Internet Protocol Stack

Routing and Switching

tzNOG

2015

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: Csnet, 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

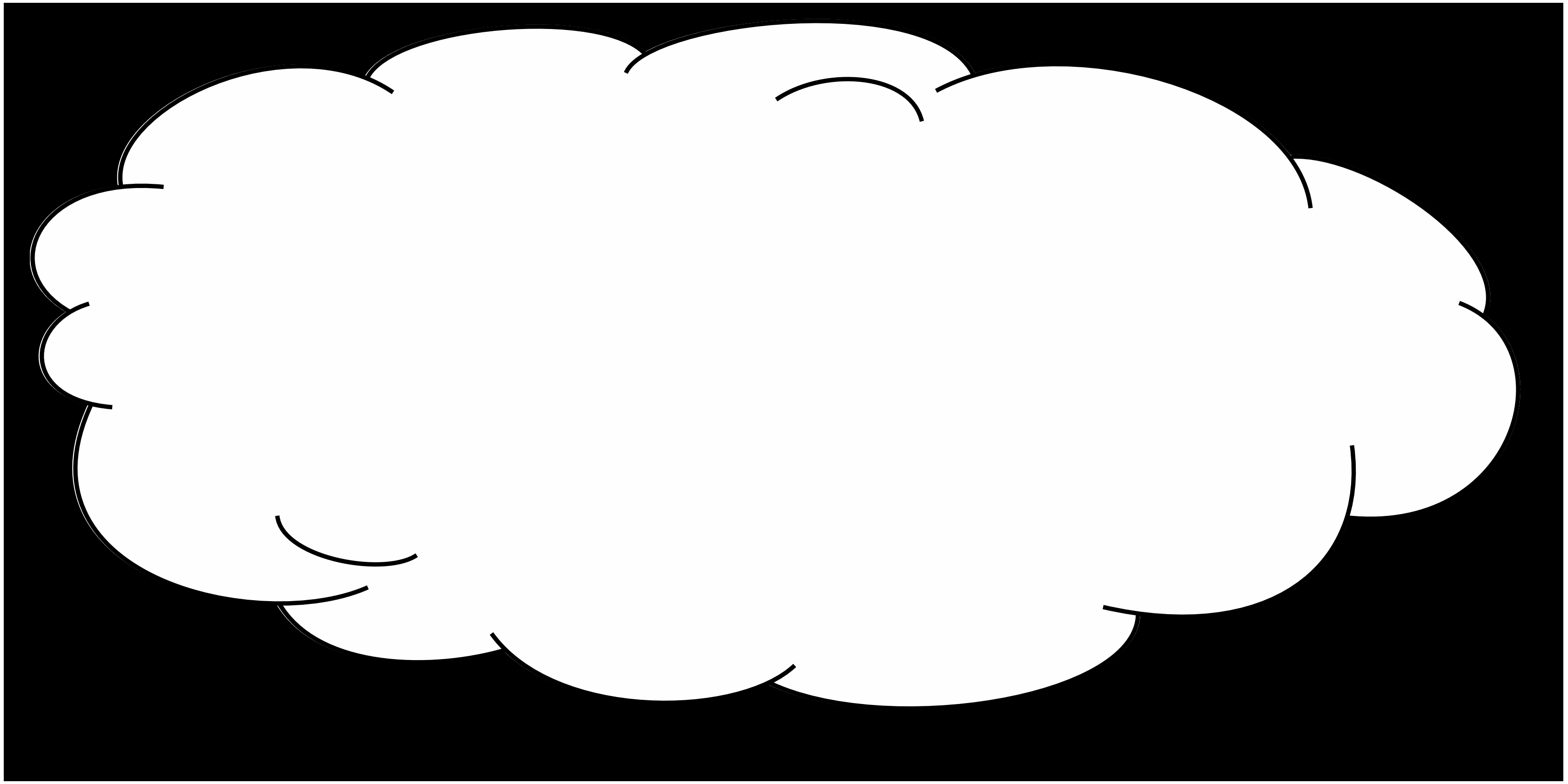
Late 1990's – 2000's:

