CS 31006: Computer Networks – History and Protocol Stack

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

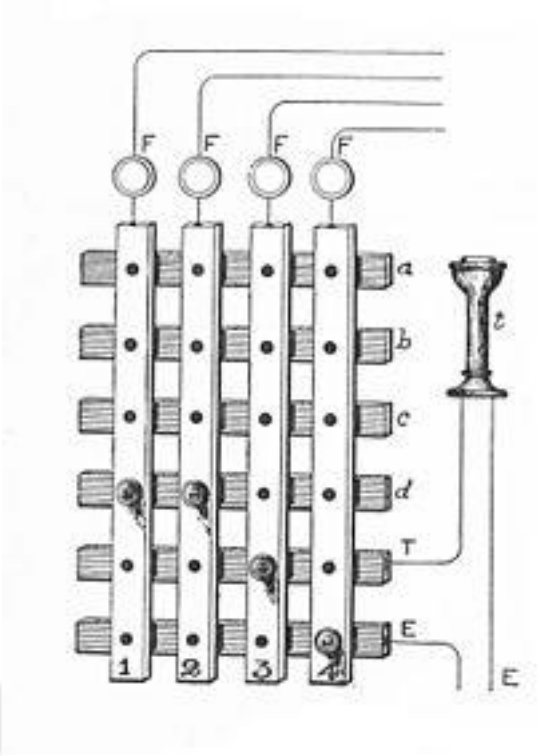
Rajat Subhra Chakraborty
rschakraborty@cse.iitkgp.ac.in

Sandip Chakraborty
sandipc@cse.iitkgp.ac.in

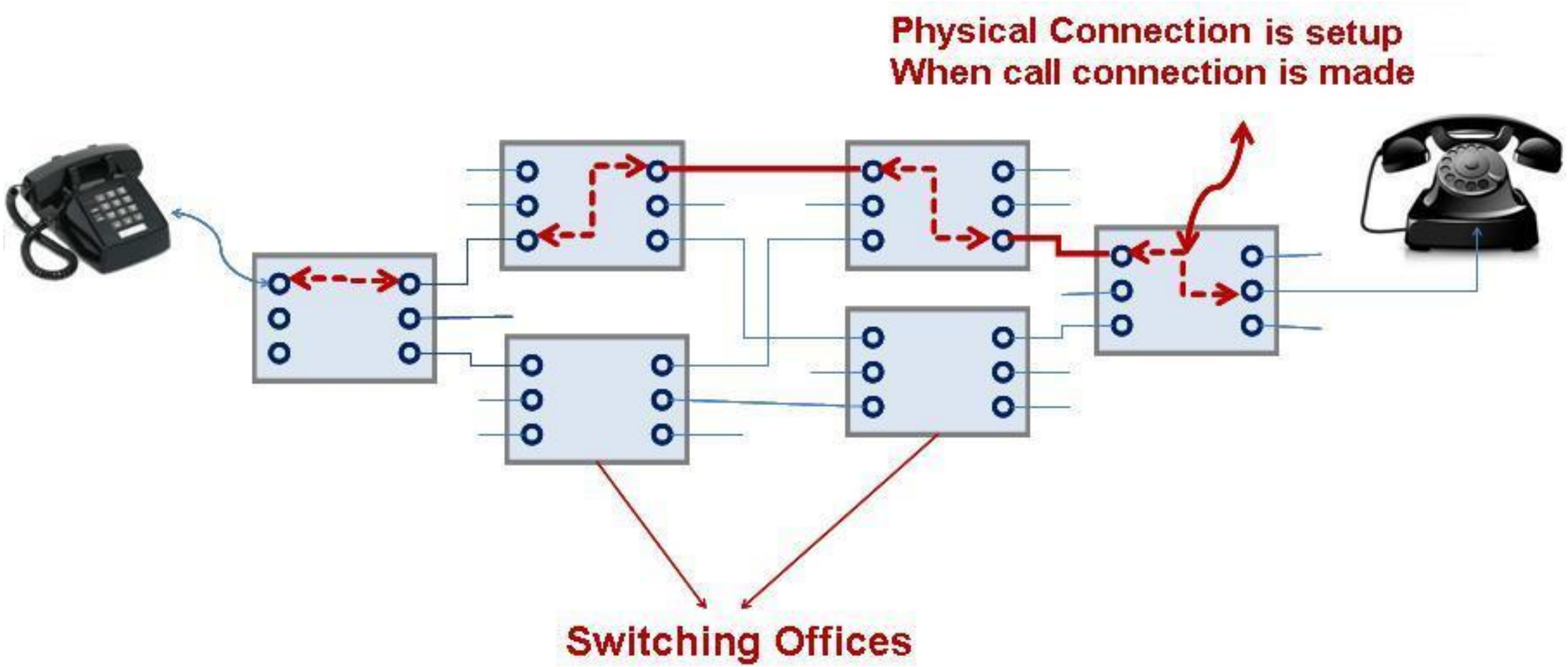
Circuit Switching – the Beginning

- The concept of how telephone switching works – creates a dedicated communication links between two communication nodes (telephones)

- In January 1878, the first telephone switch went into operation in New Haven Connecticut.



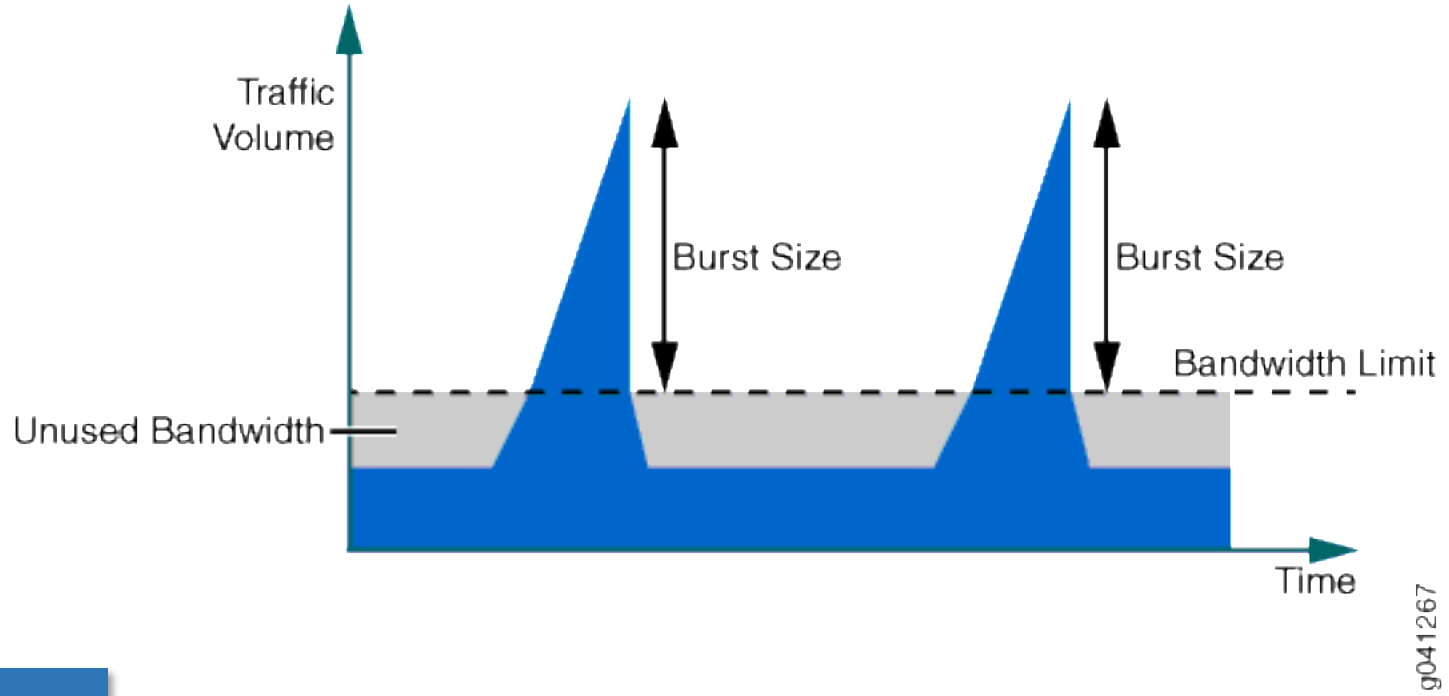
Circuit Switching



Source: http://comp380group4project1.web.unc.edu/technical-discussion/circuit-switching/

