UCDAVIS
UNIVERSITY OF CALIFORNIA

Internet of Things

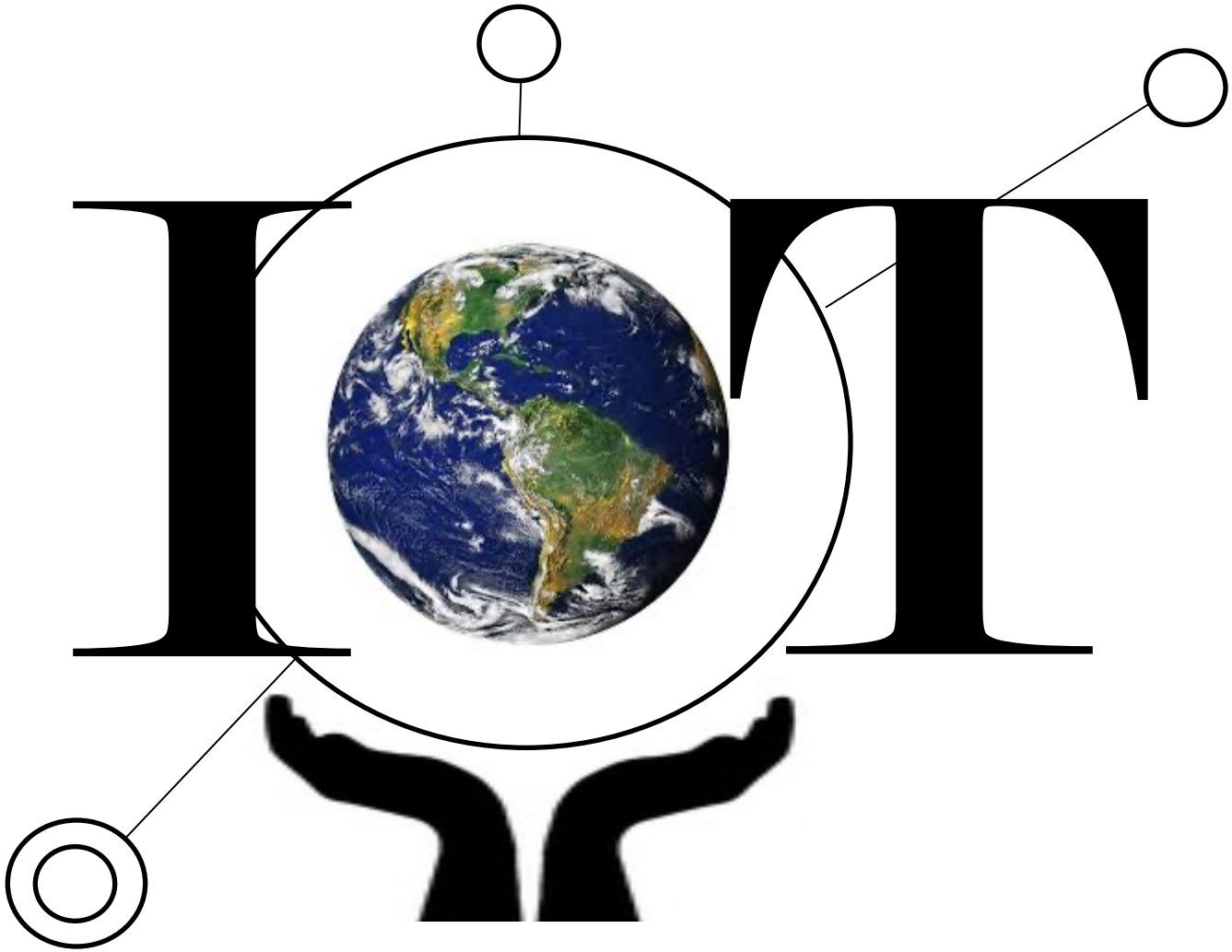
Senior Design Project Course

IP Communication

Lecturer: Avesta Sasan

University of California Davis

Lets Get Started:



Focus of Today's Lecture: (Review)

