IP and Networking Basics

Scalable Infrastructure Workshop

AfNOG 2008

Outline

- Origins of TCP/IP
- OSI Stack & TCP/IP Architecture
- IP Addressing
- IPv6
- Large Network Issues
- Routers
- Types of Links
- Address Resolution Protocol

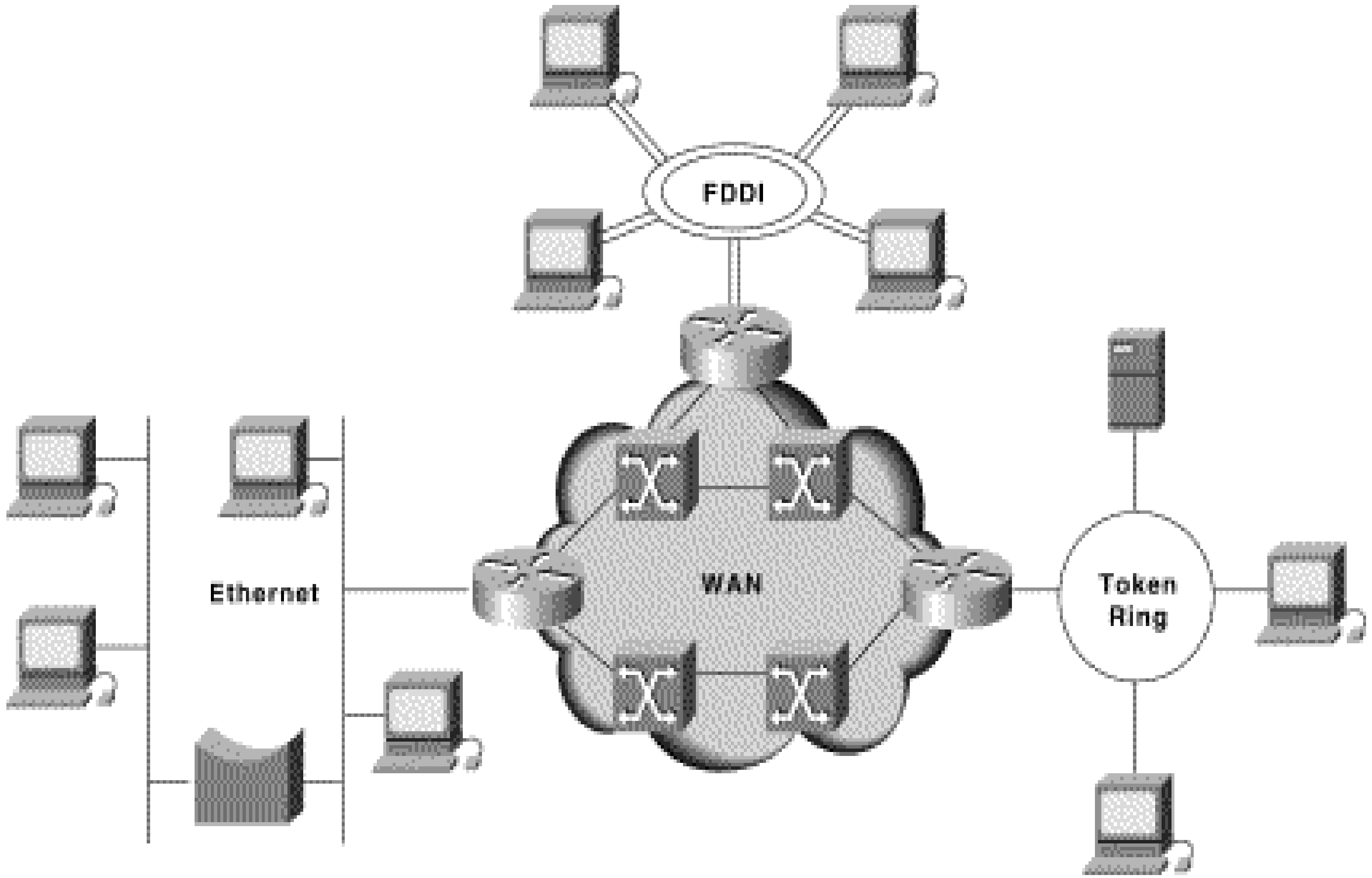
Origins of TCP/IP

- RAND Corporation (a “think tank”) & DoD formed ARPA (Advanced Research Project Agency)
- 1968 – ARPA engineers proposed Distributed network design for ARPANET Network

Distributed Network Design

- Pre-ARPANET networks
  - “connection orientated”
  - Management & control was centralized
- “New” Network – ARPANET
  - Connectionless
  - Decentralised
- Modern Internet has evolved from the ARPANET

A small internetwork or (small "i") "internet"



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

The principle of “Internetworking”

- We have lots of little networks
- Many different owners/operators
- Many different types
    - Ethernet, dedicated leased lines, dialup, ATM, Frame Relay, FDDI
- Each type has its own idea of 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)

OSI Stack & TCP/IP Architecture

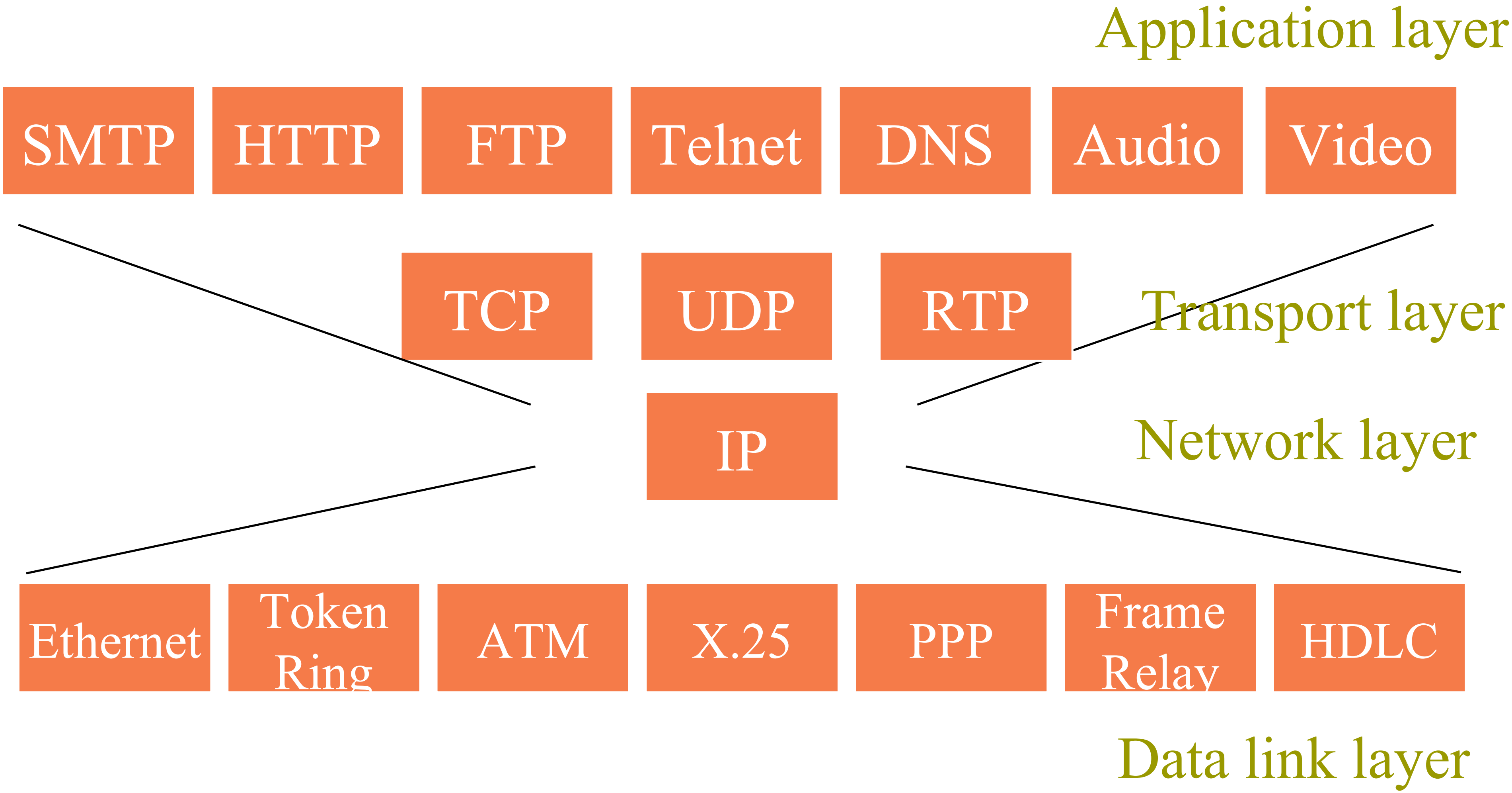
What is TCP/IP?

- In simple terms is a language that enables communication between computers
- A set of rules (protocol) that defines how two computers address each other and send data to each other
- Is a suite of protocols named after the two most important protocols TCP and IP; but includes other protocols such as UDP, RTP, etc.

Open Systems & TCP/IP

- TCP/IP formed from standardized communications procedures that were platform independent and open
- Open systems
  - open architecture - readily available to all
- What is open system networking?
  - network based on well known and standardized protocols
  - standards readily available
  - networking open systems using a network protocol

Protocol Layers: The TCP/IP Hourglass Model



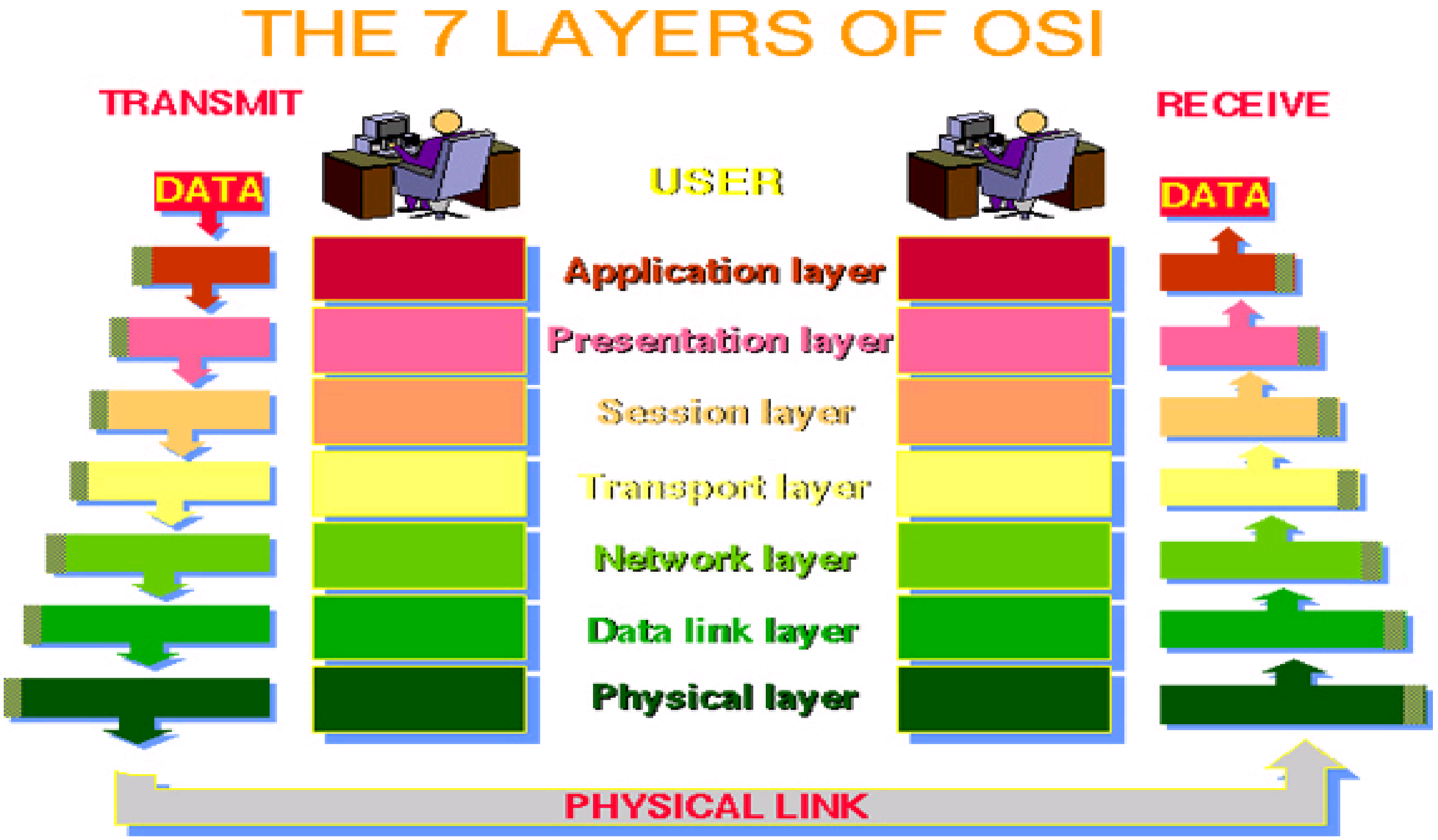
The unifying effect of the network layer

