CEN445 – Network Protocols and Algorithms

# Chapter 5 – Network Layer

# 5.6 Network Layer in the Internet

Dr. Mostafa Hassan Dahshan
Computer Engineering Department
College of Computer and Information Sciences
King Saud University
mdahshan@ksu.edu.sa
http://faculty.ksu.edu.sa/mdahshan

# Network Layer in the Internet

## Design Principles

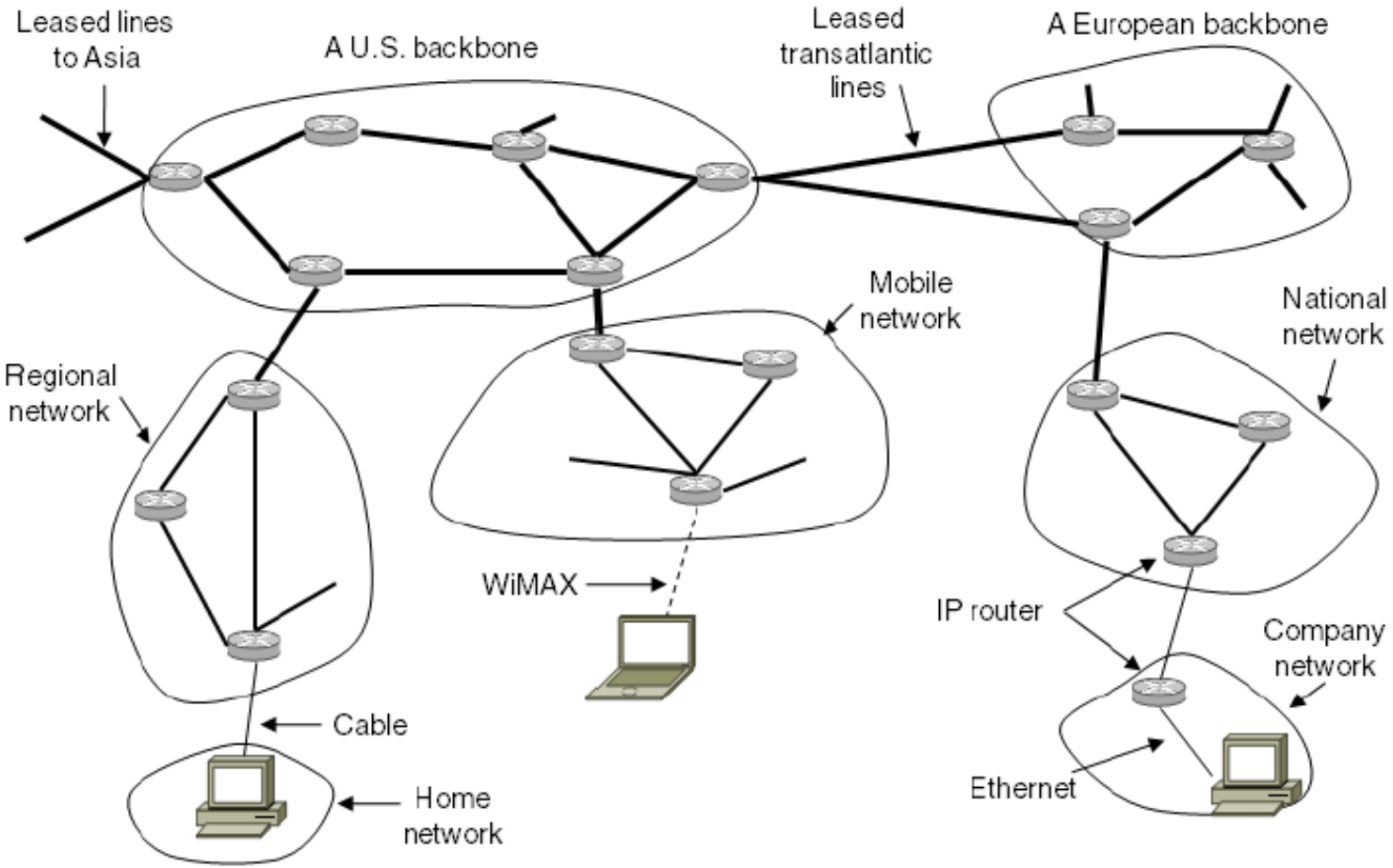
1. Make sure it works
2. Keep it simple
3. Make clear choices
4. Exploit modularity
5. Expect heterogeneity
6. Avoid static options and parameters
7. Look for a good design; it need not be perfect
8. Be strict when sending and tolerant when receiving
9. Think about scalability
10. Consider performance and cost

# Network Layer in the Internet

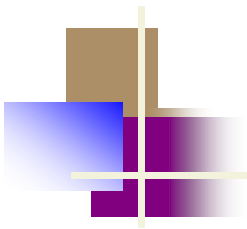
- Collection of subnetworks or Autonomous Systems (ASes) that are interconnected
- No real structure: Several major backbones, high bandwidth links, fast routers
- Tier 1 networks: biggest backbone
- Attached to backbones: ISP, regional nets
- Attached to reg. nets: more ISPs, LANs, univ
- IP is the glue to holds the Internet together
- Unlike others, IP designed for internetworking

# Network Layer in the Internet

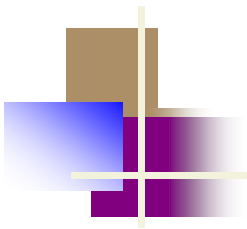
Internet is an interconnected collection of many networks that is held together by the IP protocol



# Network Layer in the Internet



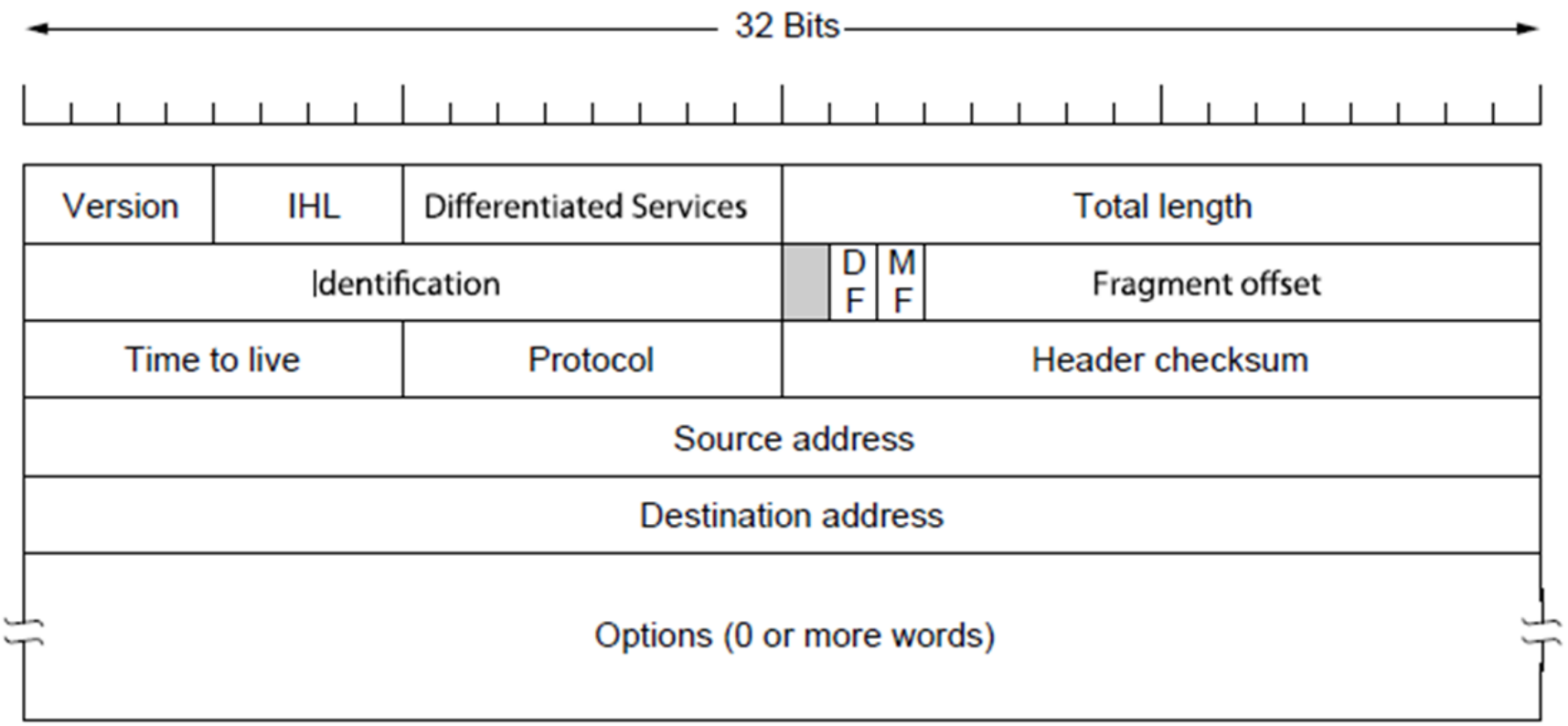
- Transport layer takes data stream
- Break it into datagrams
  - Theoretical limit is 64 KB
  - In practice, 1500 B max to fit in Ethernet
- Datagrams may be fragmented along the way, reassembled at destination
- May traverse multiple networks along path
- At destination, datagrams are handed to transport layer, then to upper process



# IP Version 4 Protocol

- IP packet consists of header and body
- Header
  - 20 bytes fixed
  - 0-40 bytes optional (variable)
- Transmitted in big endian (MSB first)
  - left-to-right, top-to-bottom

# IP Version 4 Protocol



# IP Version 4 Protocol

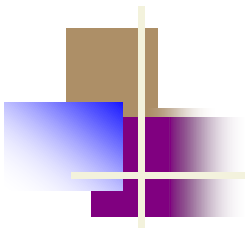
- Version
  - identifies protocol version: v4, v5 or v6
- IHL
  - header length in 32-bit words
  - 4-bits. min 5, max 15 (15x4=60 bytes max)
- Differentiated Services
  - previously TOS, left unused for years
  - 6 bits, now utilized in Differentiated Services
  - 2 bits, used for congestion notification

# IP Version 4 Protocol

- Total length:
  - full packet length (header + data)
  - up to 65,535 bytes
- Identification
  - for fragmented datagrams
  - identify which packet fragment belongs to
- DF: Don't Fragment
  - might need to use less efficient route
- MF: More Fragments on the way

# IP Version 4 Protocol

- Fragment offset
  - fragment position within a datagram
  - fragments must be multiples of 8 in size
  - 13 bits, max 8192 fragments (x8=65536)
- Time To Live (TTL)
  - limit packet lifetime
  - count time in seconds (max 255 sec)
  - in practice 255 hops
  - at 0, packet discarded, warning is sent



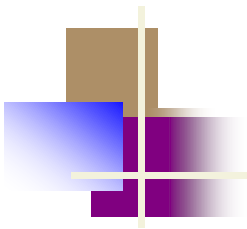
# IP Version 4 Protocol

- Protocol
  - which transport process
  - TCP, UDP, ICMP, …etc
  - protocol numbers assigned by IANA
- Header checksum
  - verifies header only
  - add all 16-bit halfwords using 1's compliment
  - take 1's compliment of the result
  - initially set to zero
  - calculated at each hop (TTL changes)

# IP Version 4 Protocol

| Option | Description |
|---|---|
| Security | Specifies how secret the datagram is |
| Strict source routing | Gives the complete path to be followed |
| Loose source routing | Gives a list of routers not to be missed |
| Record route | Makes each router append its IP address |
| Timestamp | Makes each router append its address and timestamp |

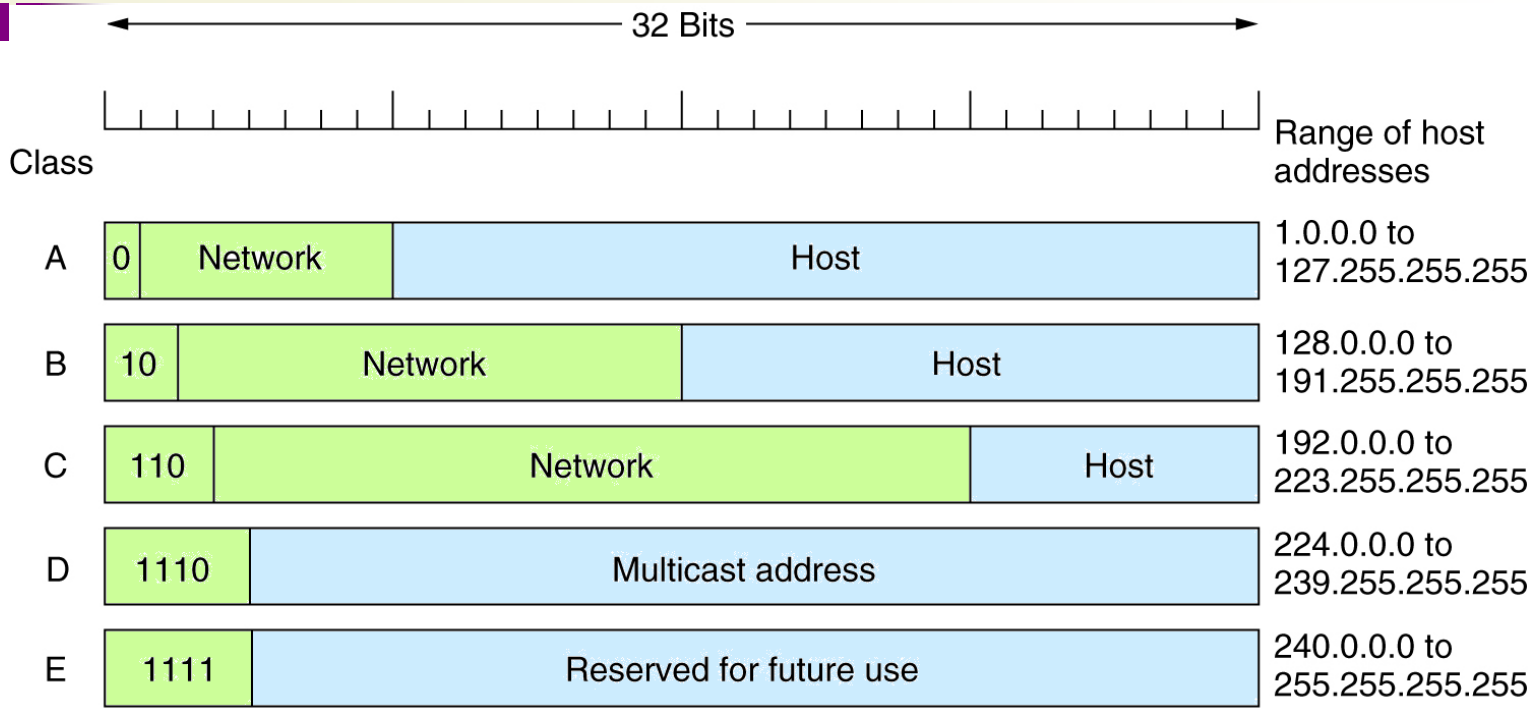
- Options
  - provides space for experimental and future use
  - many routers ignore them
  - only partly supported, rarely used