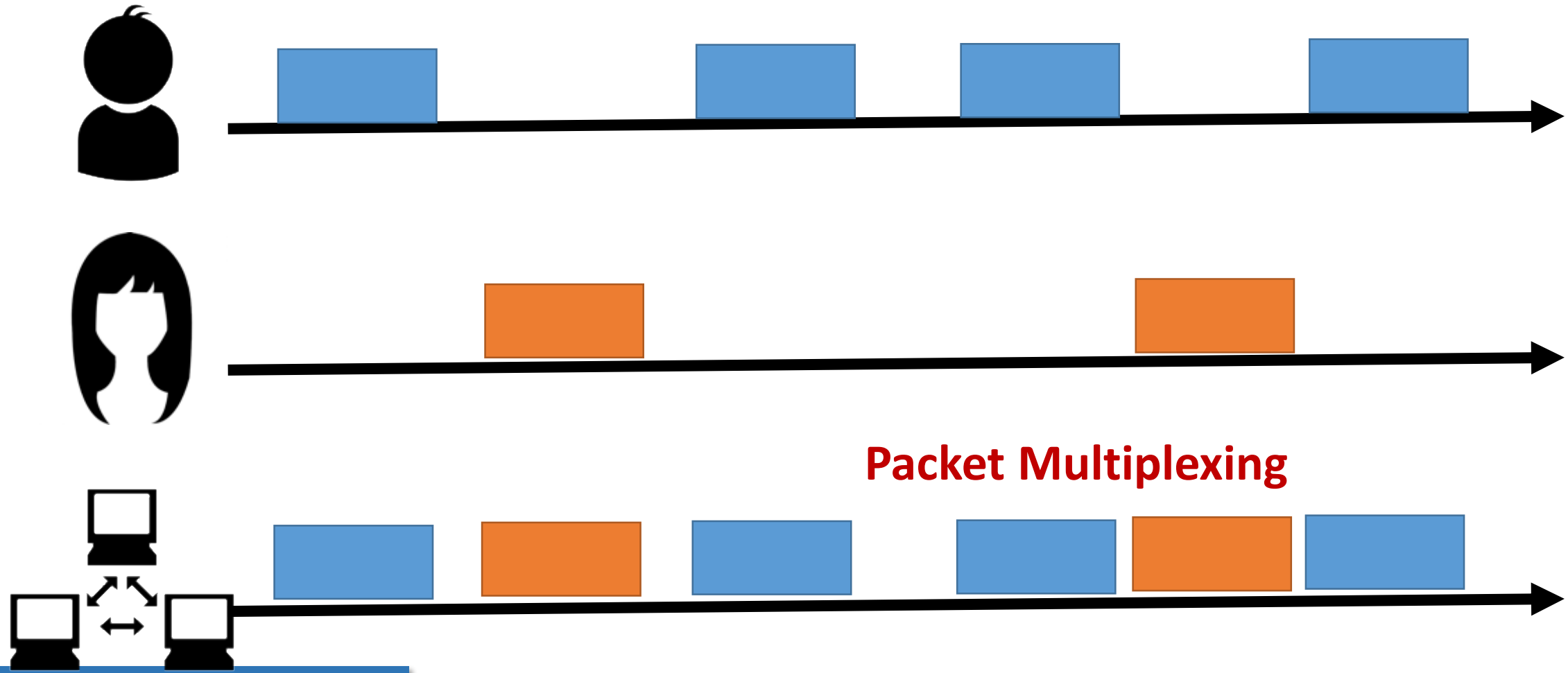
Problem with Circuit Switching for Data Communication

- Data traffic is bursty – uses an on-off pattern for data communication.



Packet Switching

- Decide data boundary from the communication of one user – data packets



The First Packet Switching Network

Robert Taylor was promoted to the head of the information processing office at **Defense Advanced Research Projects Agency (DARPA)** in June 1966. He intended to realize **Licklider's** ideas of an interconnected networking system. Bringing in **Larry Roberts** from MIT, he initiated a project to build such a network. The first ARPANET link was established between the **University of California, Los Angeles (UCLA)** and the **Stanford Research Institute** at 22:30 hours on October 29, 1969.

Kleinrock said in an interview: "We typed the L and we asked on the phone,

"Do you see the L?"

"Yes, we see the L," came the response.

We typed the O, and we asked, "Do you see the O."

"Yes, we see the O."

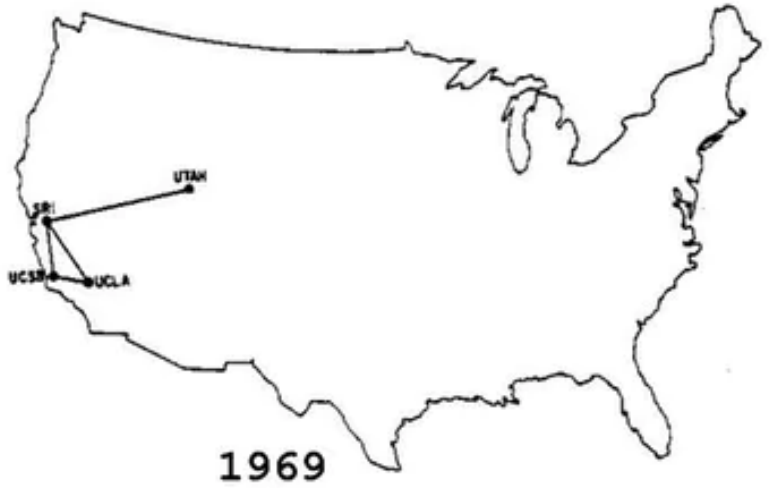
Then we typed the G, and the system crashed ...

Yet a revolution had begun"

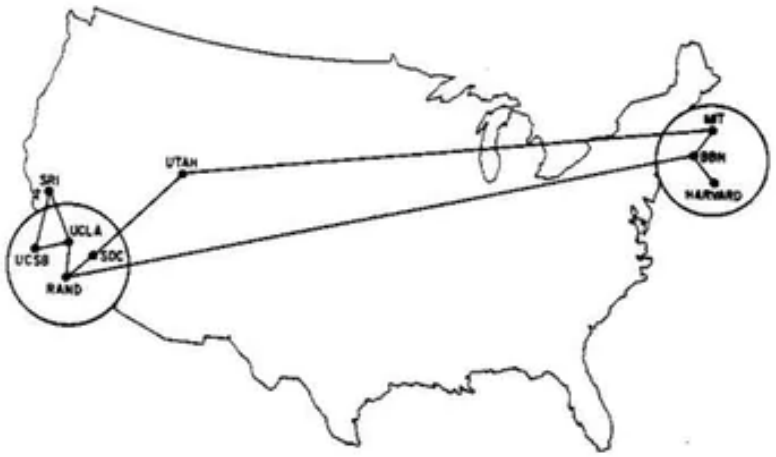
By December 5, 1969, a 4-node network was connected by adding the University of Utah and the University of California, Santa Barbara.

