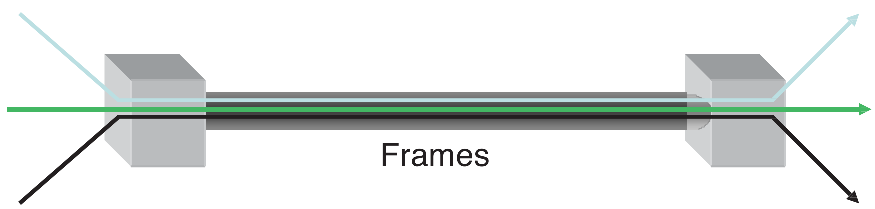
Multiplexing/Demultiplexing

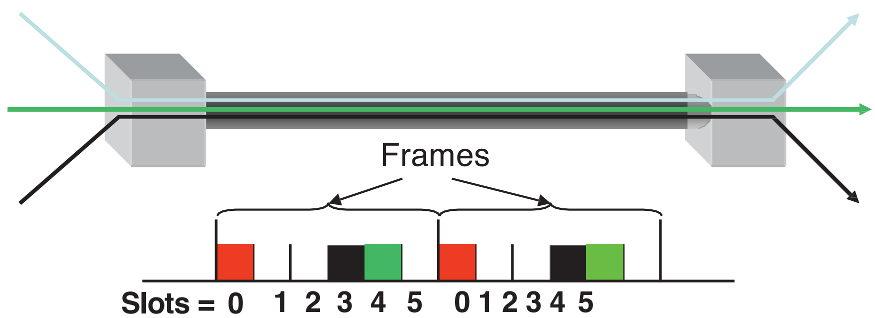


Sharing system resource among multiple users

How to achieve sharing?

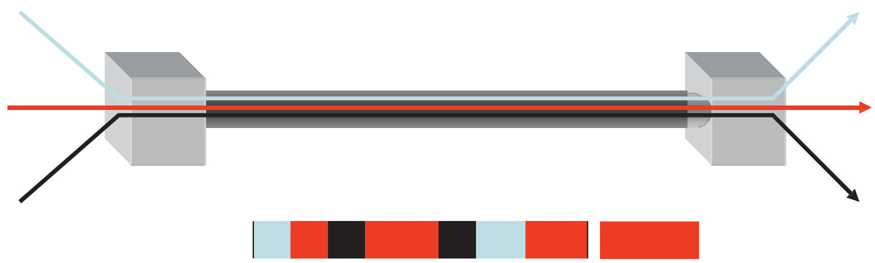
- Two broad popular approaches
- Circuit switching
    - Telephone networks
- Packet switching
    - Internet

Circuit Switching



- One popular form: Time-division multiplexing
- Time divided in frames and frames divided in slots
- Relative slot position inside a frame determines which conversation the data belongs to
  - E.g., slot 0 belongs to red conversation
- Issue:
  - If a conversation does not use its circuit the capacity is lost!
- Other forms of circuit switching have similar issues
  - E.g., frequency-division multiplexing

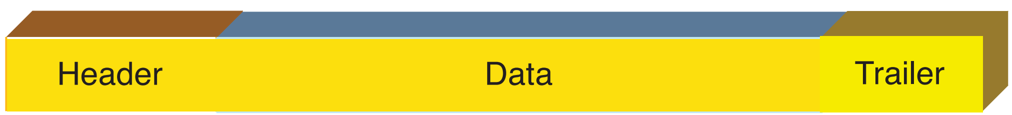
Packet Switching: Statistical Multiplexing



- Data from any user can be transmitted at any given time
    - A single user can use the entire link capacity if the only active one
- Works well when only some users active at a given time
- This form of sharing referred to as "statistical multiplexing"

