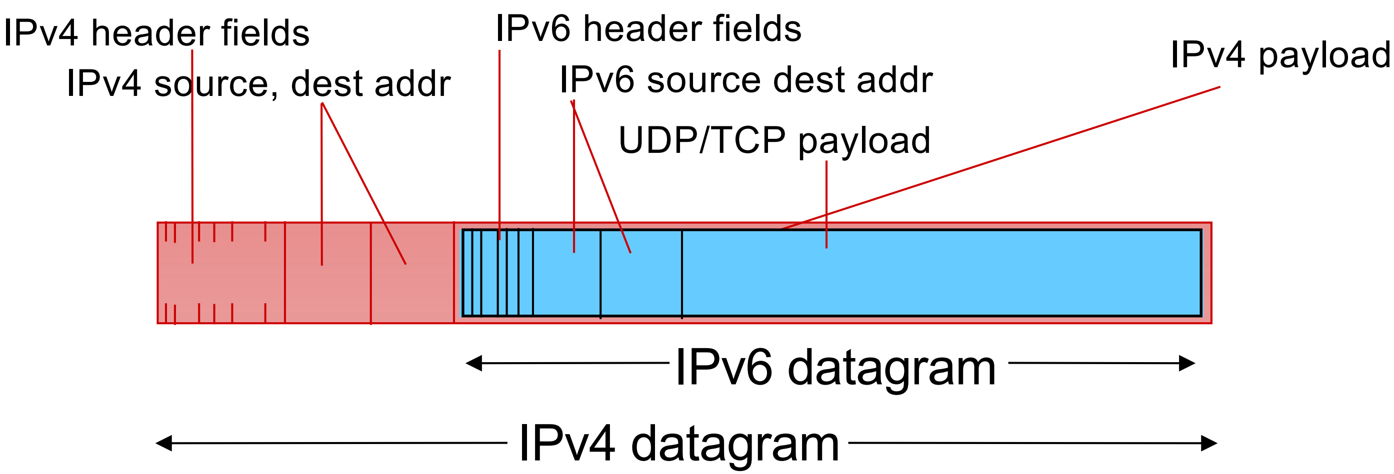
The field ***next header*** (equivalent to "Protocol" in IPv4) is used to implement specific options