- 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
- There are defined ways of using:
  - IP over Ethernet
  - IP over ATM
  - IP over FDDI
  - IP over serial lines (PPP)
  - IP over almost anything

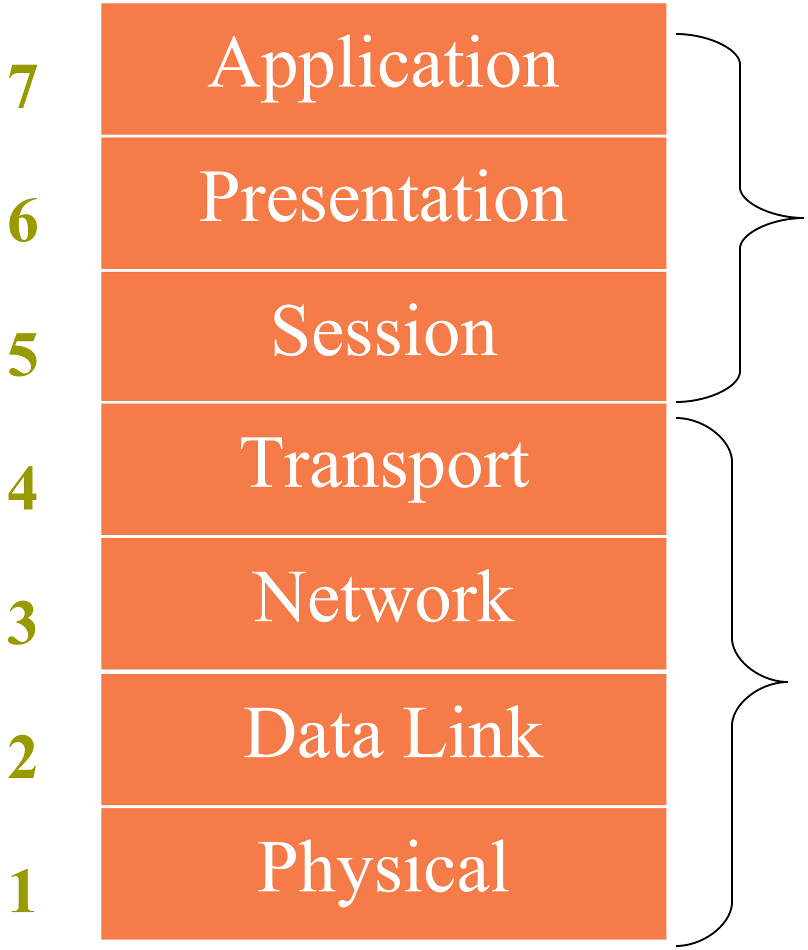
OSI - Layered Model Concept

- ISO Open Systems Interconnection Reference Model (OSI-RM) adopted as a standard for networking
- Divide-and-conquer approach
- Dividing requirements into groups, e.g. transporting of data, packaging of messages, end user applications
- Each group can be referred to as a layer
  - Upper layers are logically closer to the user and deal with more abstract data, relying on lower layer protocols to translate data into forms that can eventually be physically transmitted.

OSI Model


THE 7 LAYERS OF OSI
TRANSMIT
RECEIVE
DATA
USER
DATA
Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer
PHYSICAL LINK

OSI Model



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

Upper Layers
Application oriented
(Almost) independent of layers below

Lower Layers
Transmission of data
Independent of upper layers

Layers 7, 6, 5

- 7: Application layer
  - Provides different services to the applications
  - Uses the underlying layers to carry out work
    - e.g. SMTP (mail), HTTP (web), Telnet, FTP, DNS
- 6: Presentation layer
  - Converts data from applications into common format and vice versa
- 5: Session layer
  - organizes and synchronizes the exchange of data between application processes

Layer 4

- 4: Transport layer
  - Provides end to end transportation of segments
  - E.g. TCP
    - encapsulates TCP segments in network layer packets
    - adds reliability by detecting and retransmitting lost packets
    - uses acknowledgements and sequence numbers to keep track of successful, out-of-order, and lost packets
    - timers help differentiate between loss and delay
  - UDP is much simpler: no reliability features

Layer 3

- 3: Network layer
  - Routes the information in the network
  - E.g. IP is a network layer implementation which defines addresses in such a way that route selection can be determined.
    - Single address space for the entire internetwork
    - adds an additional layer of addressing, e.g. IP address (at the network layer) is different from MAC address (at the data link layer).

Layer 3

- 3: Network layer (e.g. IP)
  - Unreliable (best effort)
    - if packet gets lost, network layer doesn’t care for higher layers can resend lost packets
  - Forwards packets hop by hop
    - encapsulates network layer packet inside data link layer frame
    - different framing on different underlying network types
    - receive from one link, forward to another link
    - There can be many hops from source to destination

Layer 3

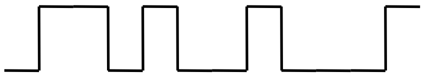
- 3: Network layer (e.g. IP)
  - Makes routing decisions
    - how can the packet be sent closer to its destination?
    - forwarding and routing tables embody “knowledge” of network topology
    - routers can talk to each other to exchange information about network topology

Layer 2

- 2: Data Link layer
  - Provides reliable transit of data across a physical network link
  - bundles bits into frames and moves frames between hosts on the same link
  - a frame has a definite start, end, size
  - often also a definite source and destination link-layer address (e.g. Ethernet MAC address)
  - some link layers detect corrupted frames while other layers re-send corrupted frames (NOT Ethernet)

Layer 1

- 1: Physical layer
  - moves bits using voltage, light, radio, etc.
  - no concept of bytes or frames
  - bits are defined by voltage levels, or similar physical properties



1101001000

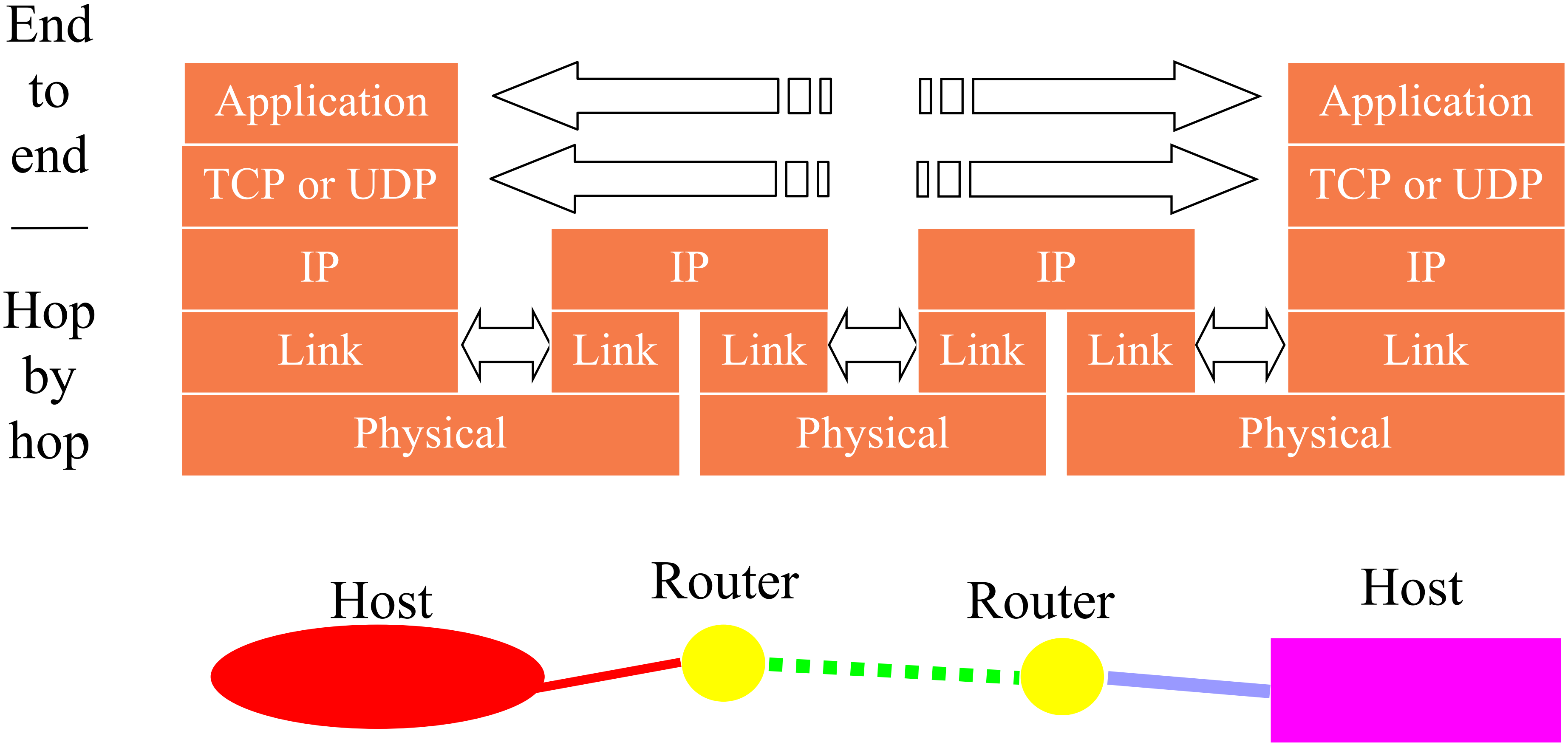
Corresponding layers in the OSI and TCP/IP models

	OSI	TCP/IP	
7	Application	Application	*Mail, Web, etc.*
6	Presentation		
5	Session		
4	Transport	Transport	*TCP/UDP – end to end reliability*
3	Network	Network	*IP - Forwarding (best-effort)*
2	Data Link	Data Link & Physical	*Framing, delivery*
1	Physical		*Raw signal*