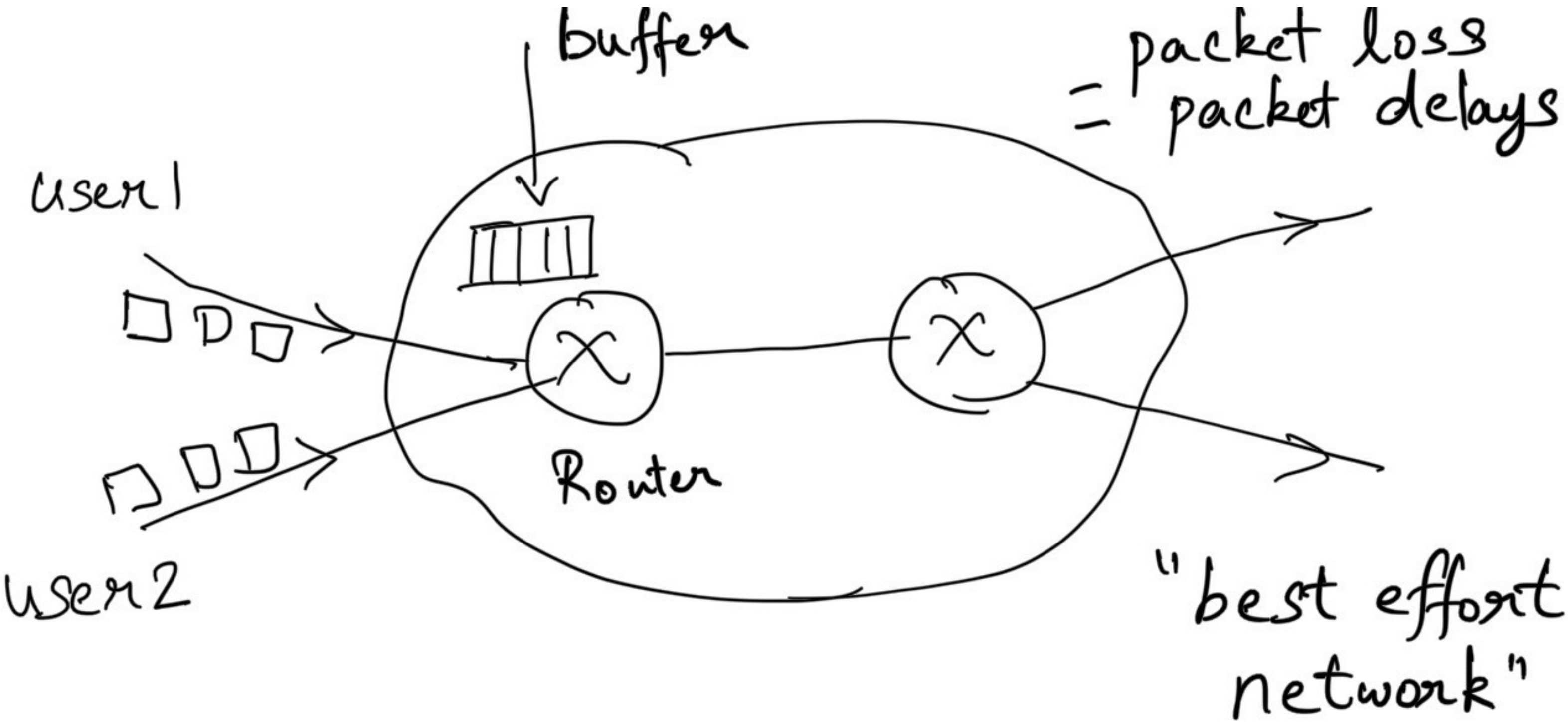
Packet Switching

- Data is sent as smaller chunks, referred to as packets.
- Packets have the following structure:



- At each node the entire packet is received, stored briefly, and then forwarded to the next node based on the header information (**Store-and-Forward Networks**)