- 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

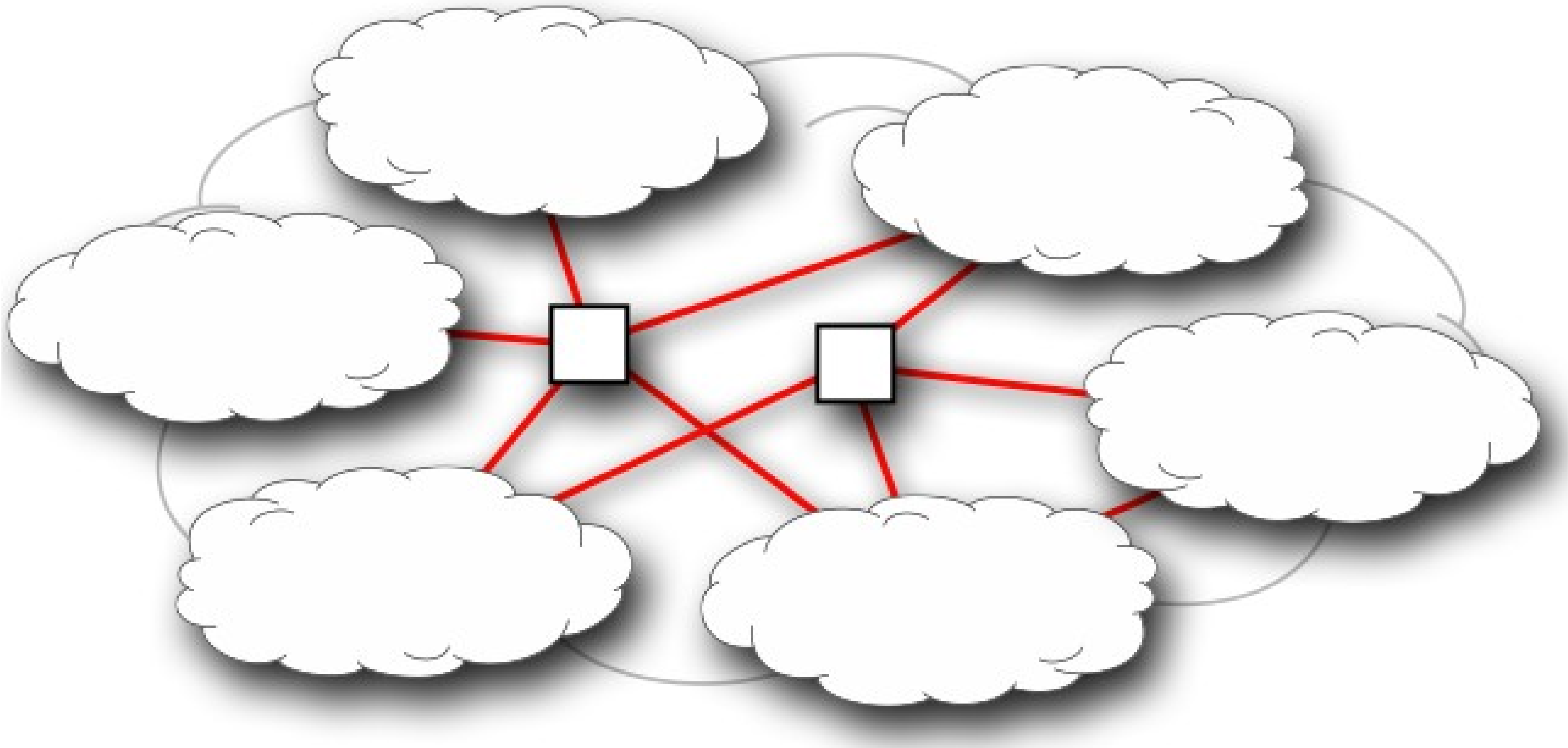
now: 40-100 Gbps

- youtube, social networking
- depletion of Ipv4 address space

The Internet – or how we see it



A more accurate representation…



‘I’nternet vs ‘i’nternet

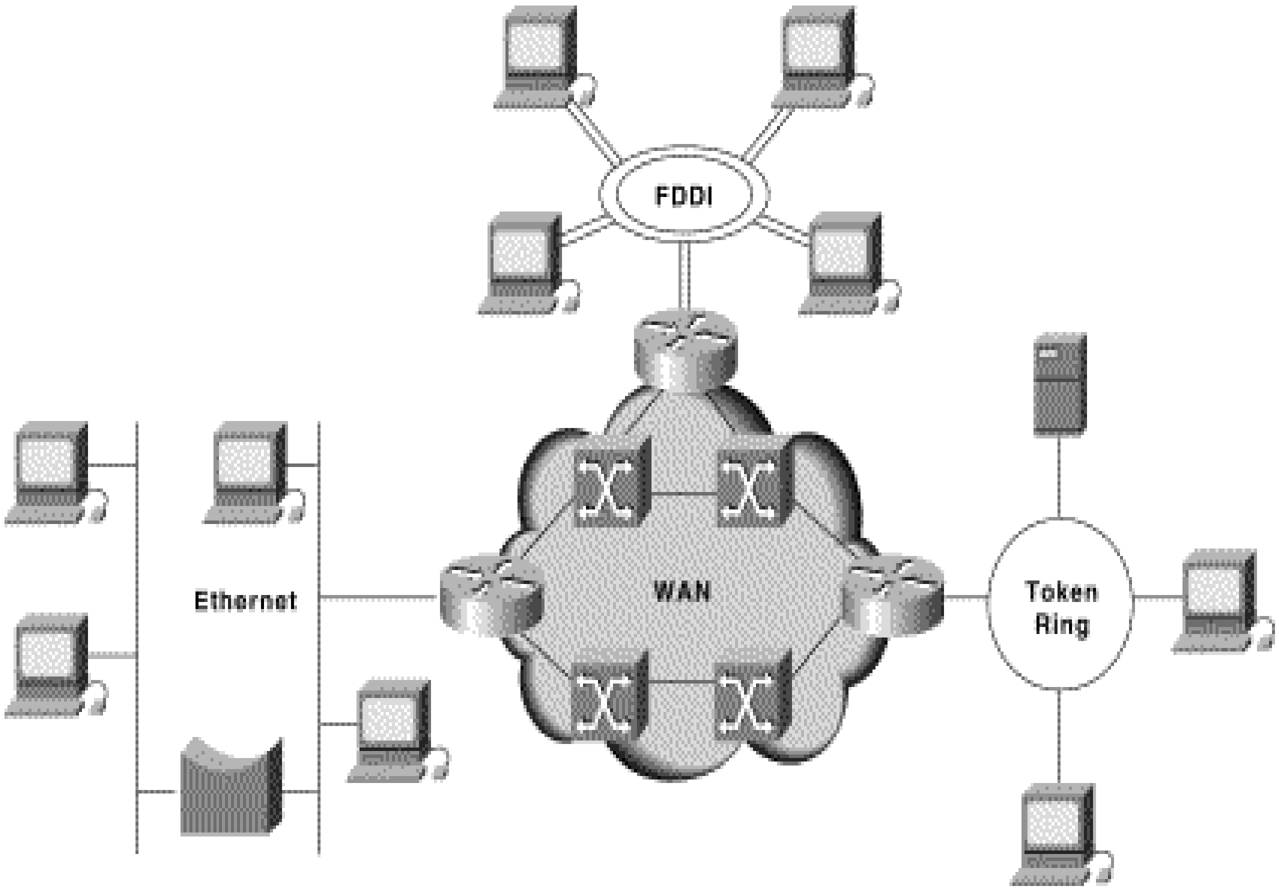
Internet: The network of networks. The proper-noun, capital-I Internet is the network of all networks which provide global end-to-end Internet Protocol connectivity between their nodes.

internet: Any set of interconnected networks. A lower-case-i internet doesn’t necessarily use Internet Protocols, nor need it be interconnected with the Internet. No longer in widespread use.

The (capital "I") Internet

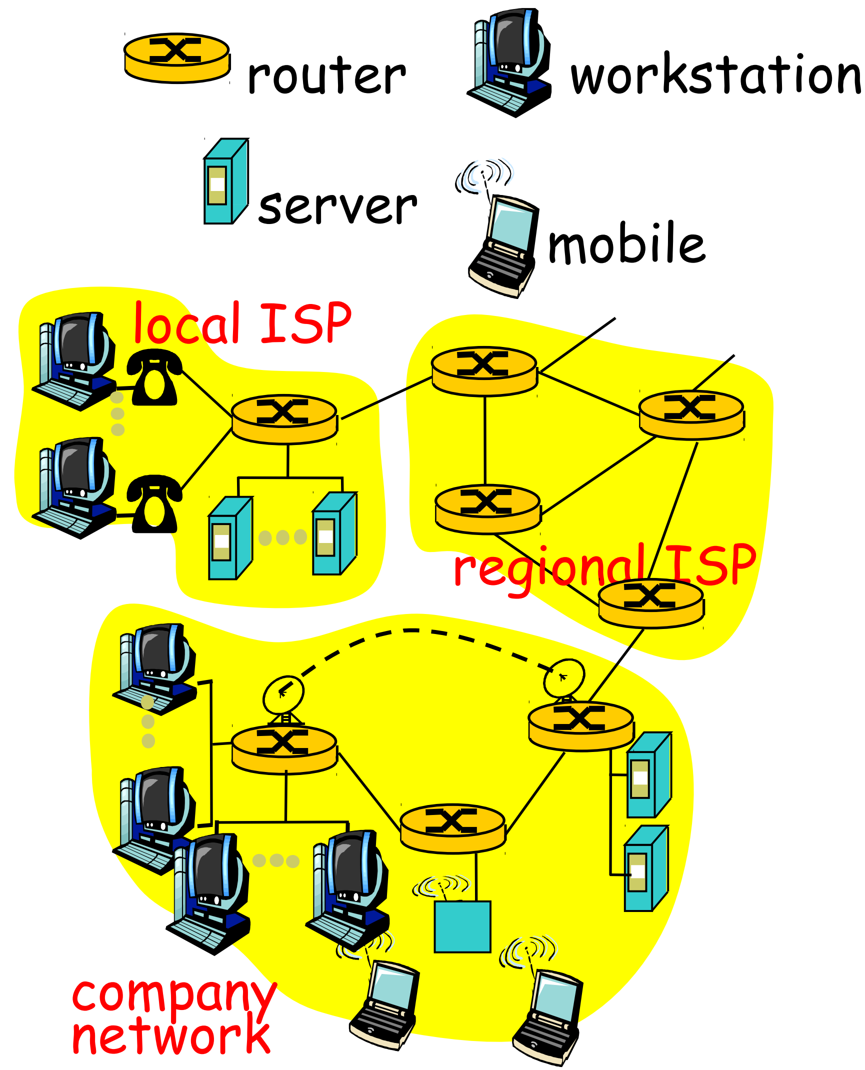
- The world-wide network of TCP/IP networks
- Different people or organisations own different parts
- Different parts use different technologies (at the "lower layers")
- Interconnections between the parts (all use IP)
- Interconnections require agreements
  - sale/purchase of service
  - contracts
  - "peering" agreements
- No central control or management