Example of an IPv6 packet including multiple headers

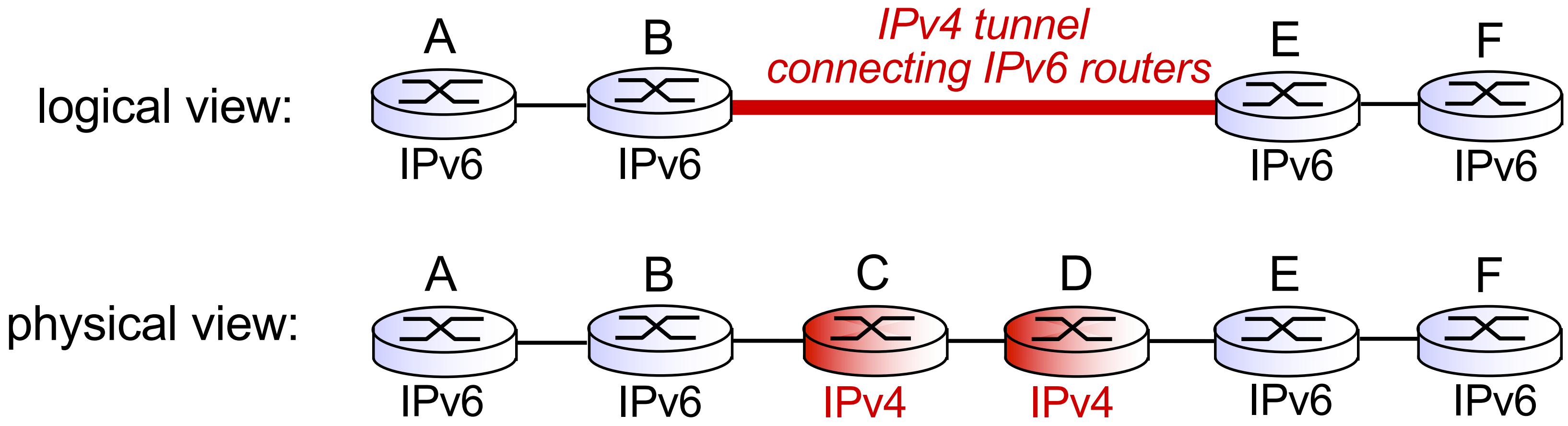


Transition from IPv4 to IPv6

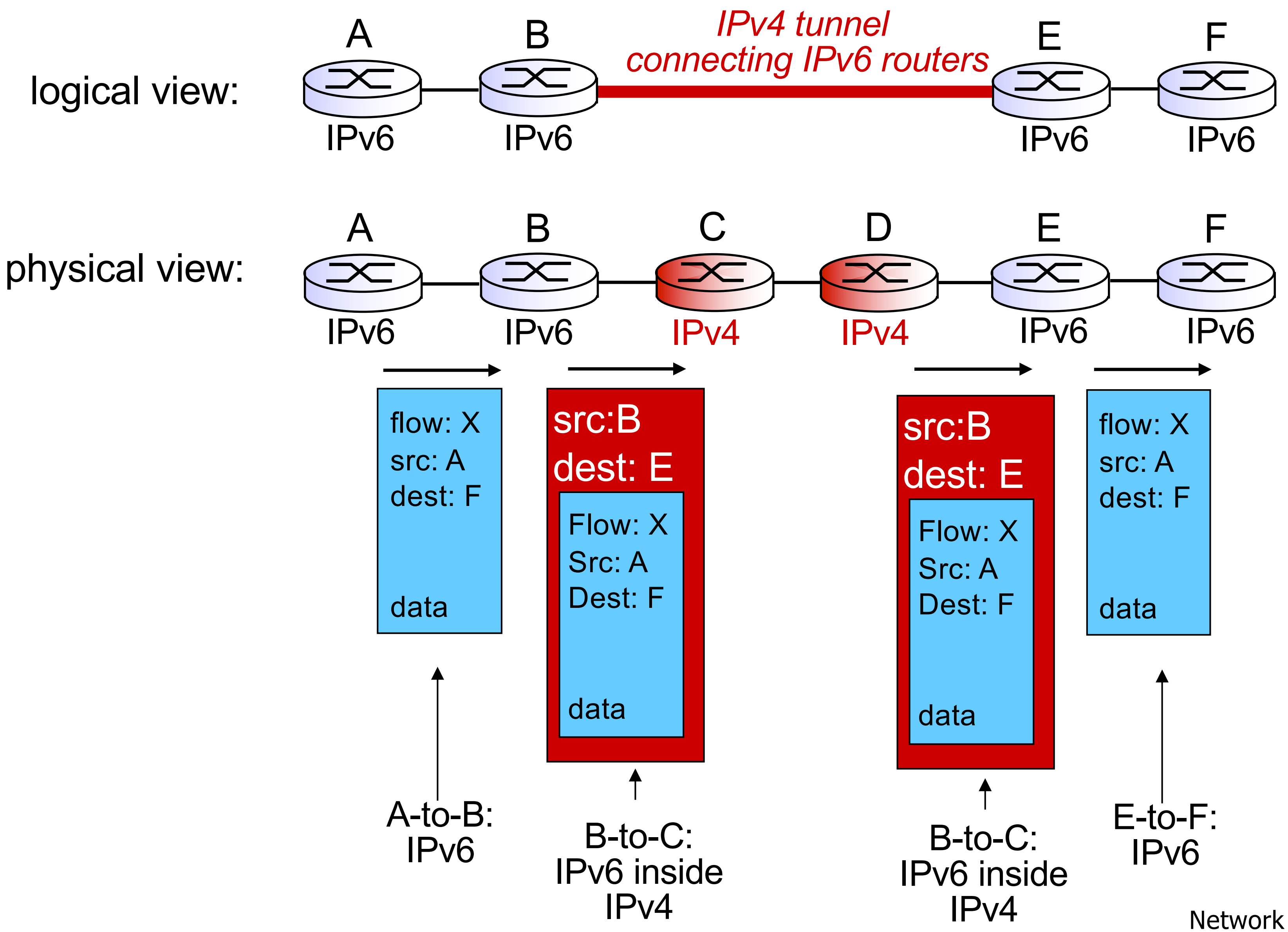
- not all routers can be upgraded simultaneously
  - no "flag days"
  - how will network operate with mixed IPv4 and IPv6 routers?
- *tunneling:* IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers



Tunneling


A
B
IPv4 tunnel
connecting IPv6 routers
E
F
logical view:
IPv6
IPv6
IPv6
IPv6
A
B
C
D
E
F
physical view:
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