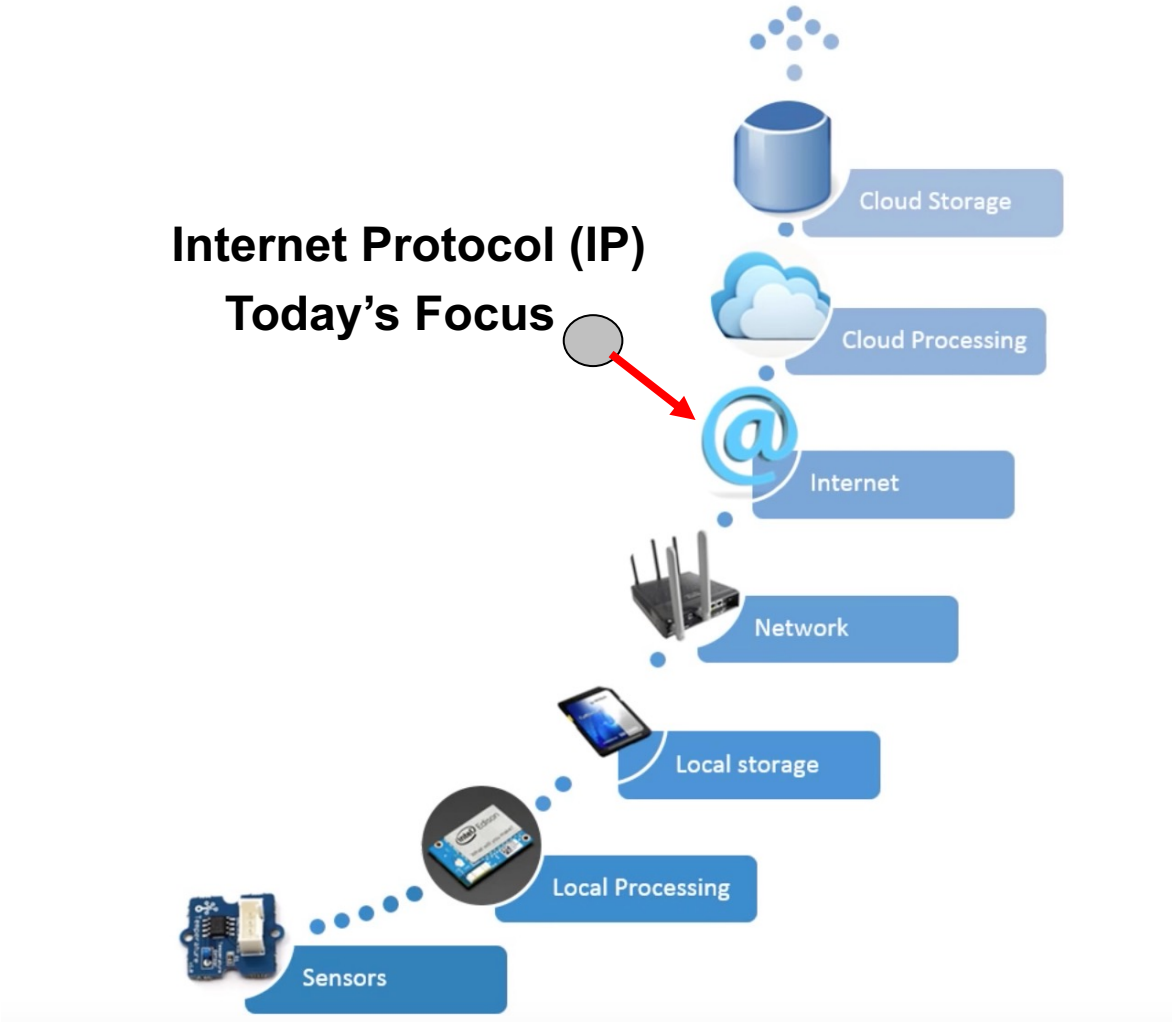
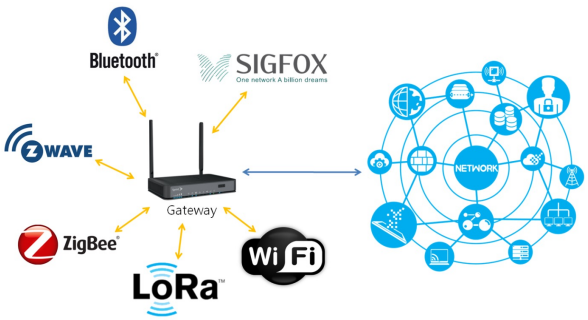


Image source: http://www.cchc.cl/informacion-a-la-comunidad/industria-de-la-construccion/personaje/