# IP Addresses

- IP address is 32 bit
- Stored in S.Addr, D.Addr fields in IP header
- Refers to network interface, not host
- Hierarchical, unlike Ethernet addresses
- Consists of network and host portions
  - network (prefix): same for all hosts in network
  - contiguous block of IP address space
- Dotted decimal notation: e.g. 128.208.2.151

# IP Addresses



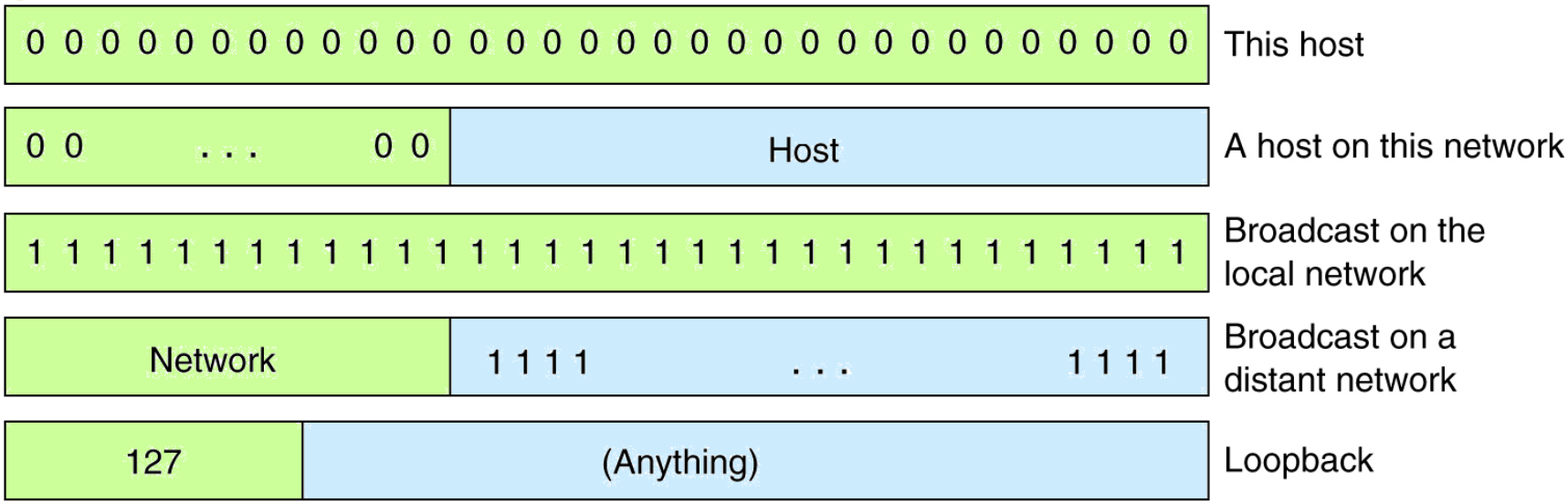

| Class | Number of Networks | Number of Hosts/ Network |
|---|---|---|
| A | ~ 128 | ~ 16 million |
| B | ~ 16 thousands | ~ 64 thousands |
| C | ~ 2 million | ~ 256 |

# IP Addresses

| Class | First Octet Range | Valid Network Numbers | Total Number of This Class of Network | Number of Hosts per Network |
|---|---|---|---|---|
| A | 1 to 126 | 1.0.0.0 to 126.0.0.0 | $2^7 - 2$ | $2^{24} - 2$ |
| B | 128 to 191 | 128.1.0.0 to 191.254.0.0 | $2^{14} - 2$ | $2^{16} - 2$ |
| C | 192 to 223 | 192.0.1.0 to 223.255.254.0 | $2^{21} - 2$ | $2^8 - 2$ |

*The Valid Network Numbers column shows actual network numbers. There are several reserved cases. For example, networks 0.0.0.0 (originally defined for use as a broadcast address) and 127.0.0.0 (still available for use as the loopback address) are reserved. Networks 128.0.0.0, 191.255.0.0, 192.0.0.0, and 223.255.255.0 also are reserved.

# IP Addresses




| Address | Meaning |
|---|---|
| 0.0.0.0 | This host |
| 255.255.255.255 | Broadcast on local network |
| 127.0.0.1 | Loopback, processed locally, not put onto wire |

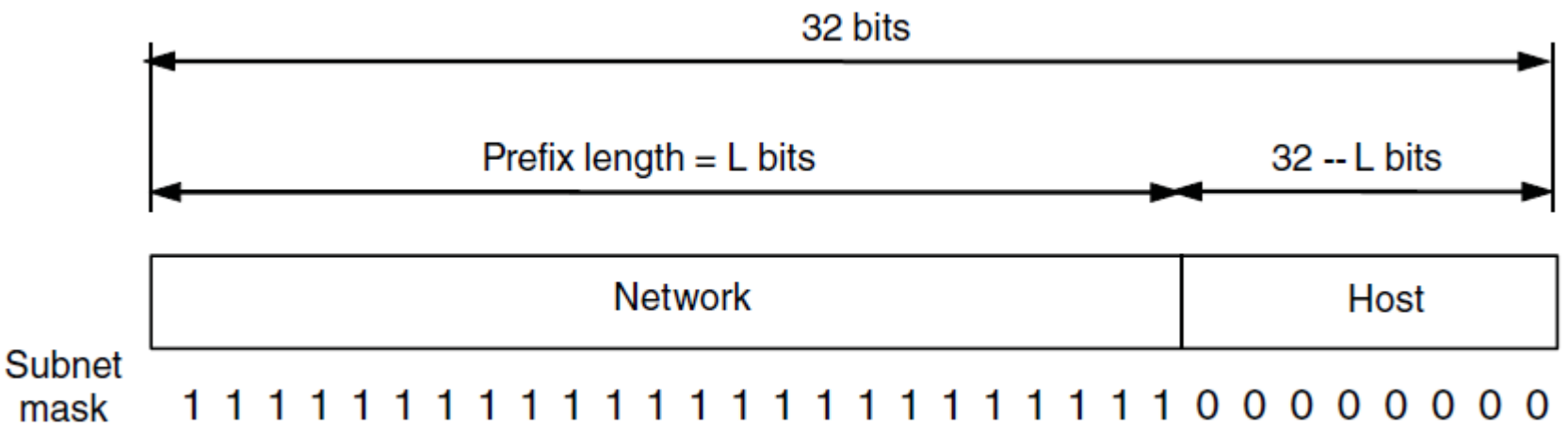
# IP Routing Process

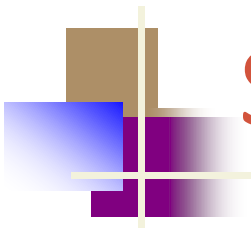
- When packet arrives, look up dest addr
  - distant network?
    →forward to next router on the interface given in routing table
  - local network?
    →send immediately to destination
  - not in the routing table?
    →forward to default gateway

# Subnets

Addresses are allocated in blocks called prefixes

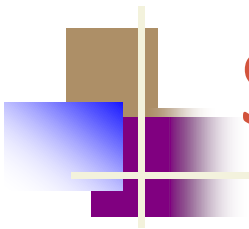
- Prefix is determined by the network portion
- Has $2^L$ addresses aligned on $2^L$ boundary
- Written address/length, e.g., 18.0.31.0/24





# Subnets

- Hierarchical addressing both good and bad
- Advantages
  - routers forward packet based on network part
  - routing table size reduced significantly
- Disadvantages
  - IP belong to specific network, depend on location
  - to keep address when moving, use Mobile IP
  - waste address space if not properly managed

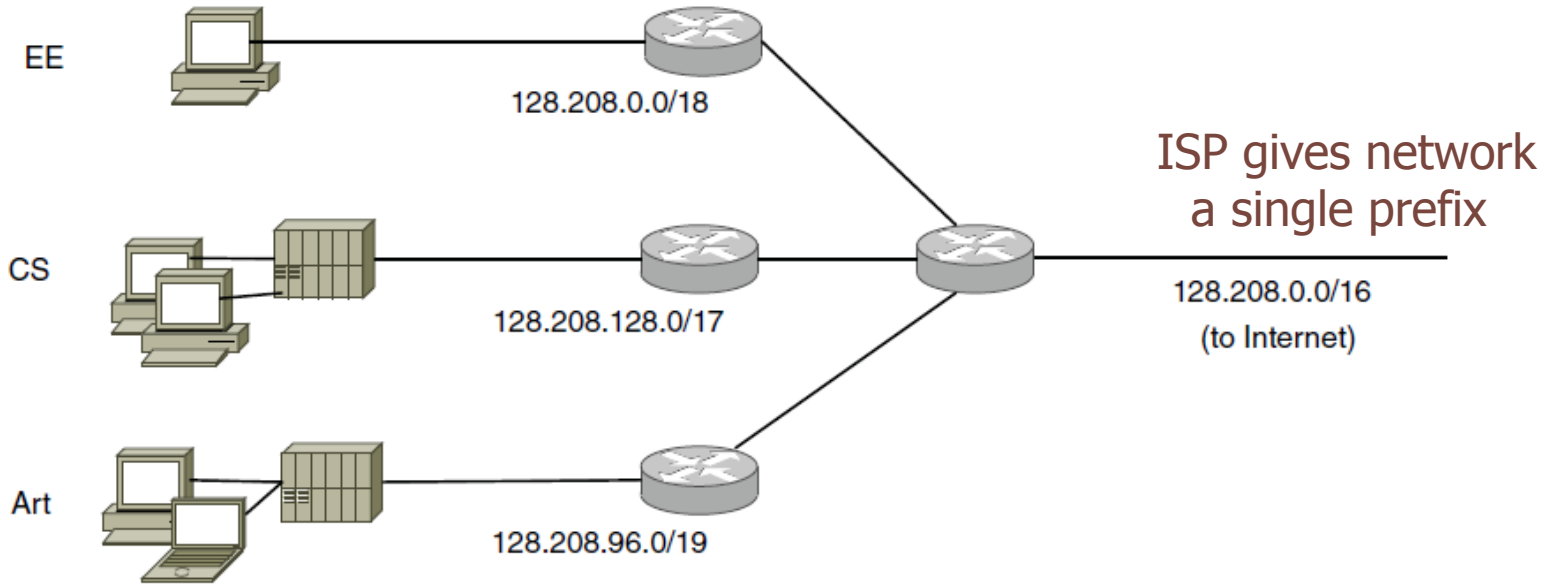


# Subnets

- Network numbers assigned by ICANN;
- Delegated parts to regional authorities
- Routing by prefix requires all hosts in network to have same network number
- Can cause problems as networks grow
- Suppose univ started with a /16 for CS dept
- Year later, EE, Art depts want access
- Requesting more blocks expensive
- /16 has enough > 60,000 hosts

# Subnets

- Subnetting allows network to be split into multiple parts (subnets) internally
- Looks like a single prefix outside the network




Network divides it into subnets internally

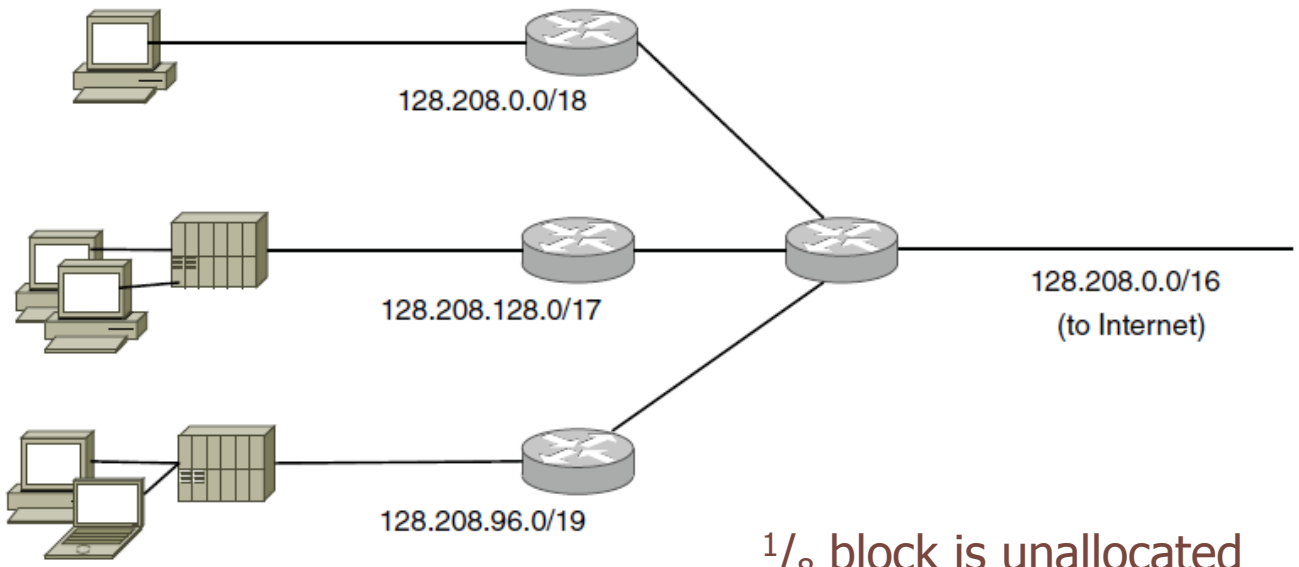
# Subnets

| Dept | Network Address | | | | |
|---|---|---|---|---|---|
| EE | 10000000 | 11010000 | 00\|xxxxxx | xxxxxxxx | 128.208.0.0 |
| CS | 10000000 | 11010000 | 1\|xxxxxxx | xxxxxxxx | 128.208.128.0 |
| Art | 10000000 | 11010000 | 011\|xxxxx | xxxxxxxx | 128.208.96.0 |

