

# IP and Networking Basics



Scalable Infrastructure  
Workshop  
AfNOG 2010

# Internet History

---

## *1961-1972: Early packet-switching principles*

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational
- 1972:
  - ARPAnet demonstrated publicly
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes

# Internet History

---

## *1972-1980: Internetworking, new and proprietary nets*

- ❑ 1970: ALOHAnet satellite network in Hawaii
  - ❑ 1973: Metcalfe's PhD thesis proposes Ethernet
  - ❑ 1974: Cerf and Kahn - architecture for interconnecting networks
  - ❑ Late 70's: proprietary architectures: DECnet, SNA, XNA
  - ❑ late 70's: switching fixed length packets (ATM precursor)
  - ❑ 1979: ARPAnet has 200 nodes
- ❑ Cerf and Kahn's internetworking principles:
    - minimalism, autonomy - no internal changes required to interconnect networks
    - best effort service model
    - stateless routers
    - decentralized control
  - ❑ define today's Internet architecture

# Internet History

---

## *1980-1990: new protocols, a proliferation of networks*

- 1983: deployment of TCP/IP
- 1982: SMTP e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: FTP protocol defined
- 1988: TCP congestion control
- New national networks: Cset, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

# Internet History

---

## *1990, 2000's: commercialisation, the Web, new apps*

- **Early 1990's:** ARPAnet decommissioned
- **1991:** NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- **early 1990s:** Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web
- **Late 1990's – 2000's:**
  - more killer apps: instant messaging, peer2peer file sharing (e.g., Napster)
  - network security to forefront
  - est. 50 million host, 100 million+ users
  - backbone links running at Gbps
- **now:** 10-40 Gbps
  - youtube, social networking

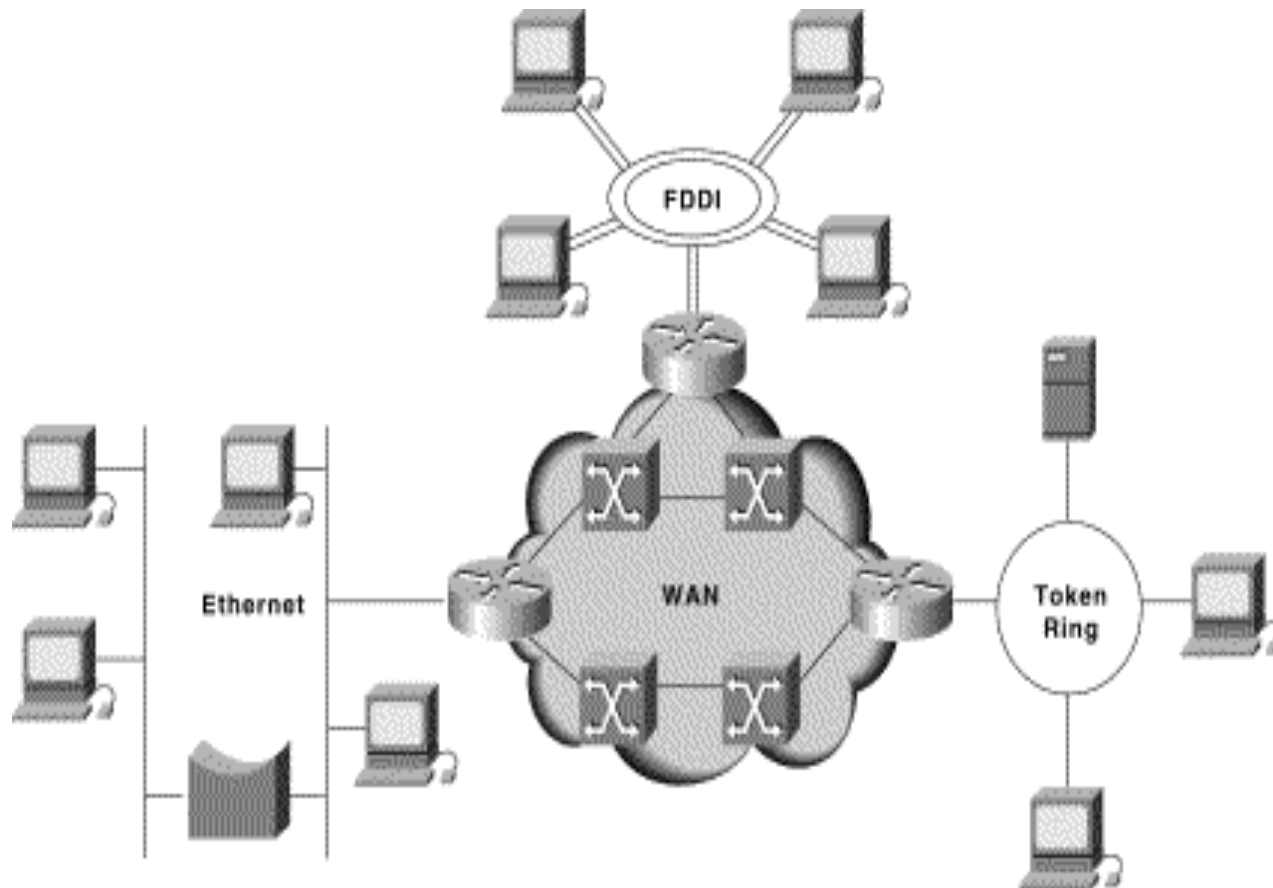
# The (capital “I”) Internet

---

- ❑ The world-wide network of TCP/IP networks
- ❑ Different people or organisations own different parts
- ❑ Different parts use different technologies
- ❑ Interconnections between the parts
- ❑ Interconnections require agreements
  - sale/purchase of service
  - contracts
  - “peering” agreements
- ❑ No central control or management

# A small internetwork or (small “i”) “internet”

---



# The principle of “Internetworking”

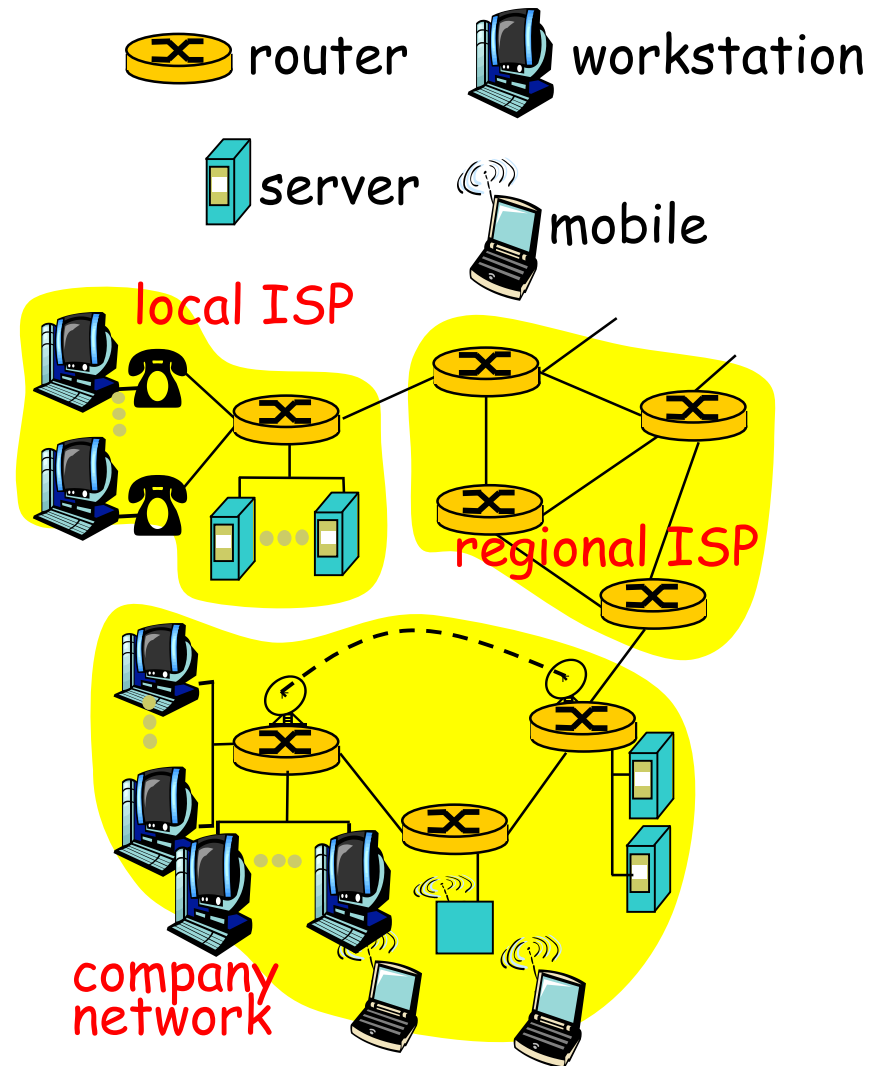
---

- ❑ We have lots of little networks
- ❑ Many different owners/operators
- ❑ Many different types
  - Ethernet, dedicated leased lines, dialup, optical, broadband, wireless, ...
- ❑ Each type has its own idea of low level addressing and protocols
- ❑ We want to connect them all together and provide a unified view of the whole lot (treat the collection of networks as a single large internetwork)



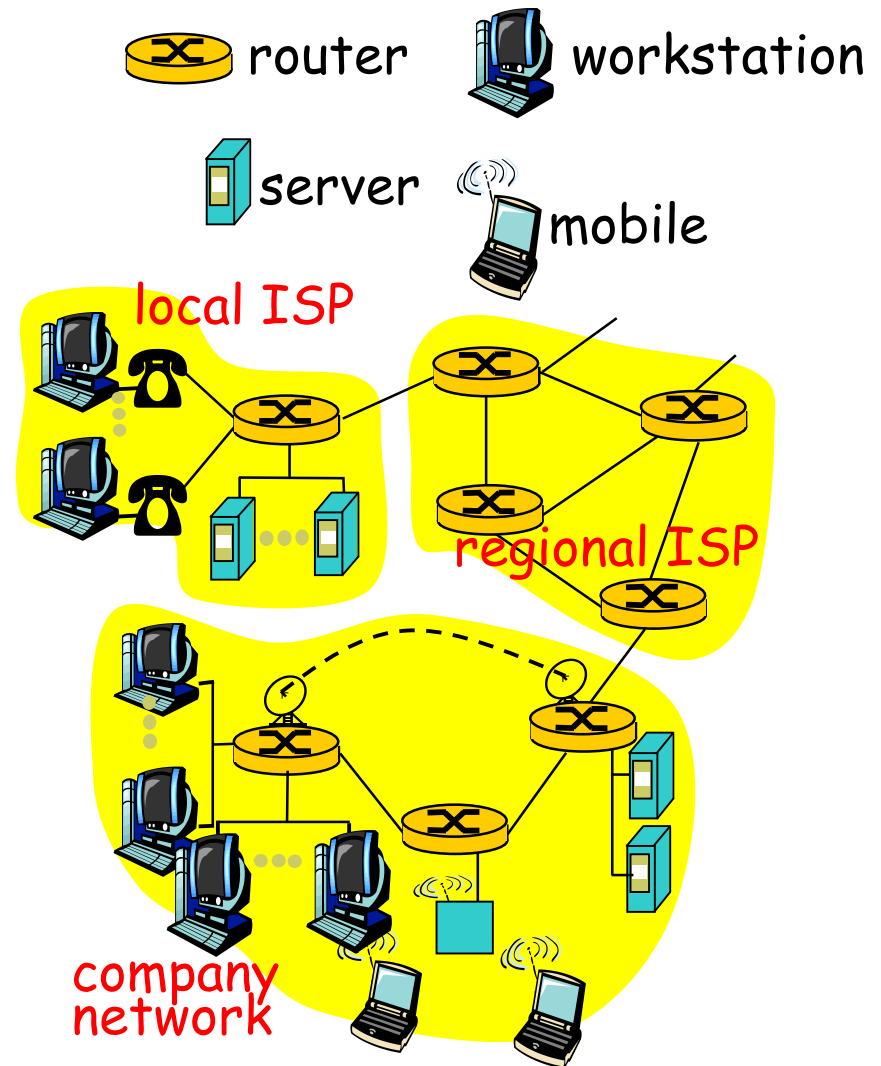
# What is the Internet: “nuts and bolts” view

- millions of connected computing devices: hosts, end-systems
  - PC's workstations, servers
  - PDA's phones, toasters
  - running network apps
- communication links
  - fiber, copper, radio, satellite
- routers: forward packets (chunks) of data through network