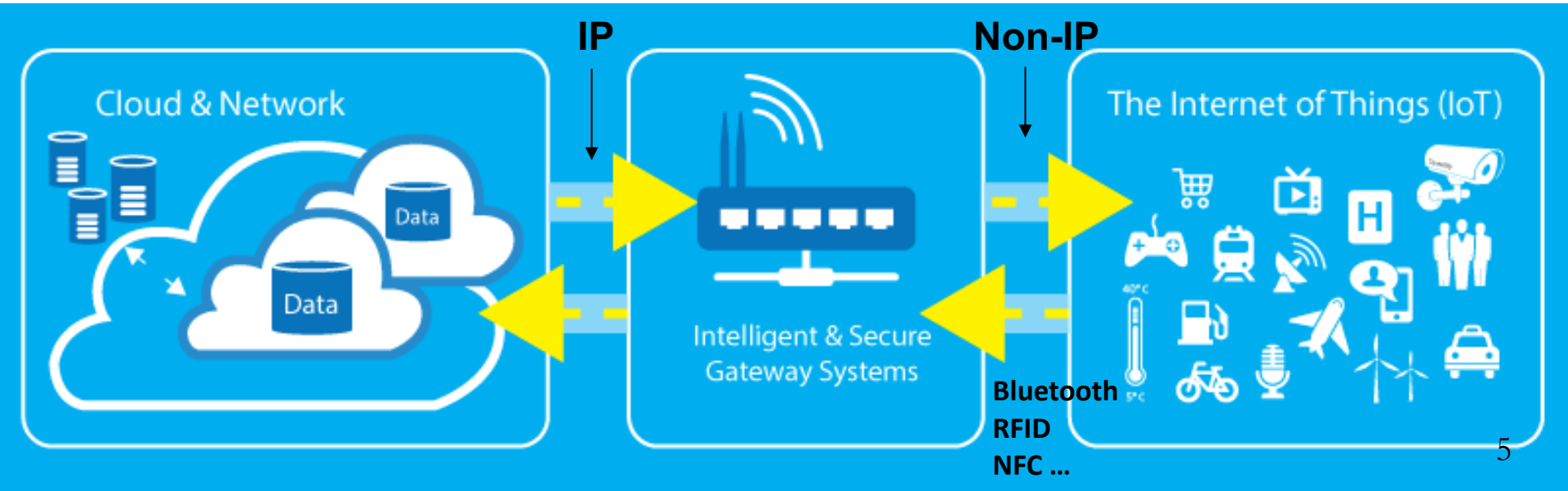
Why Using Gateway? (Review)

1. **Lowering Power**
   - Sensor send data to a gateway in short range, gate way send the data to cloud.
2. **Supporting varying** to/from sensor communication **protocols**
   - Each sensor may have a different protocol. Gateway translate it to IP
3. **Filtering the data**
   - Usually small fraction of data is usable. Filtering could be done at gateways.
4. **Reducing latency**
   - Many IoT devices too small to do the processing themseleves, and it take too long to wait for cloud. Gateway remedy this.
5. **Improving security**
   - Can afford to make data transmission through gateway more secure.
   - Prevent too many lightly secured sensors to be connected to internet.



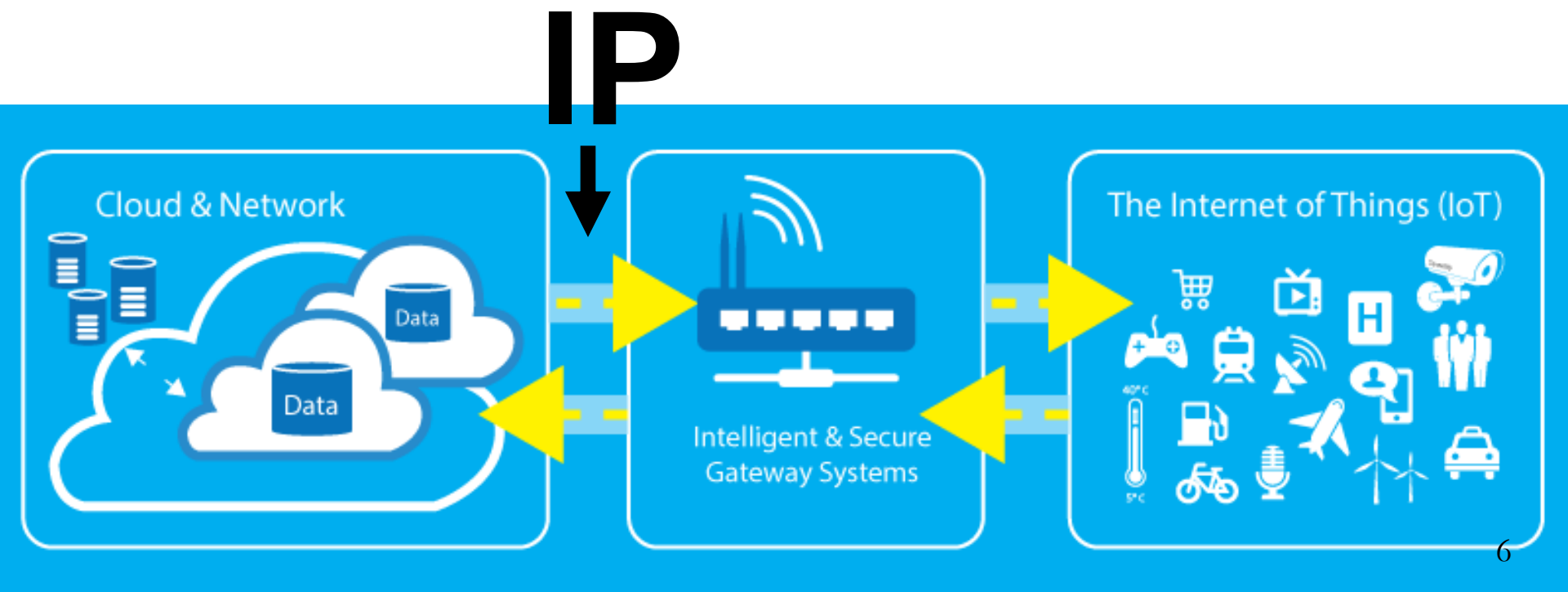
Gateway & Protocol Translation (Review)