Congestion in packet switched networks

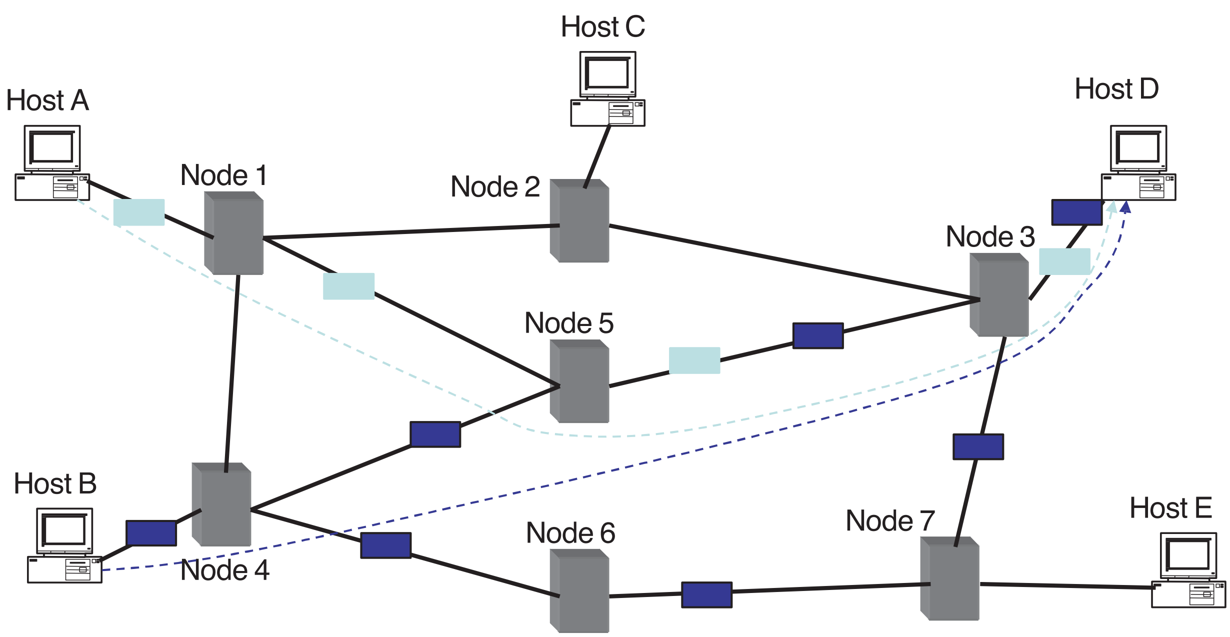


Packet vs. Circuit Switching

- Packet-switching: Benefits
  - Ability to exploit statistical multiplexing
  - More efficient bandwidth usage

- Packet switching: Concerns
  - Needs to buffer and deal with congestion:
  - More complex switches
  - Harder to provide good network services (e.g., delay and bandwidth guarantees)

Datagram Packet Switching



- Each packet is independently switched
  - Each packet header contains destination address
- Advantage: robust to failures
- Disadvantage: packets may arrive out of order.

Implications of Internet Model

- Packet-switched network
- Datagram model
- “Best-effort” network
  - Packets may be lost
  - Packets may be delayed
  - Packets may arrive out of order