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



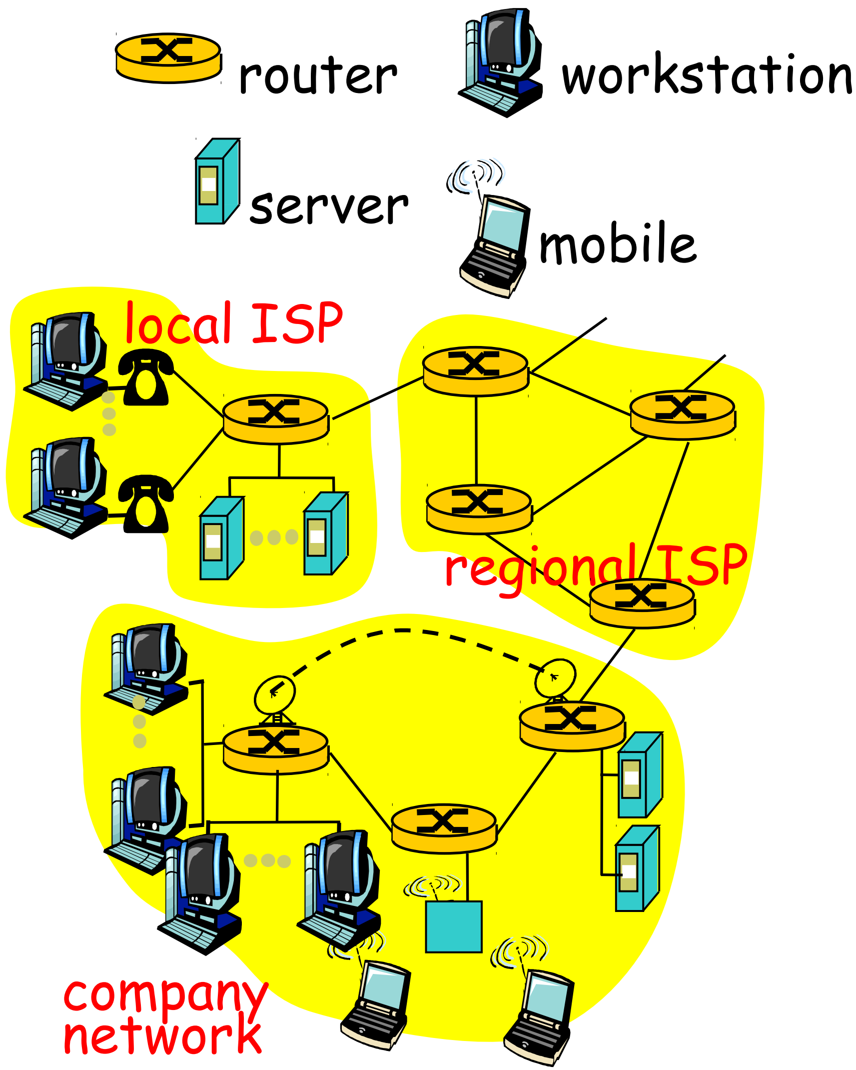
What is the Internet: "nuts and bolts" view

- 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



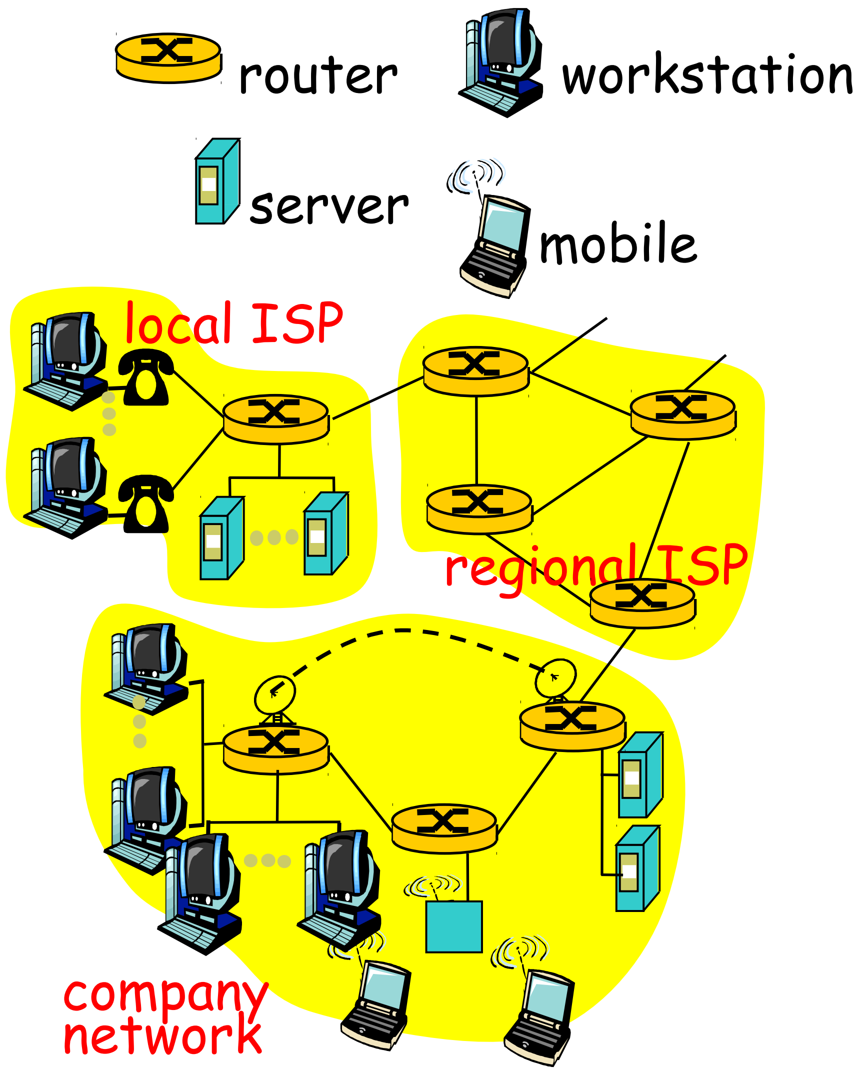
What is the Internet: "nuts and bolts" view

- 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

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



Principles of the Internet

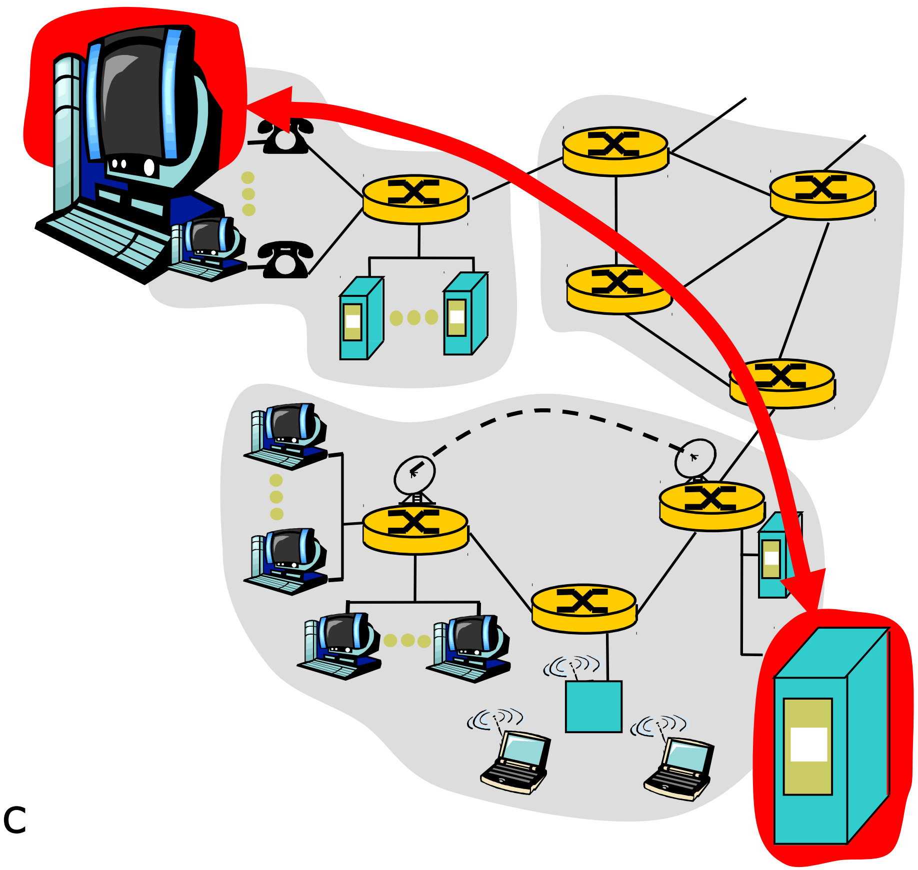
- Edge vs. core (end-systems vs. routers)
  - Dumb network
  - Intelligence at the end-systems
- Different communication paradigms
  - Connection-oriented vs. connectionless
  - Circuit switching vs. packet switching
- Layered System
- Network of collaborating networks