¼ block

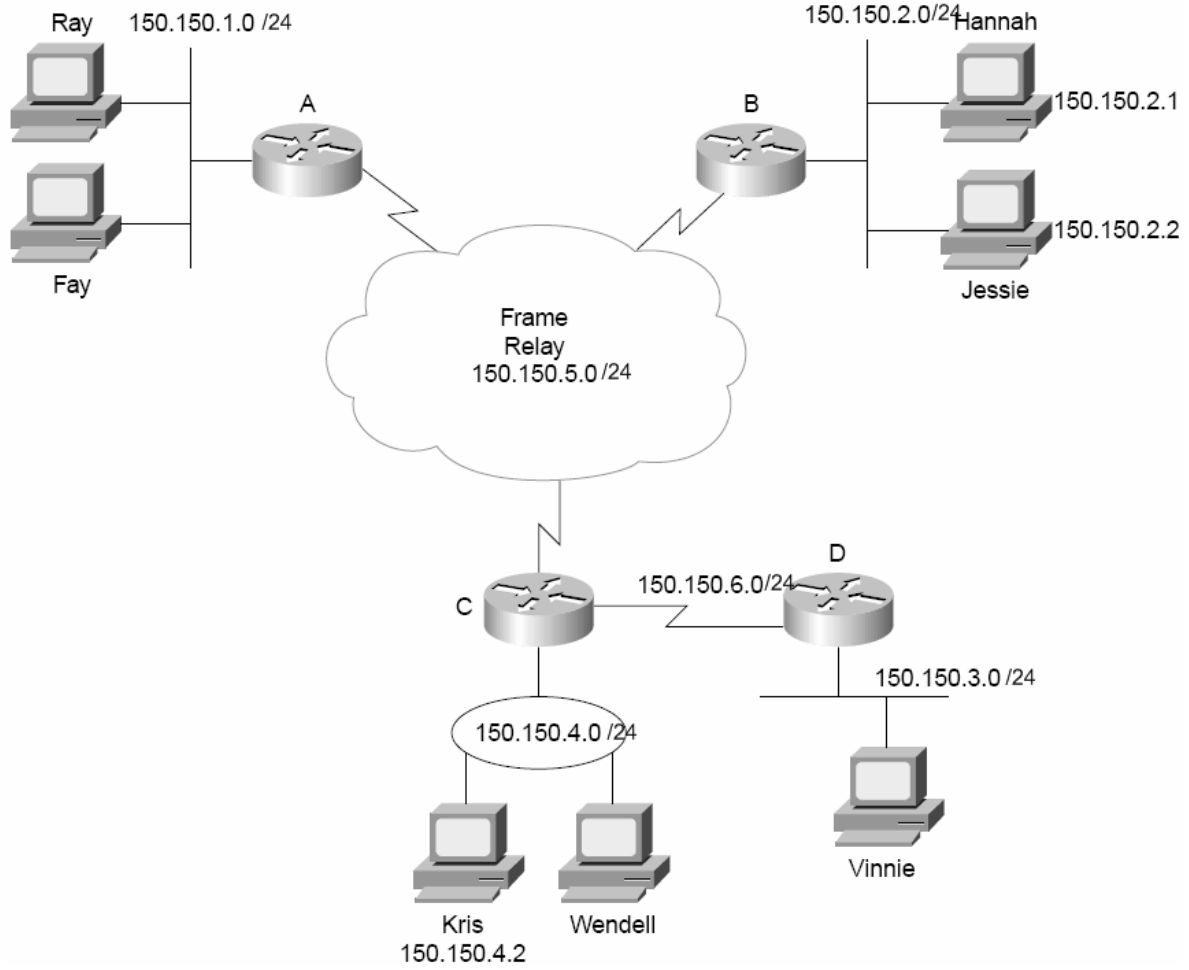
½ block

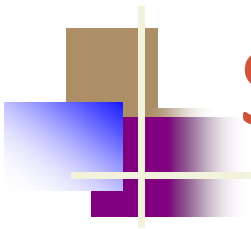
$^1/_8$ block




$^1/_8$ block is unallocated

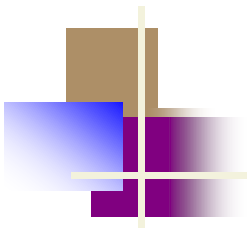
# Subnets


Ray
150.150.1.0 /24
A
Fay
150.150.2.0/24
Hannah
B
150.150.2.1
150.150.2.2
Jessie
Frame
Relay
150.150.5.0 /24
C
150.150.6.0/24
D
150.150.3.0 /24
150.150.4.0 /24
Kris
150.150.4.2
Wendell
Vinnie



# Subnets

- When packet arrives, which subnet to go to?
- Entry for each of 65,536 addresses? No!
- To identify the subnet, need **subnet mask**
  - Identifies the network part of the IP address
  - Written as (for 24 bits mask)
  - e.g. 255.255.255.0 or /24
- When packet arrives
  - AND dest addr with mask of each subnet
  - check if result is the corresponding prefix

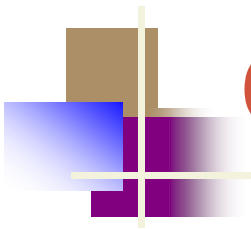


# Example

- Dest IP: 128.208.2.151
  - is CS? AND with 255.255.128.0 (first 17 bits=1)
  - 128.208.0.0 ≠ 128.208.128.0, so Not for CS
  - is EE? AND with 255.255.192.0 (first 18 bits=1)
  - 128.208.0.0 = prefix of EE, so forward to EE

# Classless Inter-Domain Routing

- Classful addressing wastes IP addresses
- Most organizations needs more than class C but less than class B
- CIDR allocates IP address blocks of variable size without regard to classes
- Example: site needs 2000 addresses → assign a block of 2048 addresses



# Classless Inter-Domain Routing

- Address lookup is more complicated
- With classful addressing, routing table is sorted and indexed according to class number
- With CIDR, routing table is sequentially scanned for a match
- Complex algorithms have been developed to speed up address matching

# Example

- University with a class B network 130.50.0.0
- They have 35 departments
- Instead of using 16 bit for host, use 6 bits for subnet and 10 bits for host ( $\lceil \log_2 (35) \rceil = 6$)
- This allows 64 subnets with 1022 hosts per subnet (2 IPs are reserved)
- Subnet mask
  - 11111111.11111111.111111|00.00000000
  - 255.255.252.0
  - /22

# Example

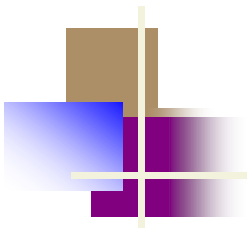
- First subnet network address
  - 10000010 00110010 000000|00 00000000
  - 130.50.0.0
- First usable IP in first subnet
  - 10000010 00110010 000000|00 00000001
  - 130.50.0.1
- Last usable IP in first subnet
  - 10000010 00110010 000000|11 11111110
  - 130.50.3.254
- Broadcast IP in first subnet
  - 10000010 00110010 000000|11 11111111
  - 130.50.3.255

# Example

- Second subnet network address
  - 10000010 00110010 000001|00 00000000
  - 130.50.4.0
- First usable IP in second subnet
  - 10000010 00110010 000001|00 00000001
  - 130.50.4.1
- Last usable IP in second subnet
  - 10000010 00110010 000001|11 11111110
  - 130.50.7.254
- Broadcast IP in second subnet
  - 10000010 00110010 000001|11 11111111
  - 130.50.7.255

# Example

- Third subnet network address
  - 10000010 00110010 000010|00 00000000
  - 130.50.8.0
- First usable IP in third subnet
  - 10000010 00110010 000010|00 00000001
  - 130.50.8.1
- Last usable IP in third subnet
  - 10000010 00110010 000010|11 11111110
  - 130.50.11.254
- Broadcast IP in third subnet
  - 10000010 00110010 000010|11 11111111
  - 130.50.11.255



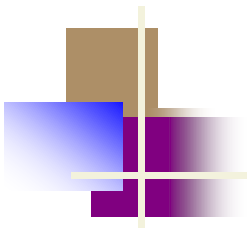
# Example

- Destination address: 130.50.10.6
- Subnet mask 255.255.252.0
- Boolean AND gives: 130.50.8.0
- This address is looked up in routing table
- Destination is in third subnet

# Example

| University | First address | Last address | How many | Written as |
|---|---|---|---|---|
| Cambridge | 194.24.0.0 | 194.24.7.255 | 2048 | 194.24.0.0/21 |
| Edinburgh | 194.24.8.0 | 194.24.11.255 | 1024 | 194.24.8.0/22 |
| (Available) | 194.24.12.0 | 194.24.15.255 | 1024 | 194.24.12/22 |
| Oxford | 194.24.16.0 | 194.24.31.255 | 4096 | 194.24.16.0/20 |

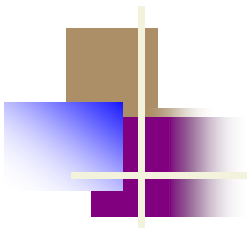
| | Address | Mask |
|---|---|---|
| C | 11000010 00011000 00000000 00000000<br>192 . 24 . 0 . 0 | 11111111 11111111 11111000 00000000<br>255 . 255 . 248 . 0 |
| E | 11000010 00011000 00001000 00000000<br>192 . 24 . 8 . 0 | 11111111 11111111 11111100 00000000<br>255 . 255 . 252 . 0 |
| O | 11000010 00011000 00010000 00000000<br>192 . 24 . 16 . 0 | 11111111 11111111 11110000 00000000<br>255 . 255 . 240 . 0 |



# Example

What happens when a packet comes with destination address 194.24.17.4?

- Binary address
  11000010 00011000 00010001 00000100
- AND with Cambridge mask yields
  11000010 00011000 00010000 00000000
  →doesn't match Cambridge's base address
- AND with Edinburgh mask yields
  11000010 00011000 00010000 00000000
  →doesn't match Edinburgh's base address



# Example

- AND with Oxford mask yields
  11000010 00011000 00010000 00000000
  →matches Oxford's base address!
- If no other matches, use Oxford's entry
- If another match found, use the match with the longest mask

# Route Aggregation

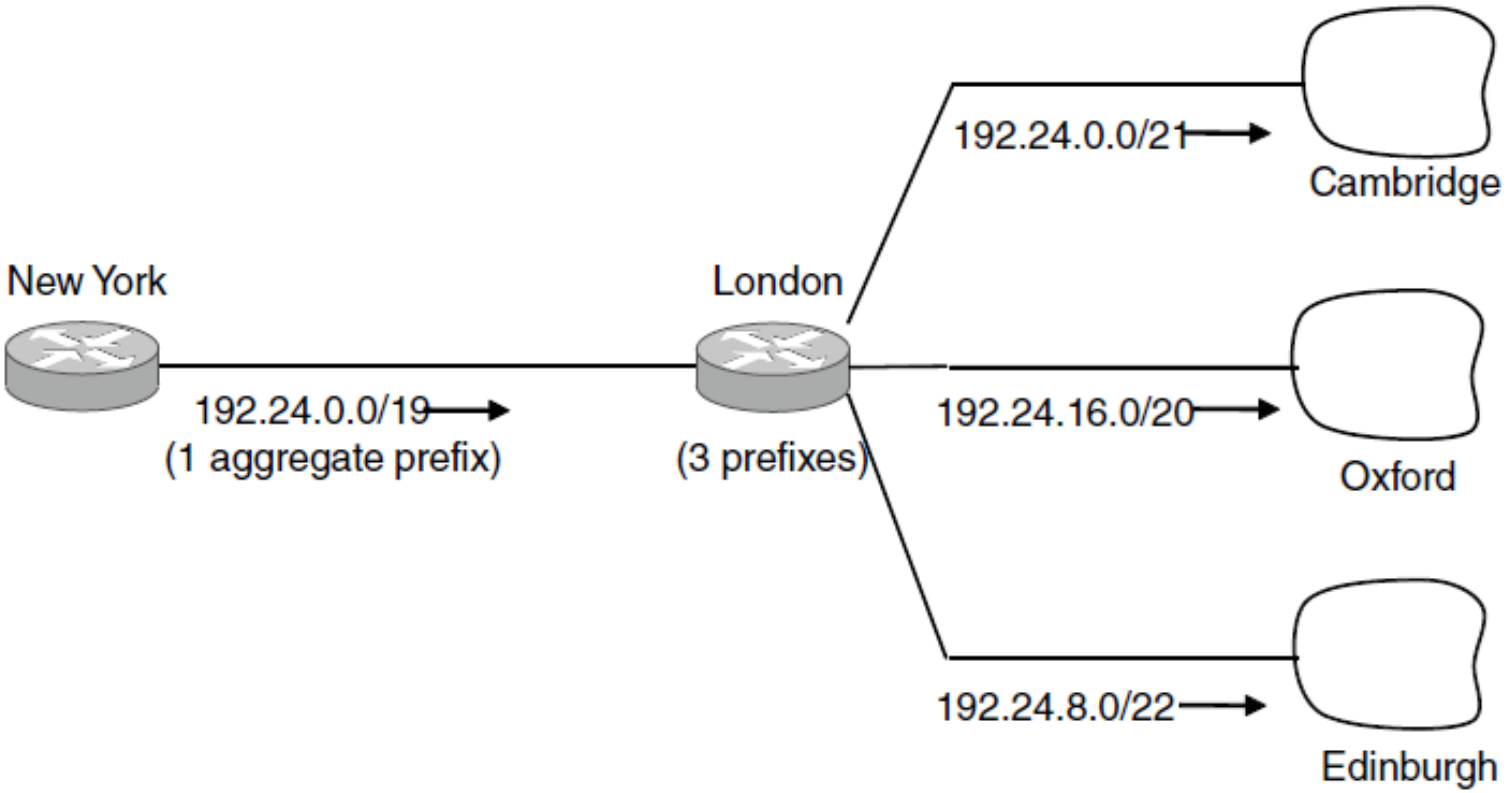
- Same outgoing line for multiple contiguous entries can be aggregated
- Single **supernet** entry
- Example, previous three subnets:
  194.24.0.0/19
  11000010 00011000 00000000 00000000
  Mask (prefix)
  11111111 11111111 11100000 00000000
- Helps reducing routing table sizes