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

IPv6: adoption

- US National Institutes of Standards estimate [2013]:
  - ~3% of industry IP routers
  - ~11% of US gov't routers
- *Long (long!) time for deployment, use*
  - 25 years and counting!
  - think of application-level changes in last 25 years: WWW, Facebook, Instagram, many more…
  - *Why?*
  - *Things are changing…*

IPv6: State of deployment 2018

- Since the World IPv6 Launch (2012), levels of IPv6 deployment in networks and service providers all over the globe have increased considerably.
- **Over 25% of all Internet-connected networks advertise IPv6 connectivity.**
- Google reports 49 **countries deliver more than 5% of traffic over IPv6, and** 24 co**untries whose IPv6 traffic exceeds 15%.**
- Major mobile networks are driving IPv6 adoption. In Japan (NTT – 7%, KDDI – 42% and Softbank – 34%), India (Reliance JIO – 87%) and the USA (Verizon Wireless – 84%, Sprint – 70%, T-Mobile USA – 93%, and AT&T Wireless – 57%).
- IPv6 is moving from the "Innovators" and "Early Adoption" stages of deployment to the "Early Majority" phase.

(Source: Internet Society https://www.internetsociety.org/resources/2018/state-of-ipv6-deployment-2018, November 2018)

IPv6: State of deployment 2018

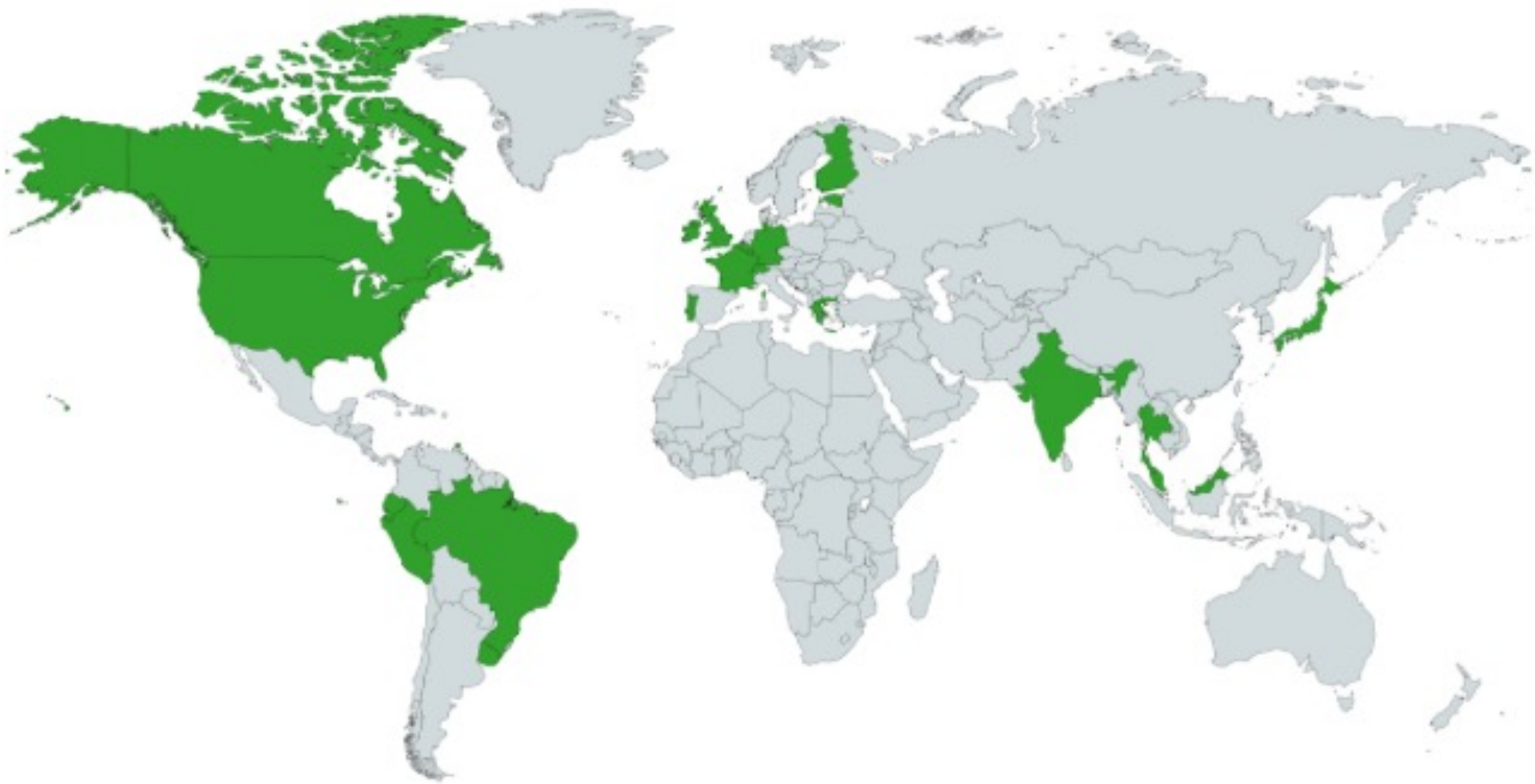
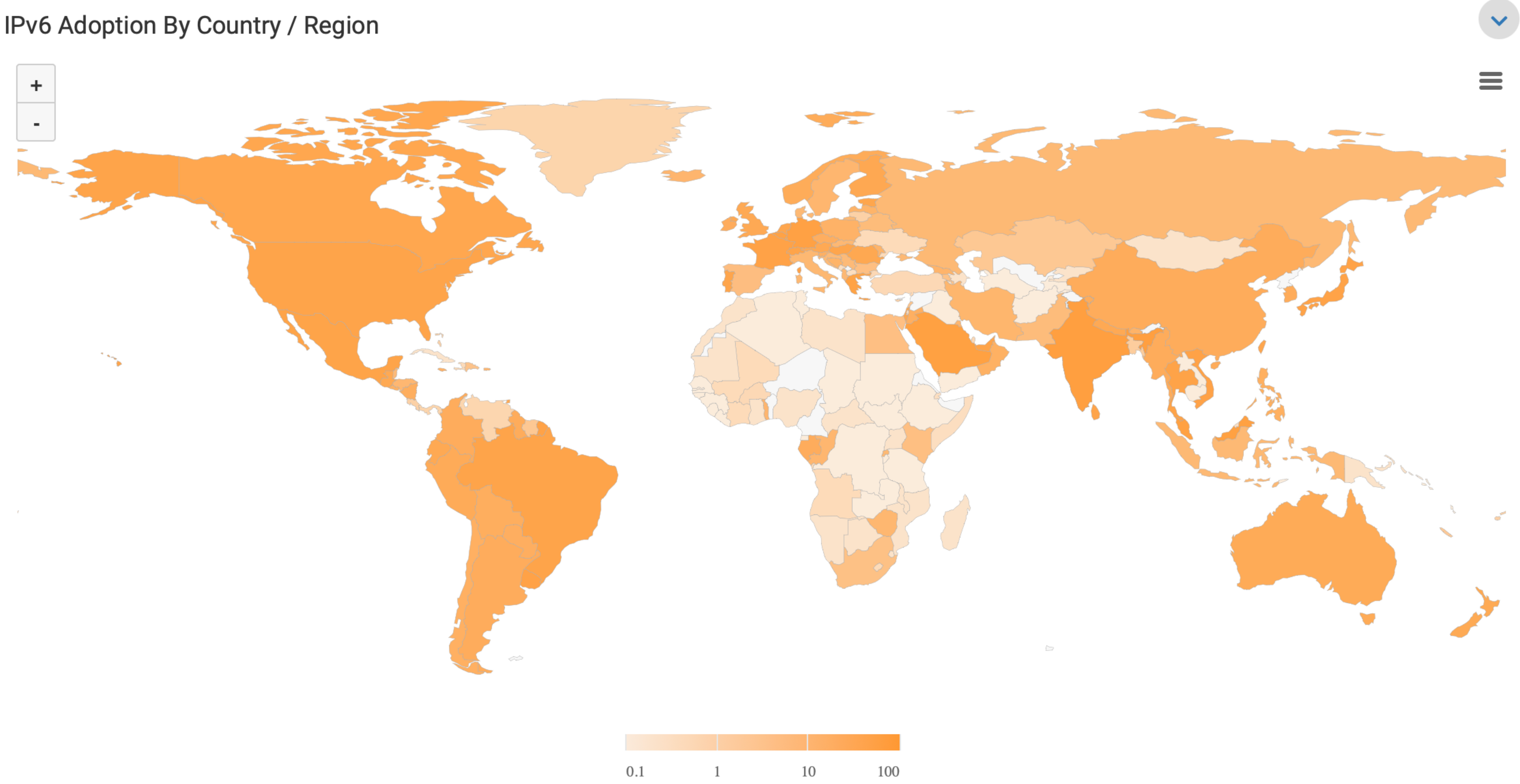