Layer Interaction

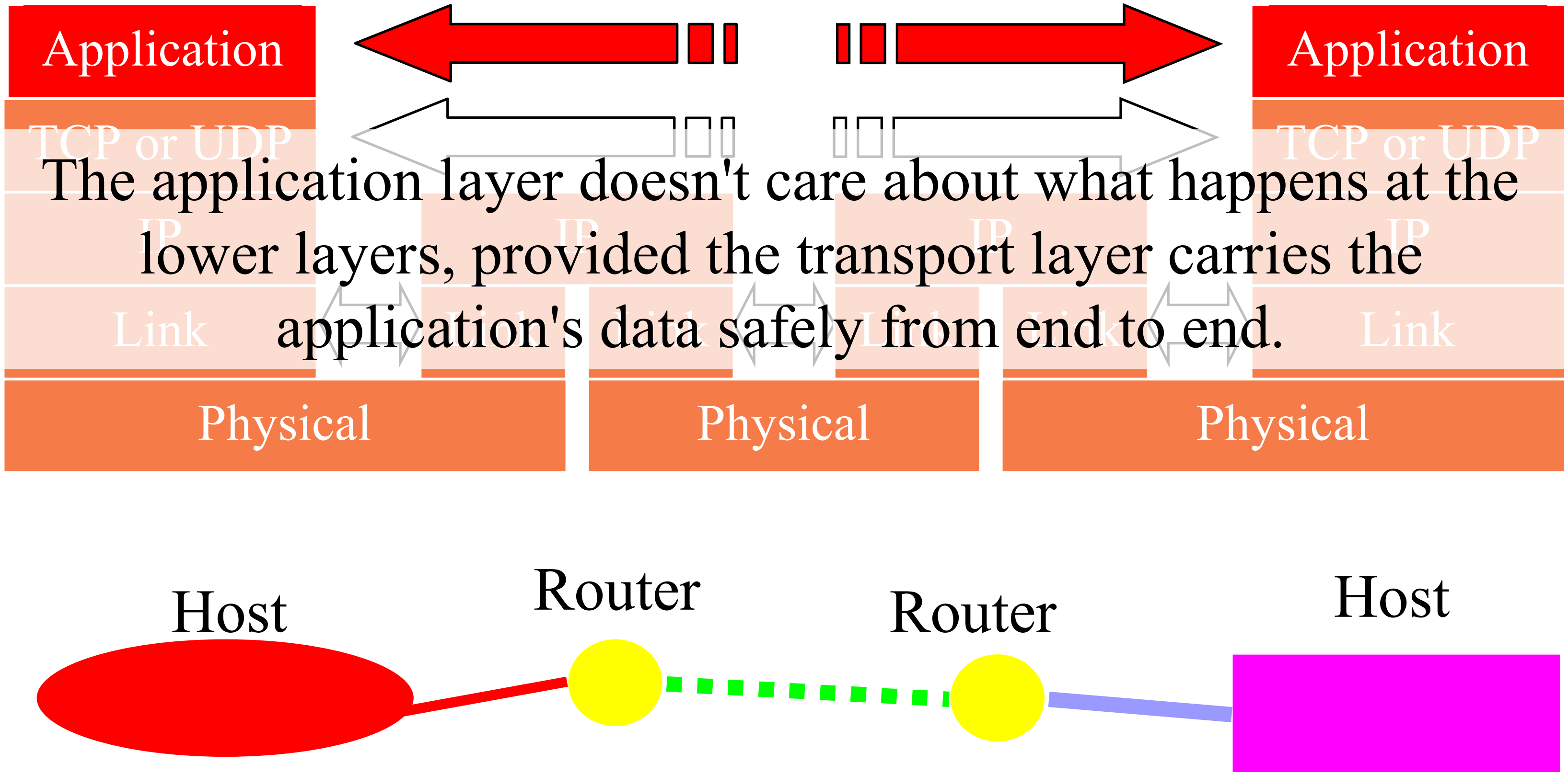
- Application, Presentation and Session protocols are end-to-end
- Transport protocol is end-to-end
  - encapsulation/decapsulation over network protocol on end systems
- Network protocol is throughout the internetwork
  - encapsulation/decapsulation over data link protocol at each hop
- Link and physical layers may be different on 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.

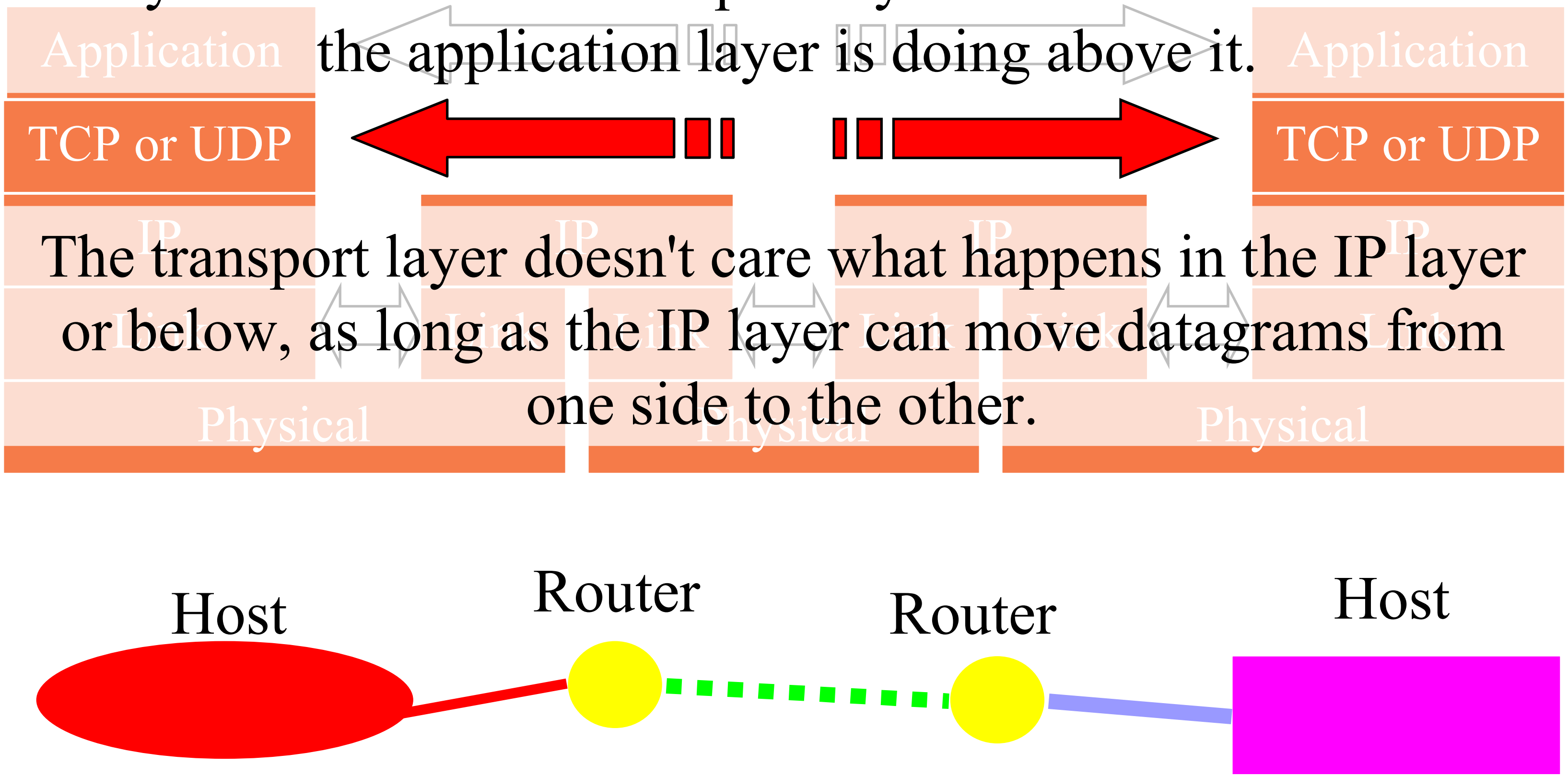


The application layer doesn't care about what happens at the lower layers, provided the transport layer carries the application's data safely from end to end.

Layer Interaction: The Transport Layer

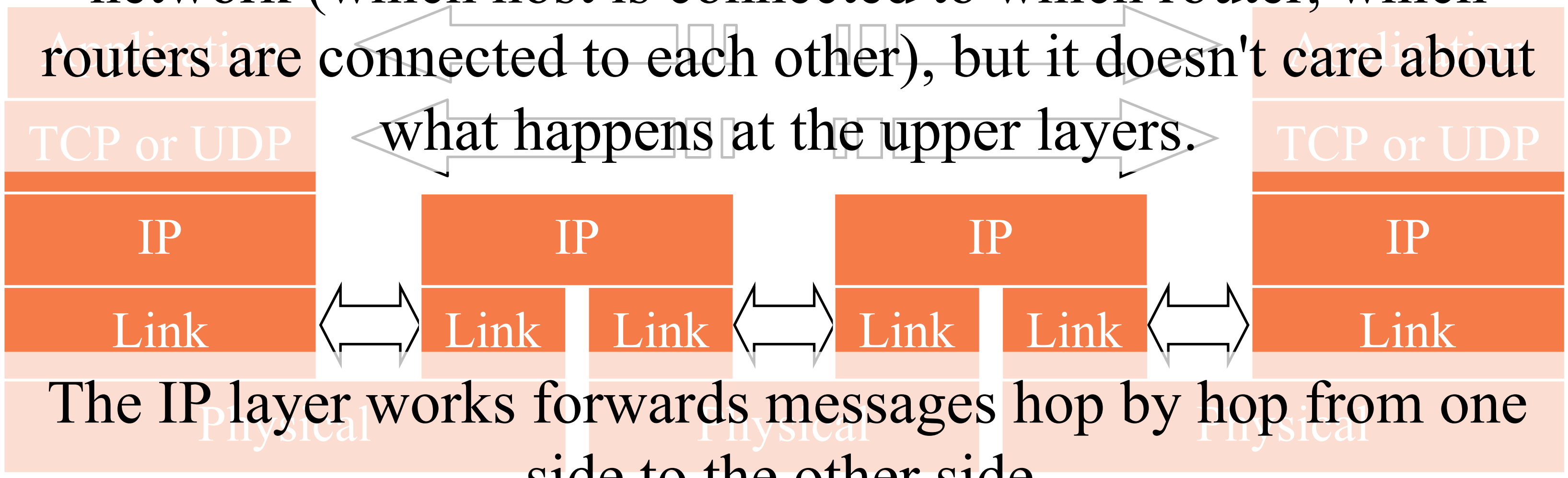
The transport layer instances at the two ends act as if they are talking to each other, but in reality they are each talking to the IP layer below it. The transport layer doesn't care about what the application layer is doing above it.

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.

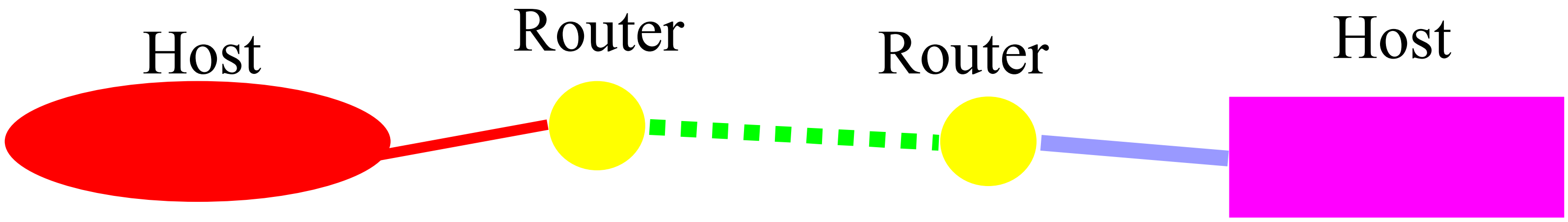


Layer Interaction: The IP Layer

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.