Source: Wikipedia

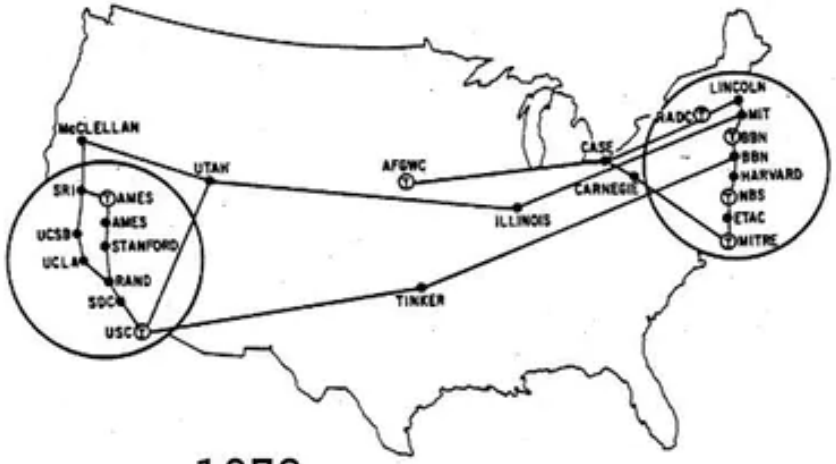
ARPANET – The First Packet Switching Network