- IoT devices can connect to Internet using Internet Protocol (IP) stack.
  - **Problem:**
    - IP stack is very **complex.**
    - Demands a large power and memory from the connecting devices.
- **Gateway removes the need for direct connection to internet.**
  - IoT devices can also connect locally through non-IP networks
  - Consume less power and offer larger mobility, and connect to the Internet via a smart gateway.



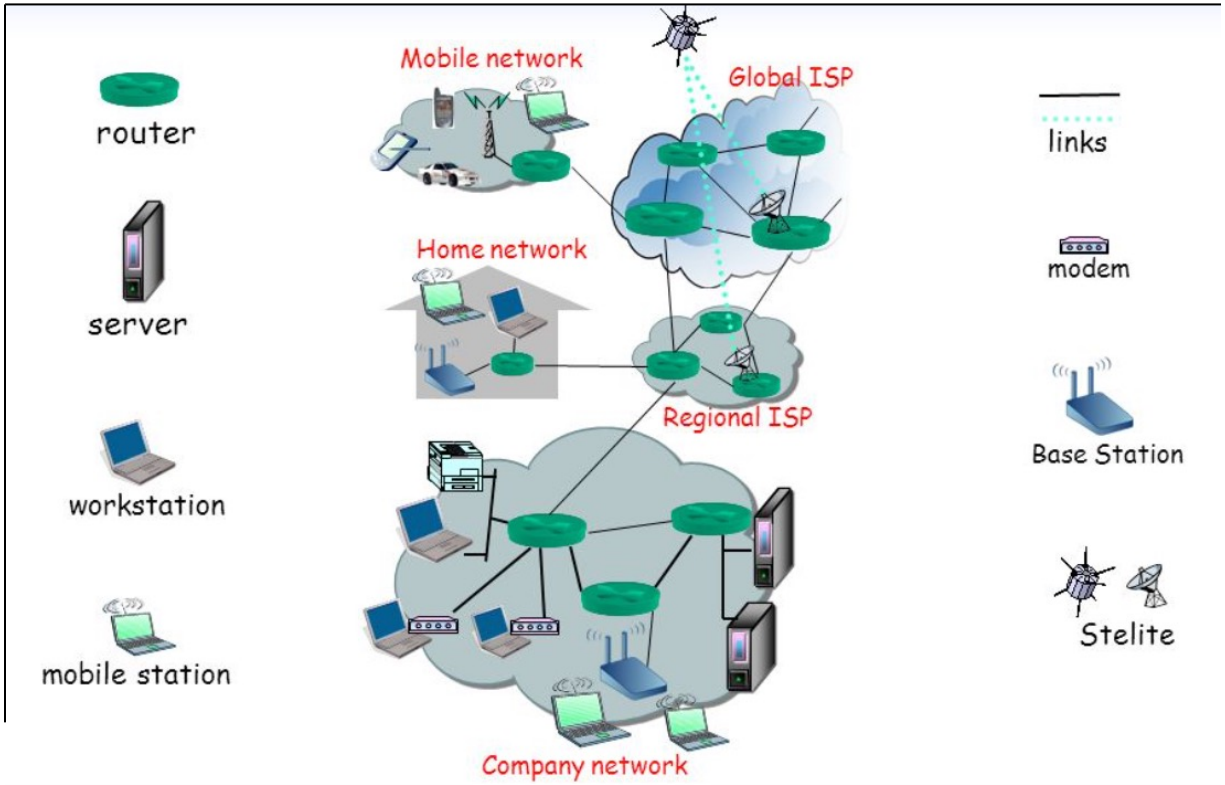
Internet Protocol

- Internet Protocol (**IP**) widely used to send/receive data over **Internet**.
- There are courses that are entirely dedicated to TCP/IP communication