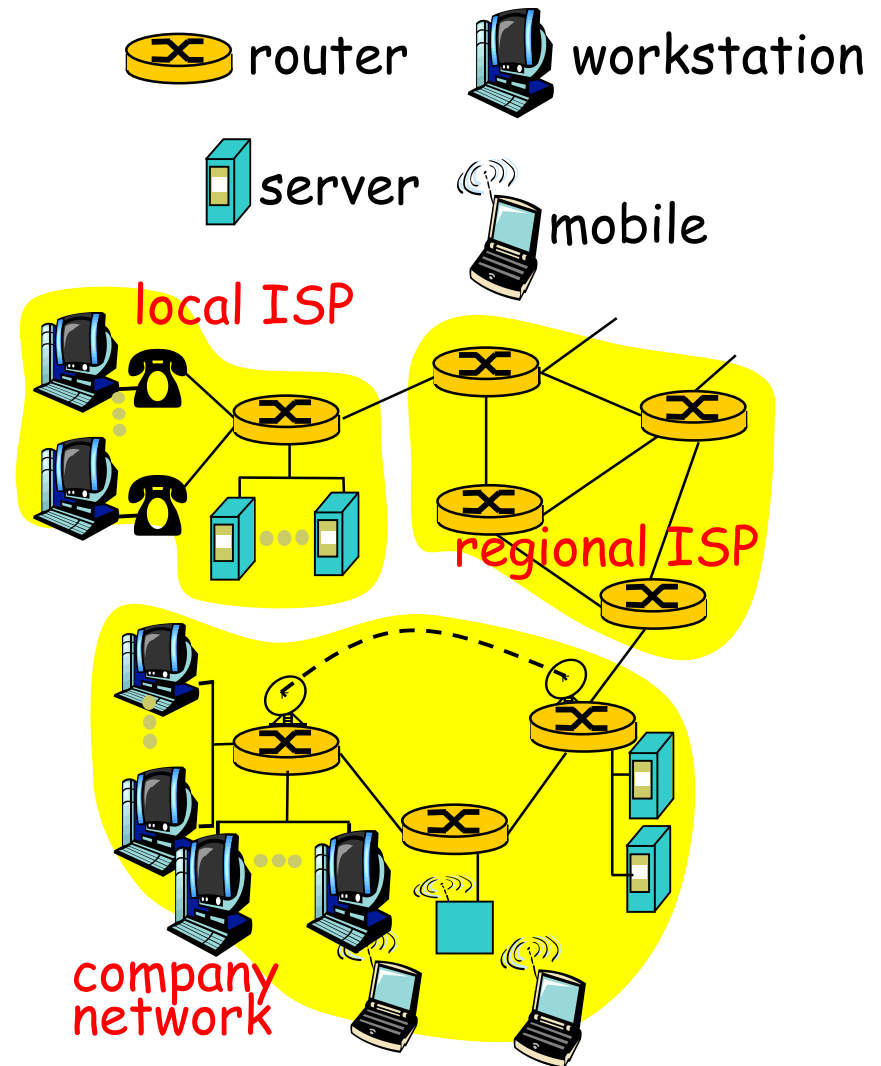
# What is the Internet: “nuts and bolts” view

- protocols: control sending, receiving of messages
  - e.g., TCP, IP, HTTP, FTP, PPP
- Internet: “network of networks”
  - loosely hierarchical
  - public Internet versus private intranet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



# What is the Internet: a service view

- communication infrastructure enables distributed applications:
  - WWW, email, games, e-commerce, database, e-voting, more?
- communication services provided:
  - connectionless
  - connection-oriented



# Connectionless Paradigm

---

- There is no “connection” in IP
  - Packets can be delivered out-of-order
  - Each packet can take a different path to the destination
  - No error detection or correction in payload
  - No congestion control (beyond “drop”)
- TCP mitigates these for connection-oriented applications
  - error correction is by retransmission

# OSI Stack & TCP/IP Architecture



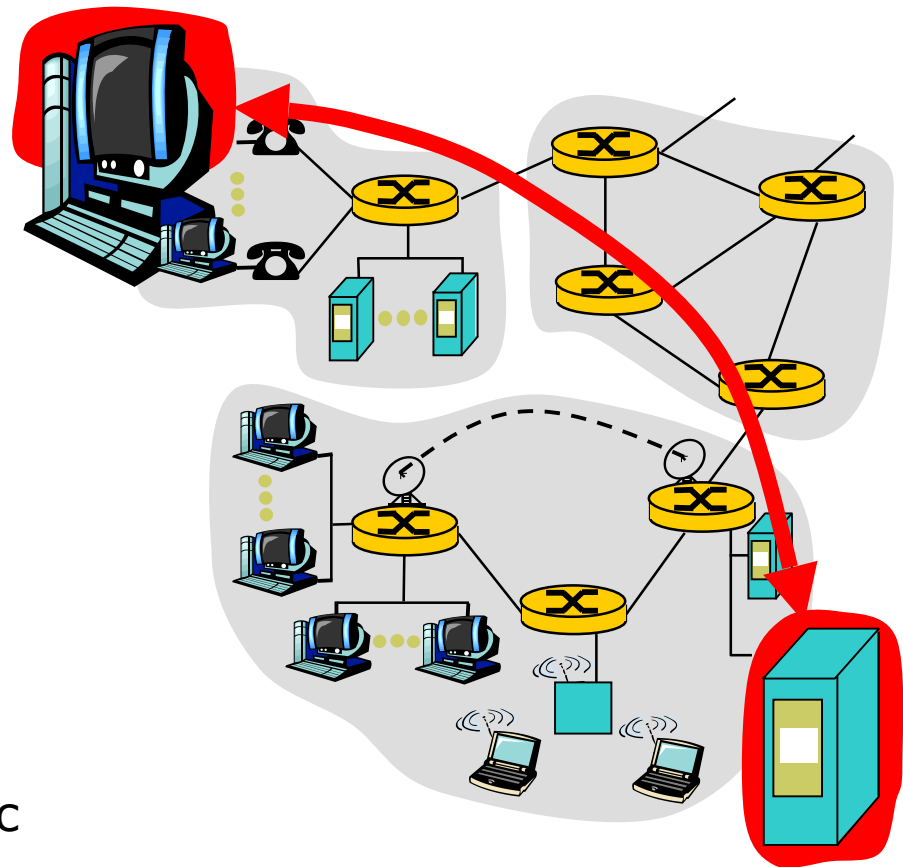
# Principles of the Internet

---

- Edge vs. core (end-systems vs. routers)
  - Dumb network
  - Intelligence at the end-systems
- Different communication paradigms
  - Connection oriented vs. connection less
  - Packet vs. circuit switching
- Layered System
- Network of collaborating networks

# The network edge

- end systems (hosts):
  - run application programs
  - e.g., WWW, email
  - at “edge of network”
- client/server model:
  - client host requests, receives service from server
  - e.g., WWW client (browser)/server; email client/server
- peer-peer model:
  - host interaction symmetric
  - e.g.: teleconferencing



# Network edge: connection-oriented service

---

- Goal: data transfer between end sys.
- handshaking: setup (prepare for) data transfer ahead of time
  - Hello, hello back human protocol
  - set up “state” in two communicating hosts
- TCP - Transmission Control Protocol
  - Internet’s connection-oriented service
- TCP service [RFC 793]
- reliable, in-order byte-stream data transfer
  - loss: acknowledgements and retransmissions
- flow control:
  - sender won’t overwhelm receiver
- congestion control:
  - senders “slow down sending rate” when network congested



# Network edge: connectionless service

---

- Goal: data transfer between end systems
  
- UDP - User Datagram Protocol [RFC 768]: Internet's connectionless service
  - unreliable data transfer
  - no flow control
  - no congestion control

# Protocol “Layers”

---

- Networks are complex!
- many “pieces”:
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software
- Question:
- Is there any hope of organizing structure of network?
- Or at least in our discussion of networks?

# The unifying effect of the network layer

---

- Define a protocol that works in the same way with any underlying network
- Call it the network layer (e.g. IP)
- IP routers operate at the network layer
- IP over anything
- Anything over IP

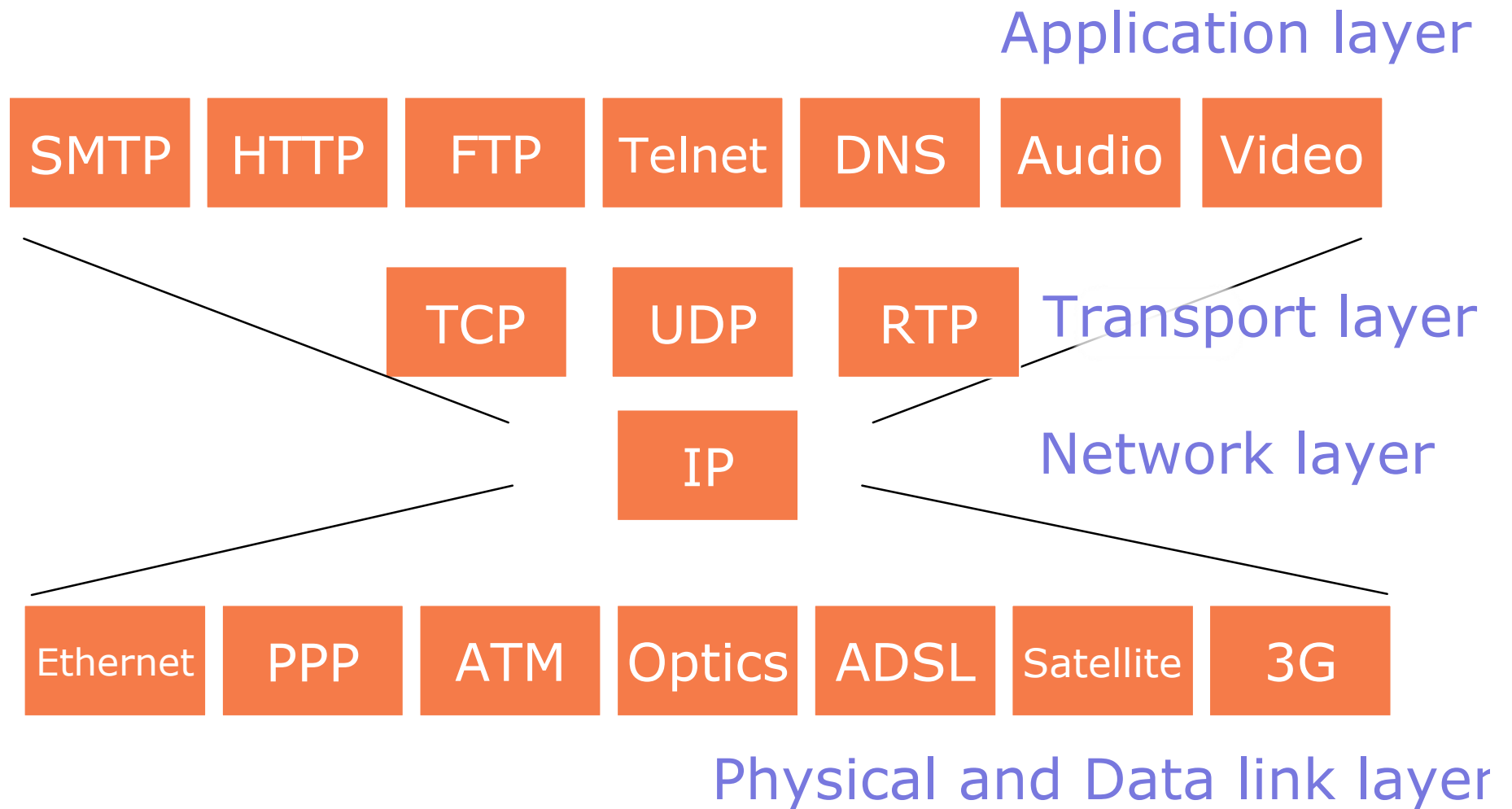
# Why layering?

---

- Dealing with complex systems:
- explicit structure allows identification, relationship of complex system's pieces
  - layered reference model for discussion
- Modularisation eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in gate procedure does not affect rest of system