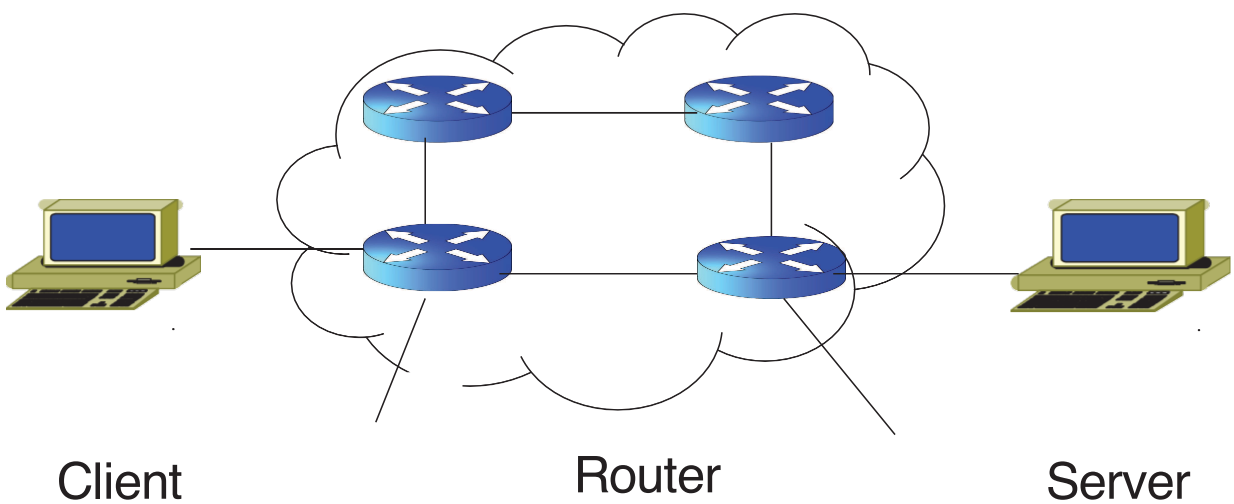
Organizing a network's tasks

- Many kinds of networking functionality
  - *Addressing: How to specify a node?*
  - *Routing: Which path should I follow?*
  - *Flow Control: How to avoid congestions?*
  - *Security: How can privacy and integrity be maintained?*

- How should they be organized?
- How should they interact?

Example

- Transfer file from node A to node B
- What's involved?

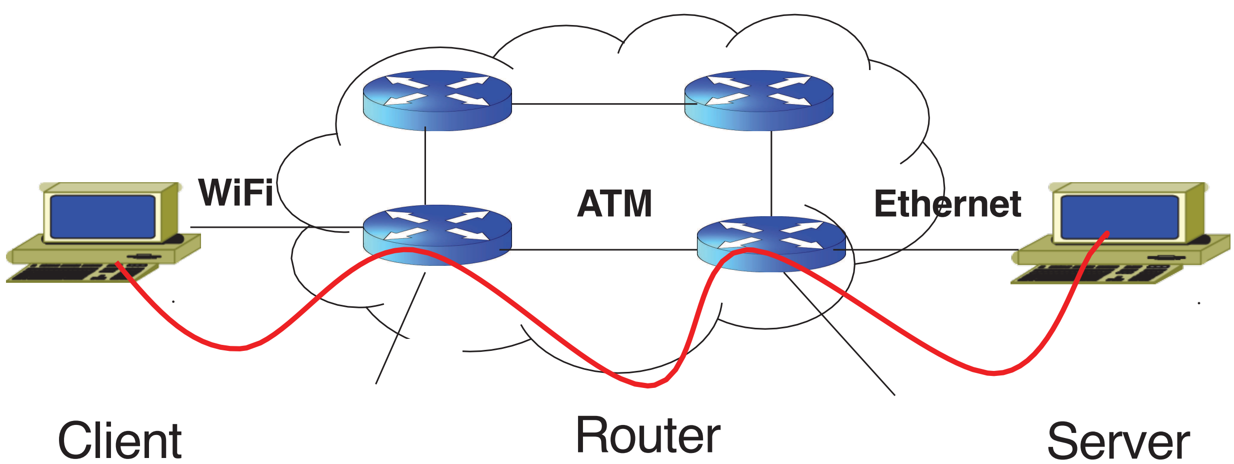


Application View

- How to authenticate client and ensure it has permission to access the record?
- What if file requested is not available?
- Should file be encrypted to ensure transmission is confidential?

Network View

- How to identify/"address" server?
- Which routers/path must be picked?
- How to ensure reliable, in-order delivery?
- How to get packet to traverse each hop?