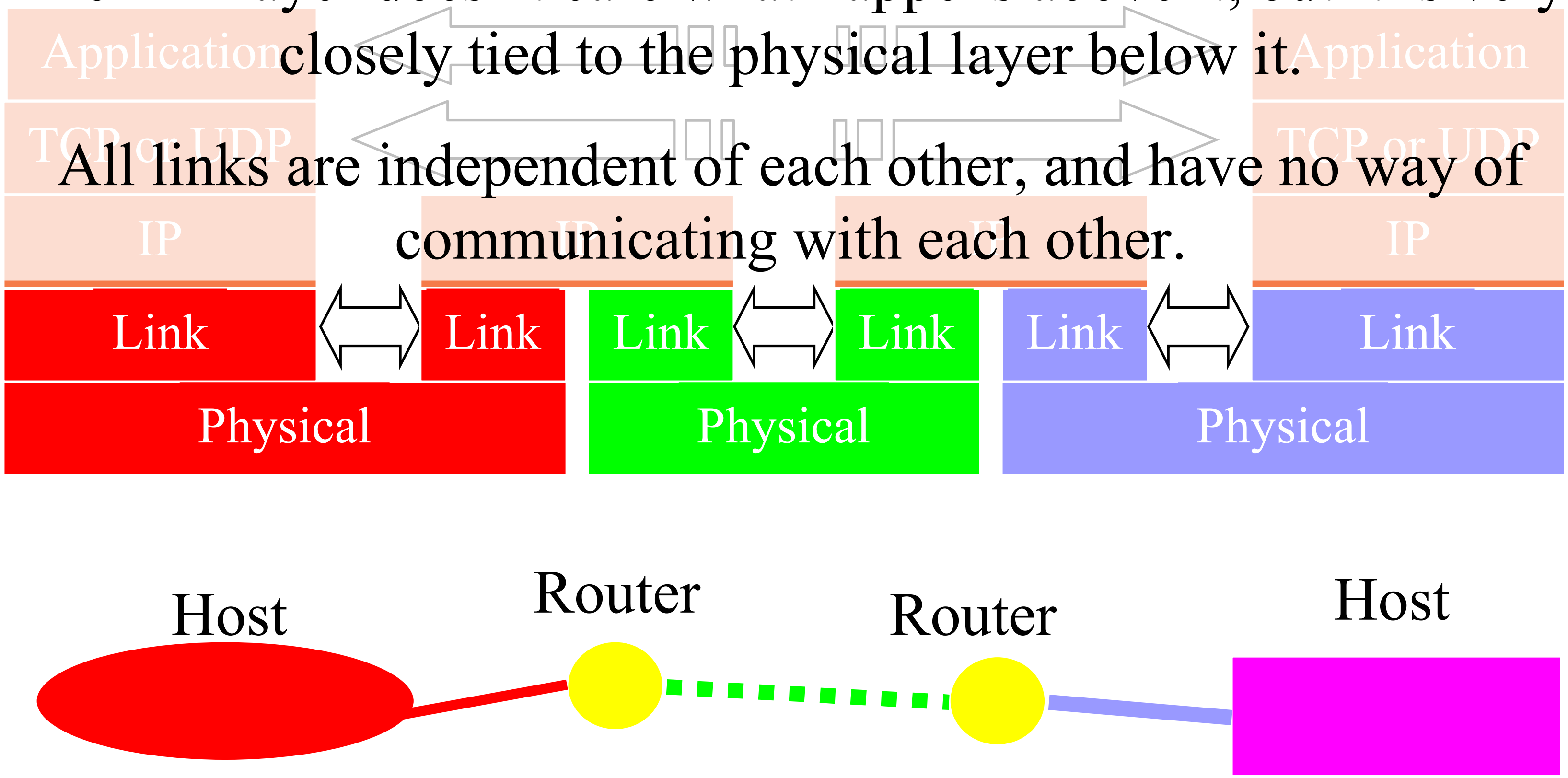
The IP layer works forwards messages hop by hop from one side to the other side.



Layer Interaction:
The Link and Physical Layers

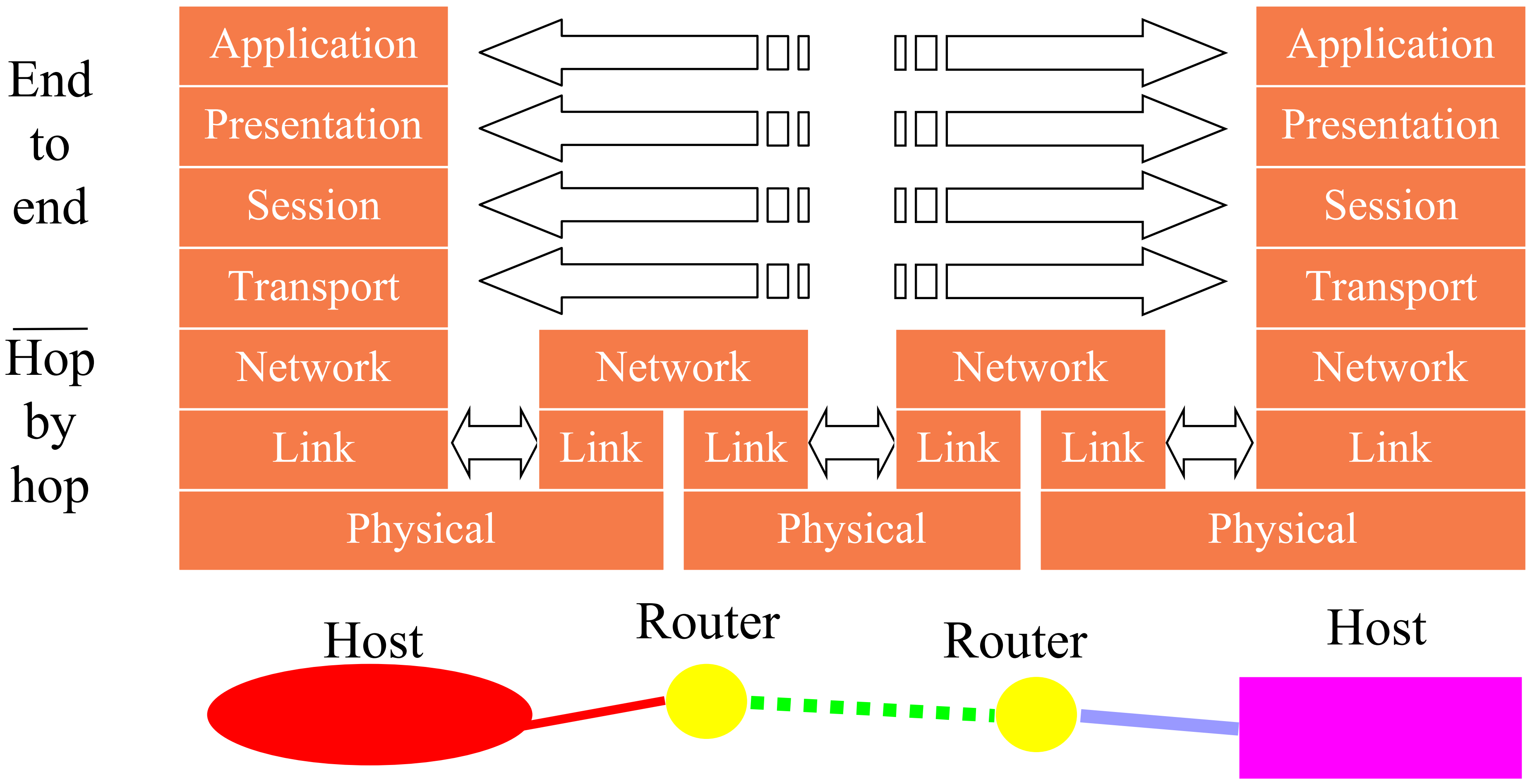
The link layer doesn't care what happens above it, but it is very closely tied to the physical layer below it.

All links are independent of each other, and have no way of communicating with each other.



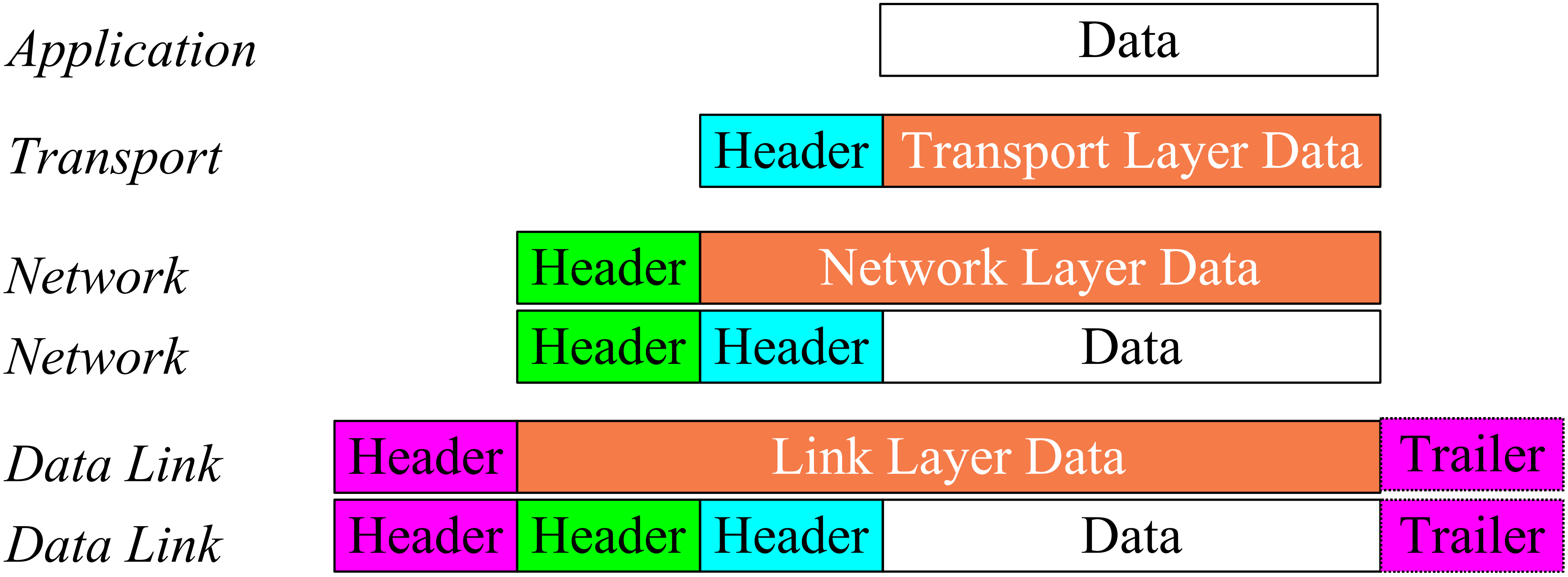
Layer Interaction: OSI 7-Layer Model

The same idea, just more layers



Encapsulation & Decapsulation

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



Application Data

Transport Header Transport Layer Data

Network Header Network Layer Data

Network Header Header Data

Data Link Header Link Layer Data Trailer

Data Link Header Header Header Data Trailer

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

Layer 2 - Ethernet frame

Preamble	Dest	Source	Length	Type	Data	CRC
	6 bytes	6 bytes	2 bytes	2 bytes	46 to 1500 bytes	4 bytes

- Destination and source are 48-bit MAC addresses
- 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.