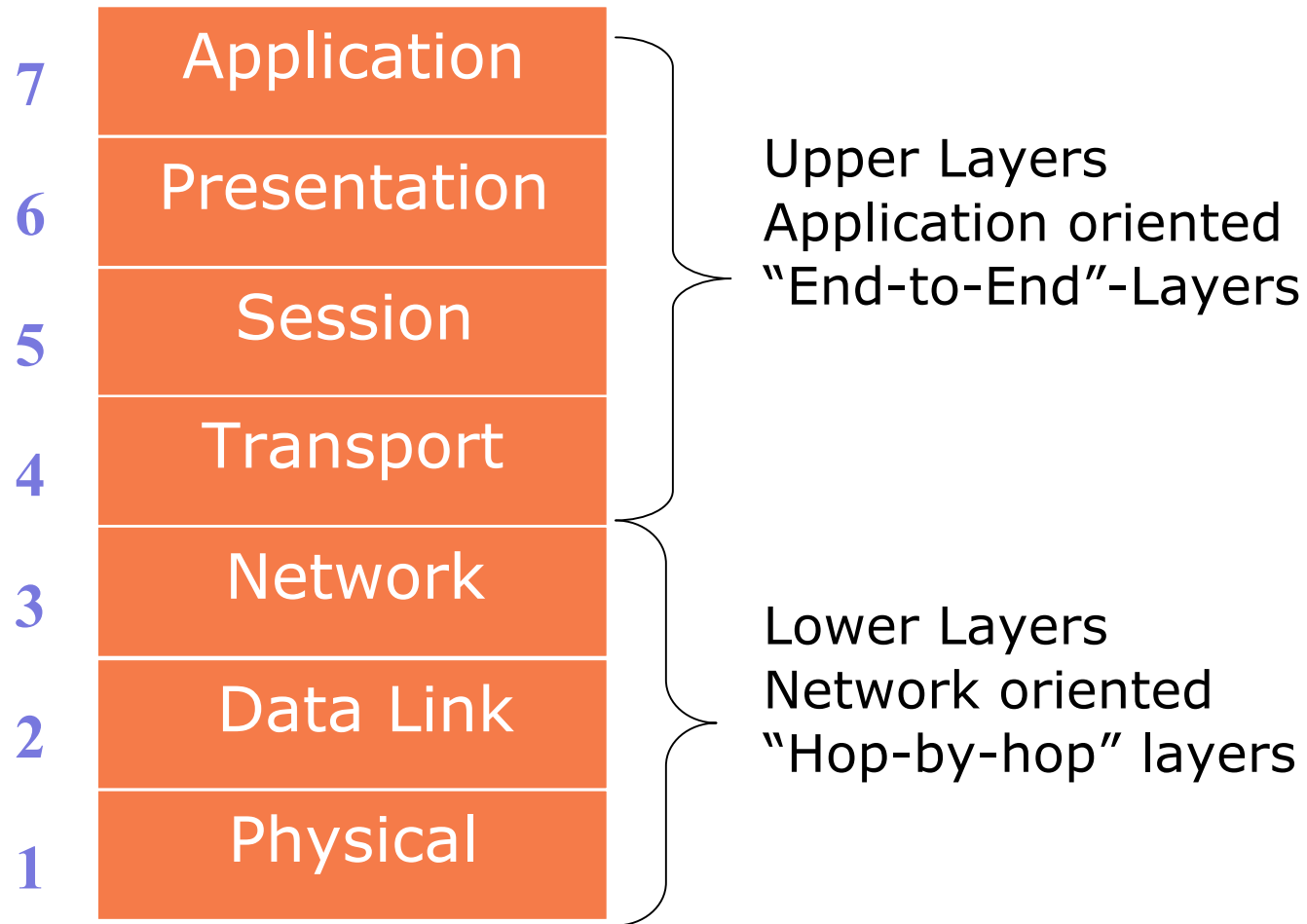
# The IP Hourglass Model

---



# The OSI Model

---

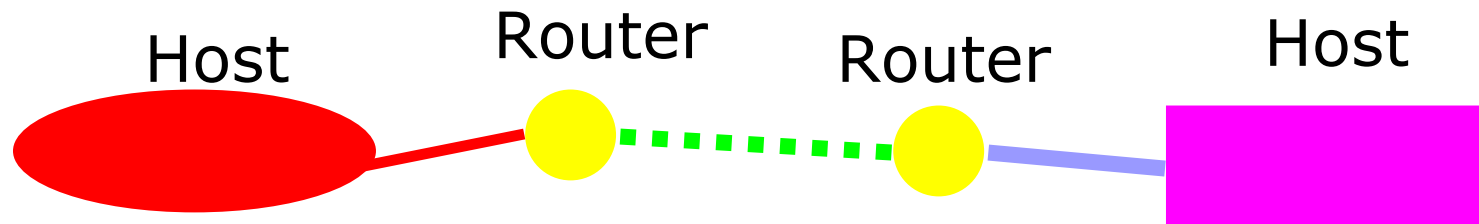
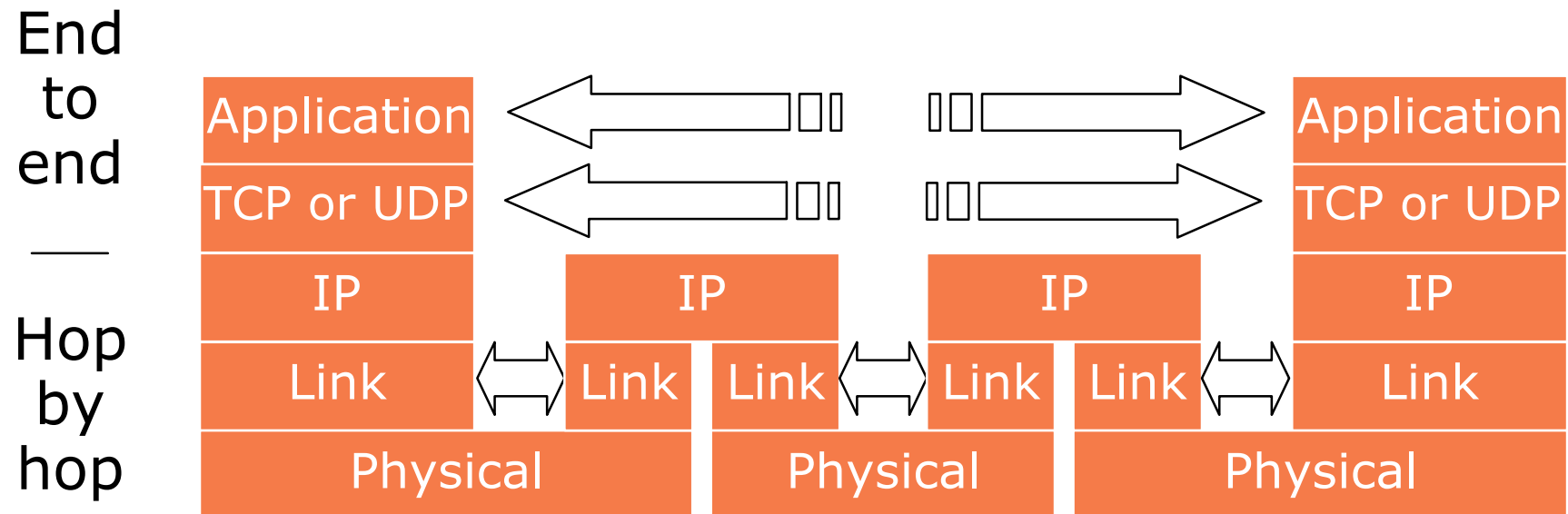


# OSI Model and the Internet

---

- ❑ Internet protocols are not directly based on the OSI model
- ❑ However, we do often use the OSI numbering system. You should at least remember these:
  - Layer 7: Application
  - Layer 4: Transport (e.g. TCP, UDP)
  - Layer 3: Network (IP)
  - Layer 2: Data link
  - Layer 1: Physical

# Layer Interaction: TCP/IP Model





# End-to-end layers

---

- Upper layers are “end-to-end”
- Applications at the two ends behave as if they can talk directly to each other
- They do not concern themselves with the details of what happens in between

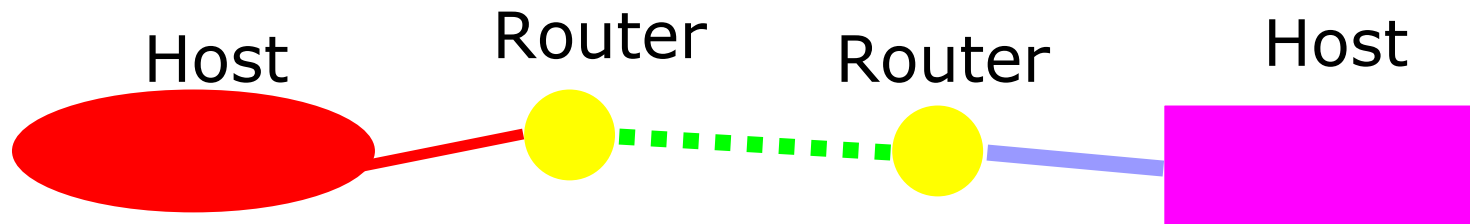
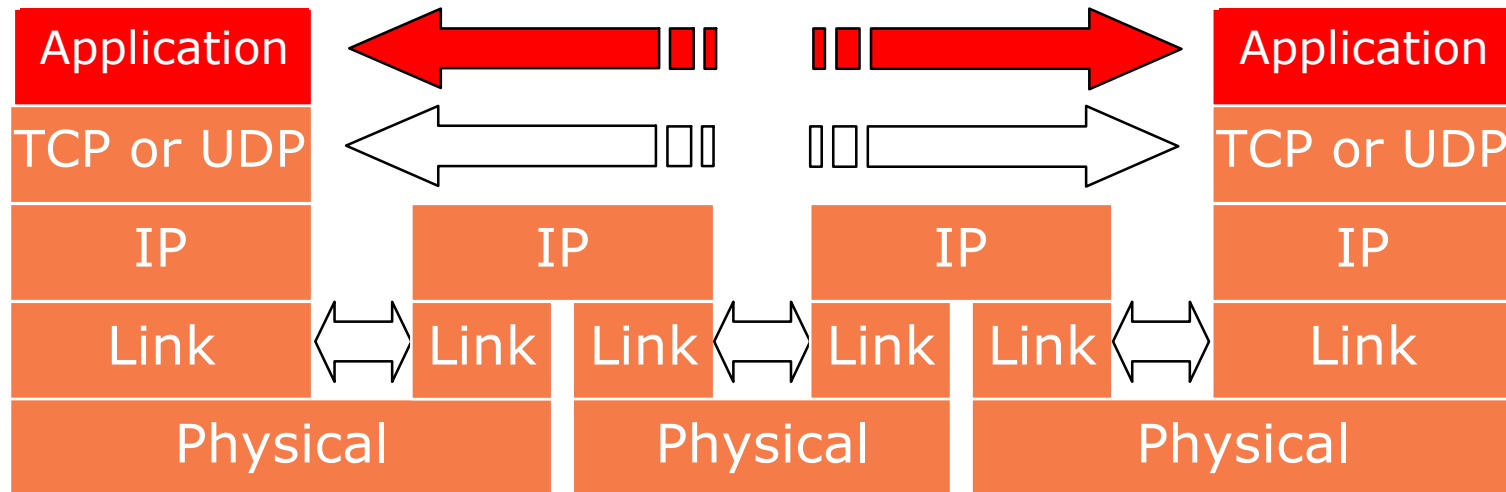
# Hop-by-hop layers

---

- ❑ At the lower layers, devices share access to the same physical medium
- ❑ Devices communicate directly with each other
- ❑ The network layer (IP) has some knowledge of how many small networks are interconnected to make a large internet
- ❑ Information moves one hop at a time, getting closer to the destination at each hop

# Layer Interaction: TCP/IP Model

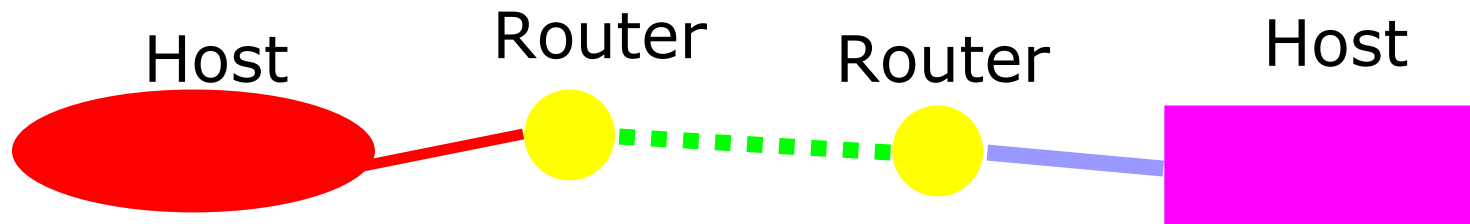
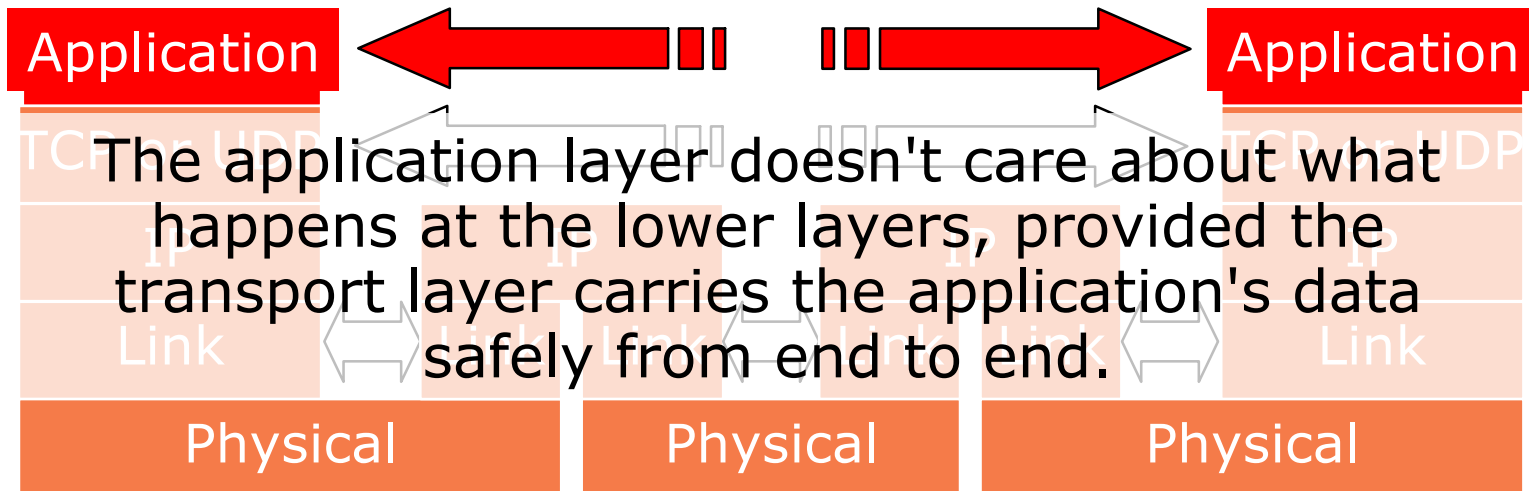
---