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
  - There is a “connection” in TCP
  - Error recovery is by retransmission
  - Packet drops as congestion signaling

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 to peer model:
  - host interaction symmetric
    e.g.: teleconferencing



Network edge: connection-oriented service

- **TCP - Transmission Control Protocol**
  - TCP service [RFC 793]
  - Internet's connection-oriented service
  - Reliable, in-order byte-stream data transfer
    - Loss: acknowledgments and retransmissions
  - Flow control:
    - Sender won't overwhelm receiver
  - Congestion control:
    - Senders "slow down sending rate"
    - When network congested

Network edge: connectionless service

- Goal: easy/fast data transfer between end systems without need for state checking.

- **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

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
  - Ethernet, WiFi, ADSL, fibre, ...
- Anything over IP
  - Mail, web, chat, ...

Why layering?

- Dealing with complex systems:
- explicit structure allows identification, relationship of complex system’s pieces
- Modularization eases maintenance, updating of system
  - change of implementation of layer’s service is transparent to rest of system
  - e.g., change in gate procedure does not affect rest of system

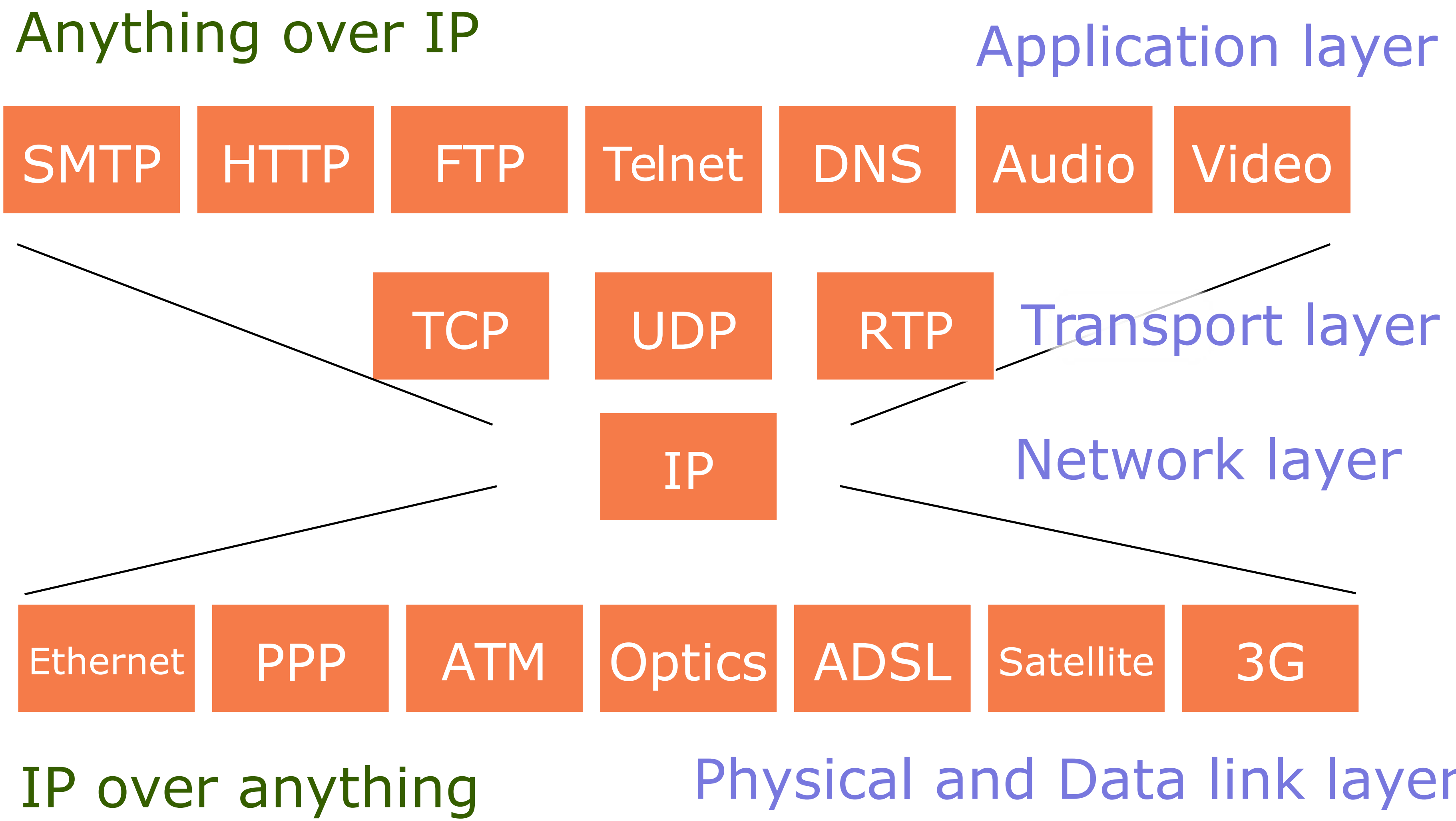
The OSI Model

7	Application	Upper Layers Application oriented “End-to-End”-Layers
6	Presentation	
5	Session	
4	Transport	
3	Network	Lower Layers Network oriented “Hop-by-hop” layers
2	Data Link	
1	Physical	