- In this lecture, we are going to just briefly review this technology



Internet

- **Definition**: A set of *inter*connected *net*works
- Underlying networks can be completely different
- (TCP/)IP is what links them

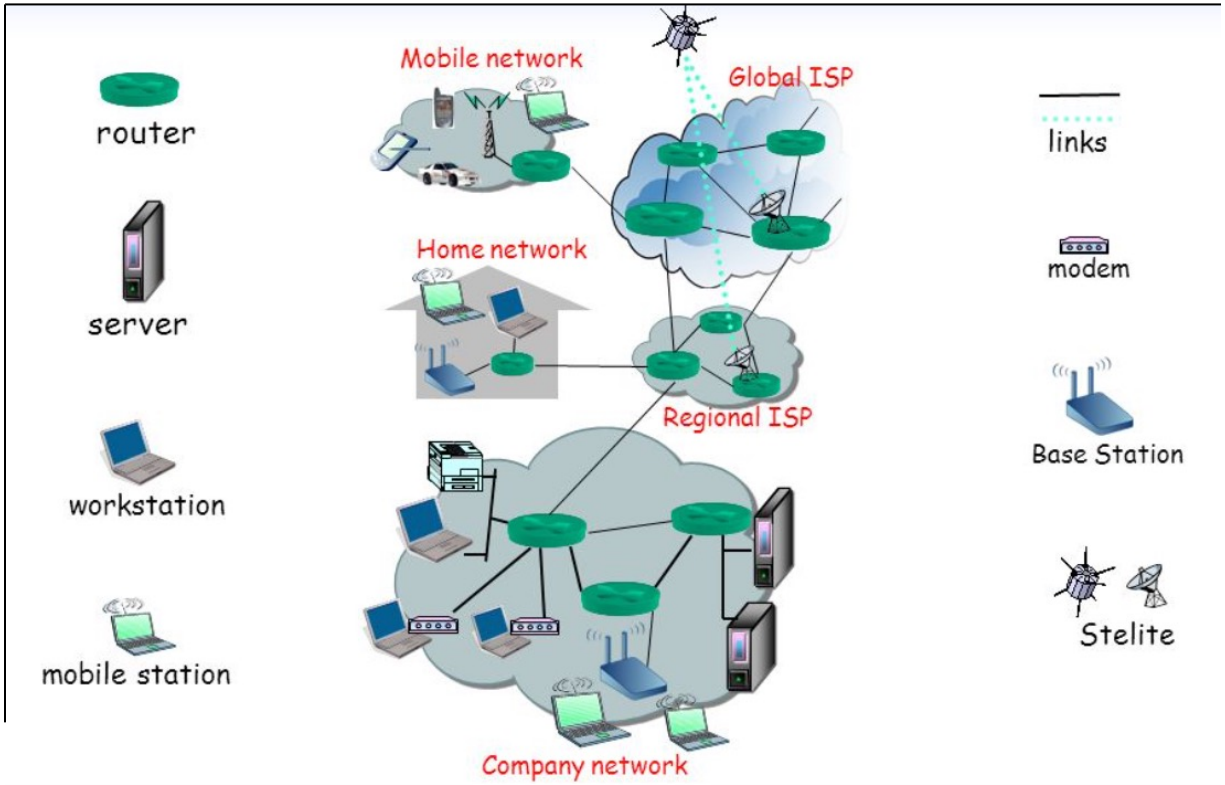


Internet

- Internet is a **distributed system** (no central control)
- Billions of connected devices
- Host: end systems that run network applications (protocols)
- Communication links
  - Fiber, copper, radio, satellite
  - Bandwidth matters the most. Delay, jitter etc. are also important
- **IP** is the glue that connects all these devices
- Analogous to sending mail using the postal service