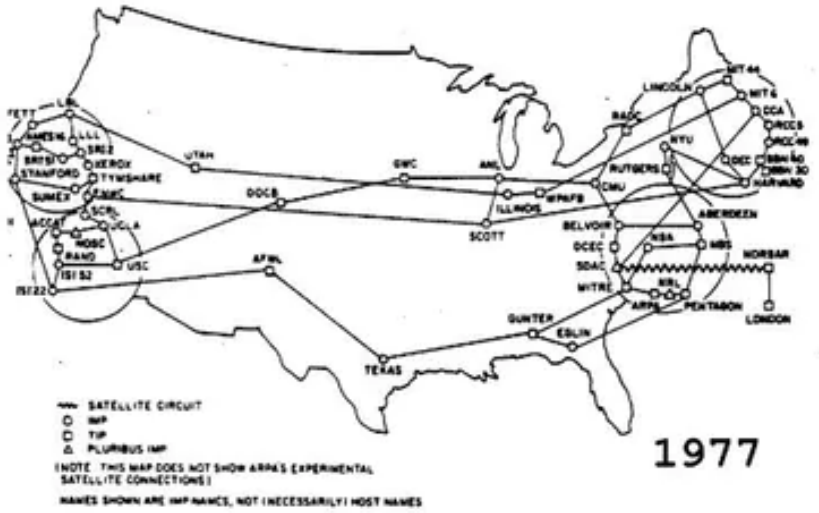
1969



1970



1972



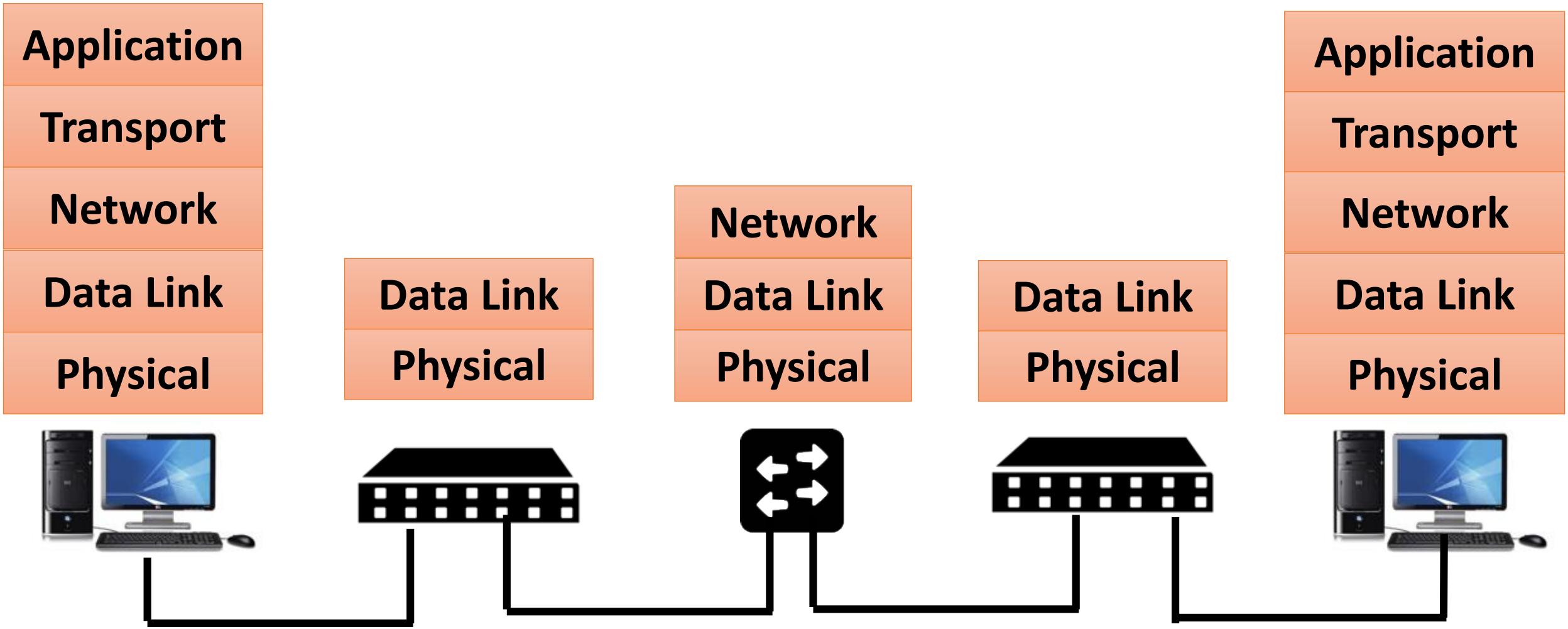
1977

History of Computer Networks

- https://www.youtube.com/watch?v=9hIQjrMHTv4

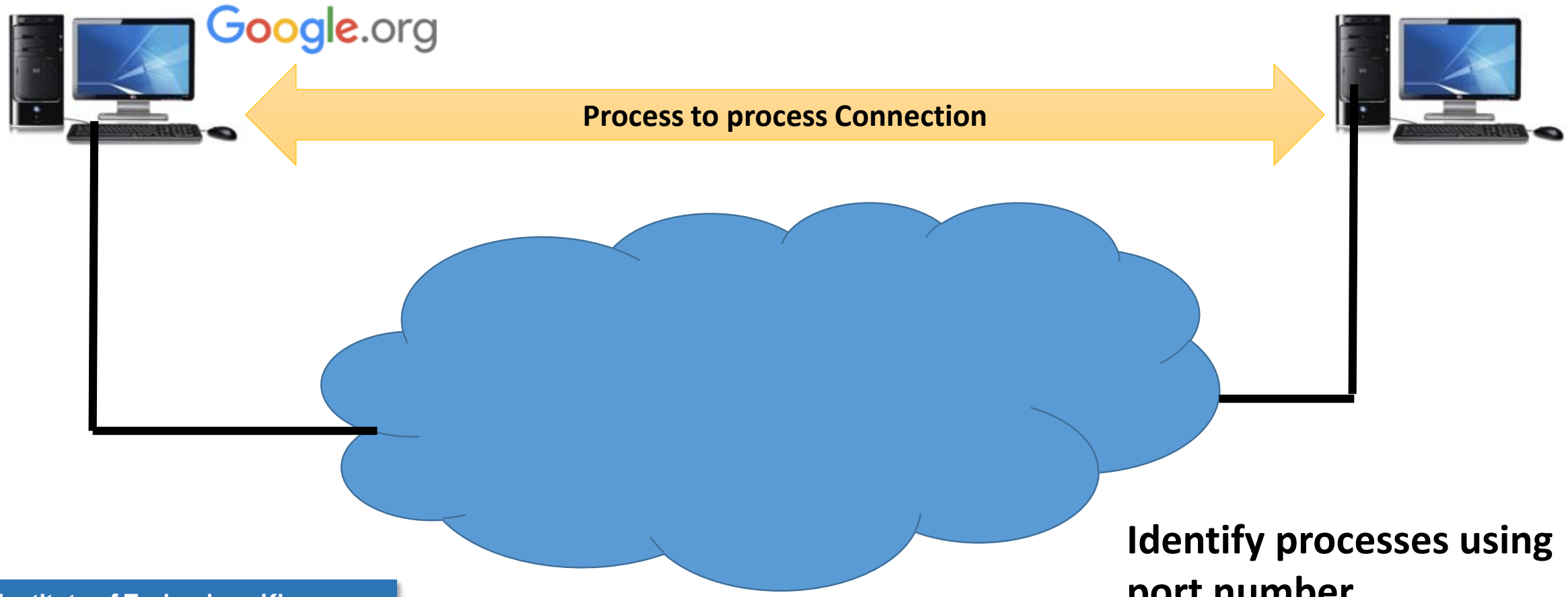
https://www.internetsociety.org/internet/history-internet

TCP/IP Protocol Stack



Transport Layer Services

- Connection oriented communication
  - Create a logical connection between your machine and Google server



Identify processes using port number

Transport Layer Services

- Connection oriented communication
  - Create a logical connection between your machine and Google server
- Ordered delivery of data packets
- Reliability
- Flow control
- Congestion avoidance

Network (Internet) Layer Services

- Connectionless communication - construct messages (datagrams) from packets and route them to the next hop
  - Why do transport layer packets need to be converted to datagrams at every hop?
- Host addressing - use IP addresses to uniquely identify a host in the network
- Datagram routing - decide the path to route every datagram from the network graph

Data Link Layer Services

- Encapsulation of network layer data packets into frames
- Frame synchronization
- Error control
- Flow control
  - Why do we need flow control at the link layer?
- Channel access / medium access, Physical addressing

Data Link Layer

Logical Link Control (LLC)	**Error control and flow control**
Medium Access Control (MAC)	**Channel access and physical layer addressing**

Addressing a Host in the Network

- Every host has two addresses –
  - Logical address (IP address) – used to find out the path towards a host
  - Physical address (MAC address) - uniquely identify a host in the Internet

- Logical address – Dr. Sandip Chakraborty, Room-311, Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur - 721302, West Bengal, India

- Physical address – My Aadhar number (XXXX XXXX XXXX)

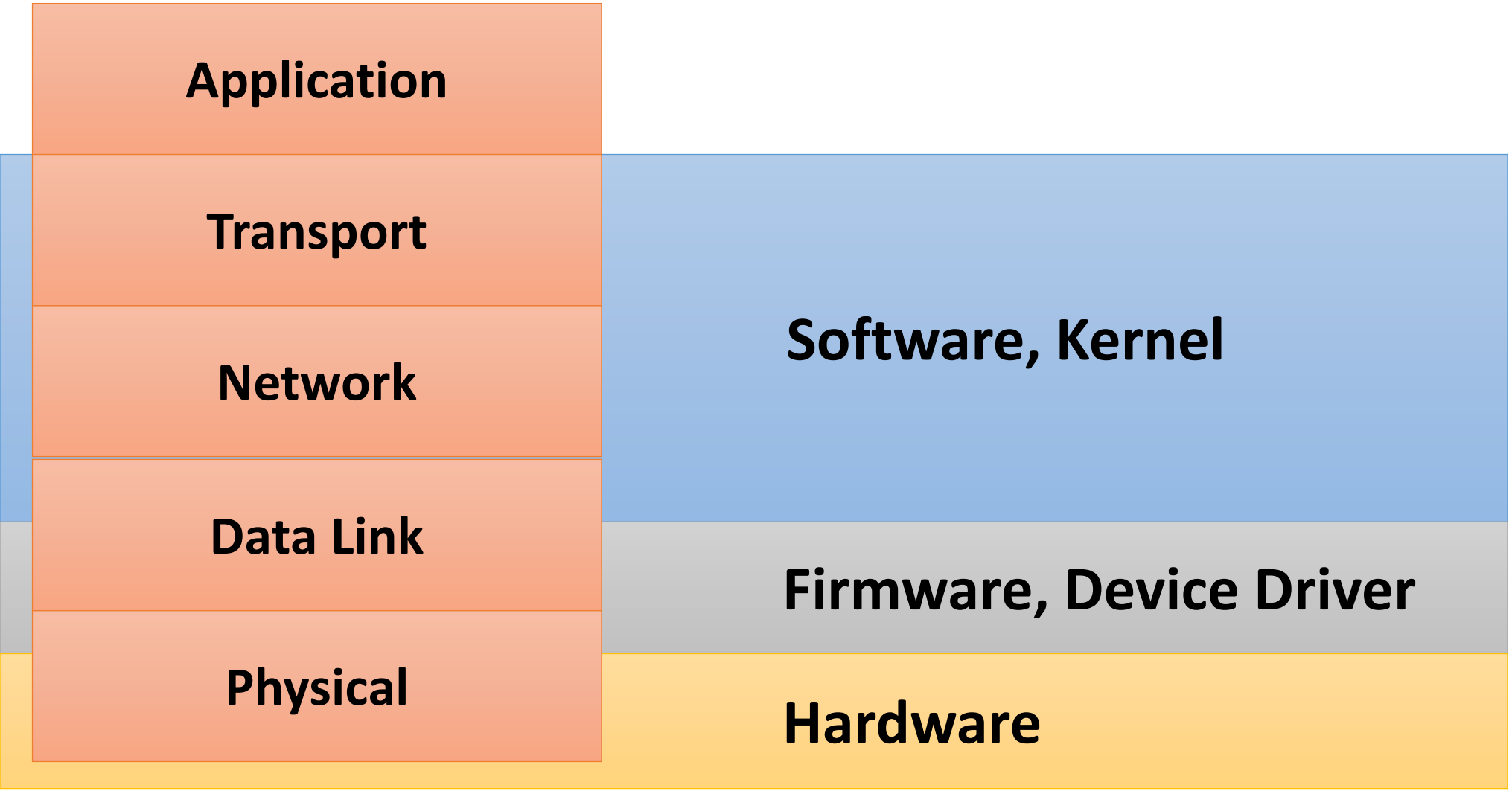
Addressing a Host in the Network

- Every host has two addresses –
  - Logical address (IP address) – used to find out the path towards a host
  - Physical address (MAC address) - uniquely identify a host in the Internet