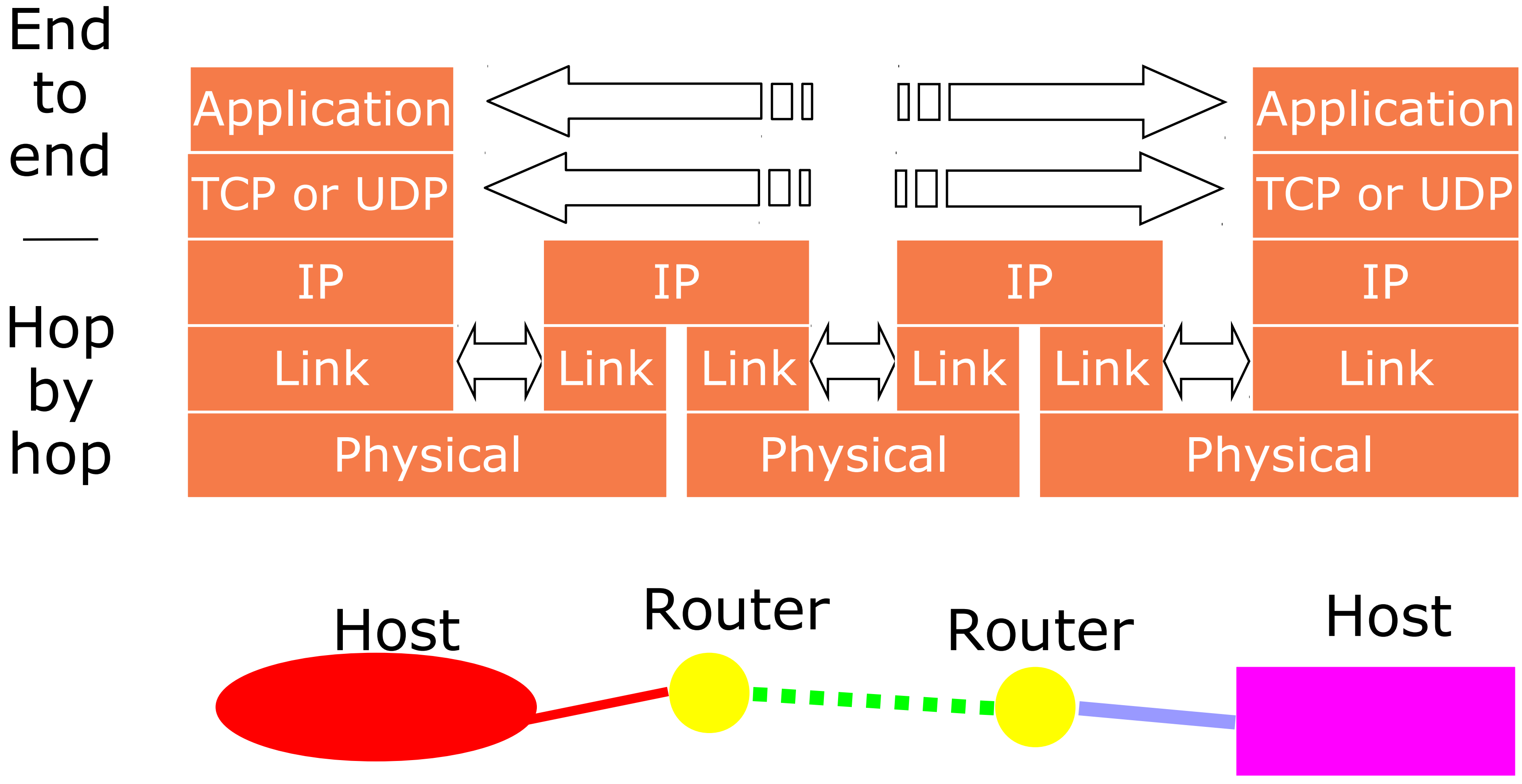
OSI Model and the Internet

- 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

The IP Hourglass Model



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

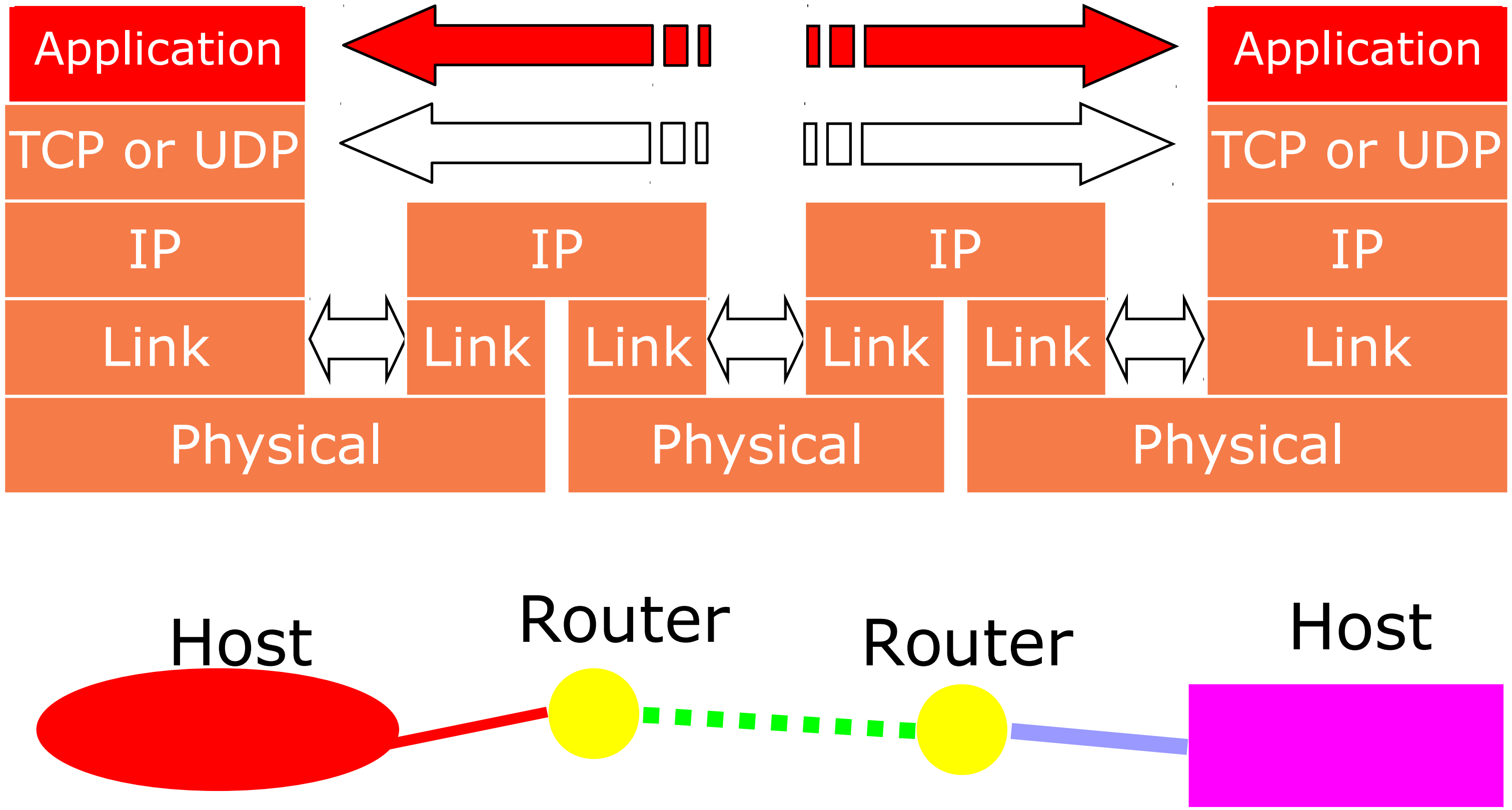
Hop-by-hop layers

- 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

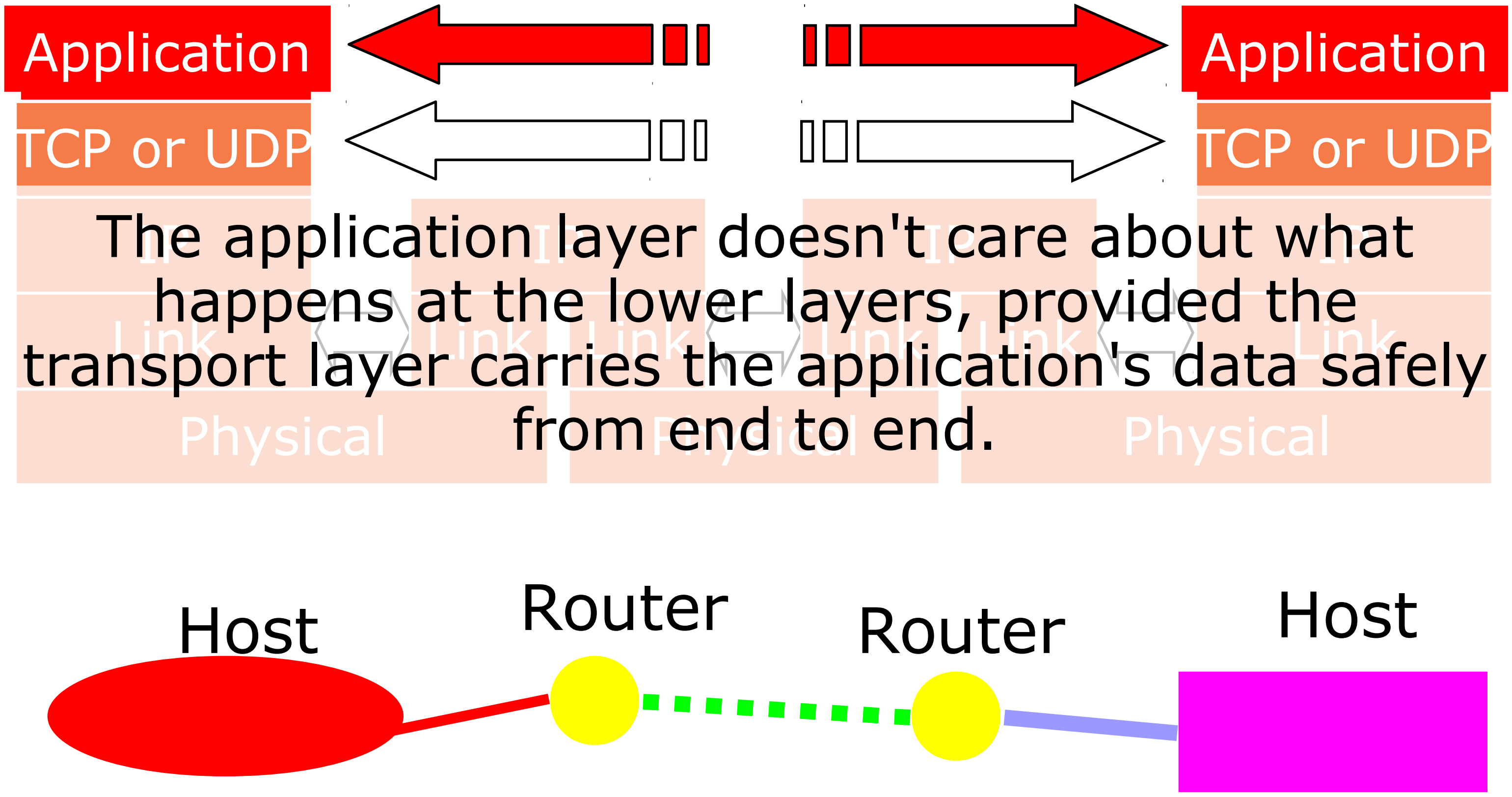
Layer Interaction: TCP/IP Model

Applications behave as if they can talk to each other. Let's look at what really happens.



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.

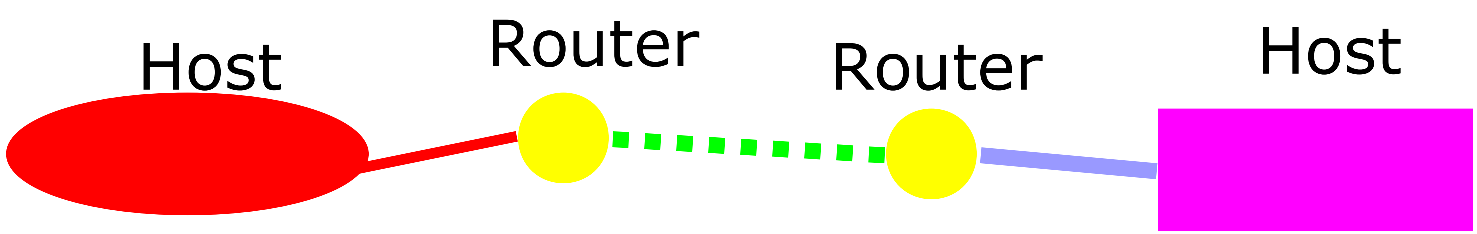