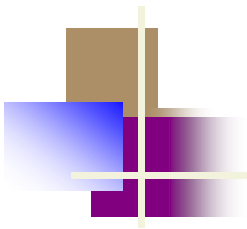
# Route Aggregation

## To determine the supernet

- Convert network IP address to binary
- Find common bit pattern from left
  - 11000010 00011000 00000000 00000000
  - 11000010 00011000 00001000 00000000
  - 11000010 00011000 00010000 00000000
- Count of bits = prefix: /19
- Fill other bits with zeros = supernet address
  - 11000010 00011000 00000000 00000000
  - 192.24.0.0

# Route Aggregation

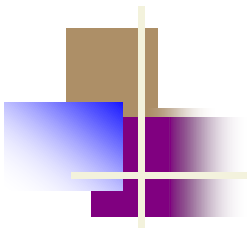




# Network Address Translation

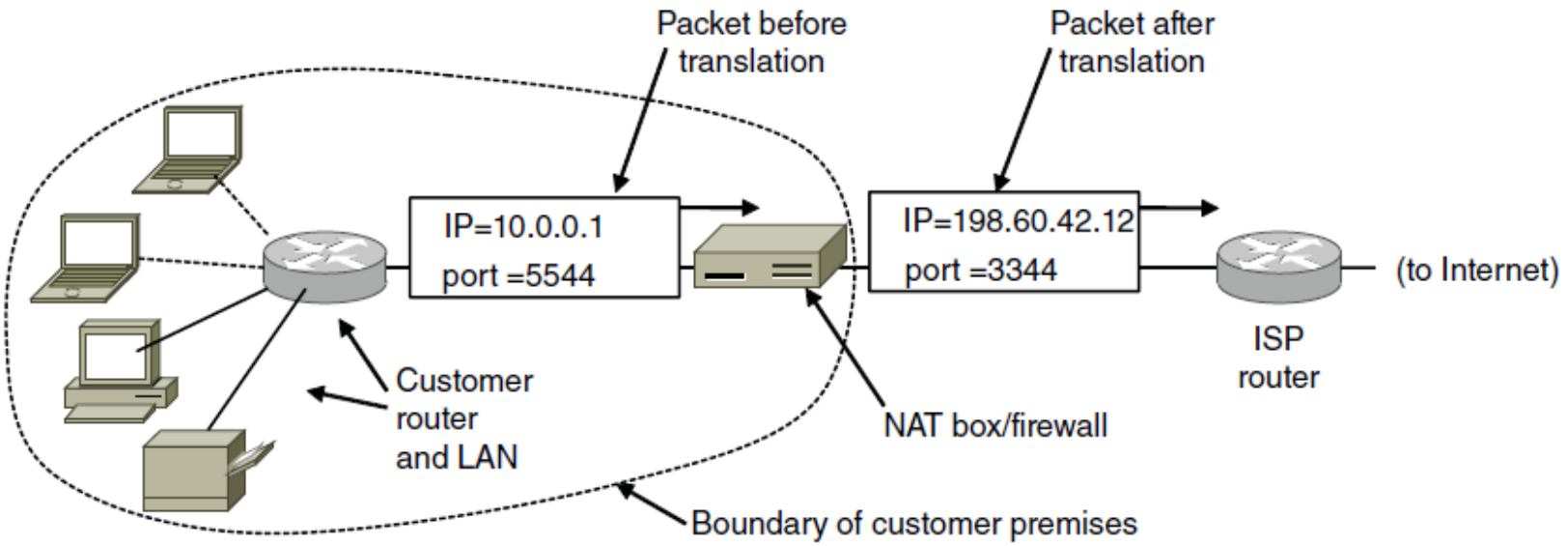
- IP addresses are scarce
- Must be used efficiently
- Assign addresses dynamically?
  - doesn't work with machines need 24/7 up
  - DSL users keep some PCs on all the time
- Home LANs now like small business LANs
- Long term solution? use IPv6
- Quick fix? use NAT

# Network Address Translation



- Assign each company 1 or few public IPs
- Used for Internet traffic
- Within LAN, use IP from private range:
  - 10.0.0.0 – 10.255.255.255/8 (~16M hosts)
  - 172.16.0.0 – 172.31.255.255/12 (~1M hosts)
  - 192.168.0.0 – 192.168.255.255/16 (~65K hosts)
- At exit router, translate to unique public IP

# Network Address Translation



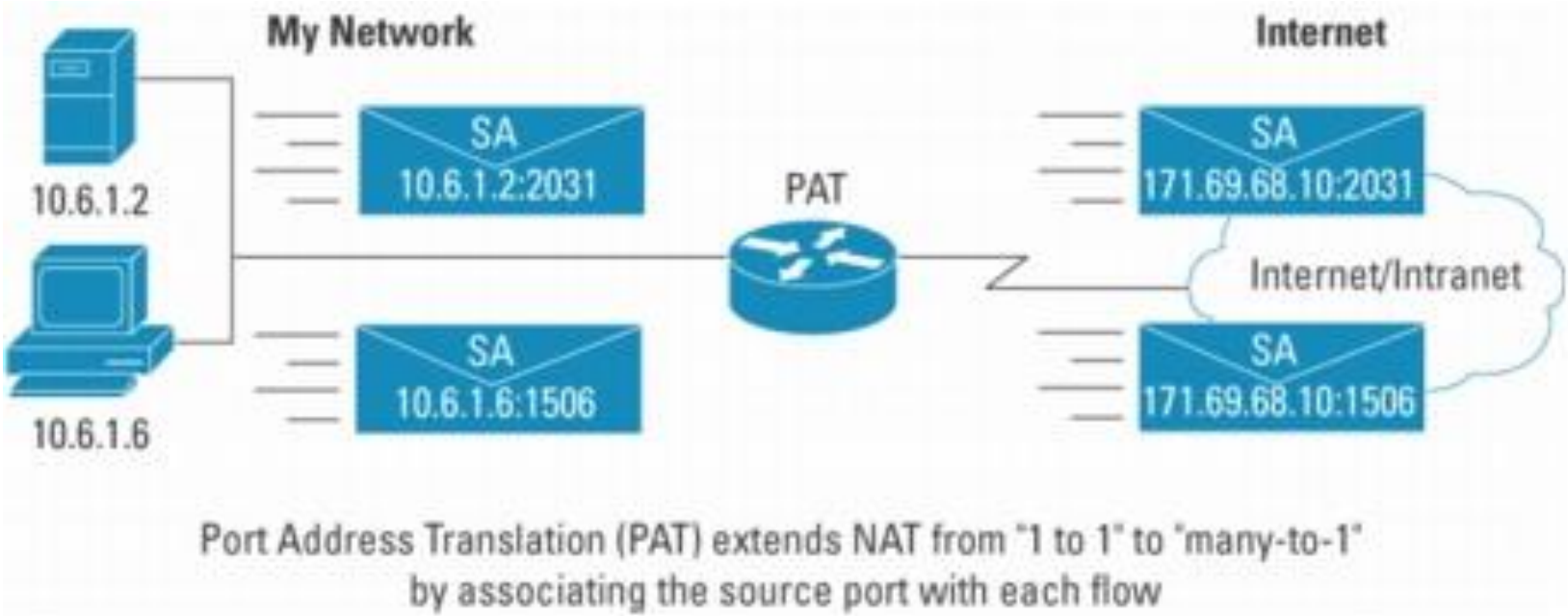

- Within LAN, each host has IP 10.x.y.z
- Before leaving LAN, pass thru NAT box
- Convert IP, e.g. 10.0.0.1 to 198.60.42.12
- NAT box can be combined with firewall

# Network Address Translation

How NAT knows where to send reply traffic?

- Most IP packets carry TCP or UDP payloads
- TCP and UDP use source and destination port numbers to distinguish connections
- NAT Box uses the source port number together with the source IP as an index for the translation database (PAT)
- Using one external IP to represent multiple internal IPs (port overloading)

# Network Address Translation




*Source: Cisco*

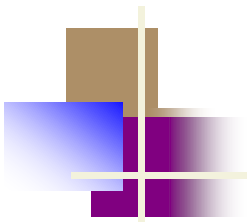
# Network Address Translation

| Pro | Inside Global | Inside Local | Outside Local | Outside Global |
|---|---|---|---|---|
| tcp | 171.69.68.5:1405 | 10.6.15.2:1405 | 204.71.200.69:80 | 204.71.200.69:80 |

PAT (Port Address Translation) includes ports in addition to IP addresses

- Many-to-one translation
- Maps multiple IP addresses to 1 or a few IP addresses
- Unique source port number identifies each session
- Conserves registered IP addresses
- Also called NAPT in IETF documents

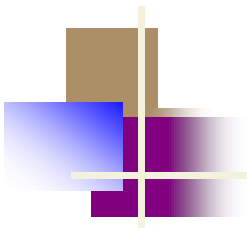
*Source: Cisco*



# Network Address Translation

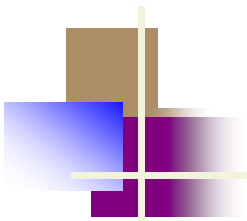
## Problems with NAT

- Violates IP architectural model: unique addresses
- Breaks end-to-end connectivity
- Changes Internet from connectionless to a kind of connection-oriented network
- Violates protocol layering (depend on TCP/UDP)
- Doesn't work with all transport protocols
- Some applications insert IP in the payload or the body of the text
- Can only map 65,536 machines to a single IP
- Delays the upgrade to IPv6



# IP Version 6

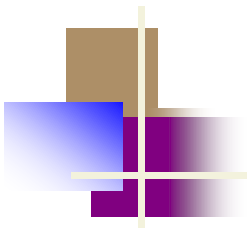
- NAT and CIDR helped conserve IP addresses
- However, IPv4 addresses almost exhausted
- IPv4 originally designed for univs , govt
- Now, different types of users, requirements
- IPv6 designed to solve these problems
- Standardized in 1998
- Yet, only 1% deployment in Internet



# IP Version 6

## Design Goals

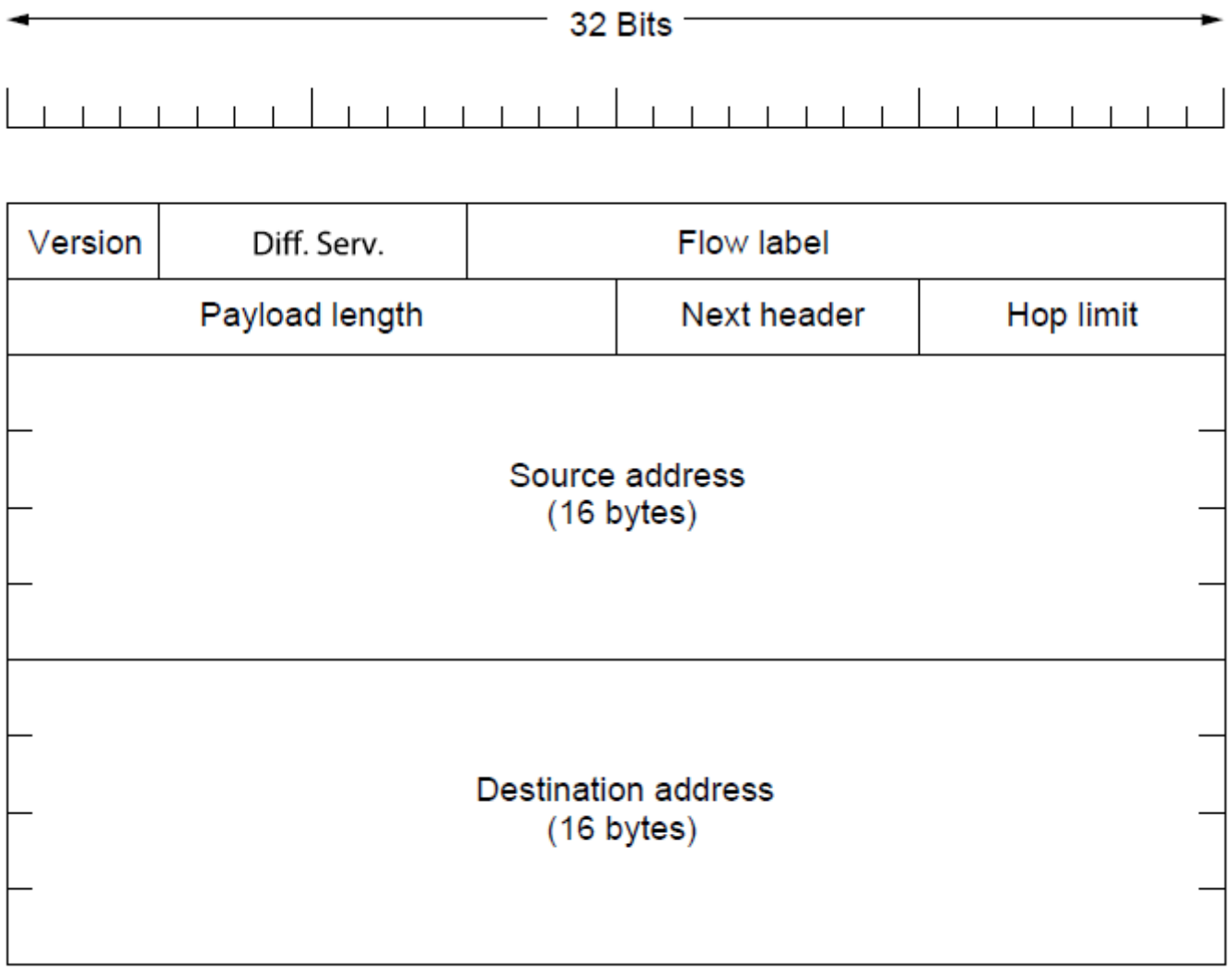
- Support billions of hosts, even inefficient allocation
- Reduce the size of the routing tables
- Simplify protocol, allow faster packets processing
- Provide better security (authentication and privacy)
- More attention to the type of service, real-time data
- Aid multicasting by allowing scopes to be specified
- Allow host to roam without changing its address
- Allow the protocol to evolve in the future
- Permit old and new protocols to coexist for years