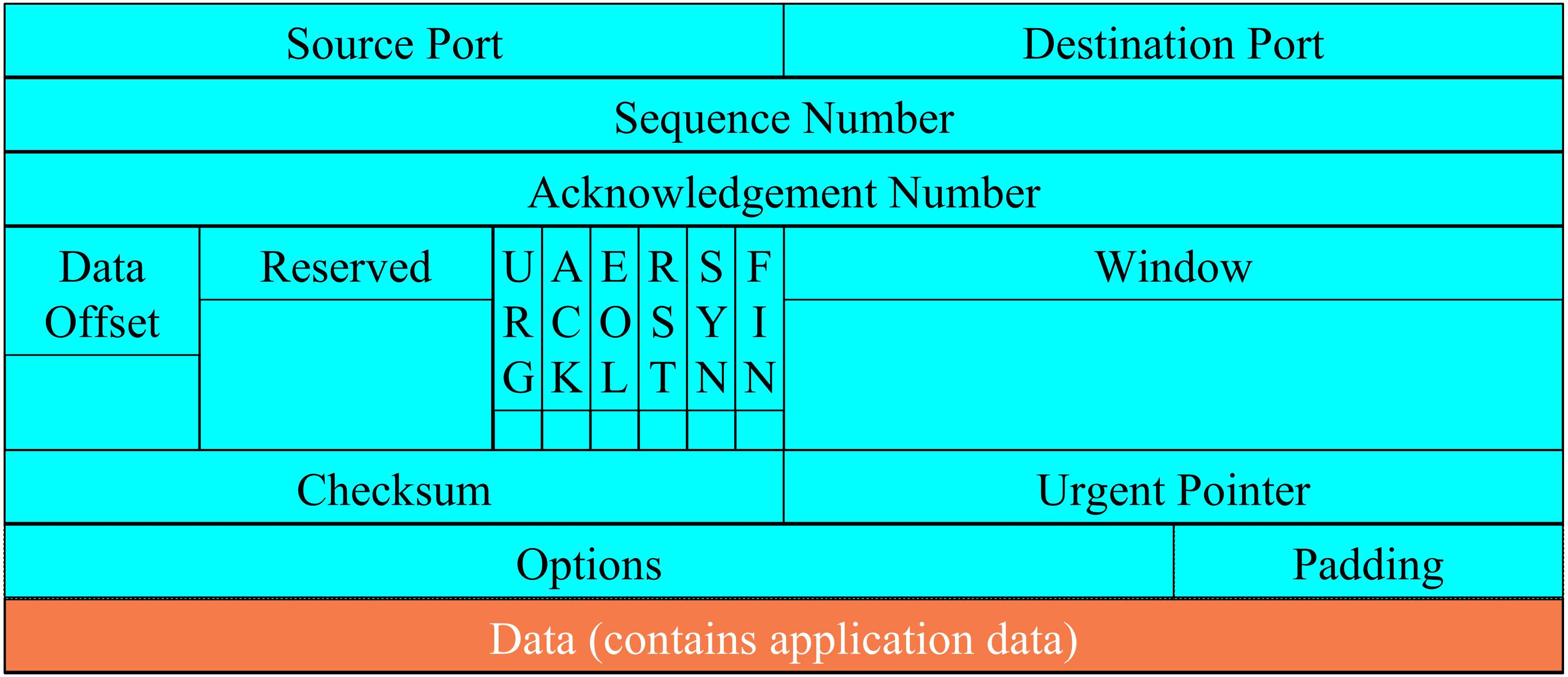
Layer 3 - IPv4 datagram

<table>
<tr><td>Version</td><td>IHL</td><td colspan="2">Type of Service</td><td colspan="4">Total Length</td></tr>
<tr><td colspan="4">Identification</td><td>Flags</td><td colspan="3">Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td colspan="2">Protocol</td><td colspan="4">Header Checksum</td></tr>
<tr><td colspan="8">Source Address (32-bit IPv4 address)</td></tr>
<tr><td colspan="8">Destination Address (32-bit IPv4 address)</td></tr>
<tr><td colspan="6">Options</td><td colspan="2">Padding</td></tr>
<tr><td colspan="8">Data (contains layer 4 segment)</td></tr>
</table>

- 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

URG

ACK

EOL

RST

SYN

FIN

Window

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 IPv4 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 IPv4 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 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

Basic Structure of an IPv4 Address

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

133	27	162	125

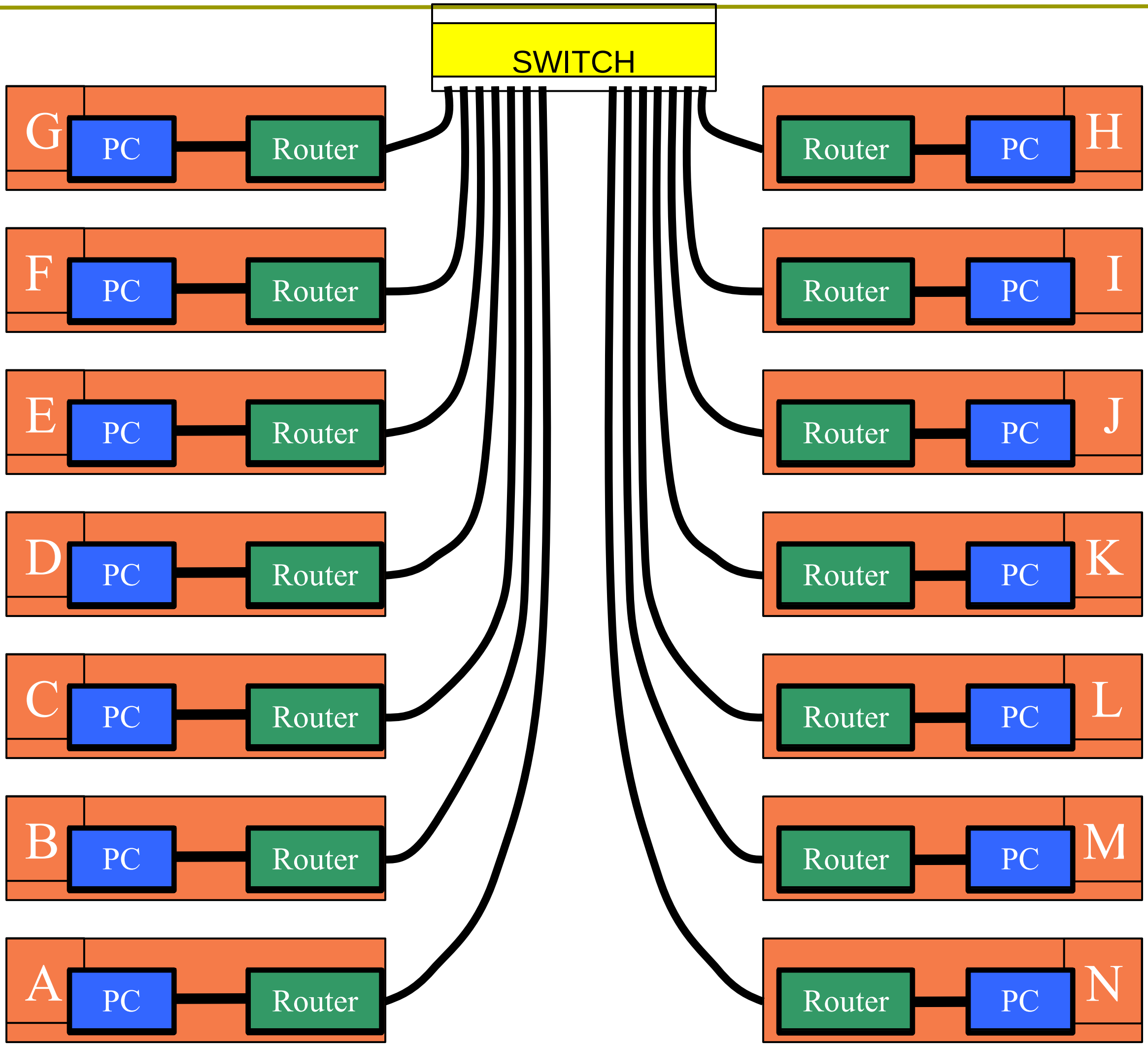
- Binary Representation:

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 computers
- Need structure in IP addresses
  - "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.

Address Structure Revisited

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

205 . 154 . 8	1
11001101 10011010 00001000	00000001
Network	Host

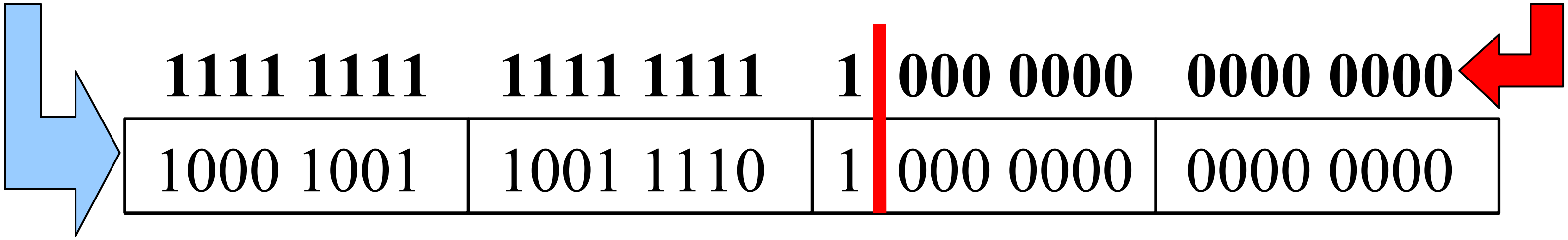
  - Boundary can be anywhere
    - very often NOT at a multiple of 8 bits
    - choose the boundary according to number of hosts

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
- Binary AND of 32 bit IP address with 32 bit netmask yields network part of address

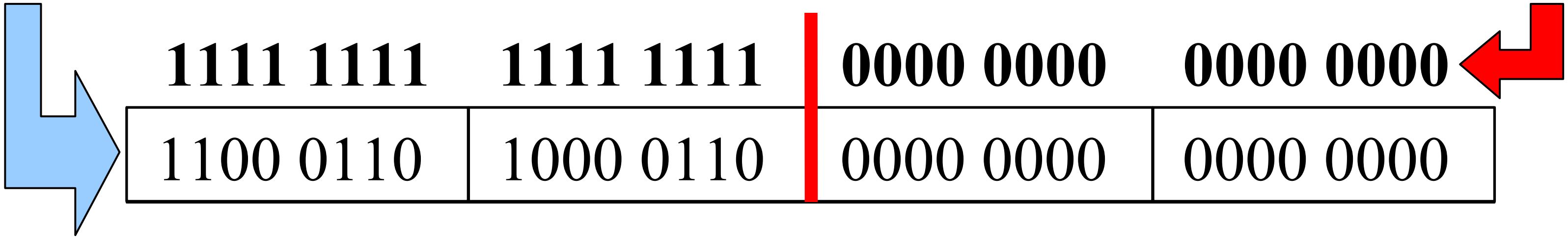
Example Prefixes

- 137.158.128.0/17 (netmask 255.255.128.0)