# Layer Interaction:

## The Application Layer

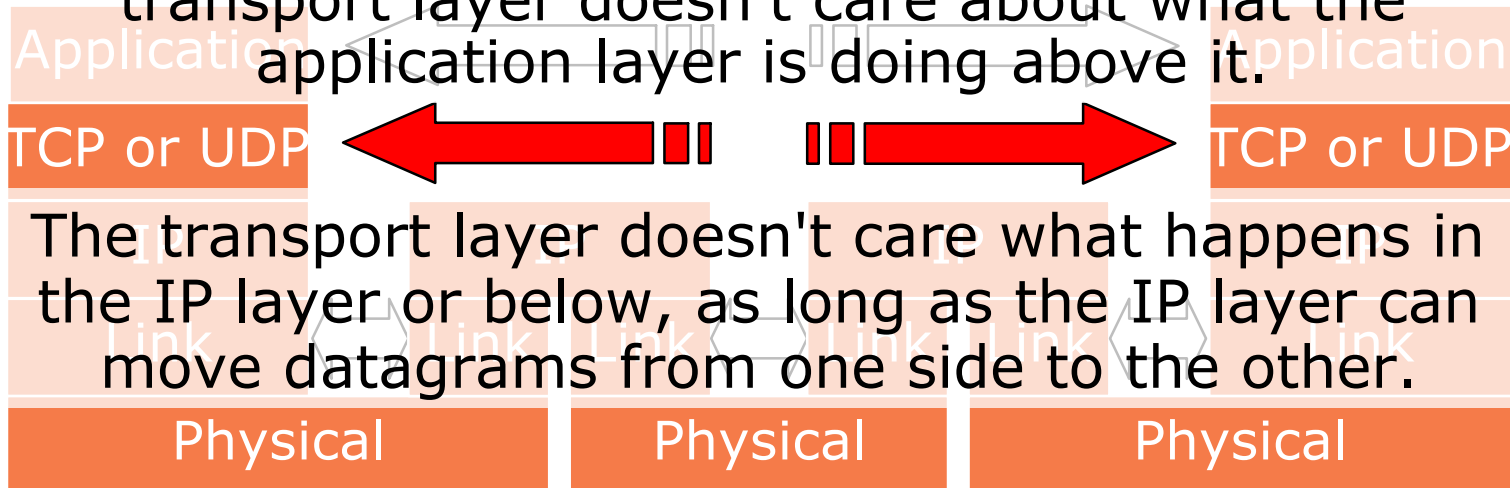
Applications behave as if they can talk to each other, but in reality the application at each side talks to the TCP or UDP service below it.



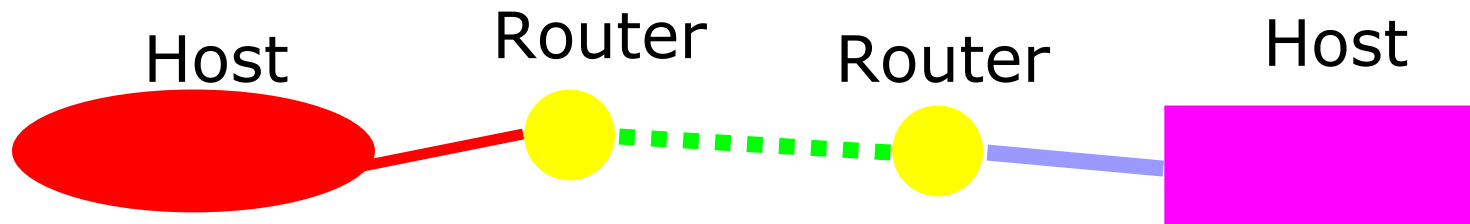
# Layer Interaction:

## The Transport Layer

The transport layer instances at the two ends act as if they are talking to each other, but in reality they are each talking to the IP layer below it. The transport layer doesn't care about what the application layer is doing above it.



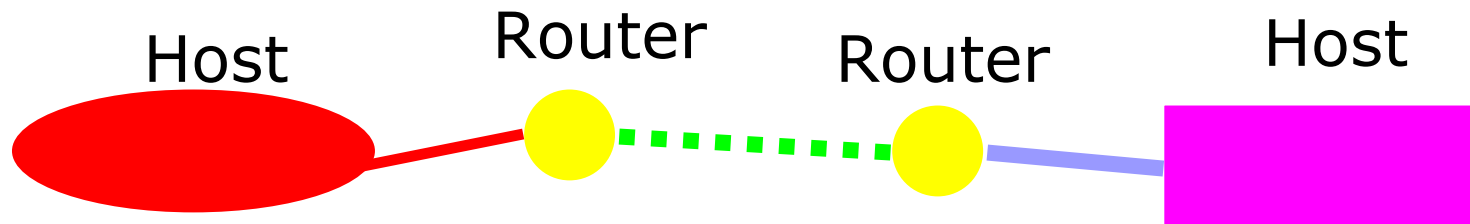
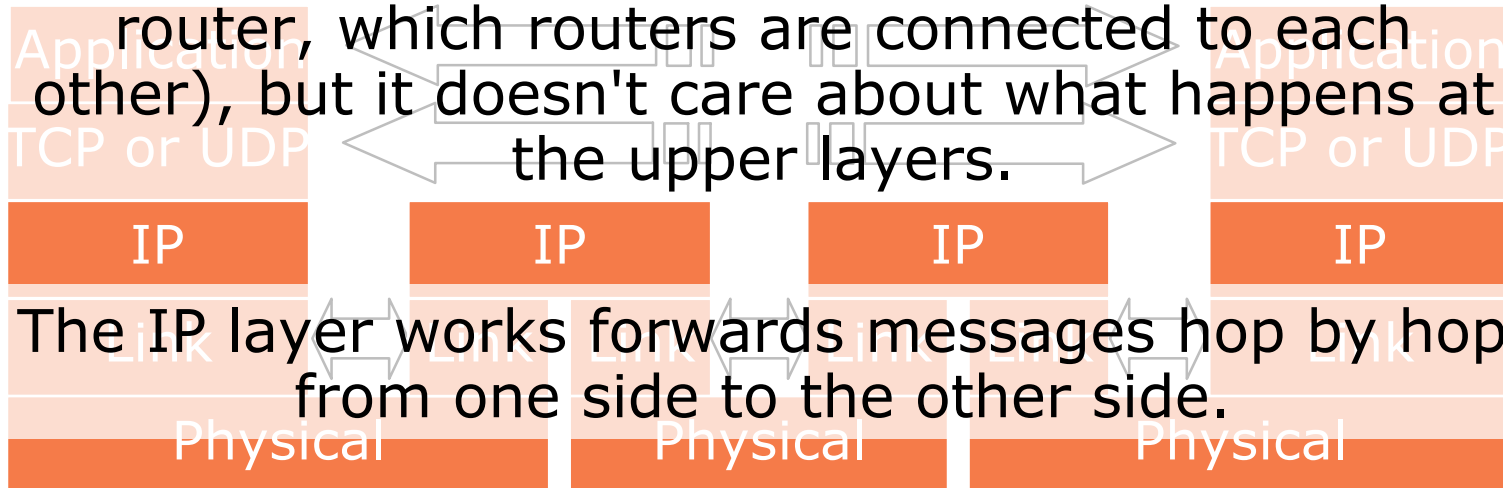
The transport layer doesn't care what happens in the IP layer or below, as long as the IP layer can move datagrams from one side to the other.



# Layer Interaction:

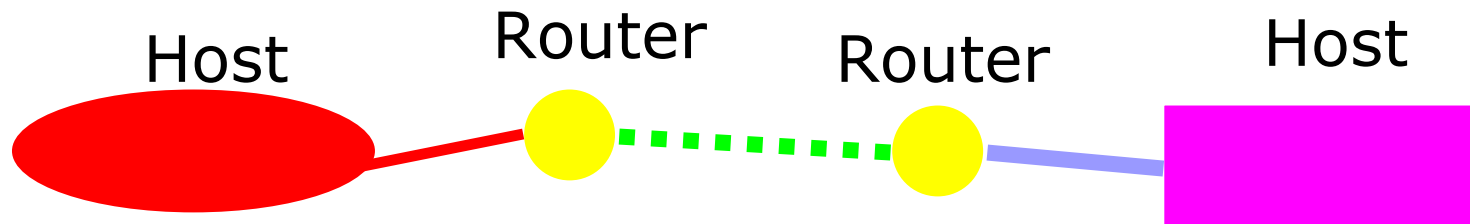
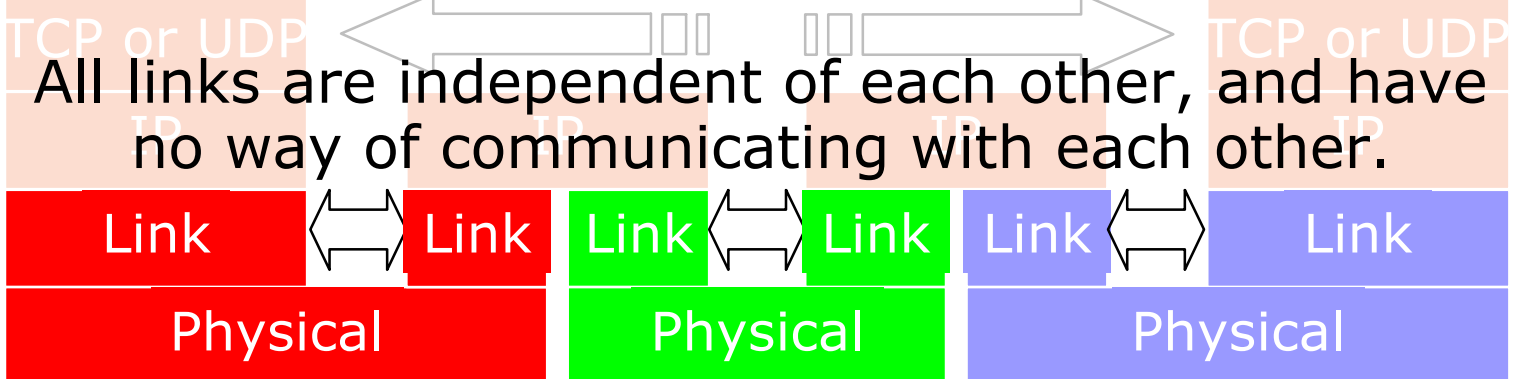
## The Network Layer (IP)

The IP layer has to know a lot about the topology of the network (which host is connected to which router, which routers are connected to each other), but it doesn't care about what happens at the upper layers.

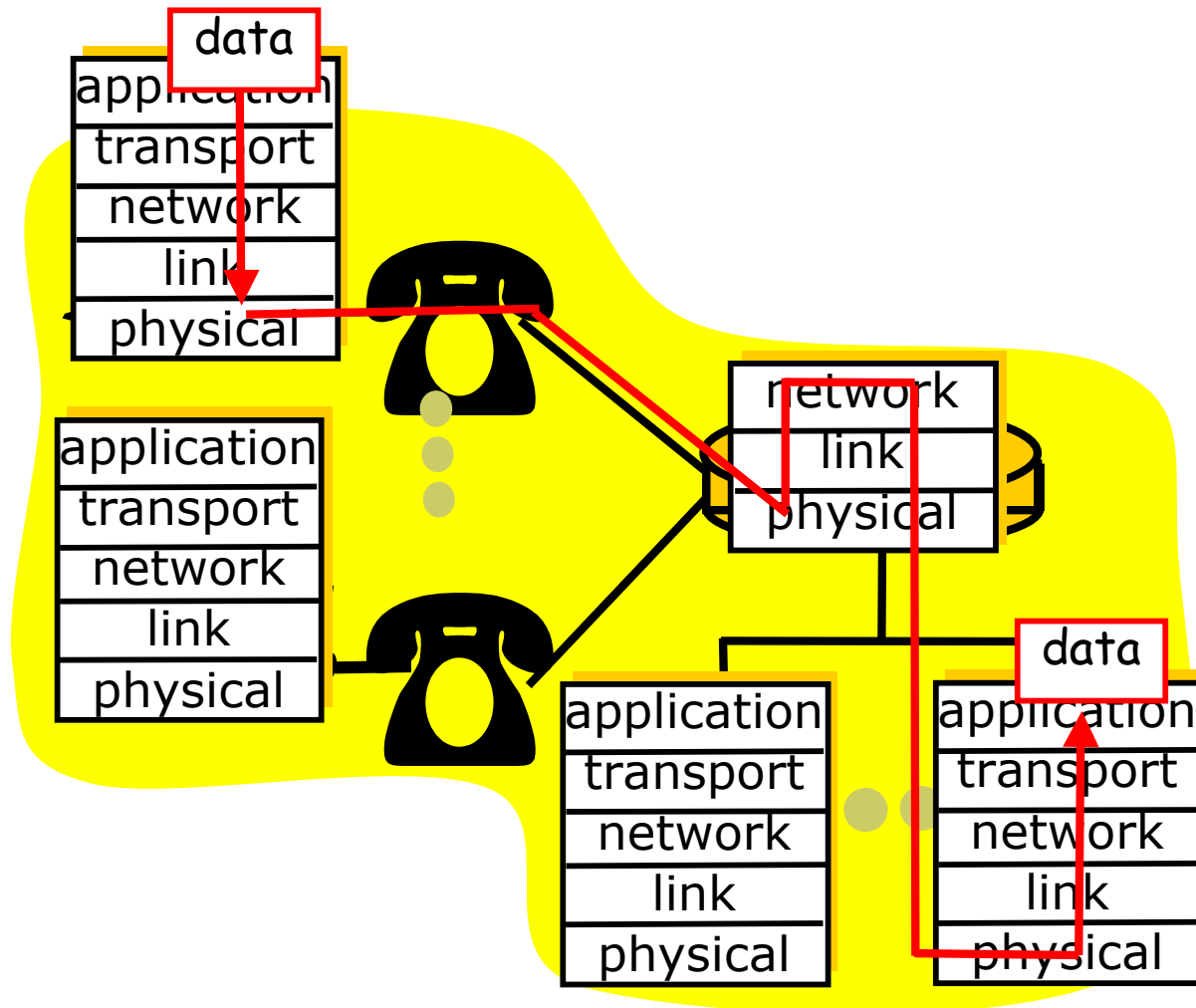


# Layer Interaction: Link and Physical Layers

The link layer doesn't care what happens above it, but it is very closely tied to the physical layer below it.