# IP Version 6

- Major improvements
  - longer addresses (128-bit): $7 \times 10^{23}$ IP/m$^2$
  - simplification of header: 7 fields instead of 13
  - better support for options: speed proc time
  - improved security: authentication, privacy

# Main IPv6 Header



# Main IPv6 Header

- Version
  - 4 for IPv4, 6 for IPv6
  - need to be examined during transition period
- Differentiated services
  - distinguish class of service
  - low 2 bits used to signal congestion
- Flow label
  - packets having same requirements
  - within source-destination

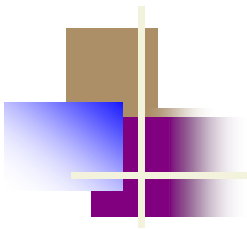
# Main IPv6 Header

- Payload length
  - how many bytes follow 40-byte header
  - header bytes not counted (unlike IPv4)
  - max length: 65,535 (was 65,515 in IPv4)
- Next header
  - can have additional optional extension headers
  - allow main header simplification
  - in last header, tells which transport protocol
- Hop count
  - keep packet from living forever
  - works same as IPv4, name reflects actual behavior

# Main IPv6 Header

## Source address and Destination address

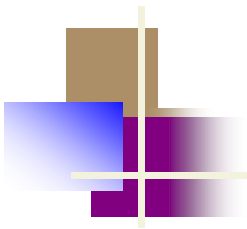
- 16 byte addresses were best compromise
- Notation:
  - 8 groups of hex digits separated by ":", e.g.:
  - 8000:0000:0000:0000:0123:4567:89AB:CDEF
  - leading 0s can be omitted
  - 0000 groups can be replaced by ":"
  - 8000::123:4567:89AB:CDEF
  - IPv4 address written as: ::192.31.20.46



# Main IPv6 Header

## Source address and Destination address

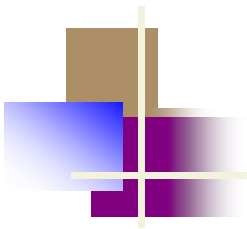
- If entire earth covered with computers
- IPv6 would allow $7 \times 10^{23}$ IP addresses/$m^2$
- In practice, addresses not used efficiently
- Most pessimistic: 1000 IP address/$m^2$



# Main IPv6 Header

## Comparison with IPv4 header

- IHL field removed
  - IPv6 header has fixed length
- Protocol field removed
  - next header field tells what after last IP header



# Main IPv6 Header

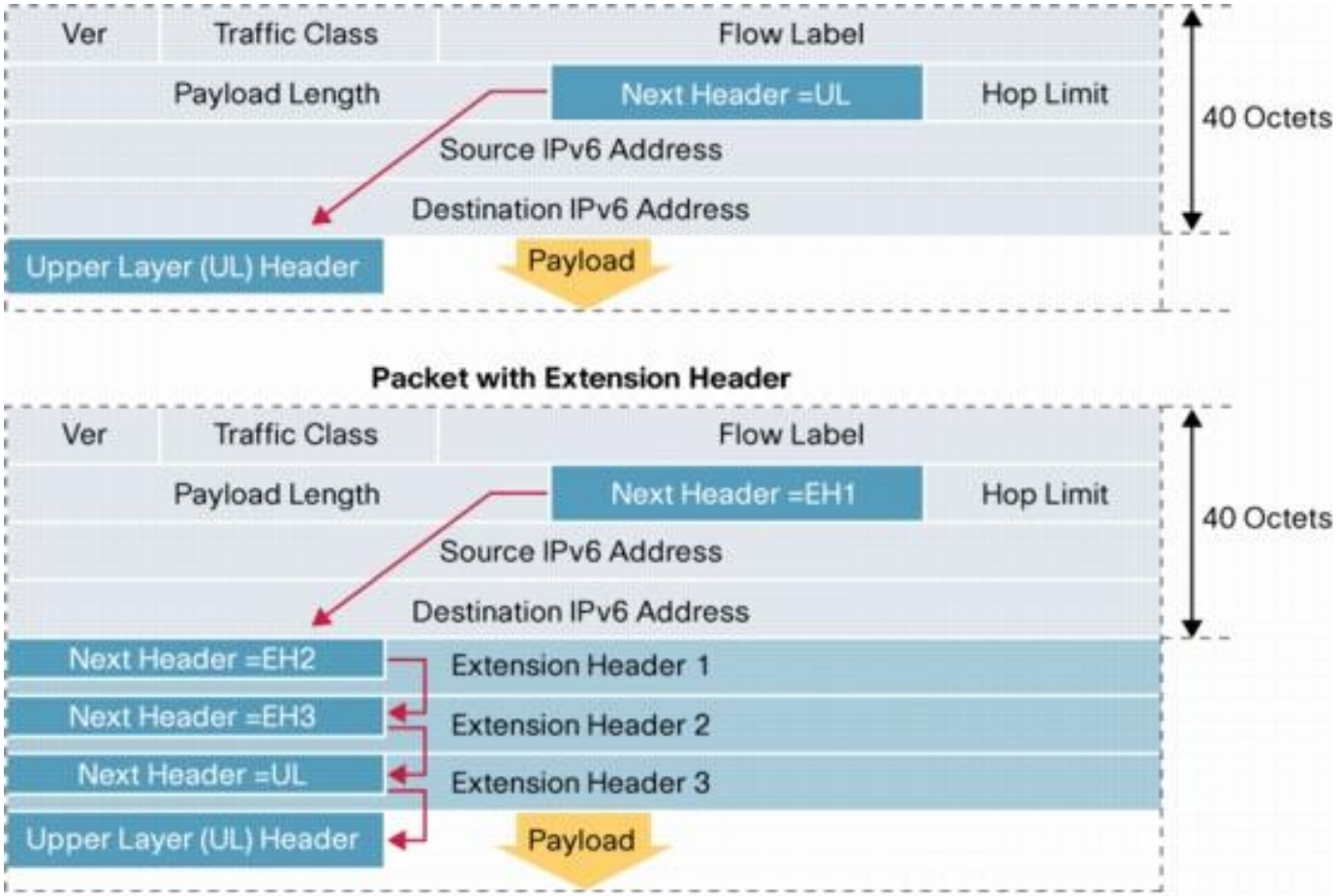
- Fragmentation related fields removed
  - IPv6 host dynamically determine packet size
  - using path MTU discovery
  - packet too large? router discard, send error msg
  - only source can fragment packets
- Checksum field removed
  - calculation reduces performance
  - networks are now reliable
  - data link, transport layers do own checksums

# Extension Headers

- Some missing IPv4 functions occasionally needed
- extension headers provide extra information
- encoded in an efficient way
- if present, must appear after fixed header

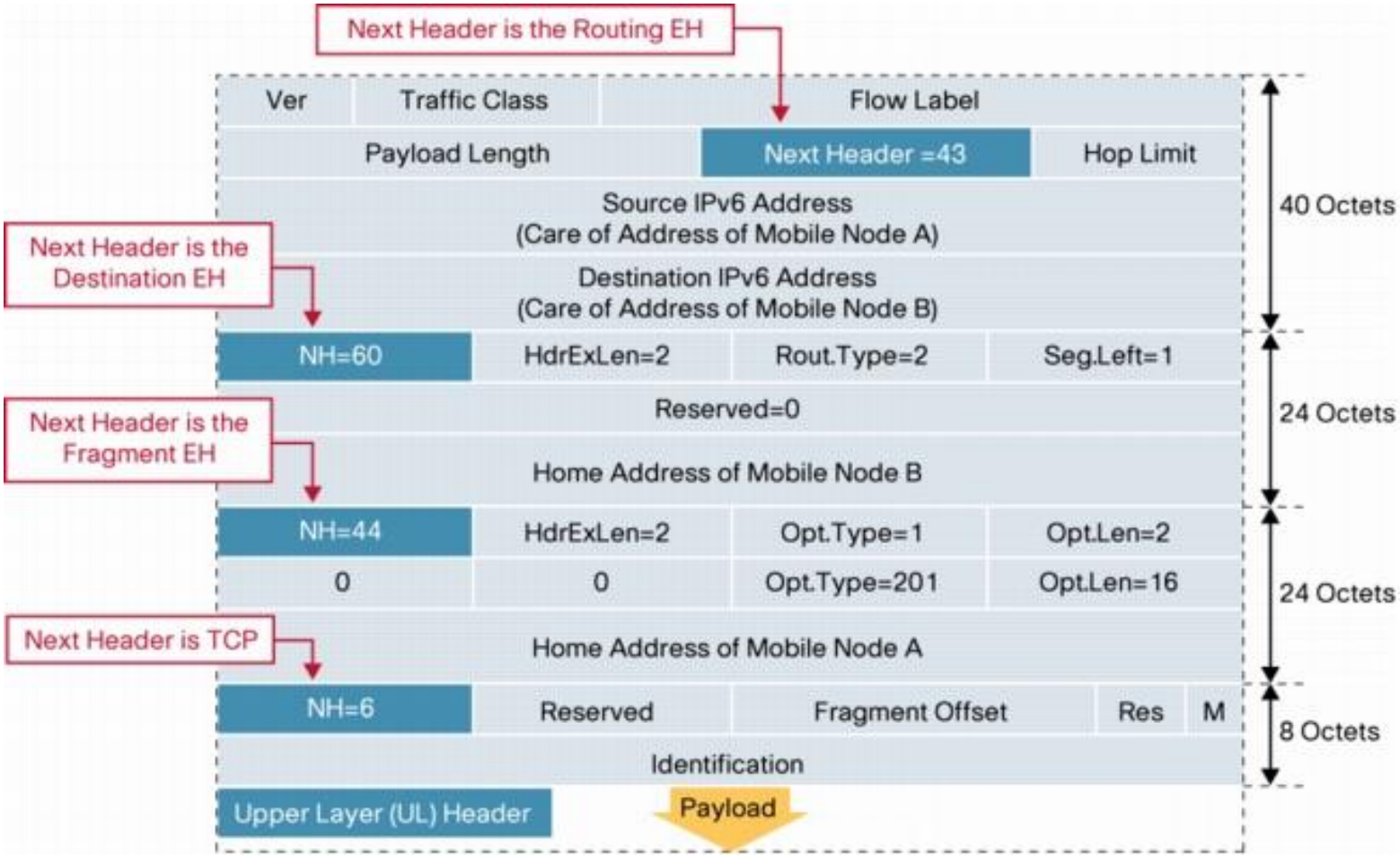
| Extension header | Description |
|---|---|
| Hop-by-hop options | Miscellaneous information for routers |
| Destination options | Additional information for the destination |
| Routing | Loose list of routers to visit |
| Fragmentation | Management of datagram fragments |
| Authentication | Verification of the sender's identity |
| Encrypted security payload | Information about the encrypted contents |