The application layer doesn't care about what happens at the lower layers, provided the transport layer carries the application's data safely from end to end.

Layer Interaction: The Transport Layer

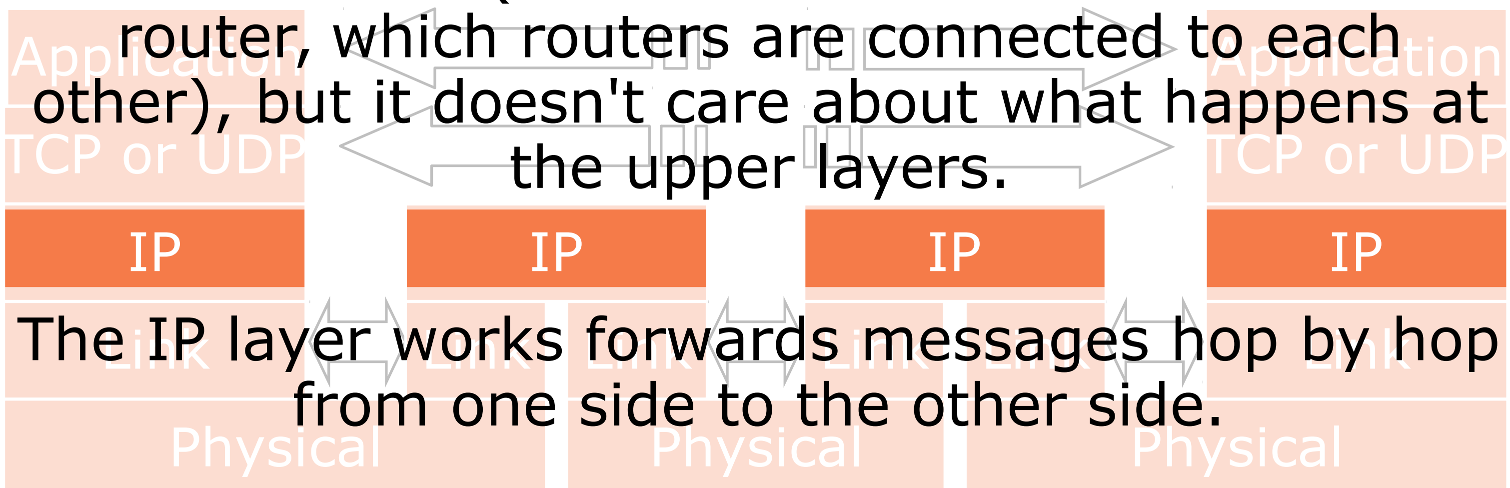
The transport layer instances at the two ends act as if they are talking to each other, but in reality they are each talking to the IP layer below it. The transport layer doesn't care about what the application layer is doing above it.

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.

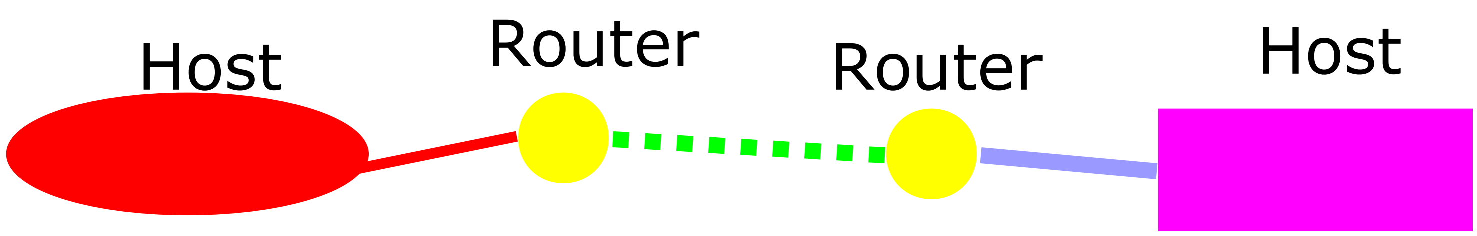


Layer Interaction: The Network Layer (IP)

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.