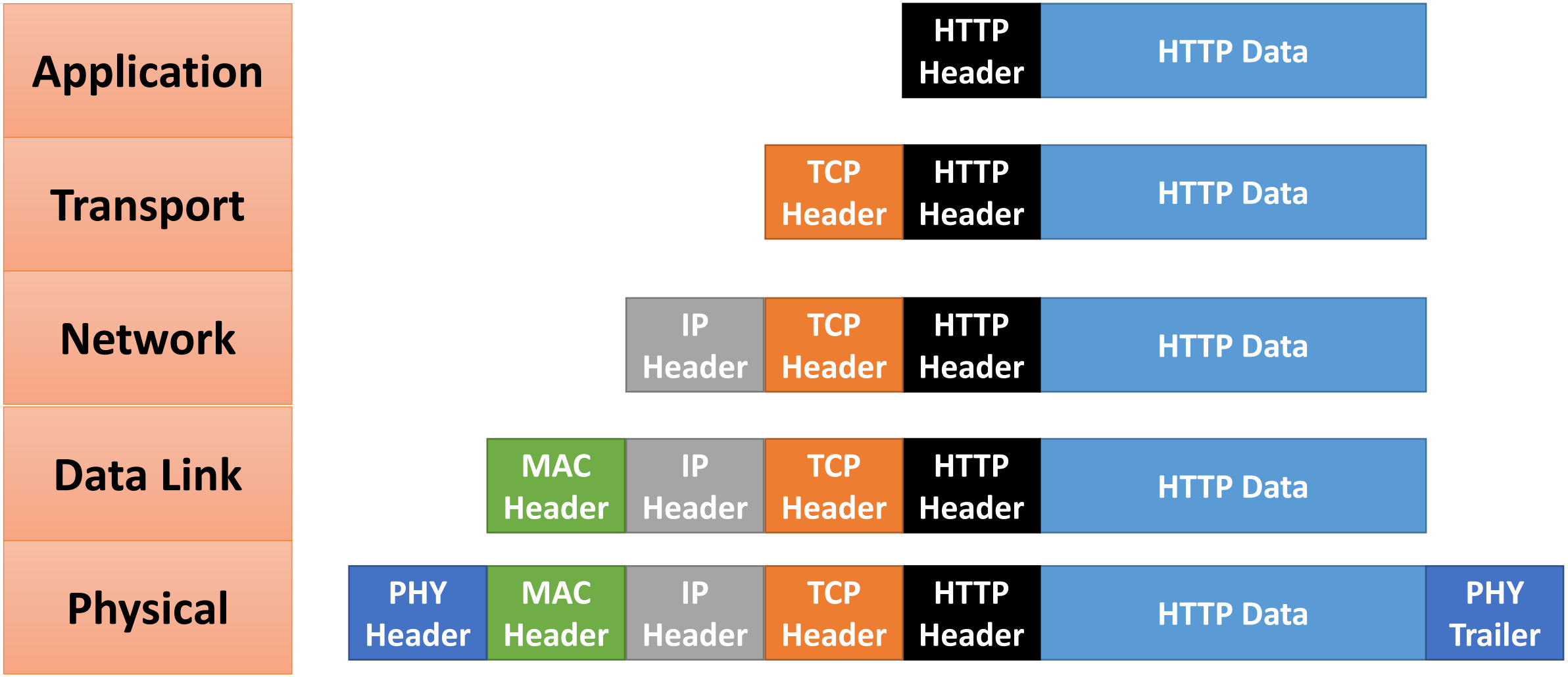
- Logical address (32 bit IP address) – 172.16.32.64

- Physical address (64 bit MAC address) – 00:00:00:14:22:01:2C:4D

Protocol Stack Implementation in a Host

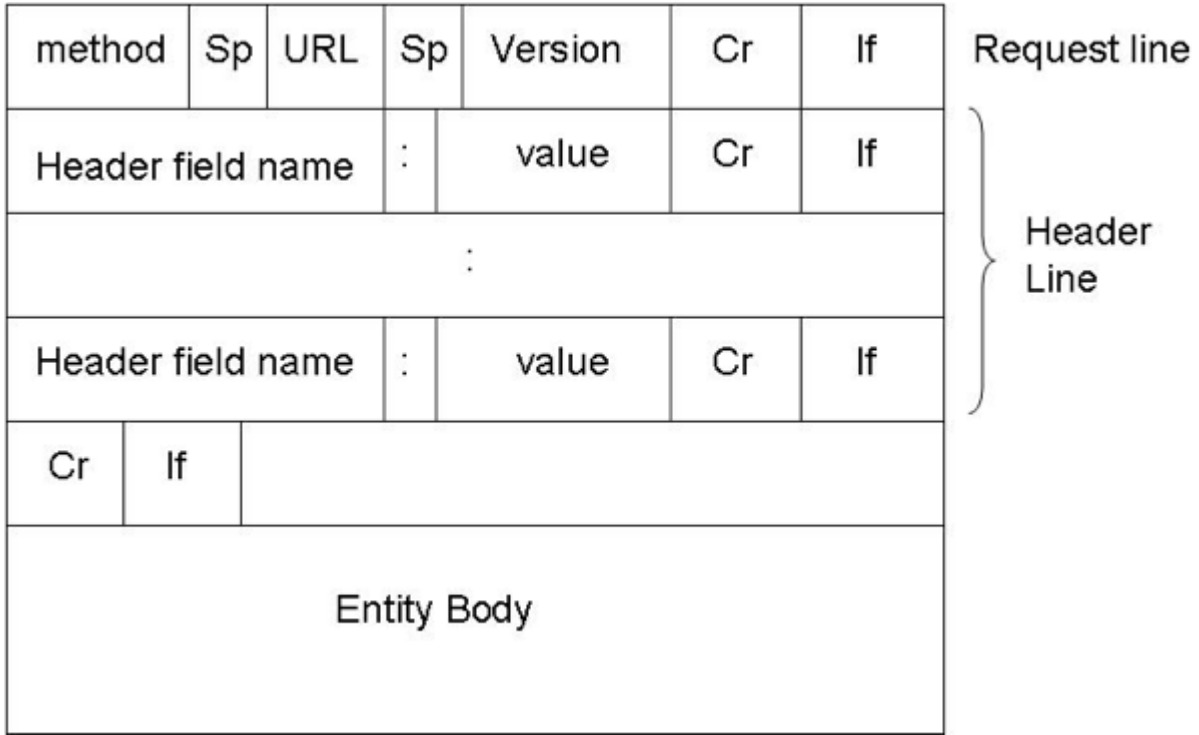


How Application Data Passes Through Different Layers



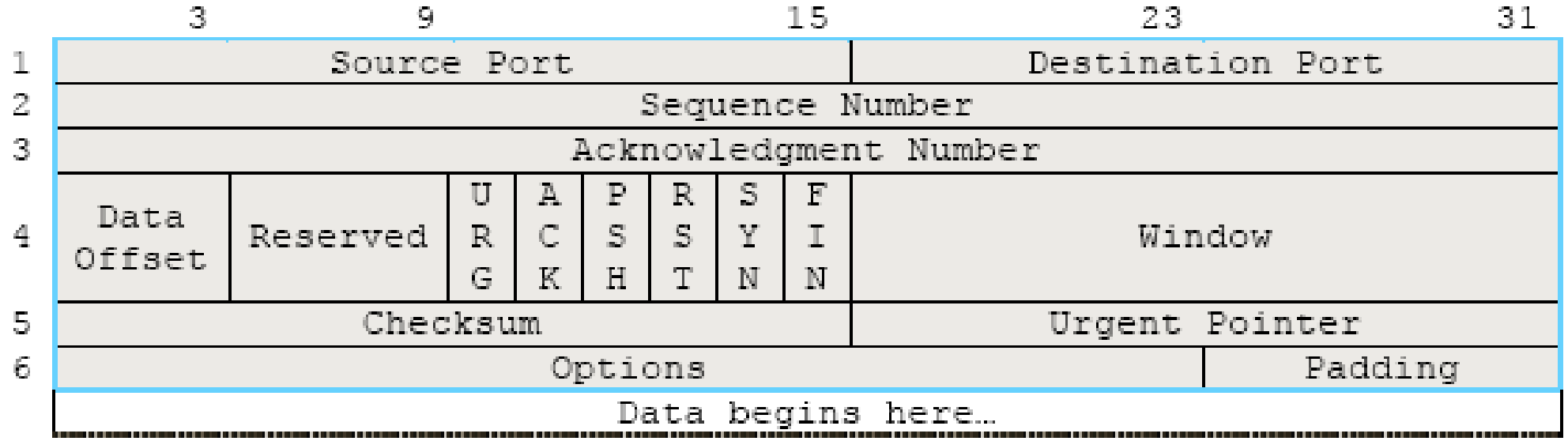
How do you access a page at www.google.com?