# Extension Headers




*Source: Cisco*

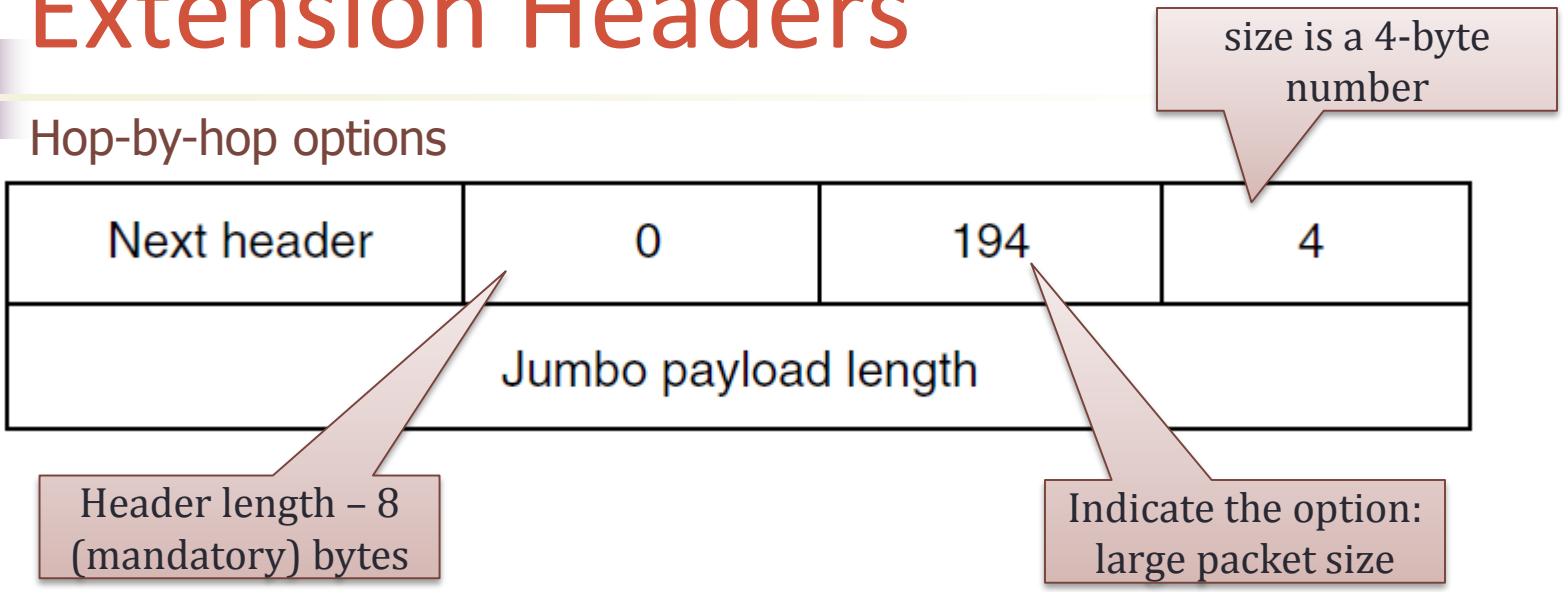
# Extension Headers




*Source: Cisco*

# Extension Headers

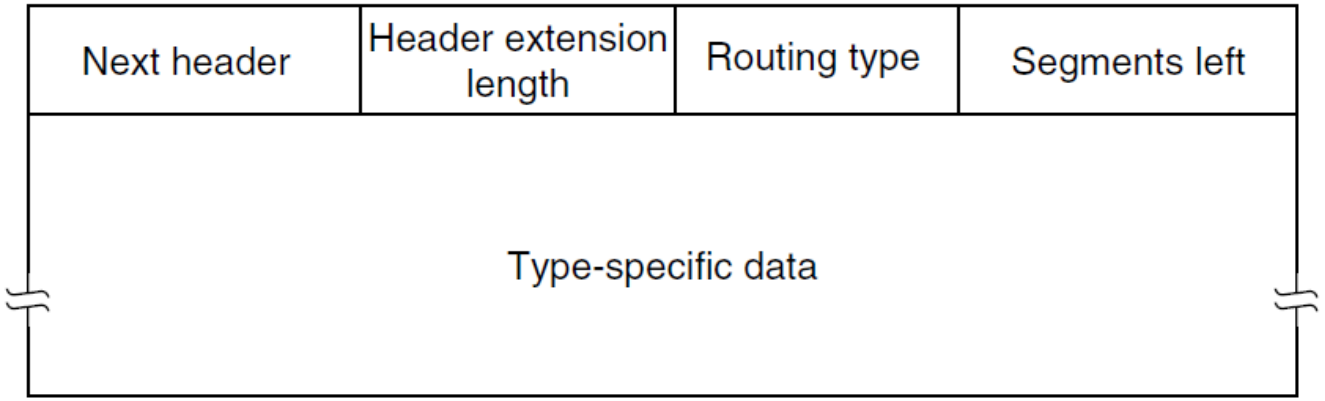
Hop-by-hop options



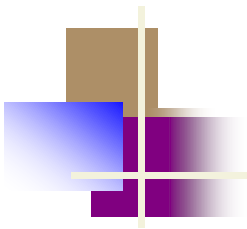

- Information all routers along path must examine
- 1 option defined so far: support for > 64 KB packets
- Size < 65,536 not allowed, discarded, ICMP error
- Datagrams with this EH called **jumbograms**
- For apps that must transfer gigabytes of data

# Extension Headers

Routing

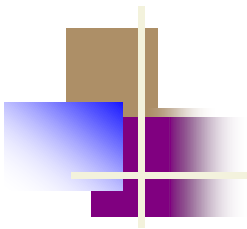



- Lists one or more routers that must be visited
- Similar to IPv4 loose routing option
- Routing type: format for the rest of the header
  - 0 indicates 32-bit word follows 1st word; then
  - number of IPv6 addresses
- Segments left: how many addresses not yet visited
  - decremented every time one is visited



# Extension Headers

- Fragmentation
  - similar to IPv4: ID, frag no., last fragment
  - only source host can fragment packets
  - simplifies work, makes routing faster
- Authentication
  - receiver can be sure who sent packet
- Encrypted security payload
  - possible to encrypt contents of packet
  - only intended recipient can read it



# Internet Control Protocols

- Internet Control Message Protocol (ICMP)
- Address Resolution Protocol (ARP)
- Dynamic Host Configuration Protocol (DHCP)

# Internet Control Message Protocol

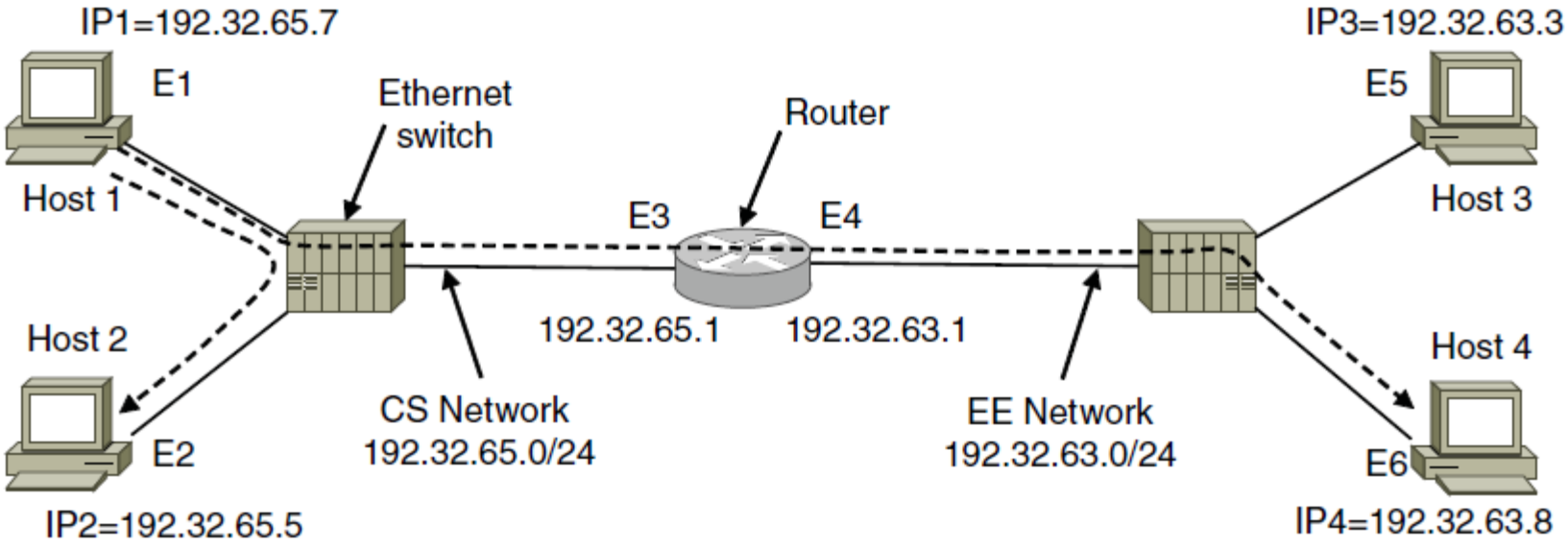
- Used by routers monitor and diagnose network problems
- Most important ICMP message types:

| Message type | Description |
|---|---|
| Destination unreachable | Packet could not be delivered |
| Time exceeded | Time to live field hit 0 |
| Parameter problem | Invalid header field |
| Source quench | Choke packet |
| Redirect | Teach a router about geography |
| Echo request | Ask a machine if it is alive |
| Echo reply | Yes, I am alive |
| Timestamp request | Same as Echo request, but with timestamp |
| Timestamp reply | Same as Echo reply, but with timestamp |

# Address Resolution Protocol

- DLL don't understand IP addresses
- LAN needs MAC address to send frames
- ARP finds MAC address of host using IP
- A broadcast: Who has the IP address XXX?
- Host in question responds with its MAC
- Improvements
  - ARP cache used to improve speed
  - timeout to allow change
  - gratuitous ARP: host announce its own MAC when configured; everybody updates cache

# Address Resolution Protocol



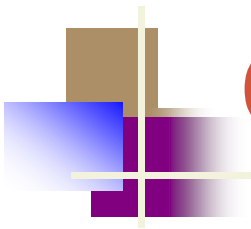

Same LAN

Different LAN

| Frame | Source IP | Source Eth. | Destination IP | Destination Eth. |
|---|---|---|---|---|
| Host 1 to 2, on CS net | IP1 | E1 | IP2 | E2 |
| Host 1 to 4, on CS net | IP1 | E1 | IP4 | E3 |
| Host 1 to 4, on EE net | IP1 | E4 | IP4 | E6 |

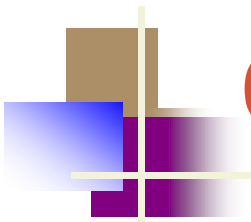
# Dynamic Host Configuration Protocol

- Allows automatic, manual IP assignment
- Every network has DHCP server
- Host broadcasts DHCP DISCOVER packet
- DHCP server responds DHCP OFFER
- IP can be leased for defined time period
- DHCP relay agent is needed on each LAN beyond broadcast domain
- Other provided information
  - subnet mask, DNS, default gateway



# OSPF – Interior Routing Protocol

- Intra-domain routing: inside an AS, org
- Also called interior gateway protocol
- Early protocols used distance vector
  - ARPANET: RIP (Routing Information Protocol)
  - suffers from count-to-infinity problem
- On 1988, IETF started work on link-state
  - OSPF (Open Shortest Path First), standard 1990
  - based on IS-IS (Intermediate System) ISO std.
  - both IS-IS, OSPF widely used, supported



# OSPF – Interior Routing Protocol

## Requirements for OSPF

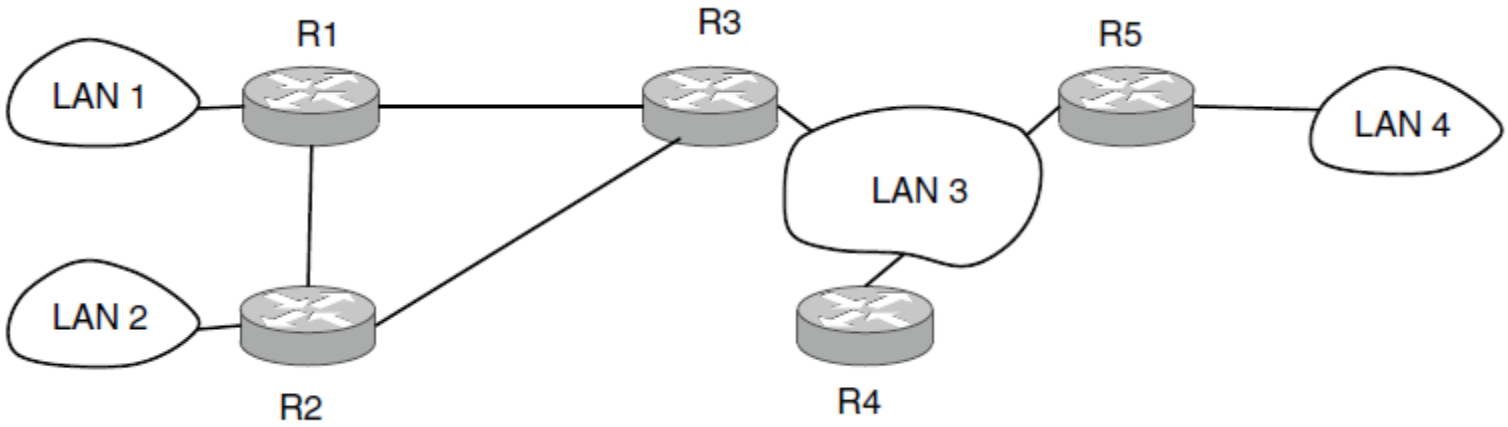
- Open (published), not proprietary
- Support variety of distance metrics
- Dynamic, adapted to changes quickly
- Support routing based on type of service
- Must do load balancing, split multiple paths
- Support hierarchical systems
- Security, prevent spoofing false routing info

# OSPF – Interior Routing Protocol

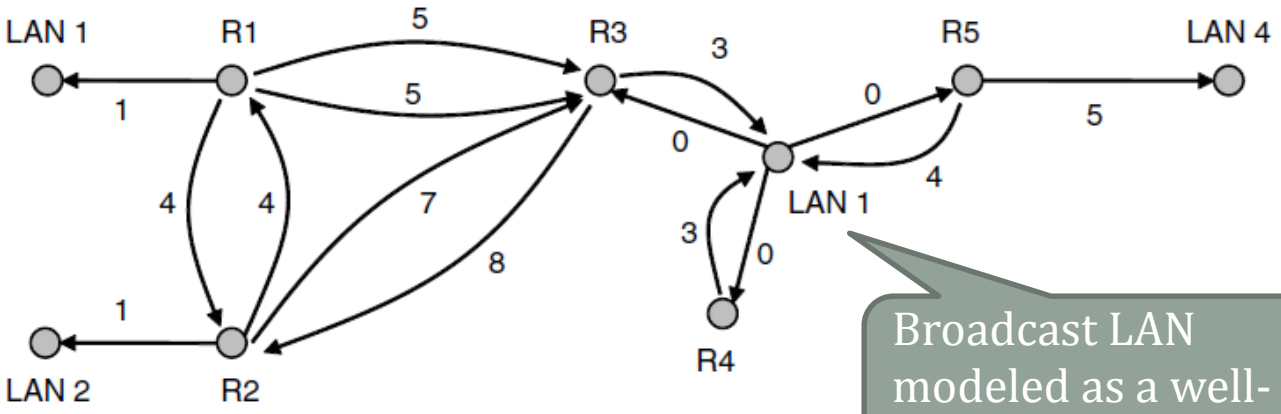
- OSPF abstract collection of actual networks
- Directed graph, arcs with weights
- Point-to-point link
  - pair of arcs ⇆ weights can be different
- Broadcast network
  - node for network itself
  - arcs **from** this node **to** routers have 0 weight
- Networks with hosts only (no routers)
  - one-way arcs, route to hosts, not through them