Routers:

- ***Routers:*** devices on multiple networks that pass traffic between them
  - Individual networks pass traffic from one router or endpoint to another
  - TCP/IP hides the details as much as possible



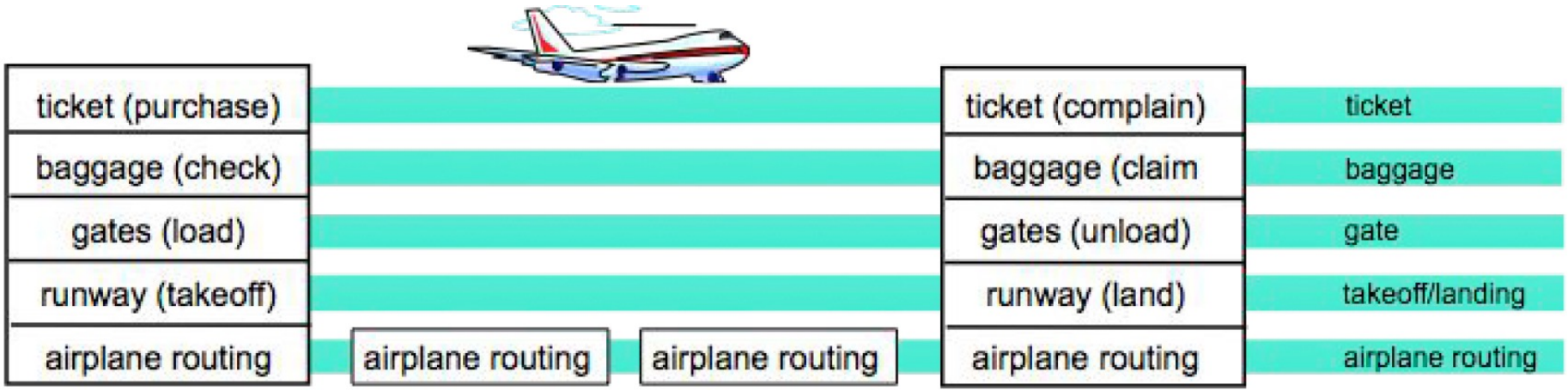
Protocol Layering

- Networks are complex! Too many pieces
  - Hosts, Routers
  - Links of various media
  - Applications
  - Protocols
  - Hardware, software

- Layers Implement service abstractions
  - Each relying on services provided by layer below