1. **Application Layer:** Use DNS to get the IP address of the Google server - DNS returns 74.125.224.72
2. **Application Layer:** Construct an HTTP GET Request –
   GET 74.125.224.72/index.html HTTP/1.1
   Construct an HTTP packet and forward it to the transport layer



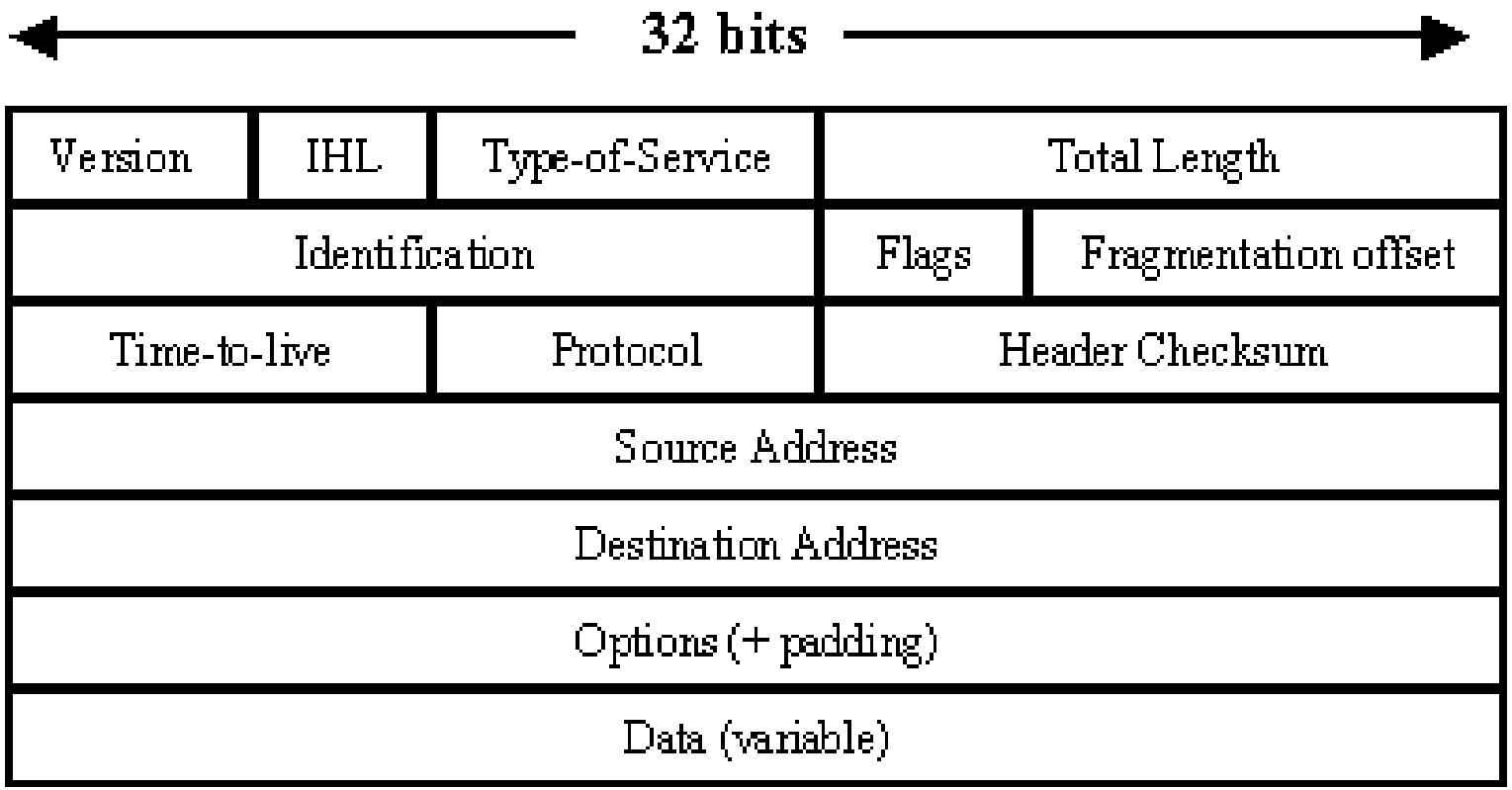
How do you access a page at www.google.com?

3. **Transport Layer:** Construct the transport layer packet (TCP packet)
   - Source port: 3324 (port address corresponding to your browser tab)
   - Destination port: 80 (port for a HTTP server)



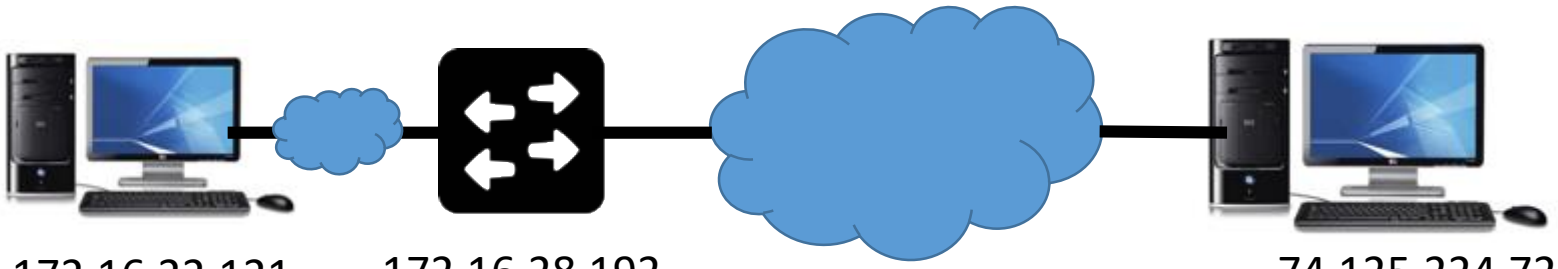
How do you access a page at www.google.com?