# OSPF – Interior Routing Protocol

Autonomous System

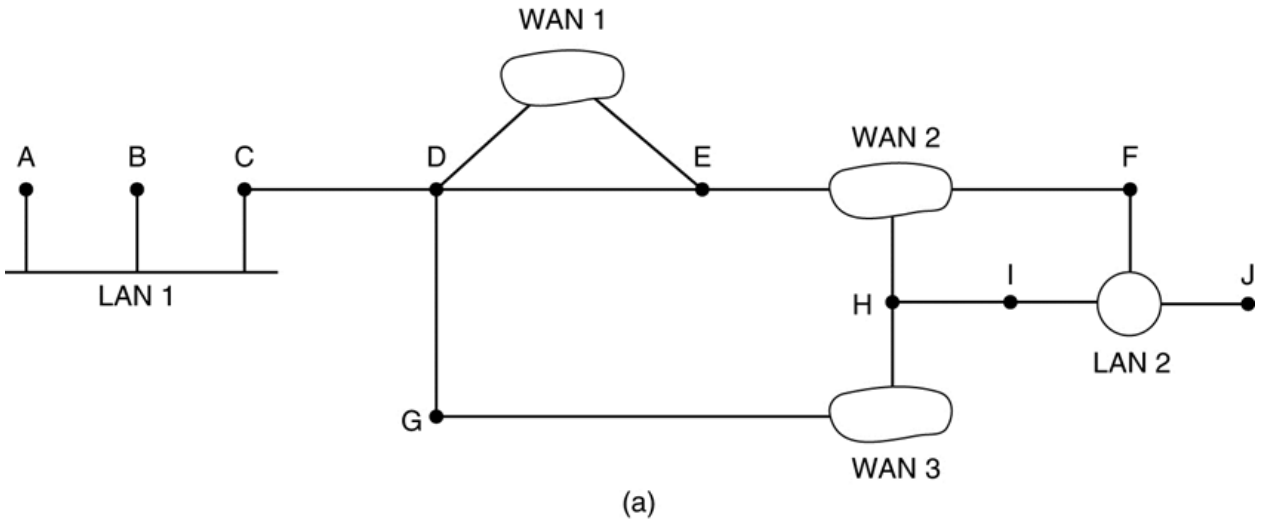



Graph representation for OSPF



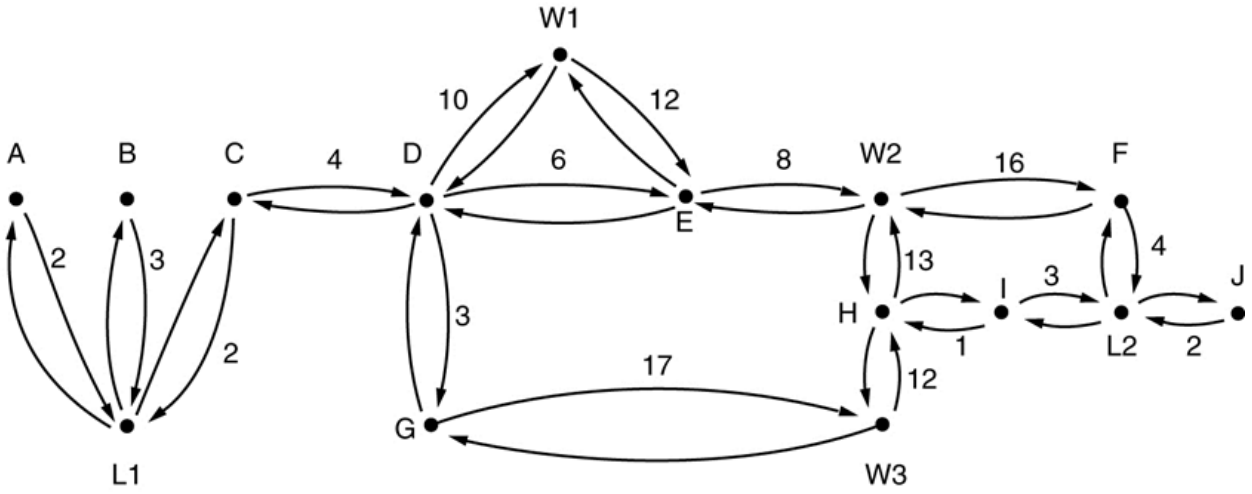
# OSPF – Interior Routing Protocol

Autonomous System

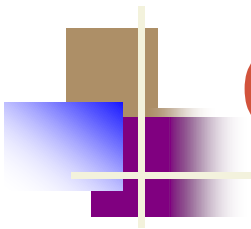


(a)

Graph representation for OSPF



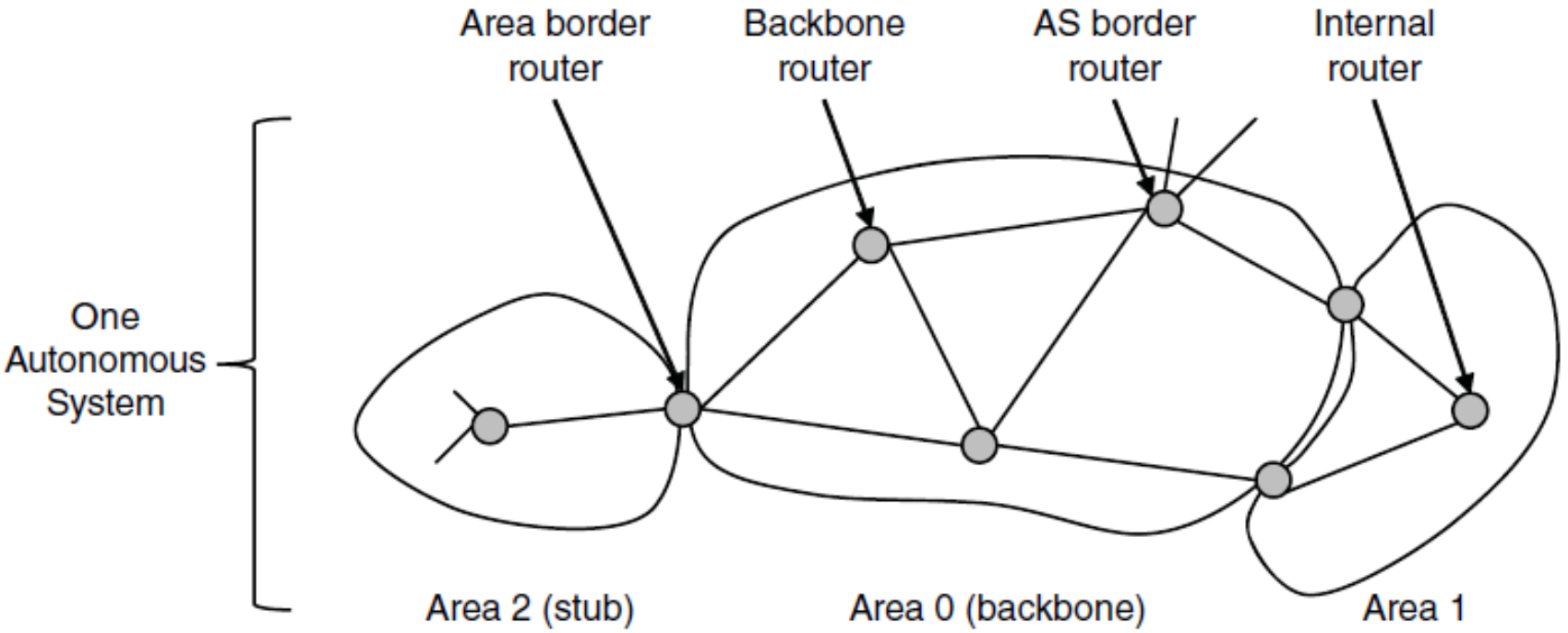
(b)



# OSPF – Interior Routing Protocol

- Many ASes are large, non trivial to manage
- OSPF allows dividing AS to numbered **areas**
- Area: network or contiguous networks
- Areas do not overlap
- **Internal routers**: wholly within an area
- Every AS has **backbone area**: area 0
- Routers in area 0: **backbone routers**
- **Area border router**: connect multiple areas

# OSPF – Interior Routing Protocol




- **AS boundary router**: injects routes to external destinations on other ASes into the area

# OSPF – Interior Routing Protocol

- Each router informs (floods) all other routers in its area of its links to other routers and networks
- Each router then runs Dijkstra to compute routes
- Routers in LAN elect **designated router**
- Avoid having all routers talk to all within LAN

| Message type | Description |
|---|---|
| Hello | Used to discover who the neighbors are |
| Link state update | Provides the sender's costs to its neighbors |
| Link state ack | Acknowledges link state update |
| Database description | Announces which updates the sender has |
| Link state request | Requests information from the partner |

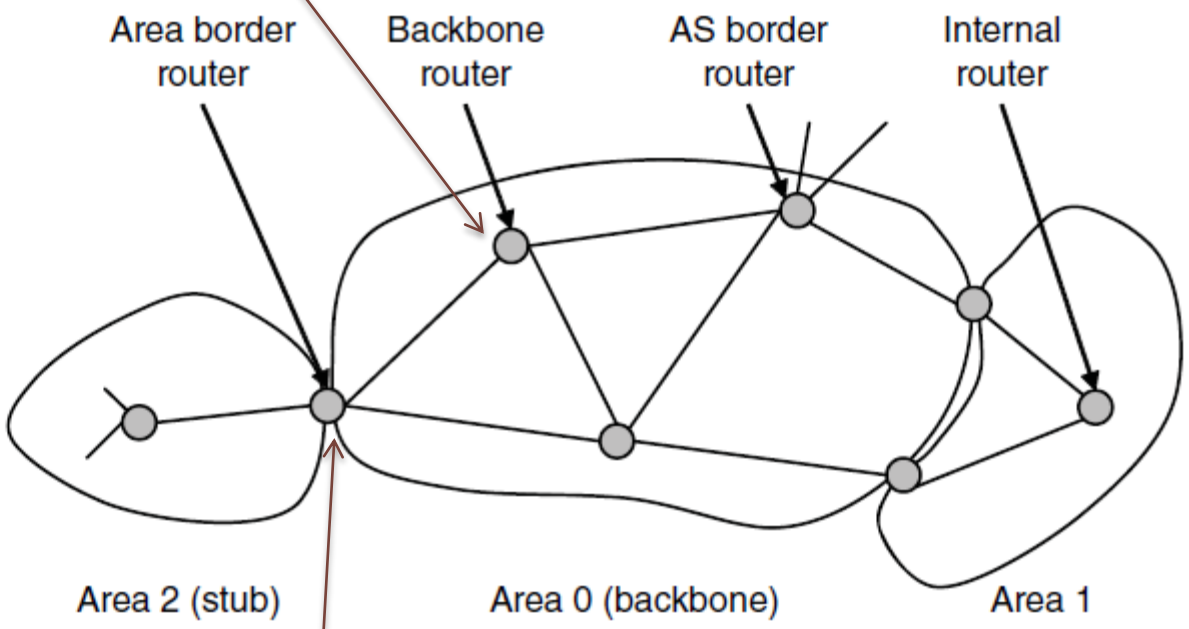
# OSPF – Interior Routing Protocol

Backbone routers accept information from area border routers

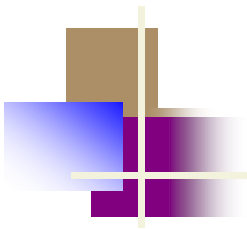
Compute best route from each backbone router to every other router

One Autonomous System

Info is propagated back to area border routers, which advertise it within their areas

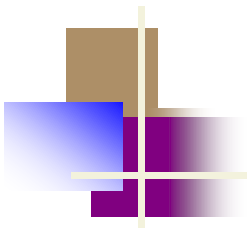



Internal routers can select the best route to a destination outside their area, including the best exit router to the backbone



# BGP – Exterior Routing Protocol

- Between independently operated networks
- Inter-domain (exterior) gateway protocol
- BGP (Border Gateway Protocol) is used
- Have to worry about politics a great deal
- Example policies:
  - Do not carry commercial traffic on the educational network
  - Never send traffic from the Pentagon on a route through Iraq
  - Use TeliaSonera instead of Verizon because it is cheaper
  - Don’t use AT&T in Australia because performance is poor
  - Traffic starting or ending at Apple should not transit Google