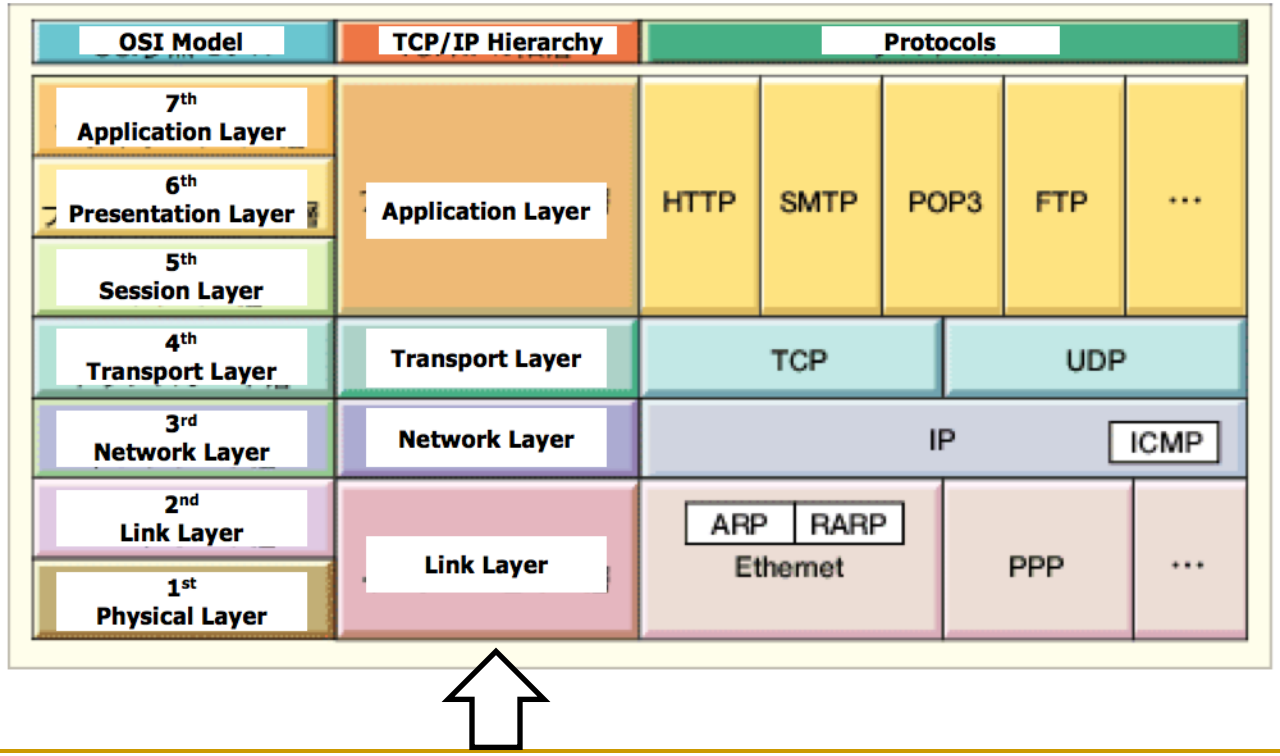
Protocol Layering Example

- Layers Implement service abstractions
  - Each relying on services provided by layer below



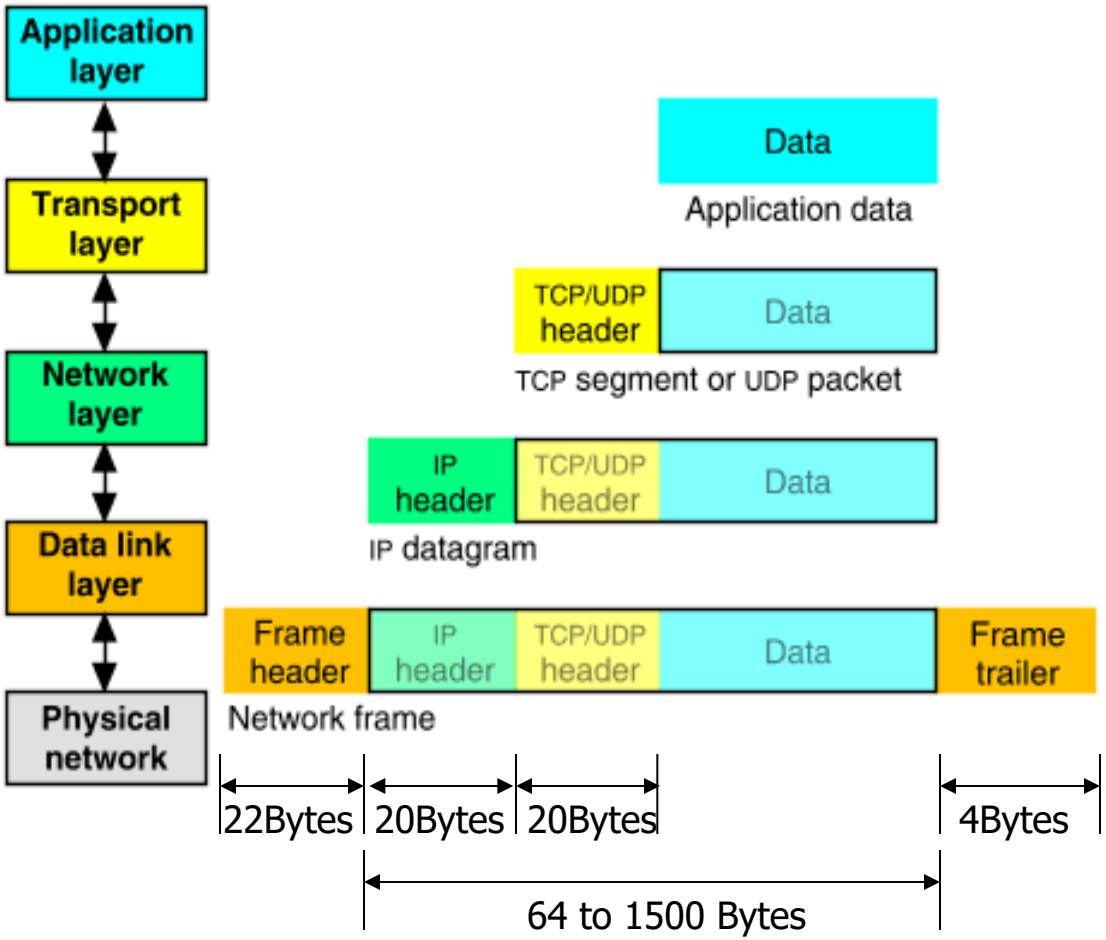
TCP/IP Network Model

- **A layered Protocol:**
  - Layer 1 : **Link :** includes device driver and network interface card
  - Layer 2 : **Network :** handles the movement of packets, e.g., Routing
  - Layer 3 : **Transport :** provides a reliable flow of data between two hosts
  - Layer 4 : **Application :** handles the details of the particular application