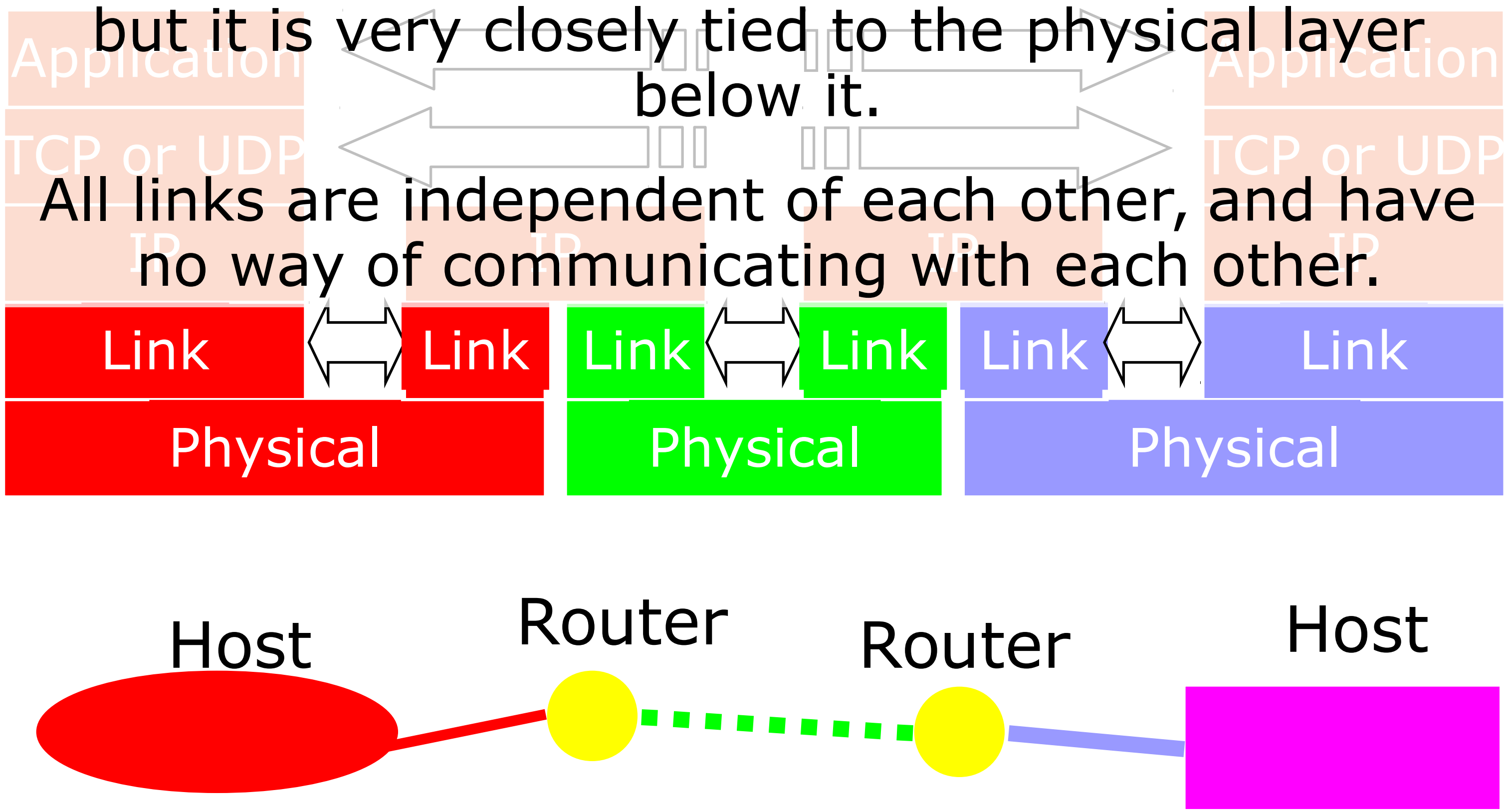
The IP layer works forwards messages hop by hop from one side to the other side.



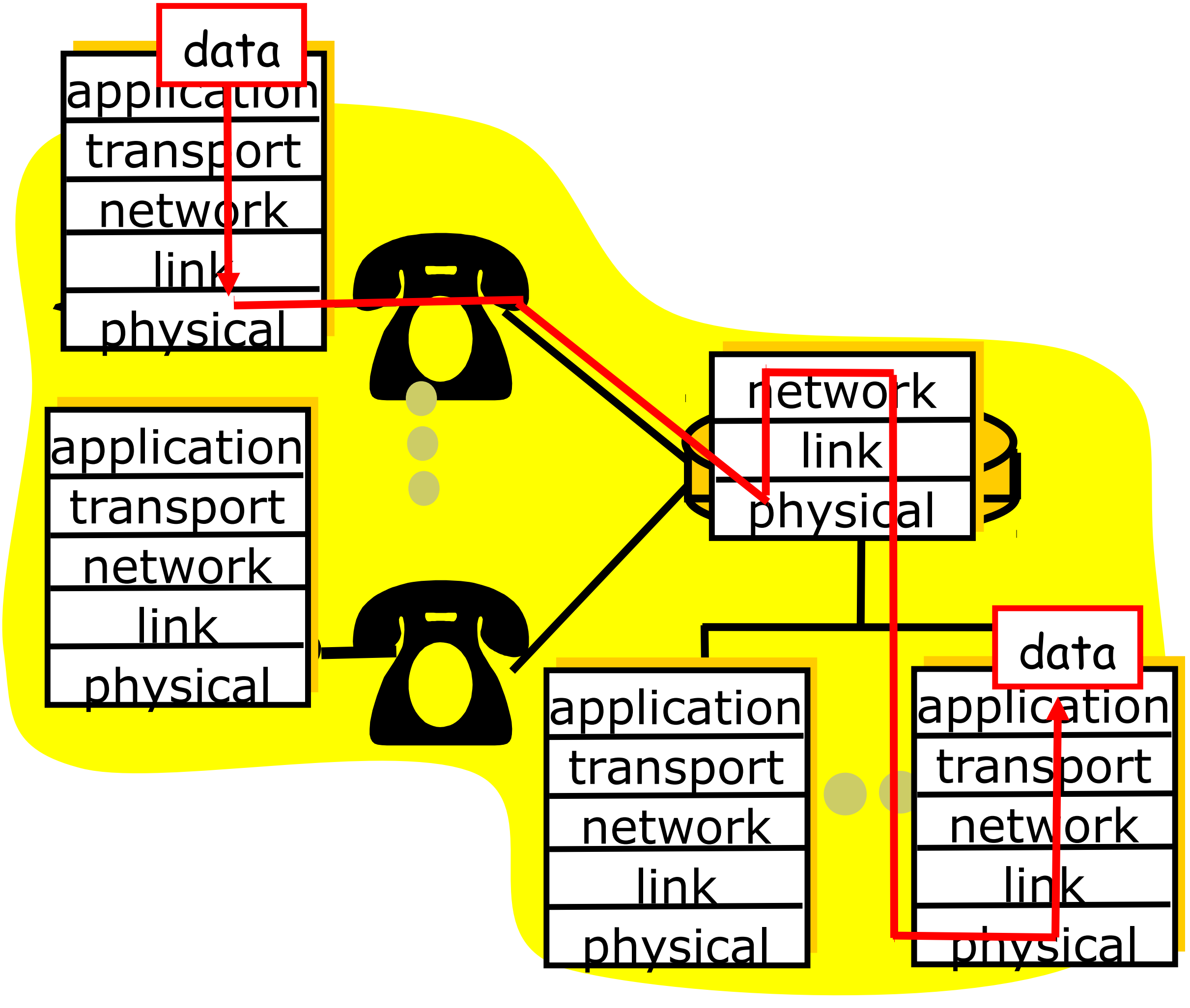
Layer Interaction:
Link and Physical Layers