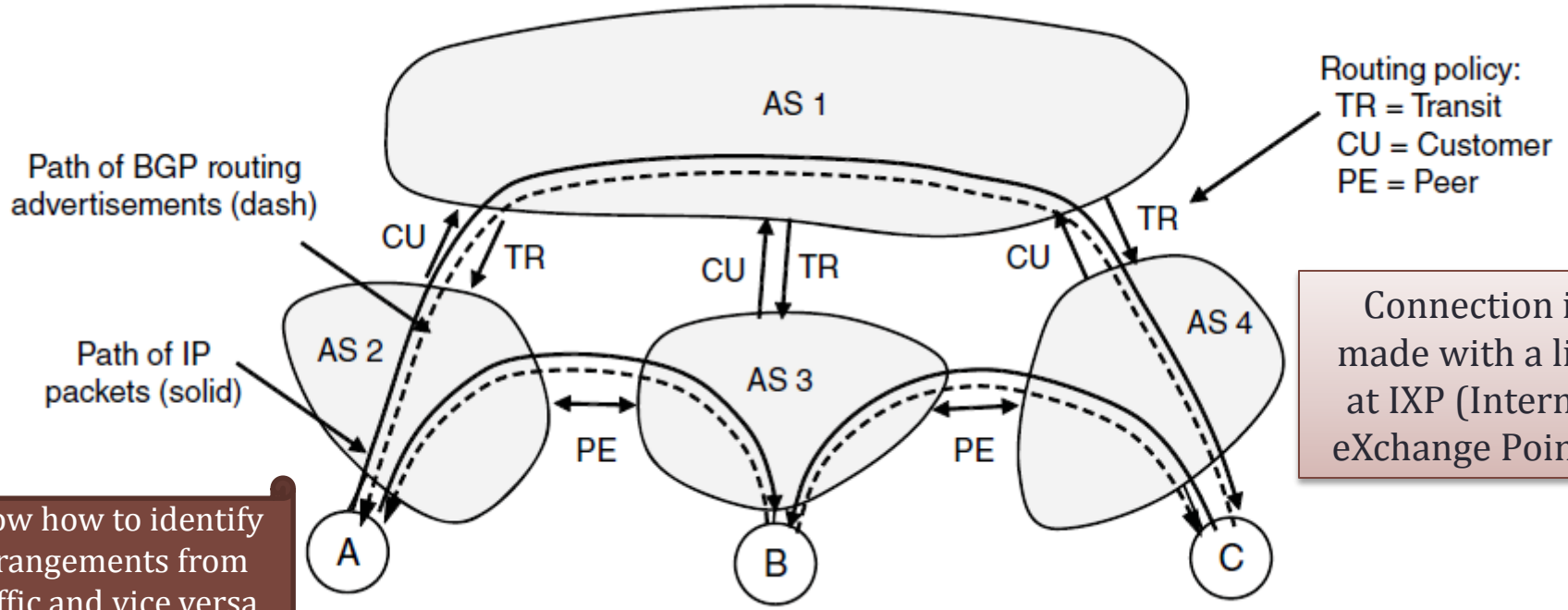


# BGP – Exterior Routing Protocol

- Routing policy is implemented by deciding what traffic can flow over which ASes' links
- ISP pay other ISP to deliver/receive packets
- Customer ISP said to buy **transit service**
- Customer ISP advertise routes only to its internal destinations to provider ISP
- Provider sends traffic only to these dests

# BGP – Exterior Routing Protocol

- AS2, AS3, AS4 are customers of AS1; pay for service
- A send to C: packets go AS2 to AS1 to AS4
- Routing adv reverse: AS4 adv C to AS1 (TR provider)
- AS1 adv route to C to its customers, including AS2

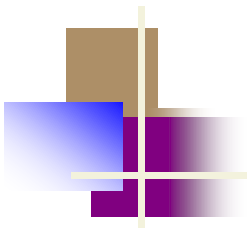



Connection is made with a link at IXP (Internet eXchange Points)

Know how to identify arrangements from traffic and vice versa

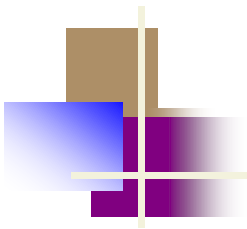
# BGP – Exterior Routing Protocol

- AS2, AS3, AS4 are buy transit svc from AS1
- Transit must be paid for
- If AS2, AS3 exc a lot of traffic, use **peering**
- Send directly for free, reduce pay to AS1
- Peering is not transitive
  - AS3 PE AS4, tfc B⟶C can go thru AS4
  - AS3 only adv route to B to AS4, not a route to A
  - what happens if C sends to A? must go thru AS1



# BGP – Exterior Routing Protocol

- A, B, C have transit arrangements
  - A can be host or LAN
  - don't need BGP; a **stub network**, one link
- Some LANs might connect to multiple ISPs
  - called **multi-homing**; improve reliability
  - in that case, use interdomain routing, e.g. BGP
  - tell other ASes which addresses via which links



# BGP – Exterior Routing Protocol

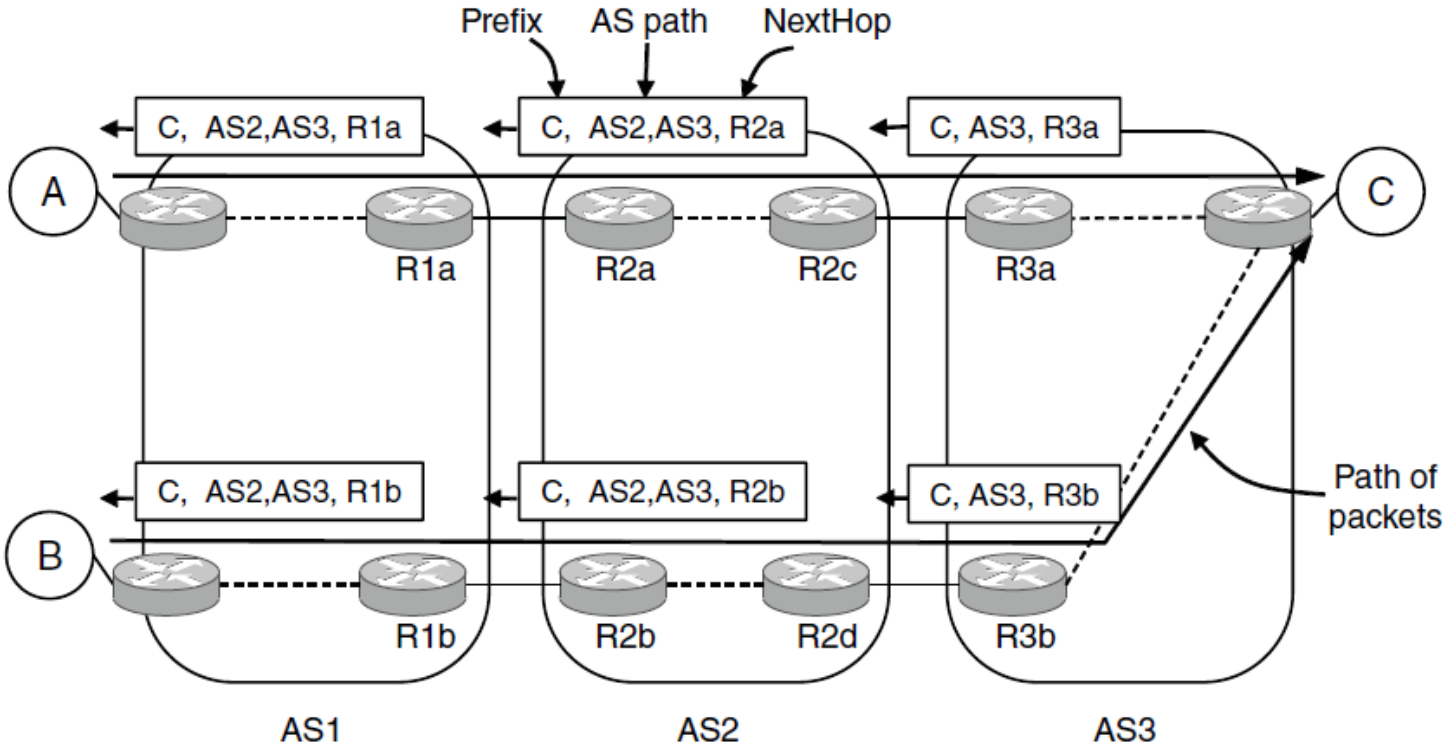
- BGP is a form of distance vector
- But, quite unlike intra-domain DV s as RIP
  - policy, not min distance used to pick route
  - keep track of path, not just host (**path vector**)
- Path vector: (prefix: *dest*, next hop, AS path)
- Pairs of BGP routers communicate using TCP
  - configured manually
  - provide reliable communication
  - hide details of network lower layers

# BGP – Exterior Routing Protocol

## BGP Route Advertisement

- Messages propagate along policy-compliant routes
- Carrying path helps to detect loops
- Router discards router if its own # in AS path

Know how to do the vector



# BGP – Exterior Routing Protocol

Giving a list of ASes is a very coarse way to specify a path.

An AS might be a small company, or an international backbone network. There is no way of telling from the route. BGP does not even try because different ASes may use different intradomain protocols whose costs cannot be compared.

Even if they could be compared, an AS may not want to reveal its internal metrics. This is one of the ways that interdomain routing protocols differ from intradomain protocols.

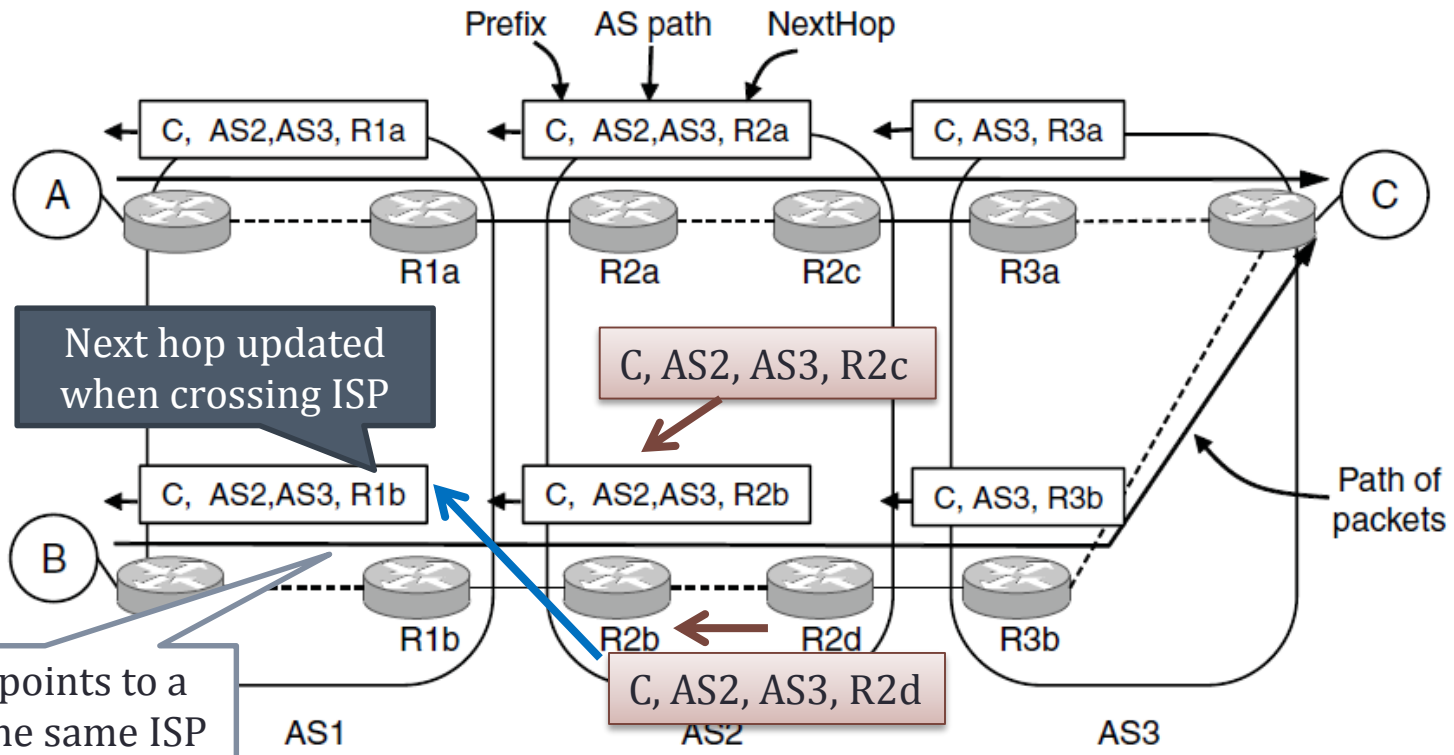
# BGP – Exterior Routing Protocol

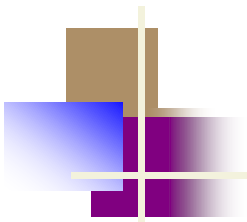
- How to propagate BGP routes from one ISP side to another?: use iBGP (internal BGP)
- Regular BGP called eBGP to distinguish it
- Rule: every router at ISP boundary learns of all routes seen by all other boundary routers
- BGP router may learn route from router of next ISP or from other boundary routers
- How BGP routers choose which route to use for each destination?: Based on ISP policy

# BGP – Exterior Routing Protocol

iBGP (Internal BGP) Route Advertisement

- R2b know it can reach C via R2c or R2d
- Must decide which is the best route to use





# BGP – Exterior Routing Protocol

## Common strategies

- Choose peer network over transit network
  - peer is free, transit costs money
- Customer routes given highest preference
  - good to send traffic directly to paying CU
- Shorter AS paths are better
  - debatable: path thru 3 small ASes may be worse than path thru 1 big AS
- Prefer route with lowest cost within ISP
  - known as early exit, hot-potato routing
  - makes routes asymmetric

# BGP – Exterior Routing Protocol

- BGP router chooses its own best route
- BGP at AS level, OSPF within AS? Not the case
- BGP, interior (OSPF) integrated more deeply
- Lot of freedom
- ISP must be careful, make compatible choices

# External References

- 1's complement addition, http://mathforum.org/library/drmath/view/54379.html
- Supernet Calculator, www.subnet-calculator.org/supernets.php
- Cisco IOS Network Address Translation Overview, http://www.cisco.com/en/US/technologies/tk648/tk361/tk438/technologies_white_paper09186a0080091cb9.html
- IPv6 Extension Headers Review and Considerations, http://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.html
- Explaining BGP Slow Table Transfers: Implementing a TCP Delay Analyzer, http://fmdb.cs.ucla.edu/Treports/110020.pdf