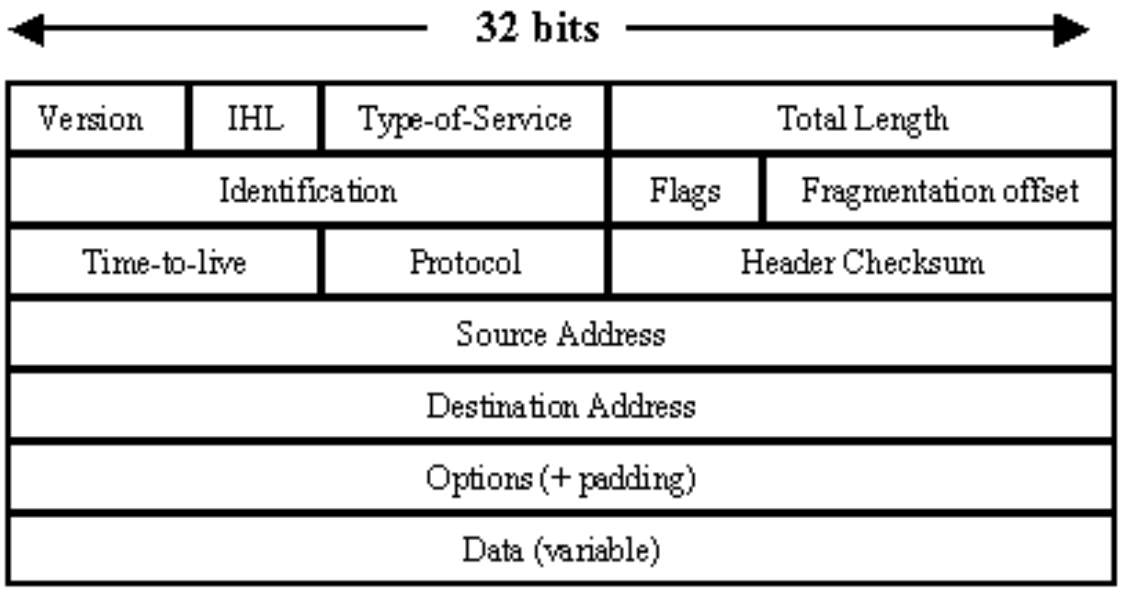
3. **Network Layer:** Construct the network layer packet (IP packet)
   - Source IP: 172.16.22.121 (IP of your machine)
   - Destination IP: Next hop IP for 74.125.224.72 (use the routing procedure to get this IP)



How do you access a page at www.google.com?

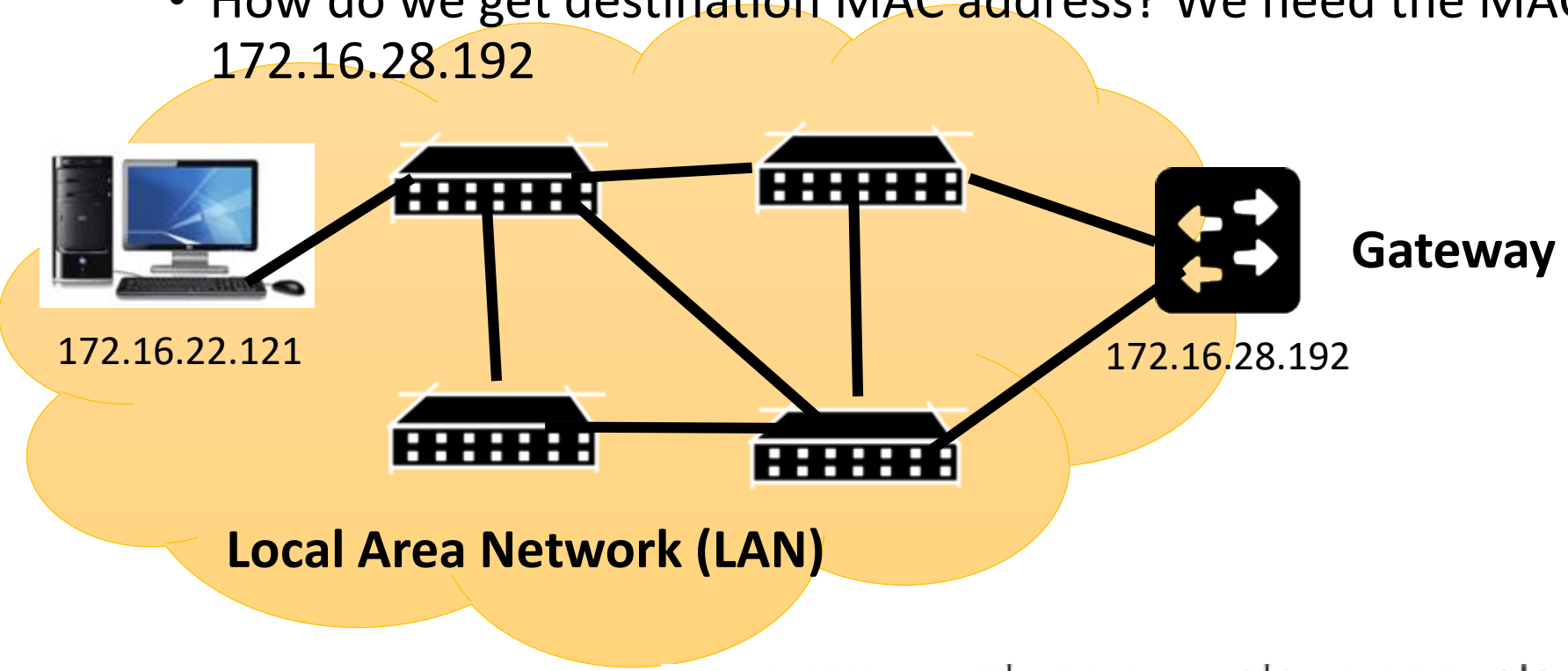
4. **Network Layer:** Use the routing procedure to find out the next hop IP to reach the Google server at 74.125.224.72. Let this IP be 172.16.28.192

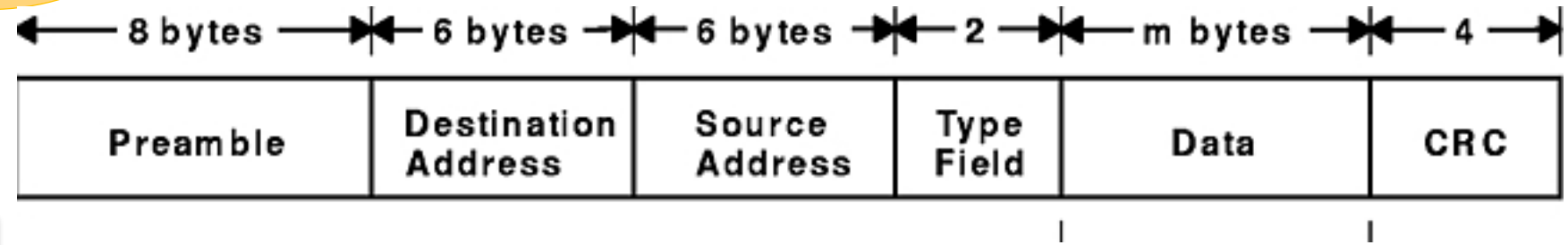




How do you access a page at www.google.com?