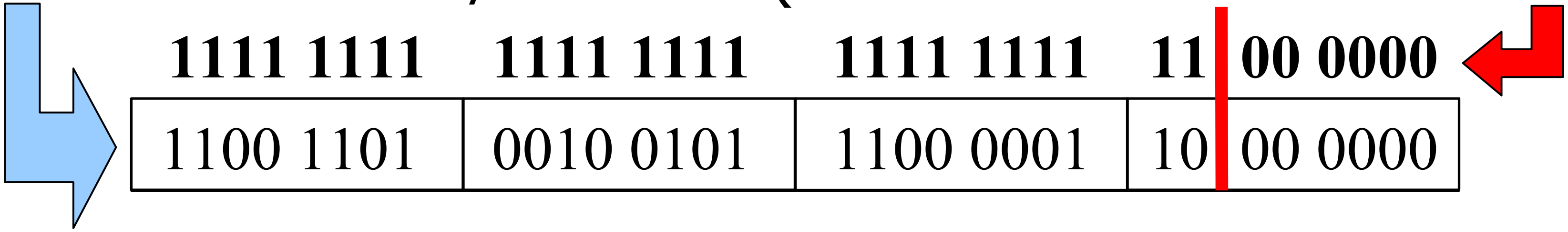
1111 1111	**1111 1111**	**1 000 0000**	**0000 0000**
1000 1001	1001 1110	1 000 0000	0000 0000

- 198.134.0.0/16 (netmask 255.255.0.0)



1111 1111	**1111 1111**	**0000 0000**	**0000 0000**
1100 0110	1000 0110	0000 0000	0000 0000

- 205.37.193.128/26 (netmask 255.255.255.192)



1111 1111	**1111 1111**	**1111 1111**	**11 00 0000**
1100 1101	0010 0101	1100 0001	10 00 0000

Special Addresses

- All 0’s in host part: Represents Network
  - e.g. 193.0.0.0/24
  - e.g. 138.37.128.0/17
- All 1’s in host part: Broadcast
  - e.g. 137.156.255.255 (137.156.0.0/16)
  - e.g. 134.132.100.255 (134.132.100.0/24)
  - e.g. 190.0.127.255   (190.0.0.0/17)
- 127.0.0.0/8: Loopback address (127.0.0.1)
- 0.0.0.0: Various special purposes

Allocating IP Addresses

- The subnet mask is used to define size of a network
- 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

- Assuming there are 15 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

- We can also group several networks into a larger block, or divide a large block into several smaller blocks
  - 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
- Old systems used restrictive rules (obsolete)
  - Called “Class A”, “Class B”, “Class C” networks
- These days (since 1994), no restriction
  - Called “classless”

A little History: Classes of IP addresses

- Different classes were used to represent different sizes of network (small, medium, large)
- Class A networks (large):
  - 8 bits network, 24 bits host (/8, 255.0.0.0)
  - First byte of IP address in range 0-127
- Class B networks (medium):
  - 16 bits network, 16 bits host (/16 ,255.255.0.0)
  - First byte of IP address in range 128-191
- Class C networks (small):
  - 24 bits network, 8 bits host (/24, 255.255.255.0)
  - First byte of IP address in range 192-223

A little History: Classes of IP addresses

- Just look at the address to tell what class it is.
  - Class A: 0.0.0.0 to 127.255.255.255
    - binary 0xxxxxxxhhhhhhhhhhhhhhhhhhhhhhhh
  - Class B: 128.0.0.0 to 191.255.255.255
    - binary 10xxxxxxxxxxxxxxhhhhhhhhhhhhhhhh
  - Class C: 192.0.0.0 to 223.255.255.255
    - binary 110xxxxxxxxxxxxxxxxxxxxxhhhhhhhh
  - Class D: (multicast) 224.0.0.0 to 239.255.255.255
    - binary 1110xxxxxxxxxxxxxxxxxxxxxxxxxxxx
  - Class E: (reserved) 240.0.0.0 to 255.255.255.255

Netmasks of classful addresses

- 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. It needs explicit configuration.

Traditional subnetting of classful networks

- Old routing systems allowed a classful network to be divided into subnets
  - All subnets (of the same classful net) had to be the same size and have the same netmask
  - Subnets could not be divided into sub-sub-nets
- None of these restrictions apply in modern systems
- You should never use old routing systems that have these restrictions (e.g. RIP version 1)

Traditional Supernetting

- Some traditional routing systems allowed supernets to be formed by combining adjacent classful nets.
  - e.g. combine two Class C networks (with consecutive numbers) into a supernet with netmask 255.255.254.0
- Modern systems use more general classless mechanisms
  - Today we'd talk about a "/23 prefix", not a "supernet".

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 class A, B, C implies prefix length /8, /16, /24
- VLSM = Variable-Length Subnet Masks
  - routing does not assume that all subnets are the same size

Classless Addressing

- IP address with the subnet mask defines the range of addresses in the block
  - E.g 10.1.1.32/28 (subnet mask 255.255.255.240) defines the range 10.1.1.32 to 10.1.1.47
    - 28-bit network part, 4-bit host part
  - 10.1.1.32 is the network address
    - the host part, in binary, is all zeros
  - 10.1.1.47 is the broadcast address
    - the host part, in binary, is all ones
  - 10.1.1.33 to 10.1.1.46 are available for use as host addresses in this subnet

Classless addressing example

- A large ISP gets a large block of addresses
  - e.g., a /16 prefix, or 65536 separate addresses
- Allocate smaller blocks to customers
  - e.g., a /22 prefix (1024 addresses) to one customer, and a /28 prefix (16 addresses) to another customer
- 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

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?

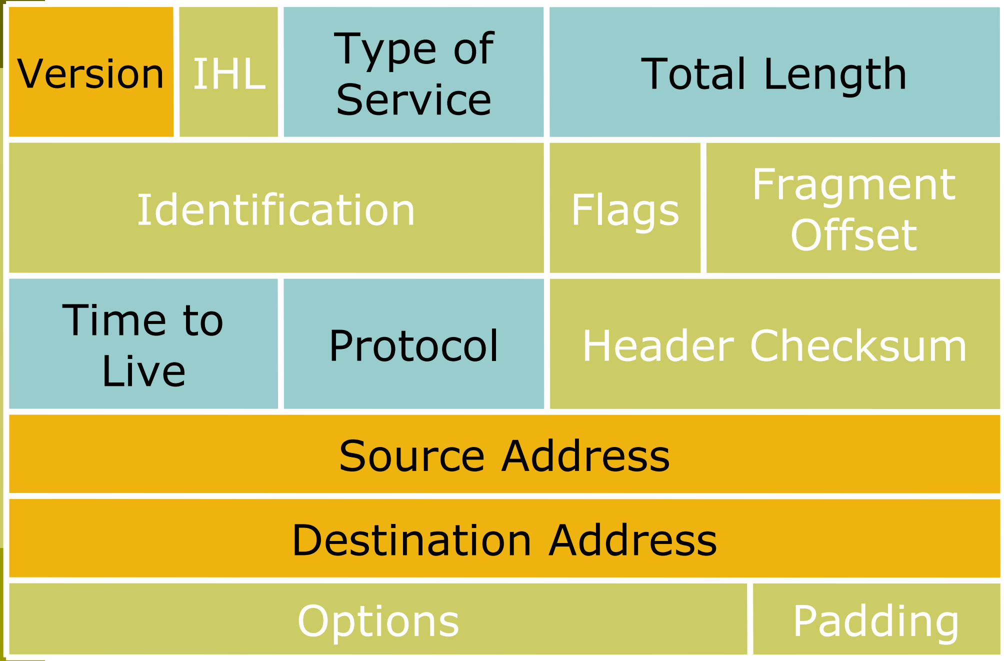
IPv6

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

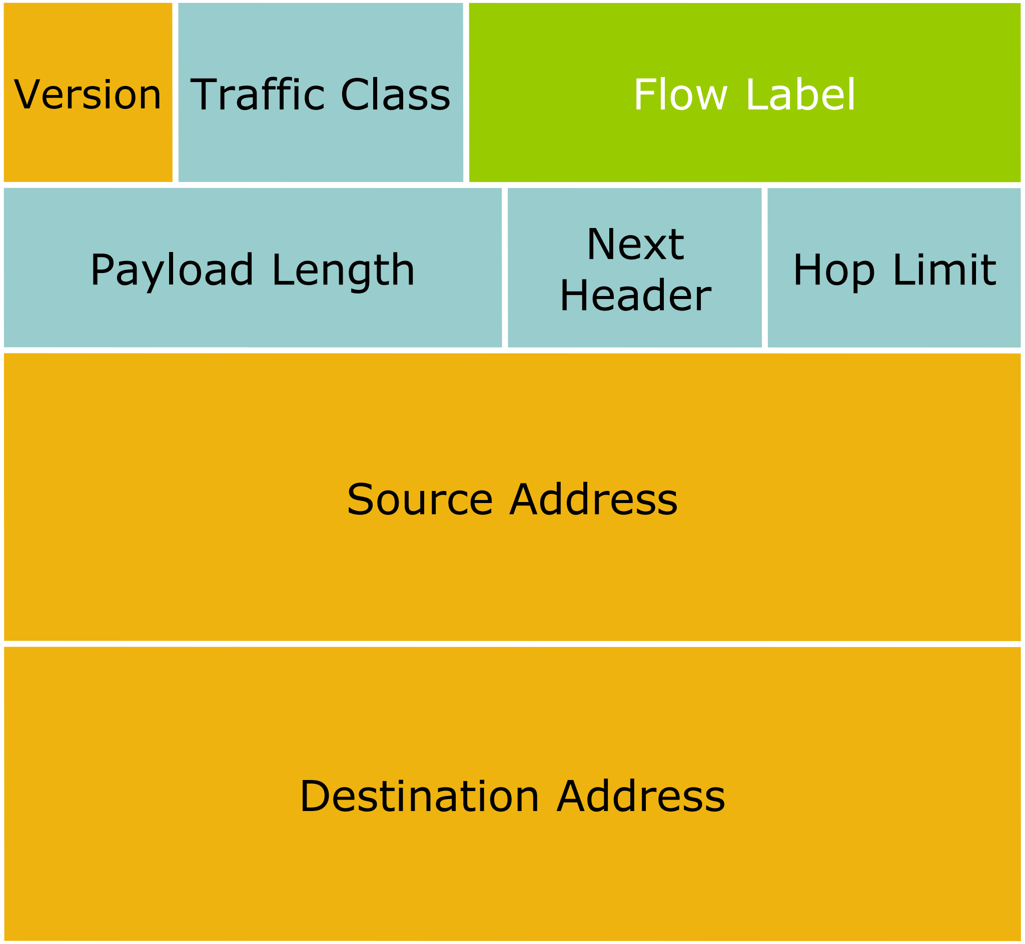
IPv4 and IPv6 Header Comparison

IPv4 Header



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

IPv6 Header

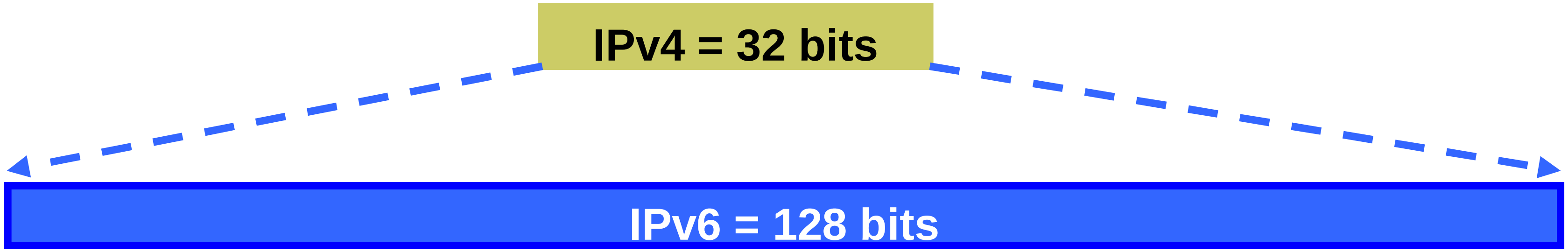


Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address

Legend

- 🟧 **Field's name kept from IPv4 to IPv6**
- 🟨 **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 x 10^{38} possible addressable devices
  - = 340,282,366,920,938,463,463,374,607,431,768,211,456
  - ~ 5 x 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)