Figure 1 – Countries with IPv6 deployment greater than 15%

(Source: Internet Society https://www.internetsociety.org/resources/2018/state-of-ipv6-deployment-2018, November 2018)

IPv6: State of deployment 2022

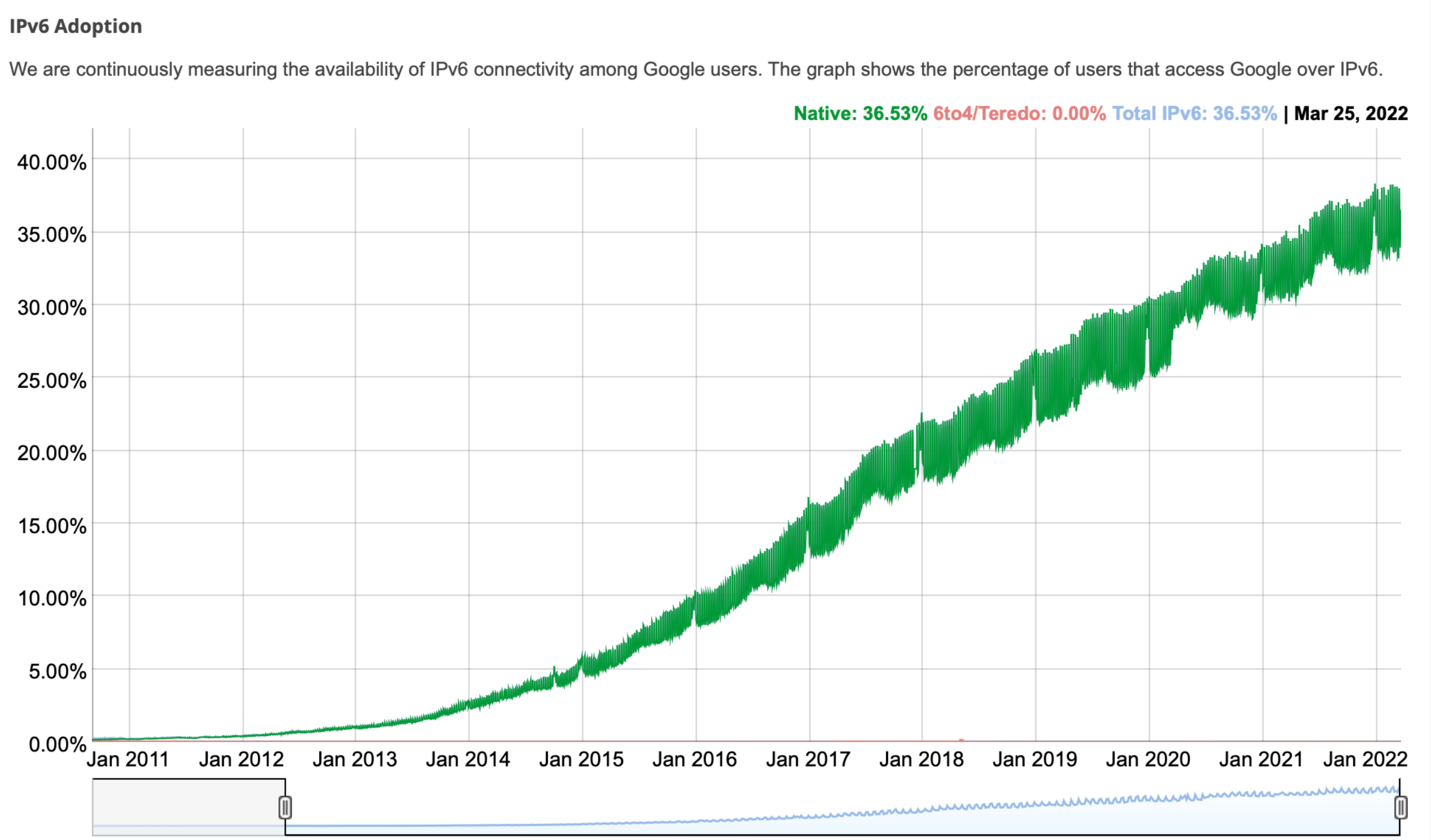


(Source: AKAMAI, https://www.akamai.com/visualizations/state-of-the-internet-report/ipv6-adoption-visualization, March 2022)

IPv6: State of deployment 2022

Google[1]: ~ 40% of clients access services via IPv6

NIST: 1/3 of all US government domains are IPv6 capable



[1] https://www.google.com/intl/en/ipv6/statistics.html