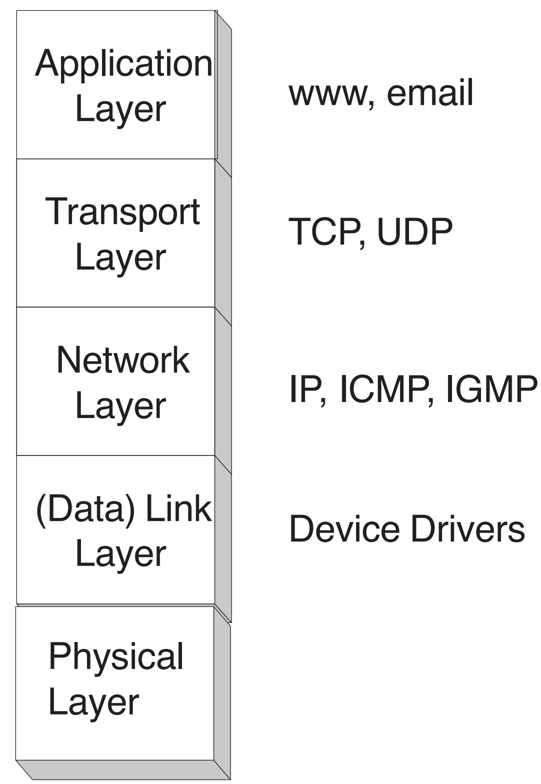


Internet: Layered Architecture

- Network functionality organized into layers
- ISO OSI Reference Model
  - ISO – International Standard Organization
  - OSI – Open System Interconnection
  - 7 layer protocol stack
- In practice today: TCP/IP stack
  - Effectively 5 layers.

Practice: TCP/IP Layering

- The TCP/IP suite has five layers
- Computers (hosts) implement all five layers. Routers (gateways) only have the bottom three layers.

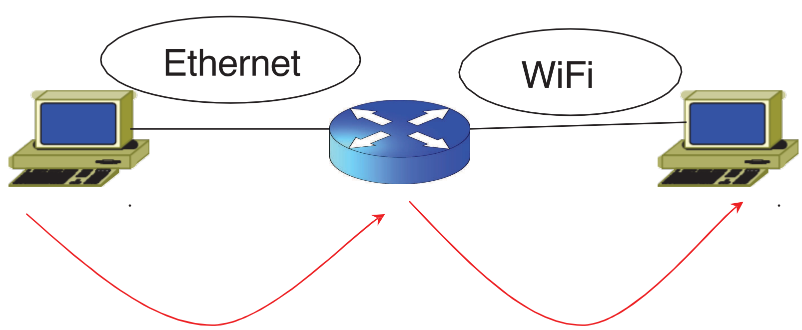


Data Link Layer

– Service: Transfer of frames over a link or “hop”

– Example function:

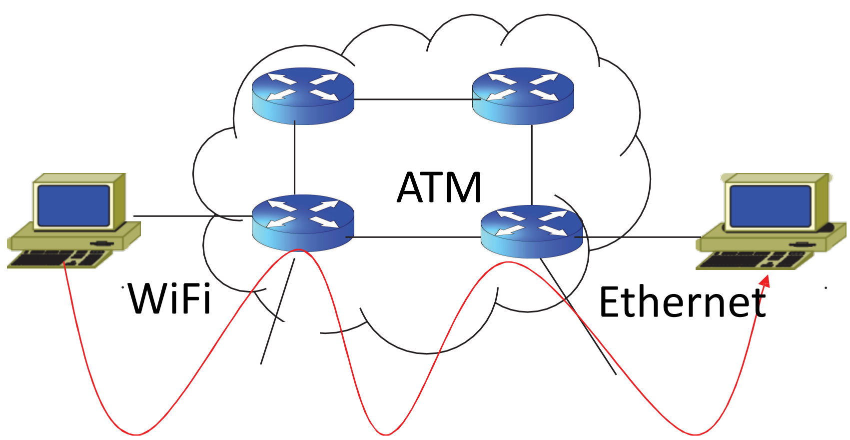
Prevent “collisions” when many users on the same WiFi transmit together.