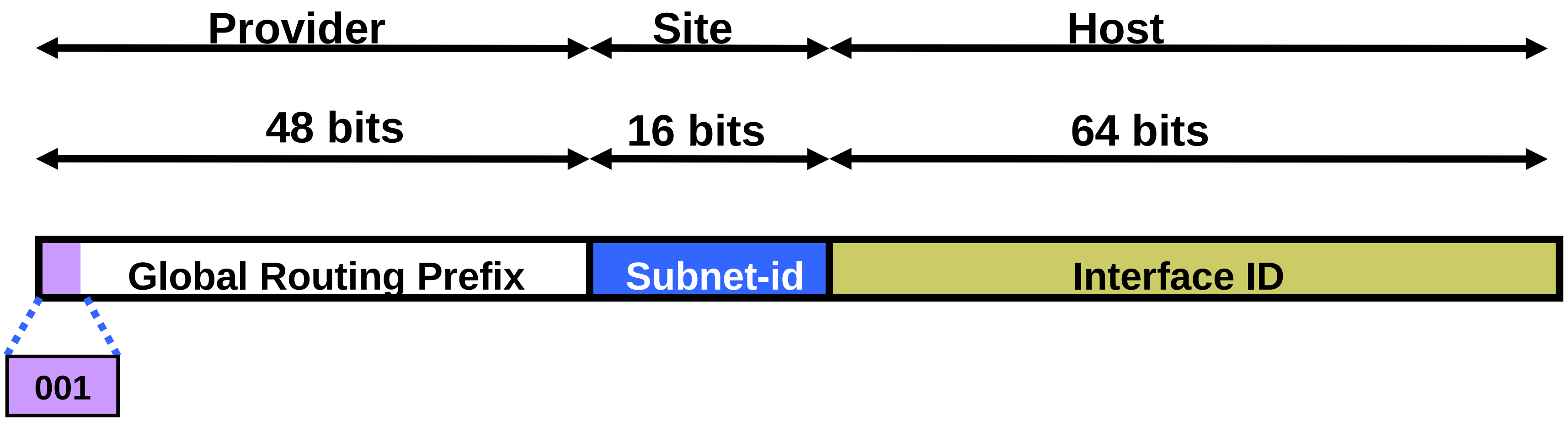
IPv6 Address Representation

- In a URL, it is enclosed in brackets (RFC3986)
  - http://[2001:db8:4f3a::206:ae14]:8080/index.html
  - Cumbersome for users
  - Mostly for diagnostic purposes
  - Use fully qualified domain names (FQDN)
- 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

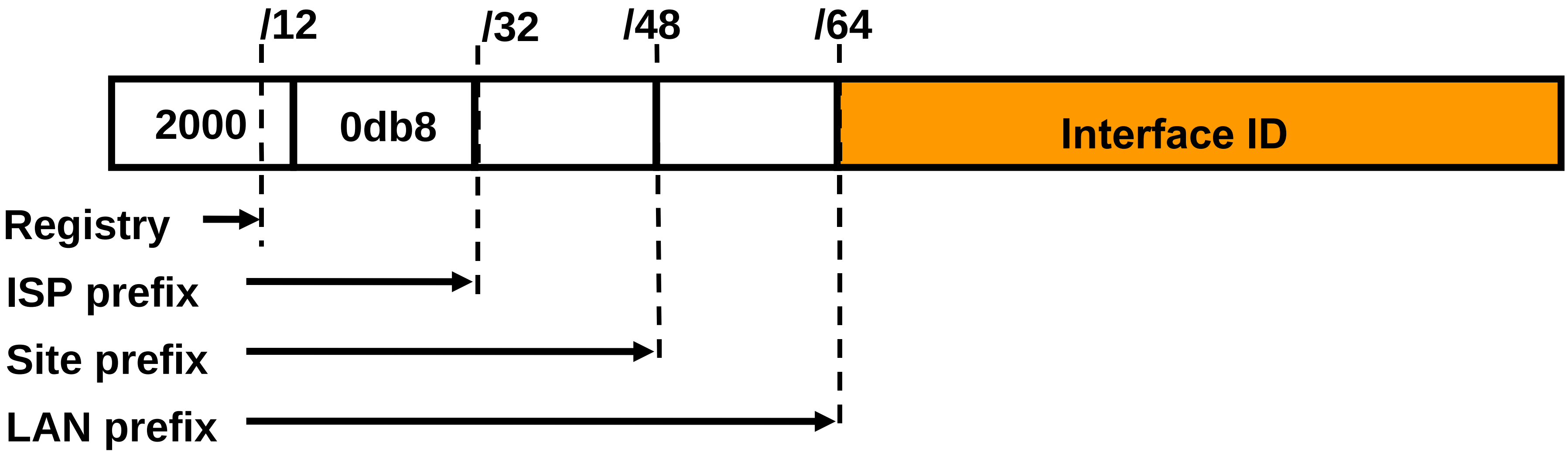
Type	Binary	Hex
Unspecified	0000…0000	::/128
Loopback	0000…0001	::1/128
Global Unicast Address	0010 ...	2000::/3
Link Local Unicast Address	1111 1110 10...	FE80::/10
Unique Local Unicast Address	1111 1100 ... 1111 1101 ...	FC00::/7
Multicast Address	1111 1111 ...	FF00::/8

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
  - Policy is that an ISP allocates 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)
- 32 bits reserved for service providers
  - Possibility of 2^{32} service providers
  - i.e. 4 billion discrete service provider networks
    - Although some service providers already are justifying more than a /32
  - Equivalent to the size 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 in internetworks

- 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

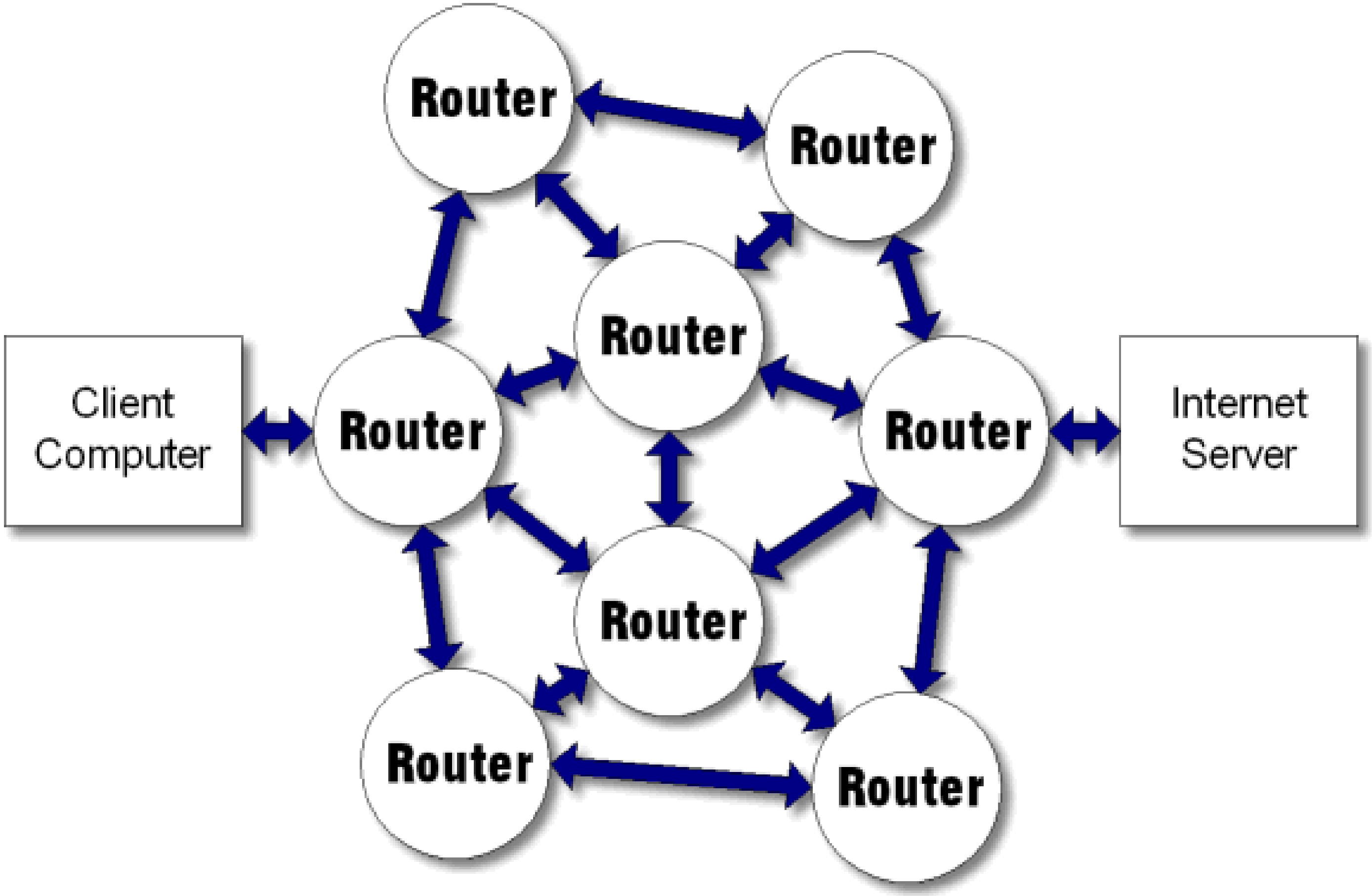
IP router - action for each packet

- 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 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
- Transport 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