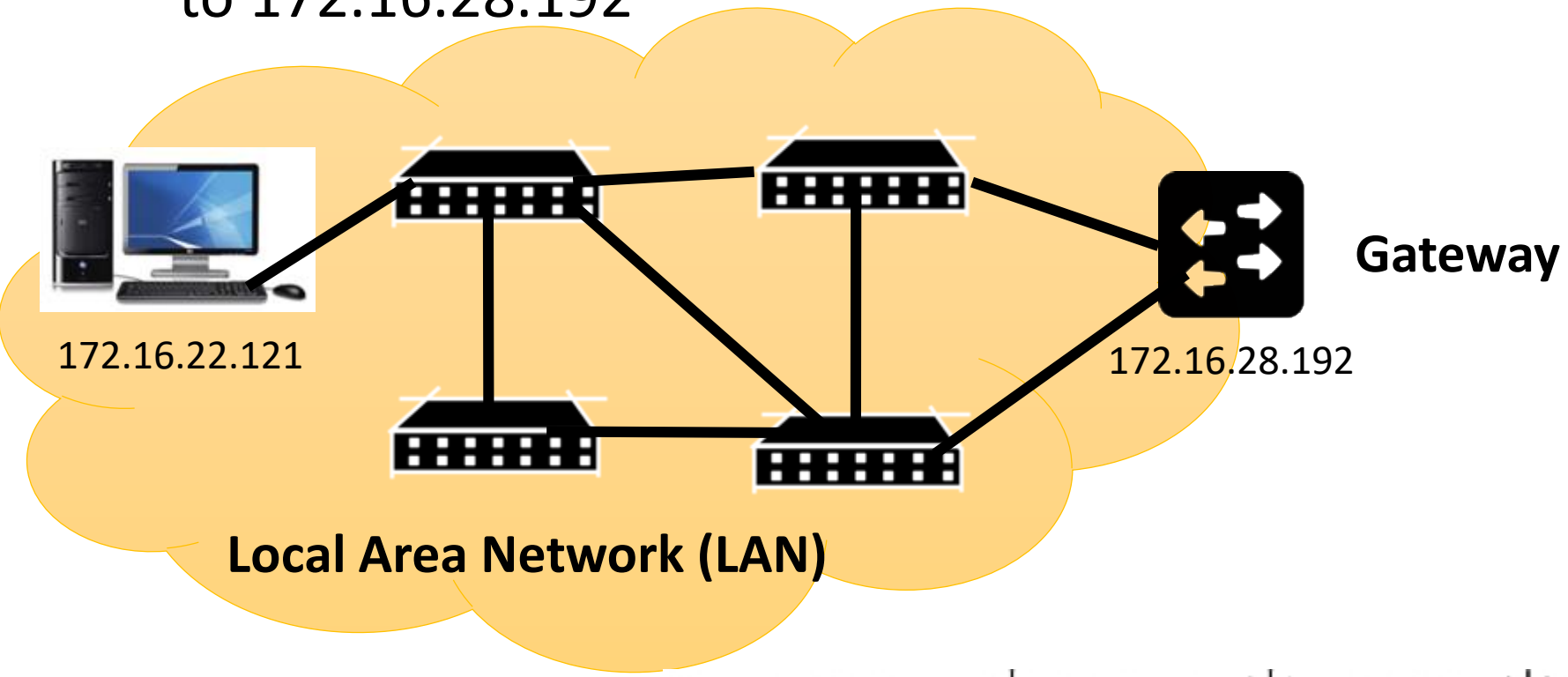
5. **Data Link Layer:** Construct the data link layer frame from the IP datagram.
   - How do we get destination MAC address? We need the MAC address corresponding to 172.16.28.192

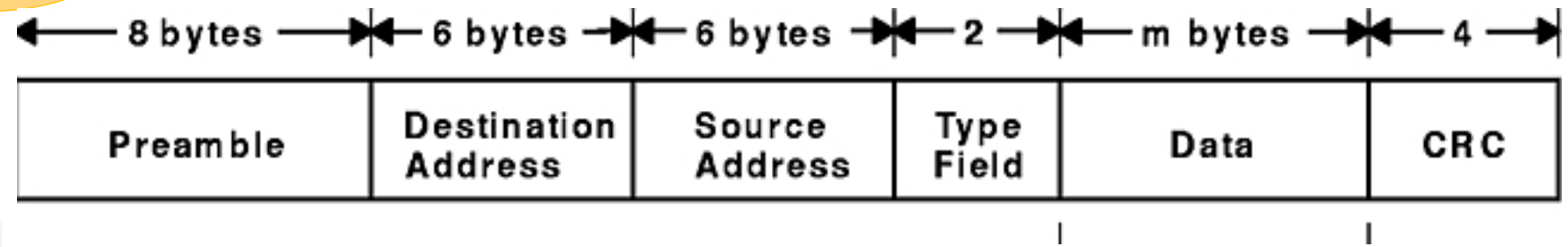




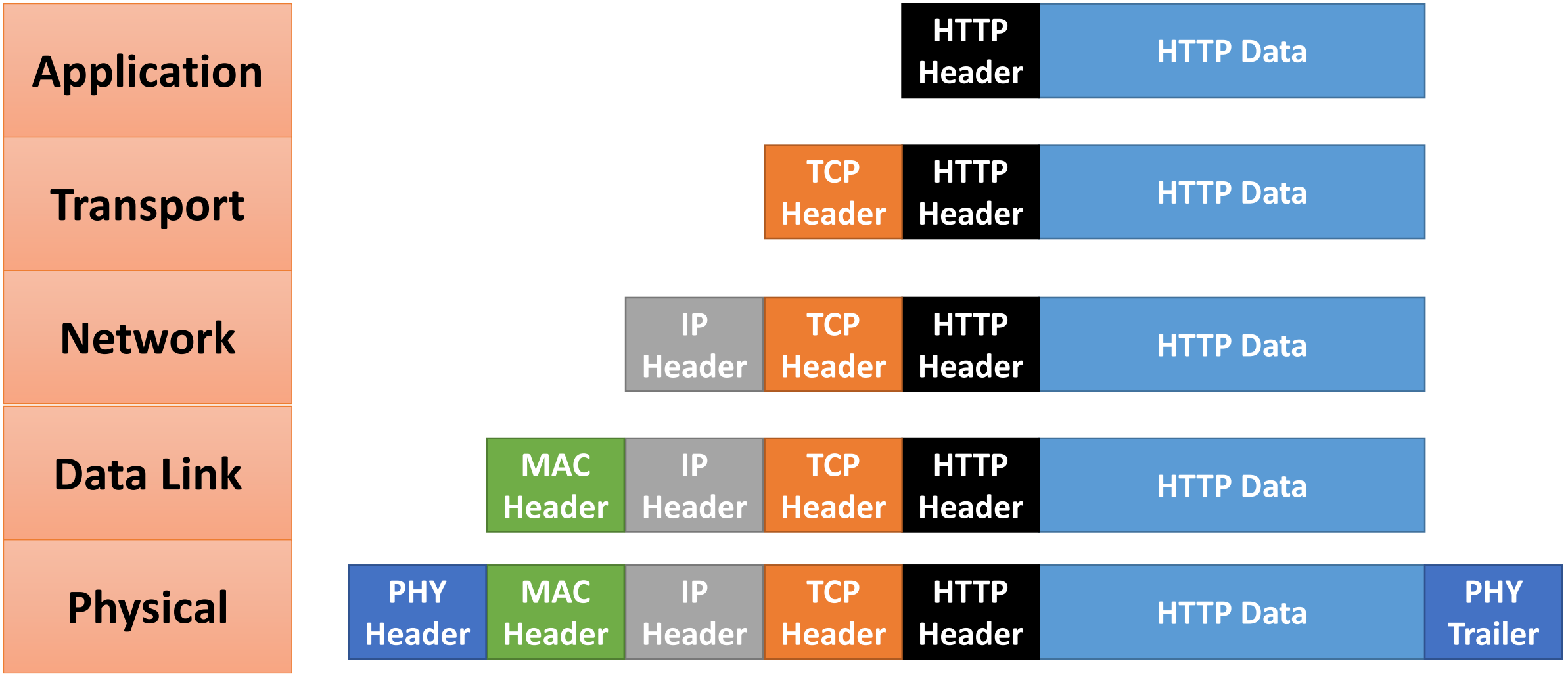
How do you access a page at www.google.com?

6. **Data Link Layer:** Use ARP protocol to get the MAC address corresponding to 172.16.28.192





In Summary


Application
Transport
Network
Data Link
Physical
HTTP Header
HTTP Data
TCP Header
HTTP Header
HTTP Data
IP Header
TCP Header
HTTP Header
HTTP Data
MAC Header
IP Header
TCP Header
HTTP Header
HTTP Data
PHY Header
MAC Header
IP Header
TCP Header
HTTP Header
HTTP Data
PHY Trailer