# Layering: physical communication





# Frame, Datagram, Segment, Packet

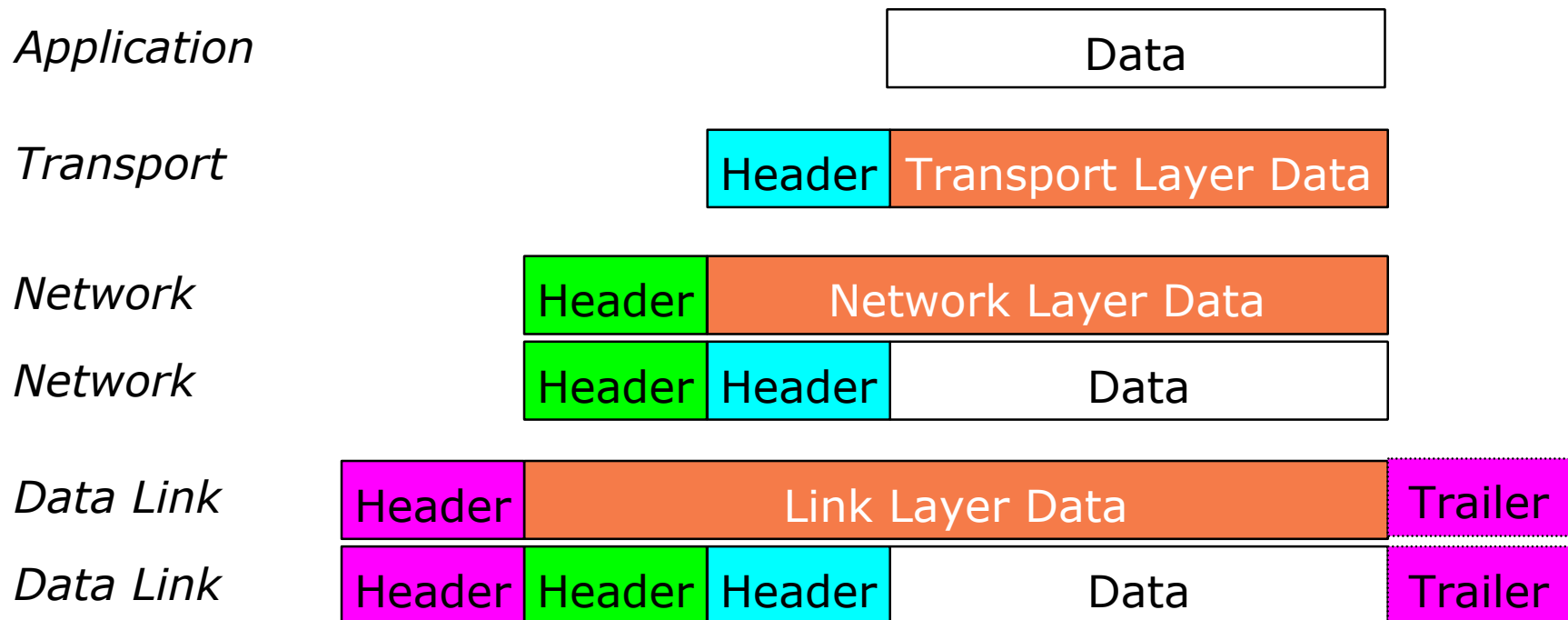
---

- Different names for packets at different layers
  - Ethernet (link layer) frame
  - IP (network layer) datagram
  - TCP (transport layer) segment
- Terminology is not strictly followed
  - we often just use the term “packet” at any layer

# Encapsulation & Decapsulation

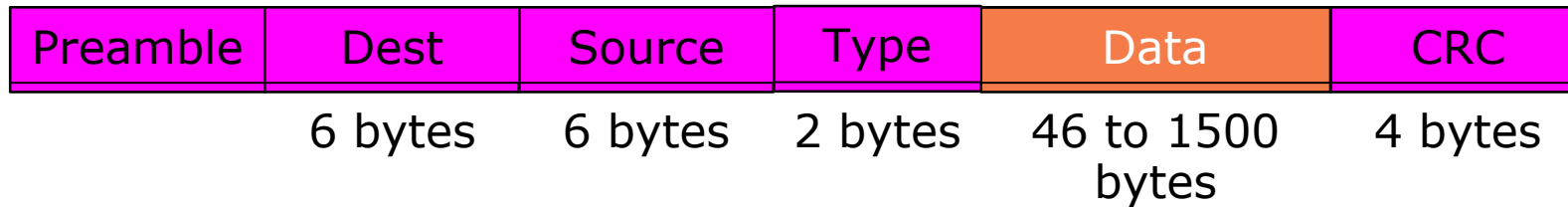
---

- Lower layers add headers (and sometimes trailers) to data from higher layers



# Layer 2 - Ethernet frame

---



- ❑ Destination and source are 48-bit MAC addresses (e.g., 00:26:4a:18:f6:aa)
- ❑ Type 0x0800 means that the “data” portion of the Ethernet frame contains an IPv4 datagram. Type 0x0806 for ARP. Type 0x86DD for IPv6.
- ❑ “Data” part of layer 2 frame contains a layer 3 datagram.

# Layer 3 - IPv4 datagram

Version	IHL	Diff Services	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address (32-bit IPv4 address)				
Destination Address (32-bit IPv4 address)				
Options				Padding
Data (contains layer 4 segment)				

- Version = 4  
If no options, IHL = 5  
Source and Destination are 32-bit IPv4 addresses
- Protocol = 6 means data portion contains a TCP segment.  
Protocol = 17 means UDP.

# Layer 4 - TCP segment

Source Port				Destination Port				
Sequence Number								
Acknowledgement Number								
Data Offset	Reserved	U	A	E	R	S	F	Window
		R	C	O	S	Y	I	
		G	K	L	T	N	N	
Checksum				Urgent Pointer				
Options						Padding		
Data (contains application data)								

- Source and Destination are 16-bit TCP port numbers (IP addresses are implied by the IP header)
- If no options, Data Offset = 5 (which means 20 octets)

# IP Addressing



# Purpose of an IP address

---

- Unique Identification of:
  - Source
    - So the recipient knows where the message is from
    - Sometimes used for security or policy-based filtering of data
  - Destination
    - So the networks know where to send the data
- Network Independent Format
  - IP over anything

# Purpose of an IP Address

---

- ❑ Identifies a machine's connection to a network
- ❑ Physically moving a machine from one network to another requires changing the IP address
- ❑ Unique; assigned in a hierarchical fashion
  - IANA (Internet Assigned Number Authority)
  - IANA to RIRs (AfriNIC, ARIN, RIPE, APNIC, LACNIC)
  - RIR to ISPs and large organisations
  - ISP or company IT department to end users
- ❑ IPv4 uses unique 32-bit addresses
- ❑ IPv6 uses unique 128-bit addresses



# Basic Structure of an IPv4 Address

---

- ❑ 32 bit number (4 octet number):  
(e.g. 133.27.162.125)
- ❑ Decimal Representation:

133	27	162	125
-----	----	-----	-----

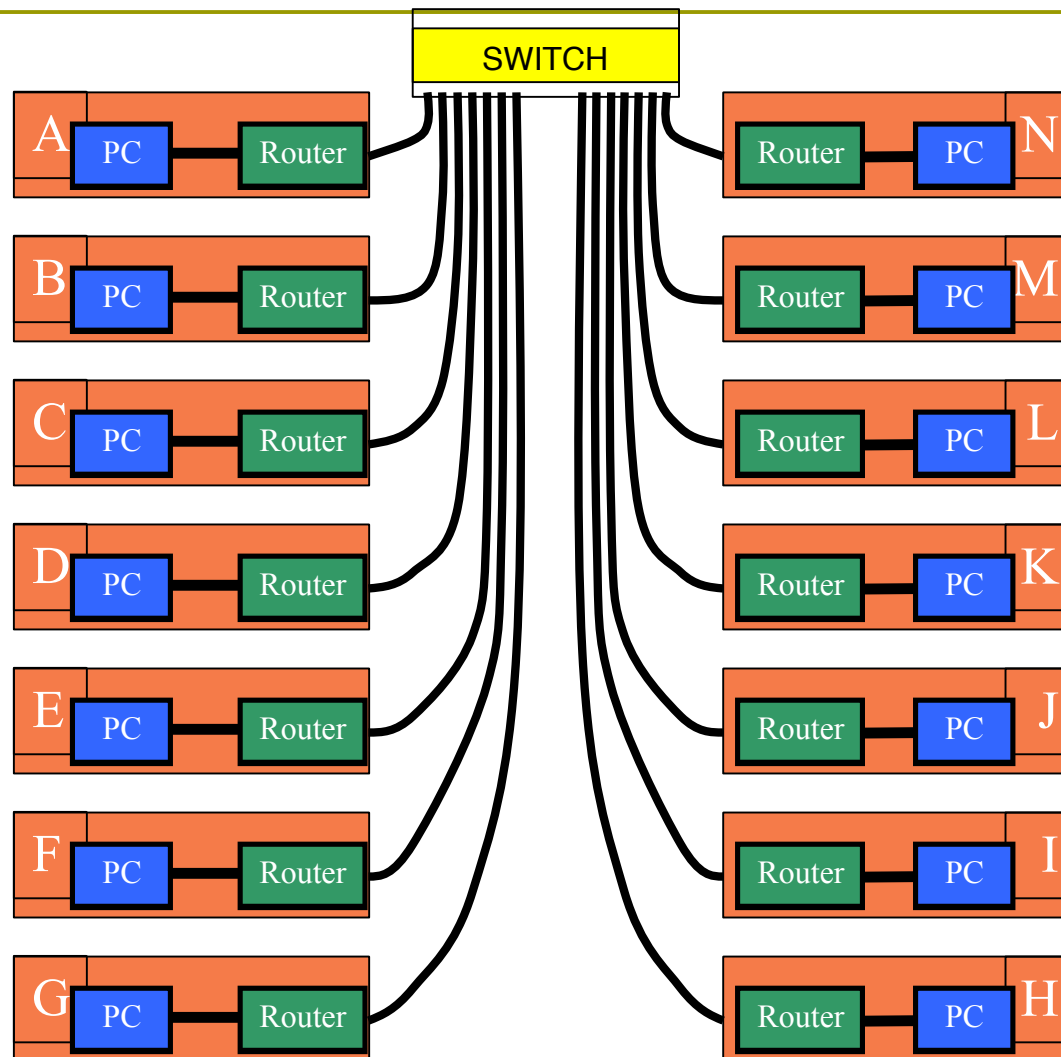
- ❑ Binary Representation:

10000101	00011011	10100010	01111101
----------	----------	----------	----------

- ❑ Hexadecimal Representation:

85	1B	A2	7D
----	----	----	----

# Address Exercise



# Address Exercise

---

- ❑ Construct an IP address for your router's connection to the backbone network.
- ❑ 196.200.220.x
- ❑ x = 1 for row A, 2 for row B, etc.
- ❑ Write it in decimal form as well as binary form.