Application Layer	Email, www
Transport Layer	TCP, UDP
Network Layer	IP, ICMP, OSPF RIP, BGP
(Data) Link Layer	Ethernet, WiFi PoS, T1

Network Layer (IP)

- Service: Moves packets ”end-to-end”
- Packet may traverse many underlying technologies
- Functions: Routing, addressing,



Application Layer	email www
Transport Layer	TCP, UDP
Network Layer	IP, ICMP, OSPF RIP, BGP
(Data) Link Layer	Ethernet, WiFi PoS, T1

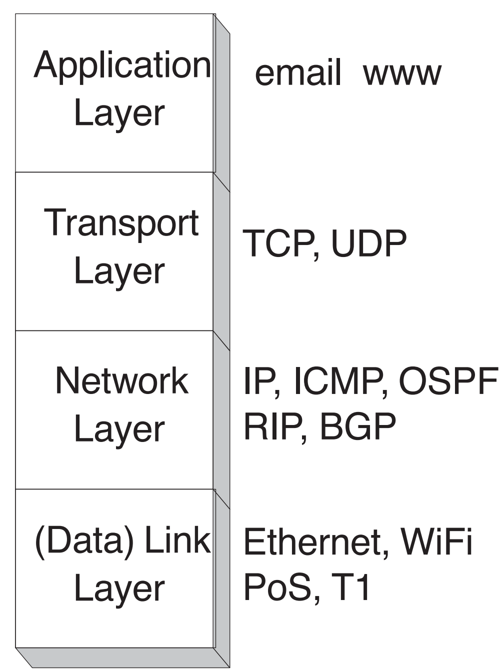
IP Delivery Model

- Best-effort delivery
- Given a packet, send to remote point but:
  - Could be lost
  - Could be reordered
  - Could be delayed.

Transport Layer

- Service: Get data across with desired properties (e.g., reliable, in-order)

- Example Functions:
  - To which application (e.g., web, email) should a packet that arrives at a computer be sent?
  - How to deal with packet losses and out-of-order packets?

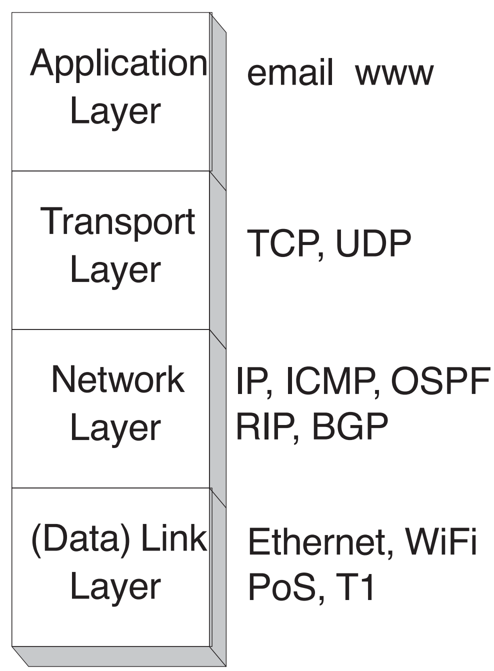


TCP and UDP

- Both sit on top of IP
- TCP:
  - Ensures reliable, in-order delivery
  - Mechanisms for "congestion control"
  - But latencies could be even higher than the IP layer
- UDP:
  - Barebones functionality
  - Retains IP delivery model
  - Application may see packet losses, and out-of-order packets. Delay not guaranteed