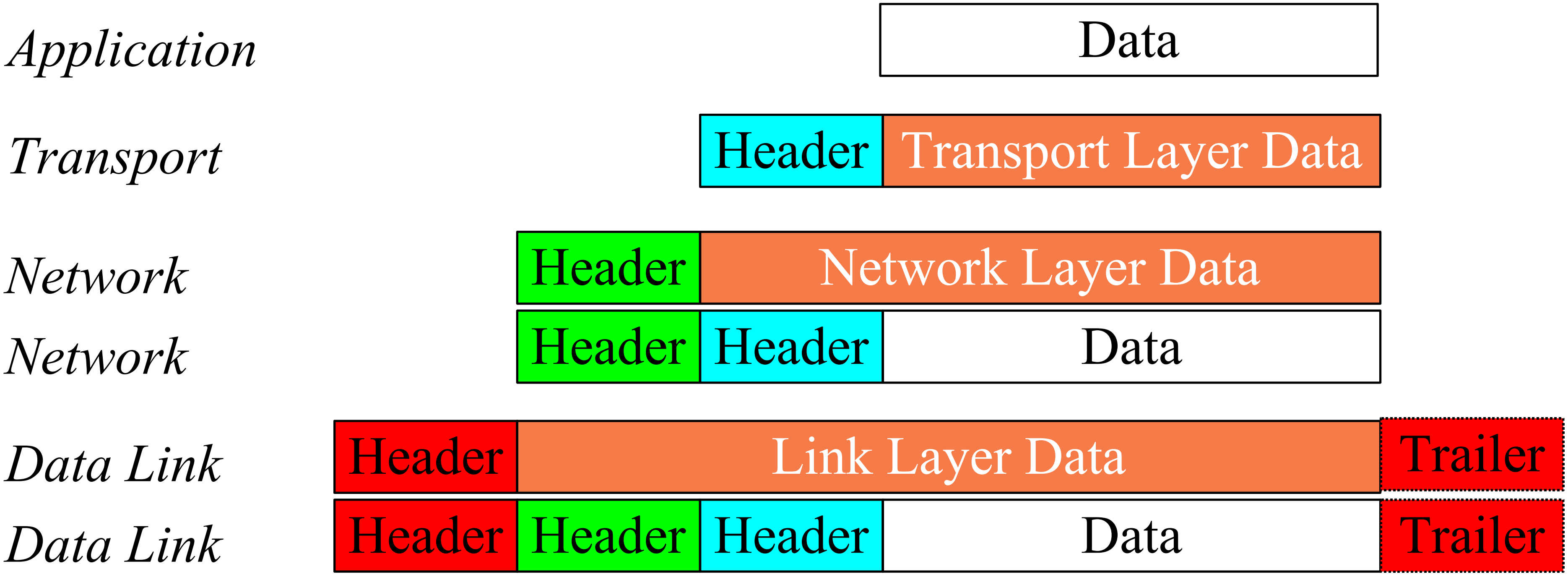
Forwarding table structure

- 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 this way"
- 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

Encapsulation and Types of Links

Encapsulation Reminder

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



Application Data

Transport Header Transport Layer Data

Network Header Network Layer Data

Network Header Header Data

Data Link Header Link Layer Data Trailer

Data Link Header Header Header Data Trailer

Classes of Links

- Different strategies for encapsulation and delivery of IP packets over different classes of links
- Point to Point (e.g. PPP)
- Broadcast (e.g. Ethernet)
- Non-broadcast multi-access (e.g. Frame Relay, X.25, ATM)

Point to Point Links

- Two hosts connected by a point-to-point link
  - data sent by one host is received by the other
- Sender takes IP datagram, encapsulates it in some way (PPP, HDLC, ...), and sends it
- Receiver removes link layer encapsulation
- Check integrity, discard bad packets, process good packets

Broadcast links

- Many hosts connected to a broadcast medium
  - Data sent by one host can be received by all other hosts
  - example: radio, ethernet

Broadcast links

- Have a mechanism for protecting against interference from simultaneous transmissions (e.g. Carrier Sense Multiple Access/Collision Detection for Ethernet)
- Address individual hosts
  - so hosts know what packets to process and which to ignore
  - link layer address is very different from network layer address
- Mapping between network and link address (e.g. ARP)

NBMA links (Non-broadcast multi-access)

- e.g. X.25, Frame Relay, SMDS
- Many hosts
- Each host has a different link layer address
- Each host can potentially send a packet to any other host
- Each packet is typically received by only one host
- Broadcast might be available in some cases

ARP

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

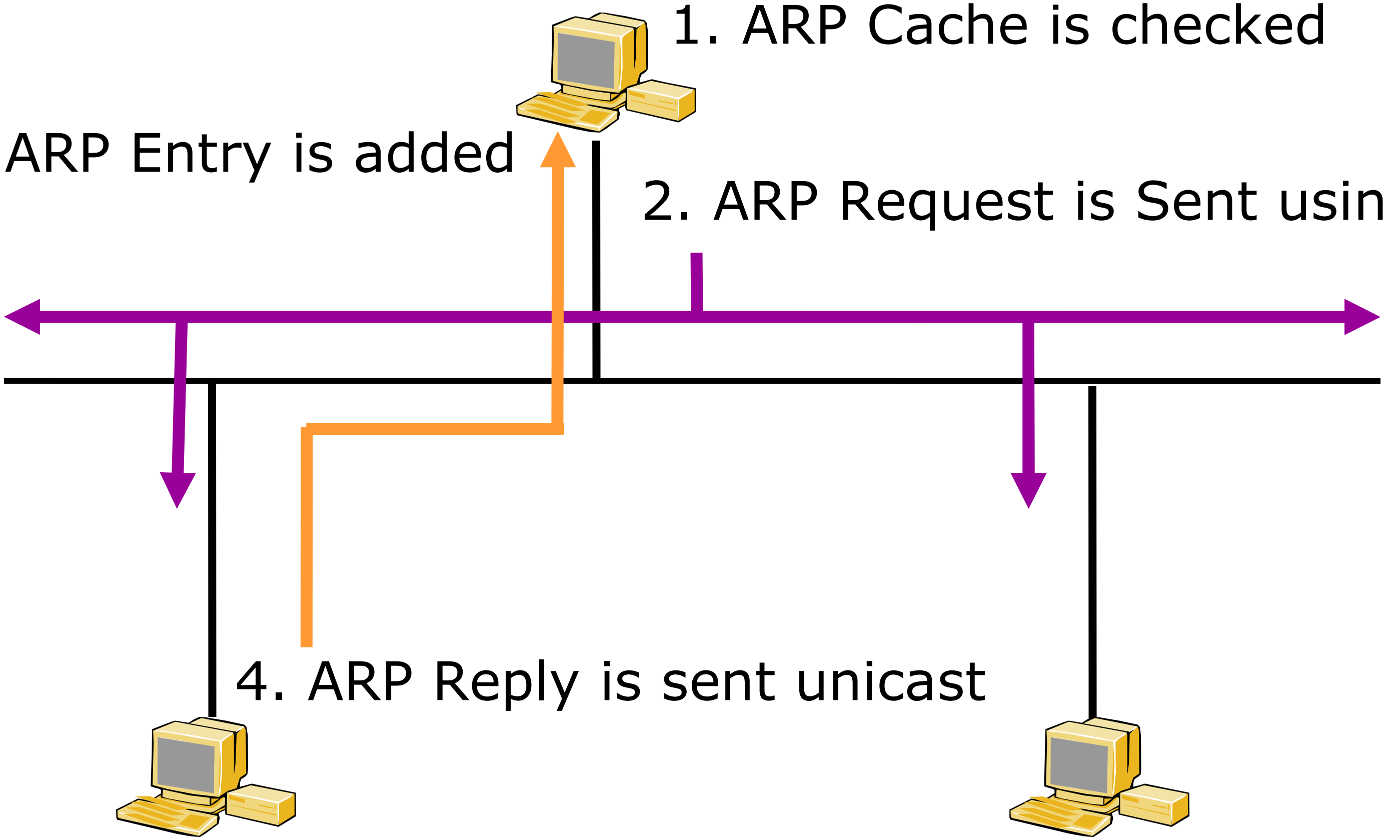
Ethernet/IP Address Resolution

- 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



1. ARP Cache is checked

2. ARP Request is Sent using broadcast

3. ARP Entry is added

4. ARP Reply is sent unicast

5. ARP Entry is added

ARP Table

IP Address	Hardware Address	Age (Sec)
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

ARP Frame

- ARP message is encapsulated in an Ethernet frame (type 0x0806)

Dest Addr	Source Addr	Frame Type	Frame Data
		0x806	Arp Message

Format of an ARP Message

<table>
<tr><th>0</th><th>8</th><th>16</th><th>31</th></tr>
<tr><td colspan="2">Hardware Type</td><td colspan="2">Protocol Type</td></tr>
<tr><td>HLEN</td><td>PLEN</td><td colspan="2">Operation</td></tr>
<tr><td colspan="4">Sender HA</td></tr>
<tr><td colspan="2">Sender HA</td><td colspan="2">Sender IP Address</td></tr>
<tr><td colspan="2">Sender IP Address</td><td colspan="2">Target HA</td></tr>
<tr><td colspan="4">Target HA</td></tr>
<tr><td colspan="4">Target IP</td></tr>
</table>

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

Reverse ARP - RARP

- For host machines that don't know their IP address – e.g diskless systems
- RARP enables them to request their IP address from the gateway's ARP cache
- Need an RARP server
- See RFC 903
- NOTE: This is not used much nowadays
  - DHCP does same function

Supplementary materials

Blocks of IP addresses

- IP addresses are often grouped according to their binary representation
- High order bits identify the block; low order bits identify an individual address in the block
  - e.g. network part and host part, but it's actually more general than that
- Each block has a size that is a power of 2
  - 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, ...
- The network administrator chooses the size of the block. This is the same as choosing the number of bits in the prefix, or choosing the netmask

Analogy with decimal numbers

- People are usually more familiar with decimal numbers than with binary numbers
- We will look at how decimal numbers can be divided into blocks whose size is a power of 10
  - 10, 100, 1000, ...
- The same principle applies to binary numbers in groups whose size is a power of 2

Grouping of decimal numbers

- Given a lot of 4-digit numbers (0000 to 9999)
  - 10^4 = 10000 numbers altogether
- Can have 10^1 (10) groups of 10^3 (1000)
- Can have 10^2 (100) groups of 10^2 (100)
- Can have 10^3 (1000) groups of 10^1 (10)
- Can have 10^4 (10000) groups of 1
- Any large group can be divided into smaller groups, recursively

Grouping of decimal numbers

- If we want a block of 100 decimal numbers:
  - 300 to 399 is a “good” block
    - we can draw a line that separates the high order part “3” from the low order part “00” to “99”
    - all numbers in the block have the same high order part “3”
    - all numbers in the block have different low order part
    - “high order digits” and “low order digits” are analogous to “network part” and “host part”
  - 307 to 406 is not a “good” block
    - even though there are 100 numbers in the block, the numbers do not all have the same high order digits (some have “3” and some have “4”)

Grouping of small binary numbers

- Given a lot of 4-bit binary numbers (0000 to 1111)
  - 2^4 = 16 numbers altogether
- Can have 2^1 (2) groups of 2^3 (8)
- Can have 2^2 (4) groups of 2^2 (4)
- Can have 2^3 (8) groups of 2^1 (2)
- Can have 2^4 (16) groups of 1
- Any large group can be divided into smaller groups, recursively

Grouping of small binary numbers

- If we want a block of 4 binary numbers:
  - 0100 to 0111 (decimal 4 to 7) is a “good” block
    - we can draw a line that separates the high order part “01” from the low order part “00” to “11”
    - all numbers in the block have the same high order part “01”
    - all numbers in the block have different low order part
  - 0110 to 1001 (decimal 6 to 9) is not a “good” block
    - even though there are 4 numbers in the block, the numbers do not all have the same high order bits (some have “01” and some have “10”)

Grouping of 32-bit binary numbers (e.g. IPv4 addresses)

- Given a lot of 32-bit numbers (0000…0000 to 1111…1111)
  - Can have 2^{0} (1) groups of 2^{32} numbers
  - Can have 2^{8} (256) groups of 2^{24} numbers
  - Can have 2^{25} groups of 2^{7} numbers
- Consider one group of 2^{7} (128) numbers
    - e.g. 1101000110100011011010010xxxxxxx
  - Can divide it into 2^{1} (2) groups of 2^{6} (64)
  - Can divide it into 2^{2} (4) groups of 2^{5} (32)
  - etc.