# Addressing in Internetworks

---

- The problem we have
  - More than one physical network
  - Different Locations
  - Larger number of hosts
  - Need a way of numbering them all
- We use a structured numbering system
  - Hosts that are connected to the same physical network have “similar” IP addresses
  - Often more than one level of structure; e.g. physical networks in the same organisation use “similar” IP addresses

# Network part and Host part

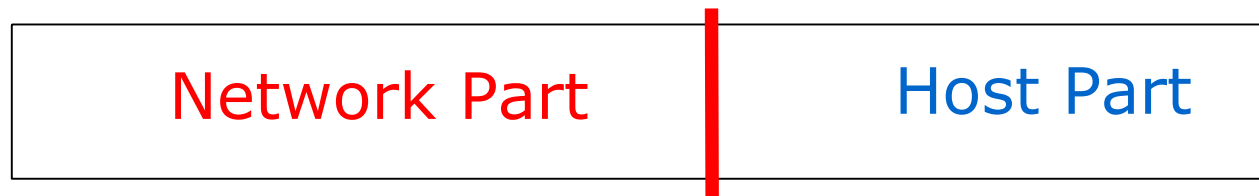
---

- Remember IPv4 address is 32 bits
- Divide it into a “network part” and “host part”
  - “network part” of the address identifies which network in the internetwork (e.g. the Internet)
  - “host part” identifies host on that network
  - Hosts or routers connected to the same link-layer network will have IP addresses with the same network part, but different host part.
  - Host part contains enough bits to address all hosts on the subnet; e.g. 8 bits allows 256 addresses

# Dividing an address

---

- Hierarchical Division in IP Address:
  - Network Part (or Prefix) – high order bits (left)
    - describes which physical network
  - Host Part – low order bits (right)
    - describes which host on that network



- Boundary can be anywhere
  - choose the boundary according to number of hosts
  - very often NOT a multiple of 8 bits

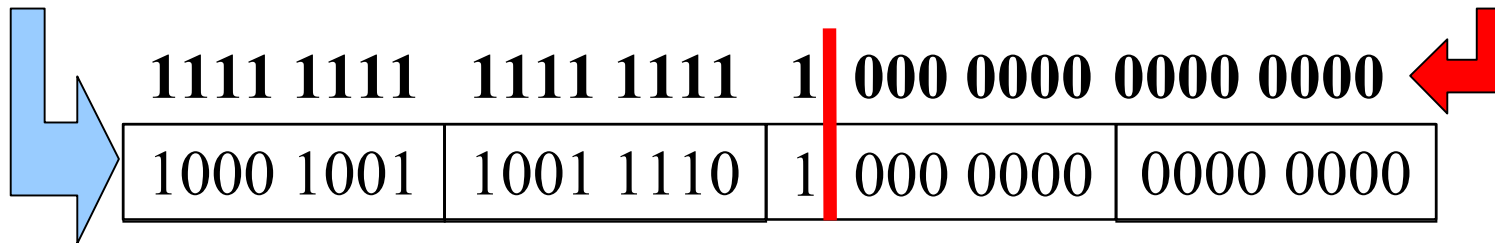
# Network Masks

---

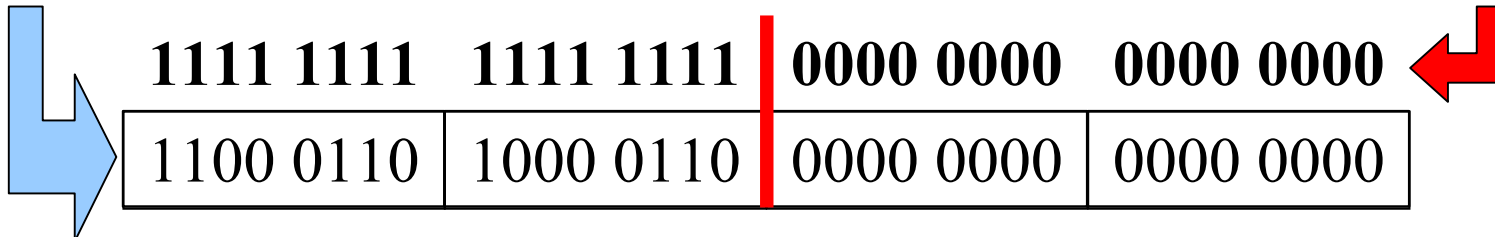
- ❑ “Network Masks” help define which bits are used to describe the Network Part and which for the Host Part
- ❑ Different Representations:
  - decimal dot notation: 255.255.224.0
  - binary: 11111111 11111111 11100000 00000000
  - hexadecimal: 0xFFFFE000
  - number of network bits: /19
    - ❑ count the 1's in the binary representation
- ❑ Above examples all mean the same: 19 bits for the Network Part and 13 bits for the Host Part

# Example Prefixes

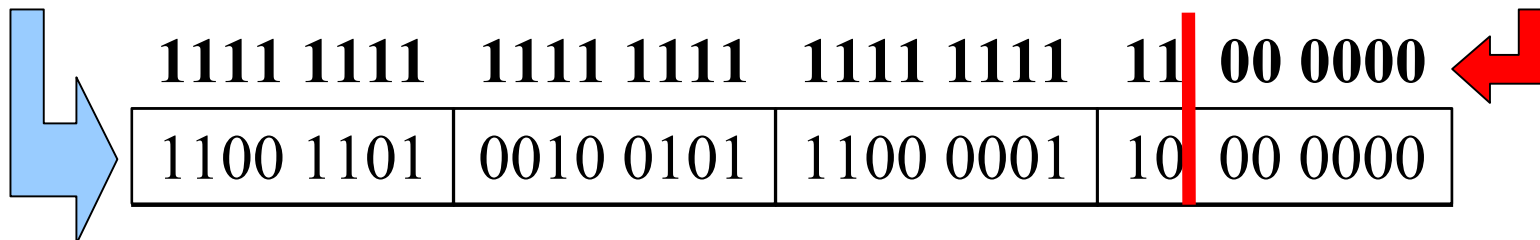
- 137.158.128.0/17 (netmask 255.254.0.0)



- 198.134.0.0/16 (netmask 255.255.0.0)



- 205.37.193.128/26 (netmask 255.255.255.192)





# Special Addresses

---

- All 0's in host part: Represents Network
  - e.g. 193.0.0.0/24
  - e.g. 138.37.64.0/18
  - e.g. 196.200.223.96/28
- All 1's in host part: Broadcast
  - e.g. 193.0.0.255 (prefix 193.0.0.0/24)
  - e.g. 138.37.127.255 (prefix 138.37.64.0/18)
  - e.g. 196.200.223.111 (prefix 196.200.223.96/28)
- 127.0.0.0/8: Loopback address (127.0.0.1)
- 0.0.0.0: Various special purposes

# Exercise

---

- Verify that the previous examples are all broadcast addresses:
  - 193.0.0.255 (prefix 193.0.0.0/24)
  - 138.37.127.255 (prefix 138.37.64.0/18)
  - 196.200.223.111 (prefix 196.200.223.96/28)
- Do this by finding the boundary between network part and host part, and checking that the host part (if written in binary) contains all 1's.

# Maximum number of hosts per network

---

- ❑ The number of bits in the host part determines the maximum number of hosts
- ❑ The all-zeros and all-ones addresses are reserved, can't be used for actual hosts
- ❑ E.g. a subnet mask of 255.255.255.0 or /24 means 24 network bits, 8 host bits (24+8=32)
  - $2^8$  minus 2 = 254 possible hosts
- ❑ Similarly a subnet mask of 255.255.255.224 or /27 means 27 network bits, 5 host bits (27+5=32)
  - $2^5$  minus 2 = 30 possible hosts

# More Address Exercises

---

- If there were 9 routers on the classroom backbone network:
  - What is the minimum number of host bits needed to address each router with a unique IP address?
  - With that many host bits, how many network bits?
  - What is the corresponding prefix length in “slash” notation?
  - What is the corresponding netmask (in decimal)?
  - With that netmask, what is the maximum number of hosts?

# More levels of address hierarchy

---

- ❑ Extend the concept of “network part” and “host part”:
  - arbitrary number of levels of hierarchy
  - blocks don't all need to be the same size
  - but each block size must be a power of 2
- ❑ Very large blocks allocated to RIRs (e.g. /8)
  - Divided into smaller blocks for ISPs (e.g. /17)
    - ❑ Divided into smaller blocks for businesses (e.g. /22)
      - Divided into smaller blocks for local networks (e.g. /26)
        - Each host gets a host address
- ❑ What if addresses overlap??

# Ancient History: Classful Addressing

---

- ❑ Nowadays, we always explicitly say where the boundary between network and host part is
  - using slash notation or netmask notation
- ❑ Old systems used restrictive rules (obsolete)
  - Called "Class A", "Class B", "Class C" networks
  - Boundary between network part and host part was implied by the class
- ❑ Nowadays (since 1994), no restriction
  - Called "classless" addressing, "classless" routing

# Ancient History: Sizes of classful networks

---

- Different classes were used to represent different sizes of network (small, medium, large)
- Class A networks (large):
  - 8 bits network part, 24 bits host part
- Class B networks (medium):
  - 16 bits network part, 16 bits host part
- Class C networks (small):
  - 24 bits network part, 8 bits host part





# Ancient History: Implied netmasks

---

- ❑ A classful network had a “natural” or “implied” prefix length or netmask:
  - Class A: prefix length /8 (netmask 255.0.0.0)
  - Class B: prefix length /16 (netmask 255.255.0.0)
  - Class C: prefix length /24 (netmask 255.255.255.0)
- ❑ Modern (classless) routing systems have explicit prefix lengths or netmasks
  - You can't just look at an IP address to tell what the prefix length or netmask should be. Protocols and configurations need explicit netmask or prefix length.

# Classless addressing

---

- ❑ Class A, Class B, Class C terminology and restrictions are now of historical interest only
  - Obsolete in 1994
- ❑ Internet routing and address management today is classless
- ❑ **CIDR = Classless Inter-Domain Routing**
  - Routing does not assume that former class A, B, C addresses imply prefix lengths of /8, /16, /24
- ❑ **VLSM = Variable-Length Subnet Masks**
  - Routing does not assume that all subnets are the same size

# Classless addressing example

---

- An ISP gets a large block of addresses
  - e.g., a /16 prefix, or 65536 separate addresses
- Assign smaller blocks to customers
  - e.g., a /22 prefix (1024 addresses) to one customer, and a /28 prefix (16 addresses) to another customer (and some space left over for other customers)
- An organisation that gets a /22 prefix from their ISP divides it into smaller blocks
  - e.g. a /26 prefix (64 addresses) for one department, and a /27 prefix (32 addresses) for another department (and some space left over for other internal networks)

# Classless addressing exercise

---

- ❑ Consider the address block 133.27.162.0/23
- ❑ Allocate 5 separate /29 blocks, one /27 block, and one /25 block
- ❑ What are the IP addresses of each block allocated above?
  - In prefix length notation
  - Netmasks in decimal
  - IP address ranges
- ❑ What blocks are still available (not yet allocated)?
- ❑ How big is the largest available block?

# Configuring interfaces – *ifconfig*

---

- ❑ `ifconfig interface [address_family] address [params]`
  - interface: network interface, e.g., eth0
  - options: up, down, netmask mask
  - address: IP address
  
- ❑ Examples:
  - `ifconfig eth0 192.168.2.2; ifconfig eth1 192.168.3.1`
  - `ifconfig eth0`
  - `ifconfig eth0 192.168.2.2 netmask 255.255.255.0`
  - `ifconfig eth0 inet6 2001:db8:bdbd::123 prefixlen 48 alias`

# IPv6 Addressing



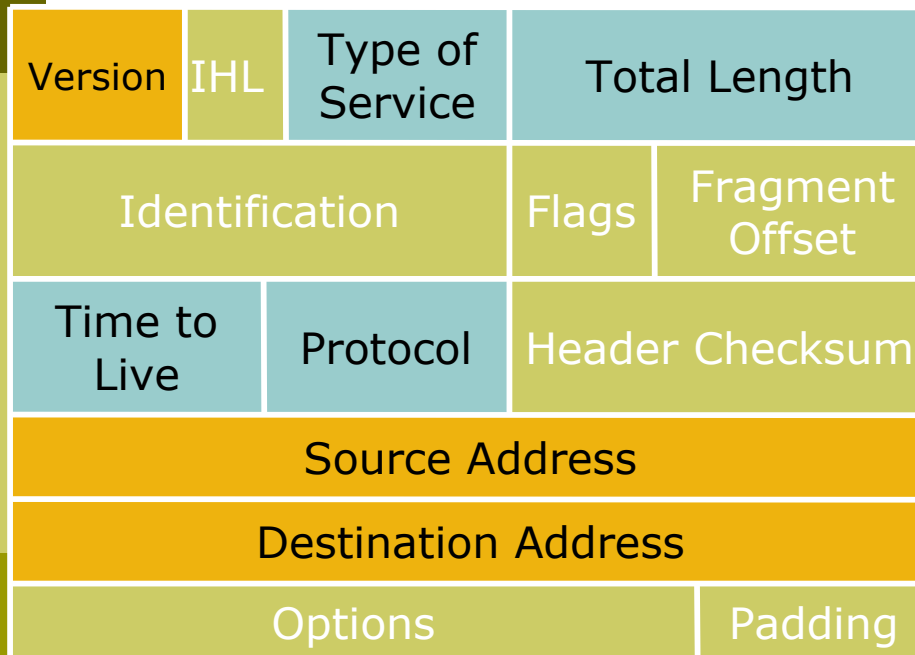
# IP version 6

---

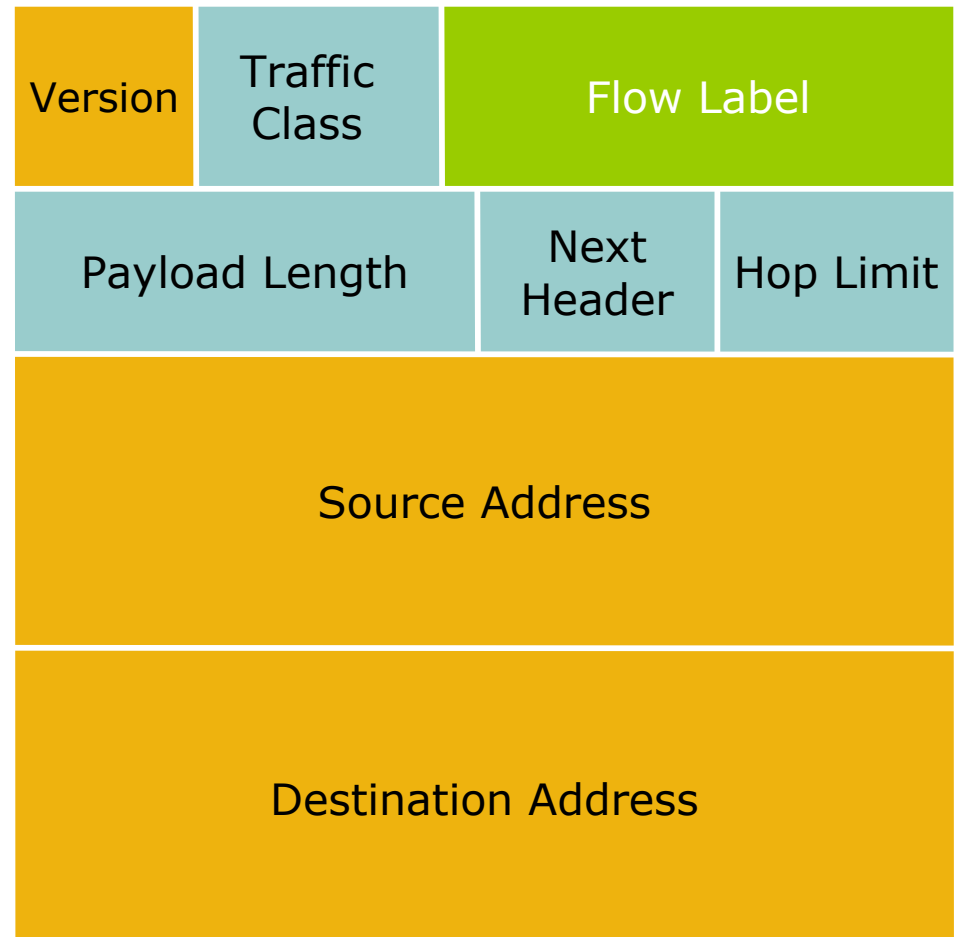
- IPv6 designed as successor to IPv4
  - Expanded address space
    - Address **length** quadrupled to 16 bytes (128 bits)
  - Header Format Simplification
    - Fixed length, optional headers are daisy-chained
  - No checksum at the IP network layer
  - No hop-by-hop fragmentation
    - Path MTU discovery
  - 64 bits aligned fields in the header
  - Authentication and Privacy Capabilities
    - IPsec is mandated
  - No more broadcast