Application Layer

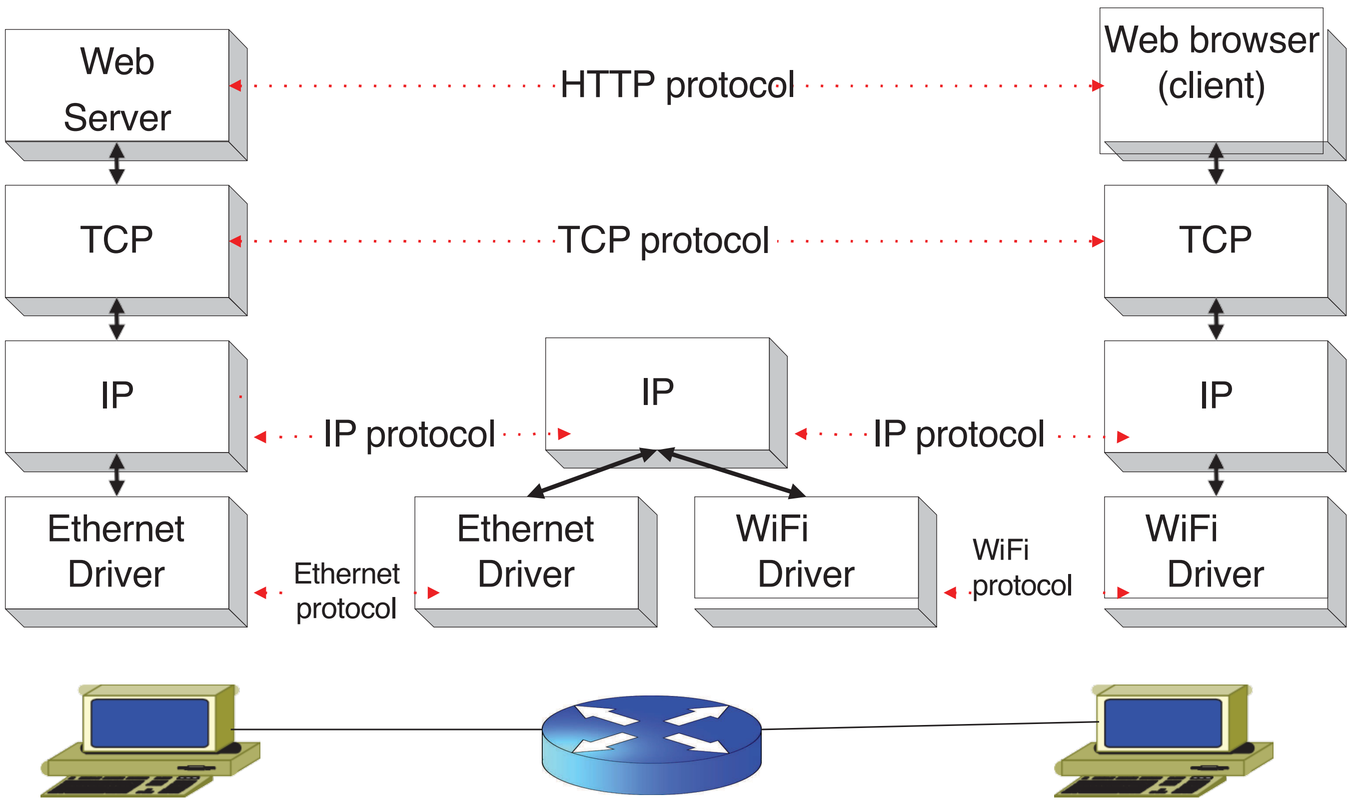
- Service: Handles details of application programs.
- Example Functions:
  - When a request for a file comes in, Check whether it is available and user has permissions.