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

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



Layering: physical communication

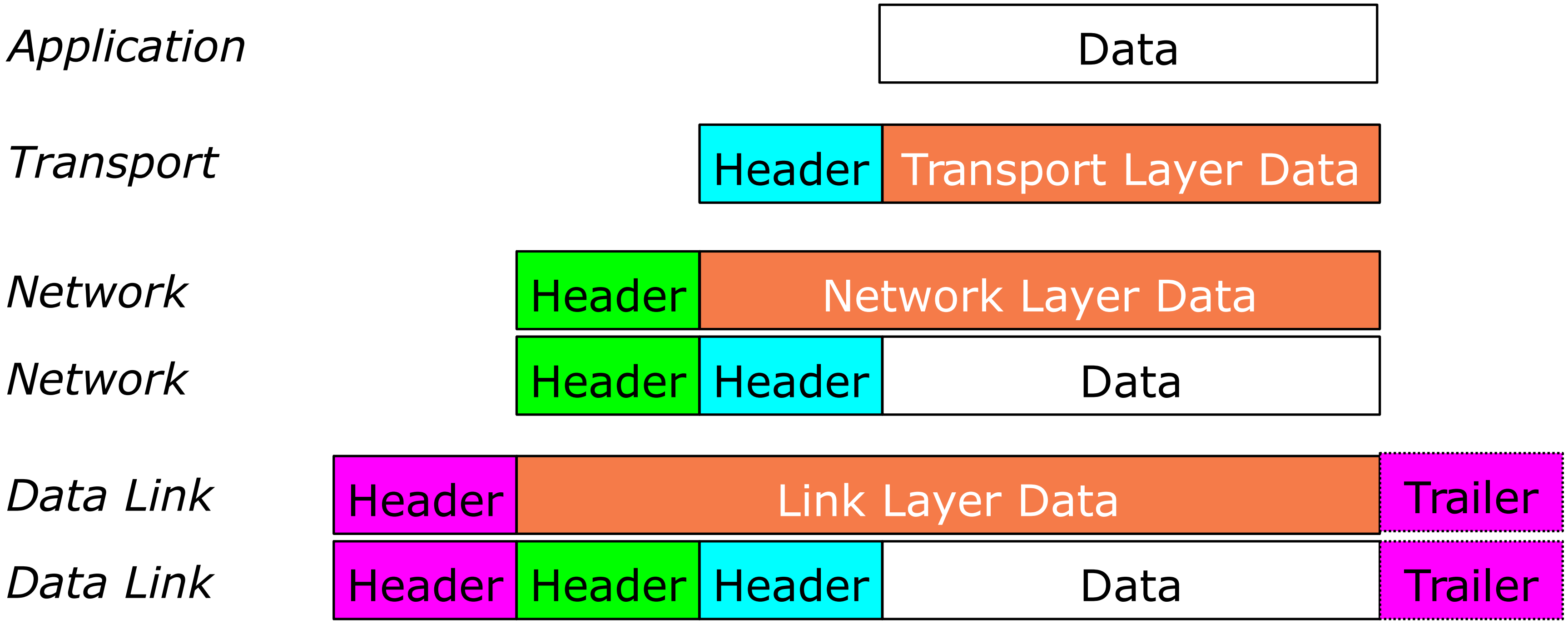


Frame, Datagram, Segment, Packet

- 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

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

- 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.

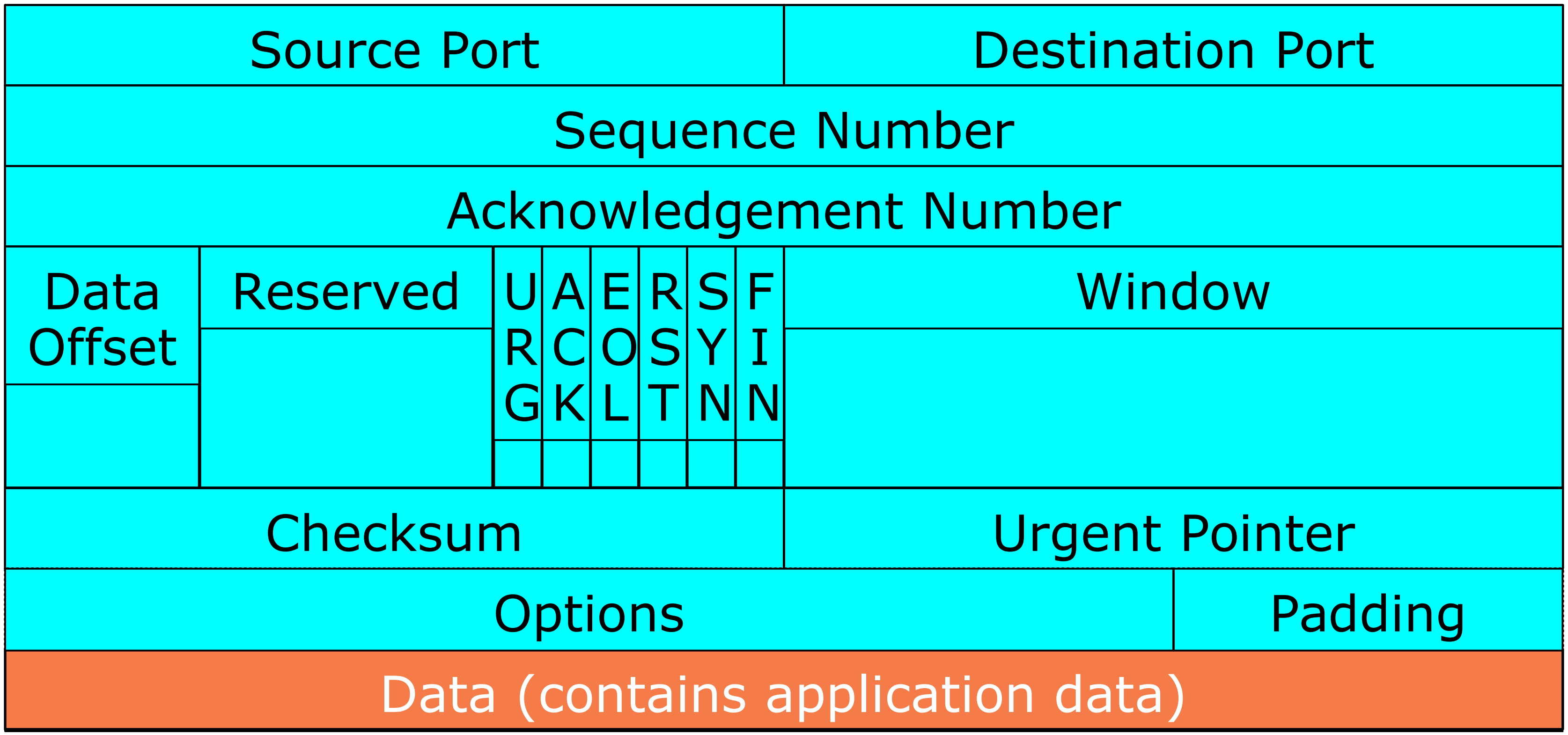
Layer 3 - IPv4 datagram

<table>
<tr><td>Version</td><td>IHL</td><td>Diff Services</td><td colspan="3">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td colspan="2">Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td>Protocol</td><td colspan="3">Header Checksum</td></tr>
<tr><td colspan="6">Source Address (32-bit IPv4 address)</td></tr>
<tr><td colspan="6">Destination Address (32-bit IPv4 address)</td></tr>
<tr><td colspan="5">Options</td><td>Padding</td></tr>
<tr><td colspan="6">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 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)