# IPv4 and IPv6 Header Comparison

## IPv4 Header



## IPv6 Header

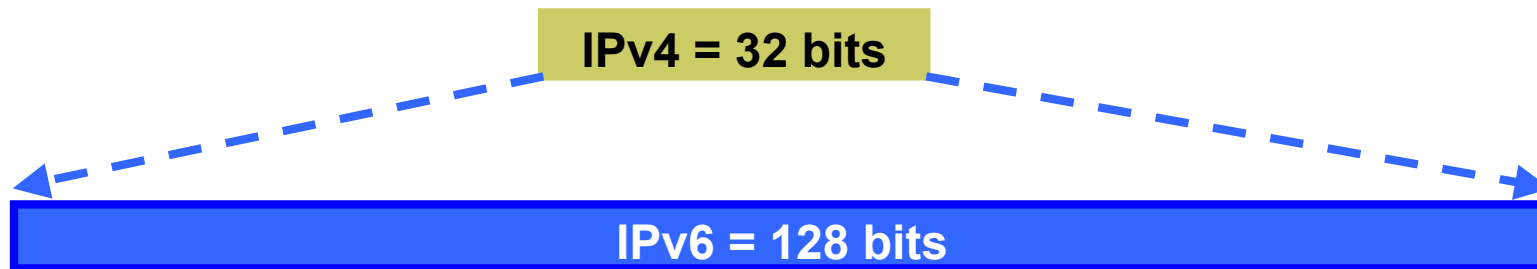


- Legend**
- Field's name kept from IPv4 to IPv6
  - Fields not kept in IPv6
  - Name and position changed in IPv6
  - New field in IPv6



# Larger Address Space


---



- IPv4
  - 32 bits
  - = 4,294,967,296 possible addressable devices
- IPv6
  - 128 bits: 4 times the size in bits
  - =  $3.4 \times 10^{38}$  possible addressable devices
  - = 340,282,366,920,938,463,463,374,607,431,768,211,456
  - ~  $5 \times 10^{28}$  addresses per person on the planet

# IPv6 Address Representation

---

- 16 bit fields in case insensitive colon hexadecimal representation
  - 2031:0000:130F:0000:0000:09C0:876A:130B
- Leading zeros in a field are optional:
  - 2031:0:130F:0:0:9C0:876A:130B
- Successive fields of 0 represented as ::, but only once in an address:
  - 2031:0:130F::9C0:876A:130B ← is ok
  - 2031::130F::9C0:876A:130B is NOT ok (two "::")
- 0:0:0:0:0:0:0:1 → ::1 (loopback address)
- 0:0:0:0:0:0:0:0 → :: (unspecified address)

# IPv6 Address Representation

---

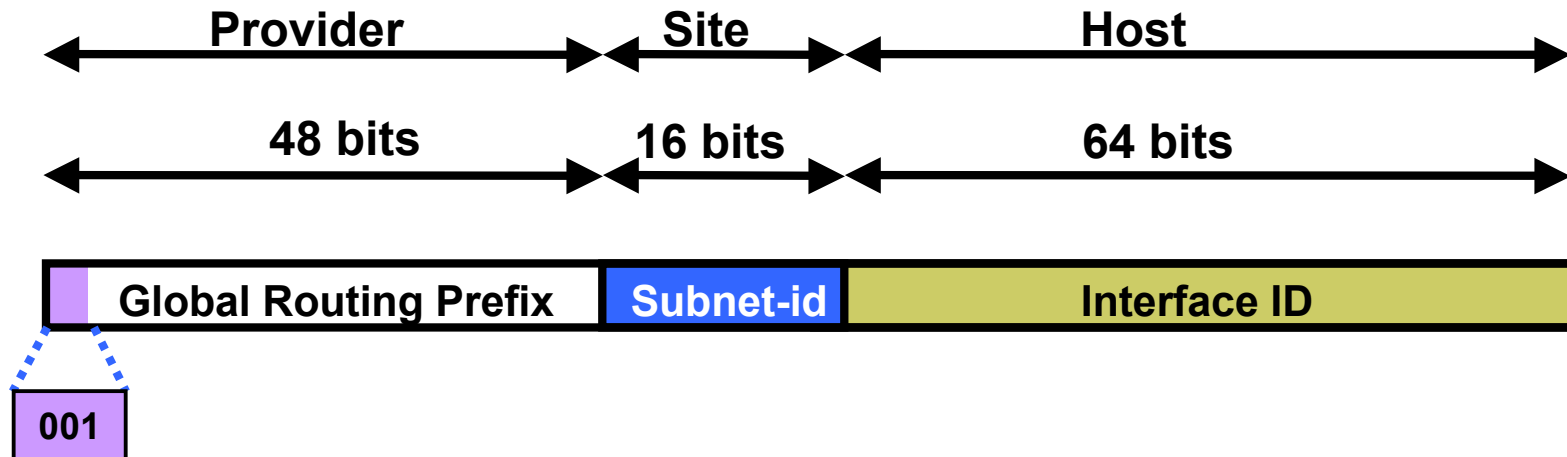
- In a URL, it is enclosed in brackets (RFC3986)
  - [http://\[2001:db8:4f3a::206:ae14\]:8080/index.html](http://[2001:db8:4f3a::206:ae14]:8080/index.html)
  - Cumbersome for users
  - Mostly for diagnostic purposes
  - Use fully qualified domain names (FQDN) instead of this
- Prefix Representation
  - Representation of prefix is same as for IPv4 CIDR
    - Address and then prefix length, with slash separator
  - IPv4 address:
    - 198.10.0.0/16
  - IPv6 address:
    - 2001:db8:12::/40

# IPv6 Addressing

Type	Binary	Hex
Unspecified	0000...0000	::/128
Loopback	0000...0001	::1/128
Global Unicast Address	0010 ...	2000::/3
Link Local Unicast Address	1111 1110 10...	FE80::/10
Unique Local Unicast Address	1111 1100 ... 1111 1101 ...	FC00::/7
Multicast Address	1111 1111 ...	FF00::/8

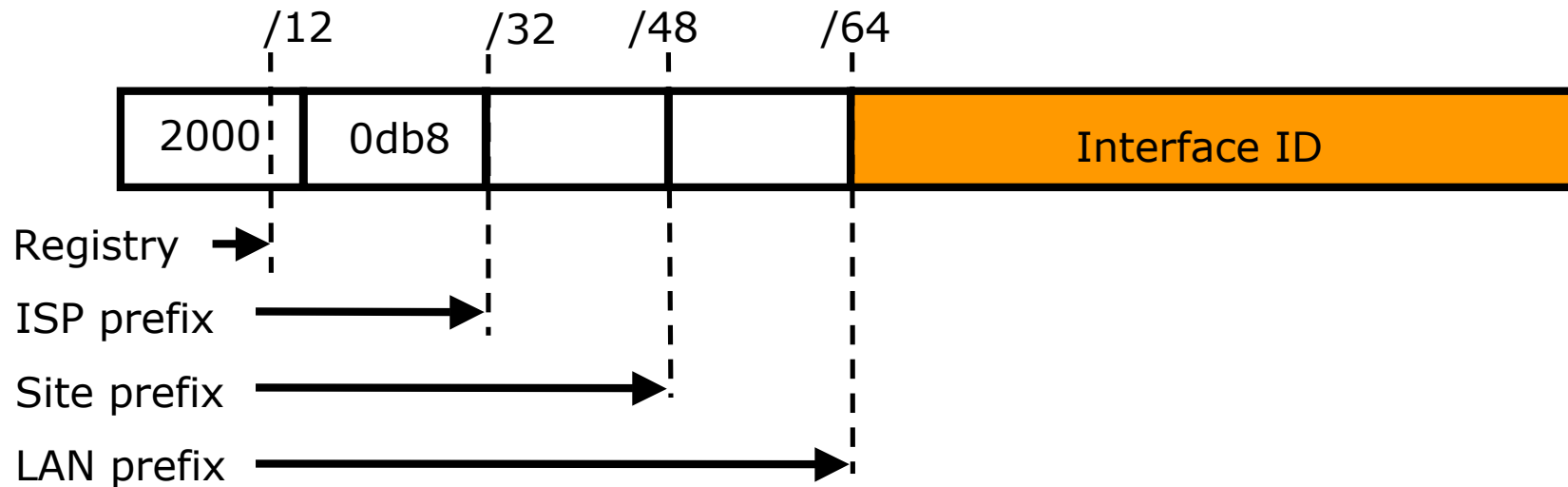
# IPv6 Global Unicast Addresses

---



- IPv6 Global Unicast addresses are:
  - Addresses for generic use of IPv6
  - Hierarchical structure intended to simplify aggregation

# IPv6 Address Allocation



- The allocation process is:
  - The IANA is allocating out of 2000::/3 for initial IPv6 unicast use
  - Each registry gets a /12 prefix from the IANA
  - Registry allocates a /32 prefix (or larger) to an IPv6 ISP
  - ISPs usually allocate a /48 prefix to each end customer

# IPv6 Addressing Scope

---

- 64 bits reserved for the interface ID
  - Possibility of  $2^{64}$  hosts on one network LAN
  - Arrangement to accommodate MAC addresses within the IPv6 address
- 16 bits reserved for the end site
  - Possibility of  $2^{16}$  networks at each end-site
  - 65536 subnets equivalent to a /12 in IPv4 (assuming 16 hosts per IPv4 subnet)

# IPv6 Addressing Scope

---

- 16 bits reserved for the service provider
  - Possibility of  $2^{16}$  end-sites per service provider
  - 65536 possible customers: equivalent to each service provider receiving a /8 in IPv4 (assuming a /24 address block per customer)
- 29 bits reserved for service providers
  - Possibility of  $2^{29}$  service providers
  - i.e. 500 million discrete service provider networks
    - Although some service providers already are justifying more than a /32
  - Equivalent to an eighth of the entire IPv4 address space



# Summary

---

- Vast address space
- Hexadecimal addressing
- Distinct addressing hierarchy between ISPs, end-sites, and LANs
  - ISPs have /32s
  - End-sites have /48s
  - LANs have /64s
- Other IPv6 features discussed later

# Large Network Issues & Routers



# The need for Packet Forwarding

---

- ❑ Many small networks can be interconnected to make a larger internetwork
- ❑ A device on one network cannot send a packet directly to a device on another network
- ❑ The packet has to be forwarded from one network to another, through intermediate nodes, until it reaches its destination
- ❑ The intermediate nodes are called “routers”

# An IP Router

---

- ❑ A device with more than one link-layer interface
- ❑ Different IP addresses (from different subnets) on different interfaces
- ❑ Receives packets on one interface, and forwards them (usually out of another interface) to get them one hop closer to their destination
- ❑ Maintains forwarding tables

# IP router - action for each packet

---

- ❑ Packet is received on one interface
- ❑ Checks whether the destination address is the router itself – if so, pass it to higher layers
- ❑ Decrement TTL (time to live), and discard packet if it reaches zero
- ❑ Look up the destination IP address in the forwarding table
- ❑ Destination could be on a directly attached link, or through another router

# Forwarding vs. Routing

---

- Forwarding: the process of moving packets from input to output
  - The forwarding table
  - Information in the packet
- Routing: process by which the forwarding table is built and maintained
  - One or more routing protocols
  - Procedures (algorithms) to convert routing info to forwarding table.
- (Much more later ...)