Protocol Standardization

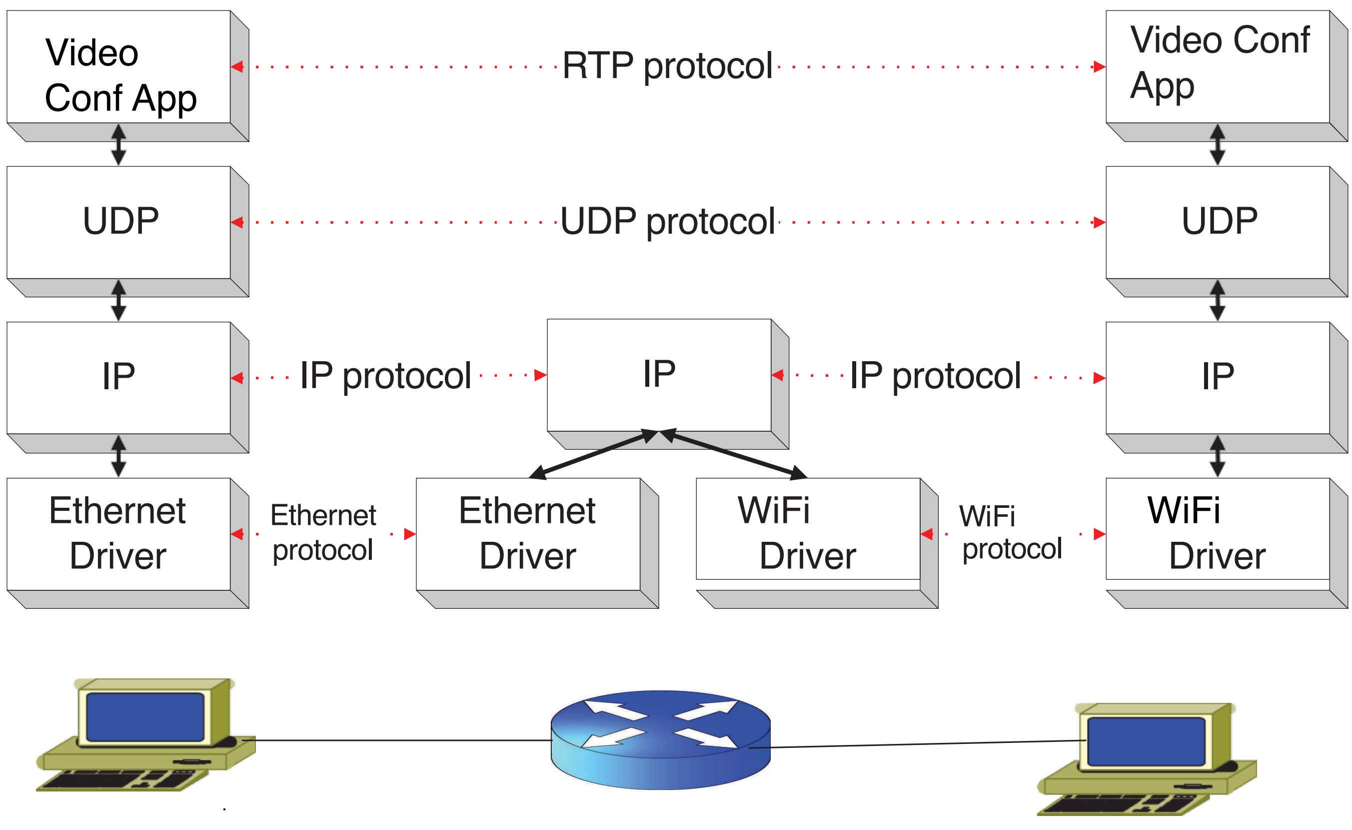
- Protocols:
  - Rules and formats that govern the communication between two peers
  - Need to be standardized.
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force
- Other standard bodies
  - ISO,ITU, IEEE,ANSI

Protocols 1



- IP protocol implemented on hosts and routers
- TCP and application only implemented on hosts

Protocols 2


Video Conf App
RTP protocol
Video Conf App
UDP
UDP protocol
UDP
IP
IP protocol
IP
IP protocol
IP
Ethernet Driver
Ethernet protocol
Ethernet Driver
WiFi Driver
WiFi protocol
WiFi Driver

Encapsulation

- As data moves down the protocol stack, each protocol adds layer-specific control information.