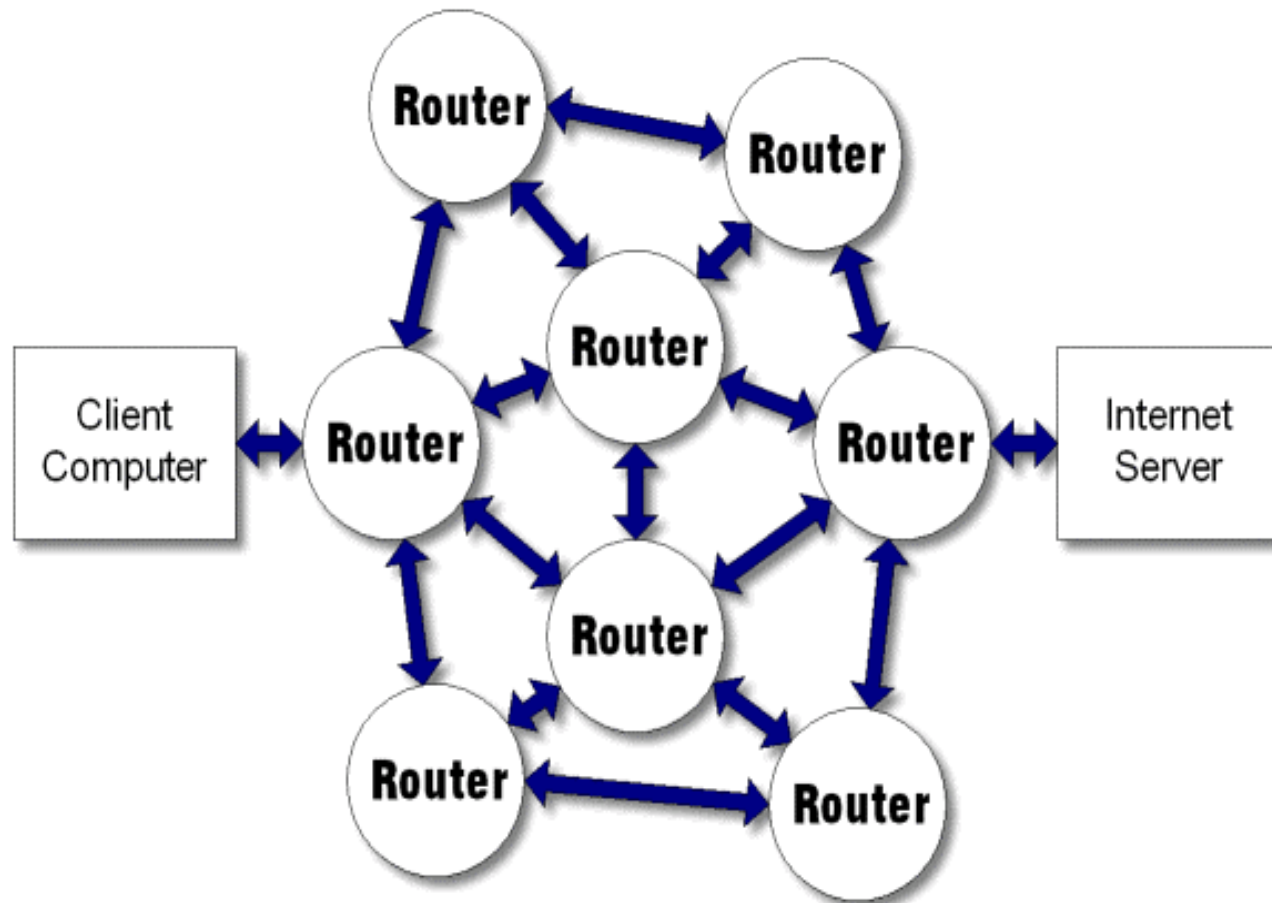
# Forwarding is hop by hop

---

- ❑ Each router tries to get the packet one hop closer to the destination
- ❑ Each router makes an independent decision, based on its own forwarding table
- ❑ Different routers have different forwarding tables and make different decisions
  - If all is well, decisions will be consistent
- ❑ Routers talk routing protocols to each other, to help update routing and forwarding tables

# Hop by Hop Forwarding

---





# Router Functions

---

- ❑ Determine optimum routing paths through a network
  - Lowest delay
  - Highest reliability
- ❑ Move packets through the network
  - Examines destination address in packet
  - Makes a decision on which port to forward the packet through
  - Decision is based on the Routing Table
- ❑ Interconnected Routers exchange routing tables in order to maintain a clear picture of the network
- ❑ In a large network, the routing table updates can consume a lot of bandwidth
  - a protocol for route updates is required

# Forwarding table structure

---

- ❑ We don't list every IP number on the Internet - the table would be huge
- ❑ Instead, the forwarding table contains prefixes (network numbers)
  - "If the first /n bits matches this entry, send the datagram thataway"
- ❑ If more than one prefix matches, the longest prefix wins (more specific route)
- ❑ 0.0.0.0/0 is "default route" - matches anything, but only if no other prefix matches

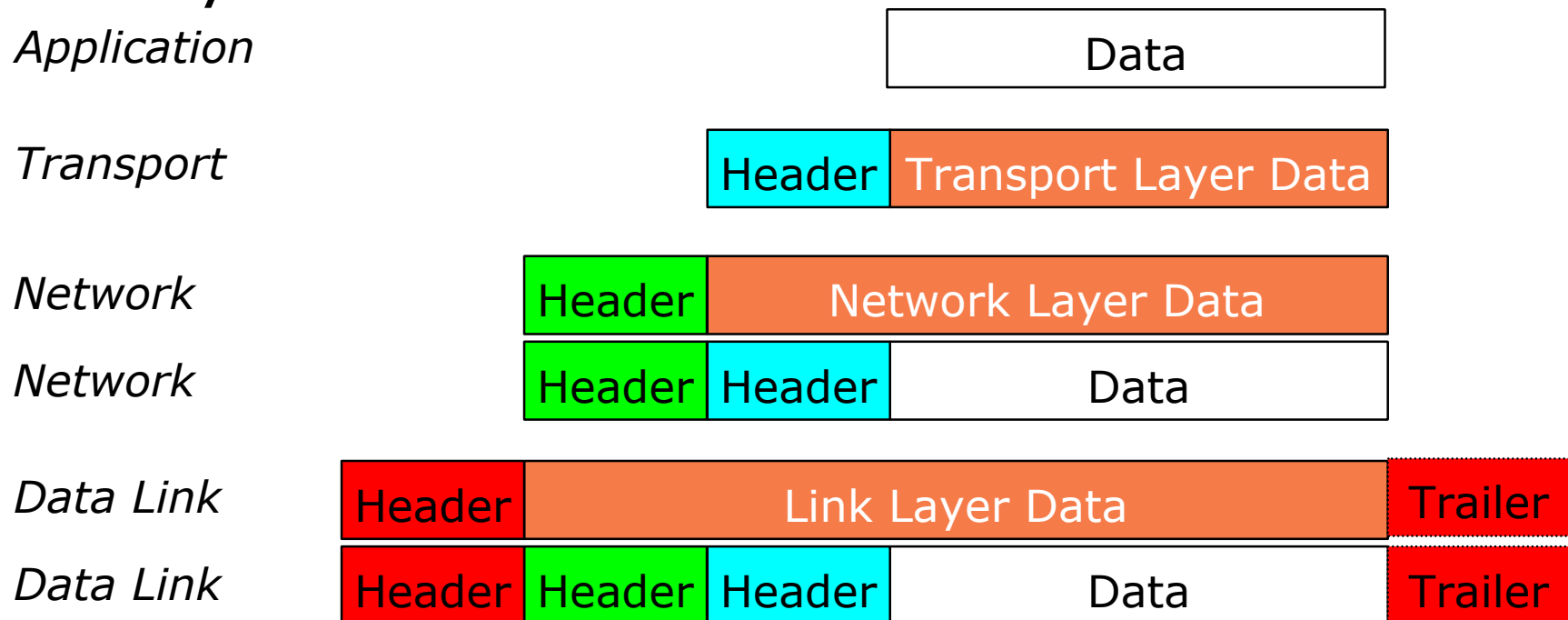
ARP



# Encapsulation Reminder

---

- Lower layers add headers (and sometimes trailers) to data from higher layers



# Ethernet Essentials

---

- ❑ Ethernet is a broadcast medium
- ❑ Structure of Ethernet frame:

Preamble	Dest	Source	Length	Type	Data	CRC
----------	------	--------	--------	------	------	-----

- ❑ Entire IP packet makes data part of Ethernet frame
- ❑ Delivery mechanism (CSMA/CD)
  - back off and try again when collision is detected

# Ethernet/IP Address Resolution

---

- Internet Address
  - Unique worldwide (excepting private nets)
  - Independent of Physical Network technology
- Ethernet Address
  - Unique worldwide (excepting errors)
  - Ethernet Only
- Need to map from higher layer to lower (i.e. IP to Ethernet, using ARP)

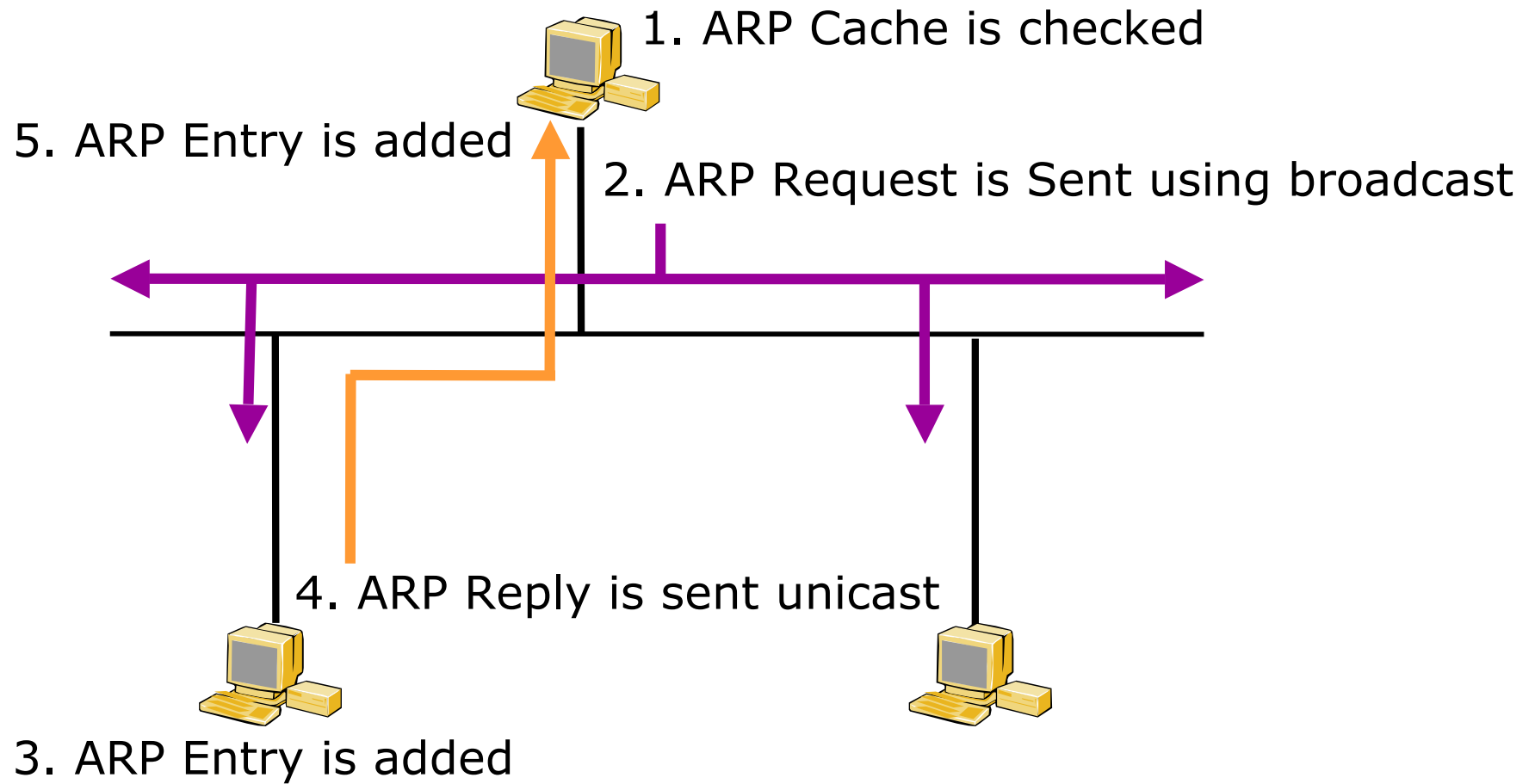
# Address Resolution Protocol

---

- ARP is only used in IPv4
  - ND replaces ARP in IPv6
- Check ARP cache for matching IP address
- If not found, broadcast packet with IP address to every host on Ethernet
- “Owner” of the IP address responds
- Response cached in ARP table for future use
- Old cache entries removed by timeout

# ARP Procedure

---





# ARP Table

---

<b>IP Address</b>	<b>Hardware Address</b>	<b>Age (Sec)</b>
192.168.0.2	08-00-20-08-70-54	3
192.168.0.65	05-02-20-08-88-33	120
192.168.0.34	07-01-20-08-73-22	43

# Types of ARP Messages

---

- ARP request
  - Who is IP addr X.X.X.X tell IP addr Y.Y.Y.Y
- ARP reply
  - IP addr X.X.X.X is Ethernet Address  
hh:hh:hh:hh:hh:hh

# Summary



# IP and Networking Basics

---

- A little bit of history
- The TCP/IP Stack
- IP Addressing
- IPv6 Addressing
- Large Network Issues & Routers
- ARP