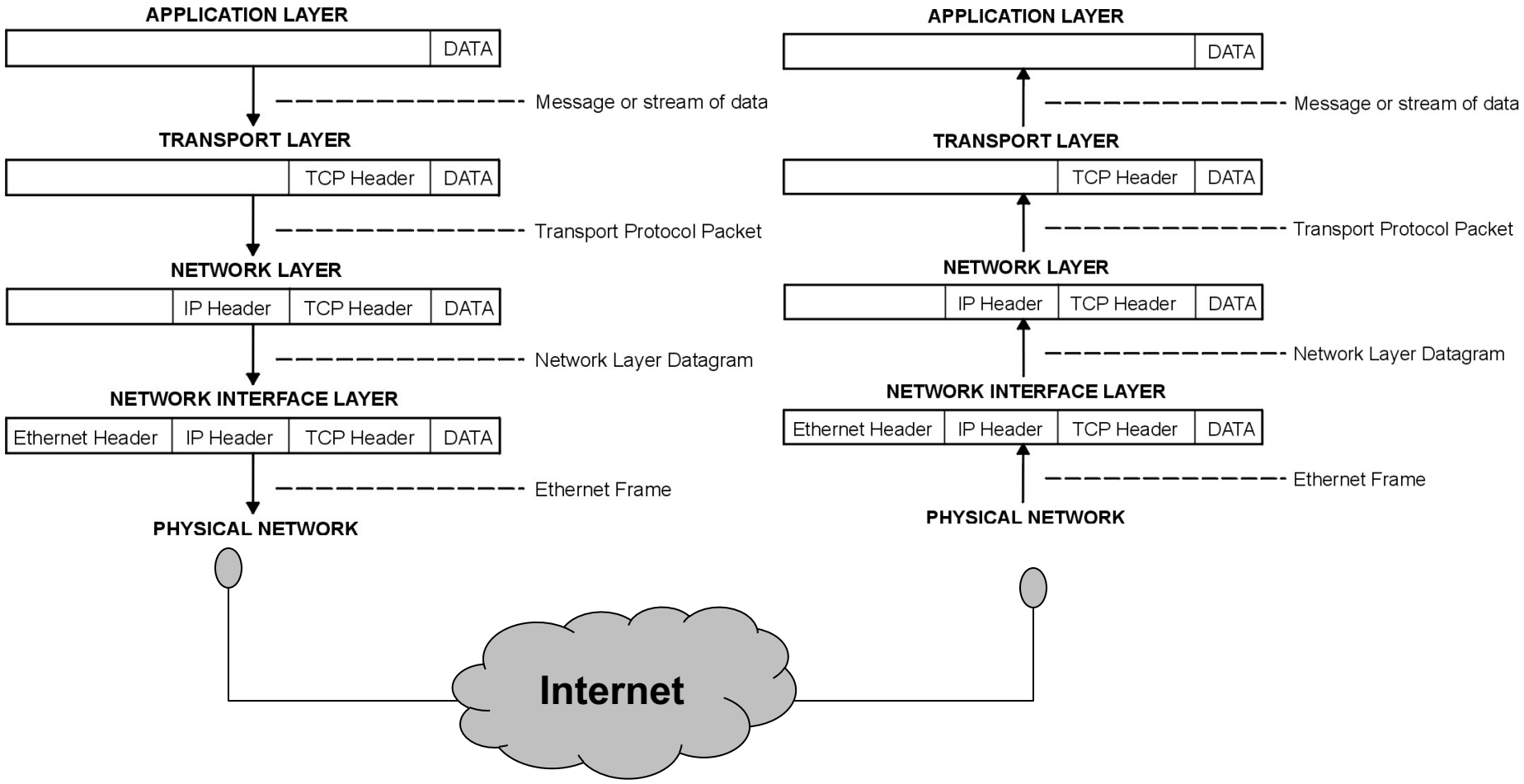
Packet Encapsulation

- The data is sent down the protocol stack
- Each layers adds to the data the required header



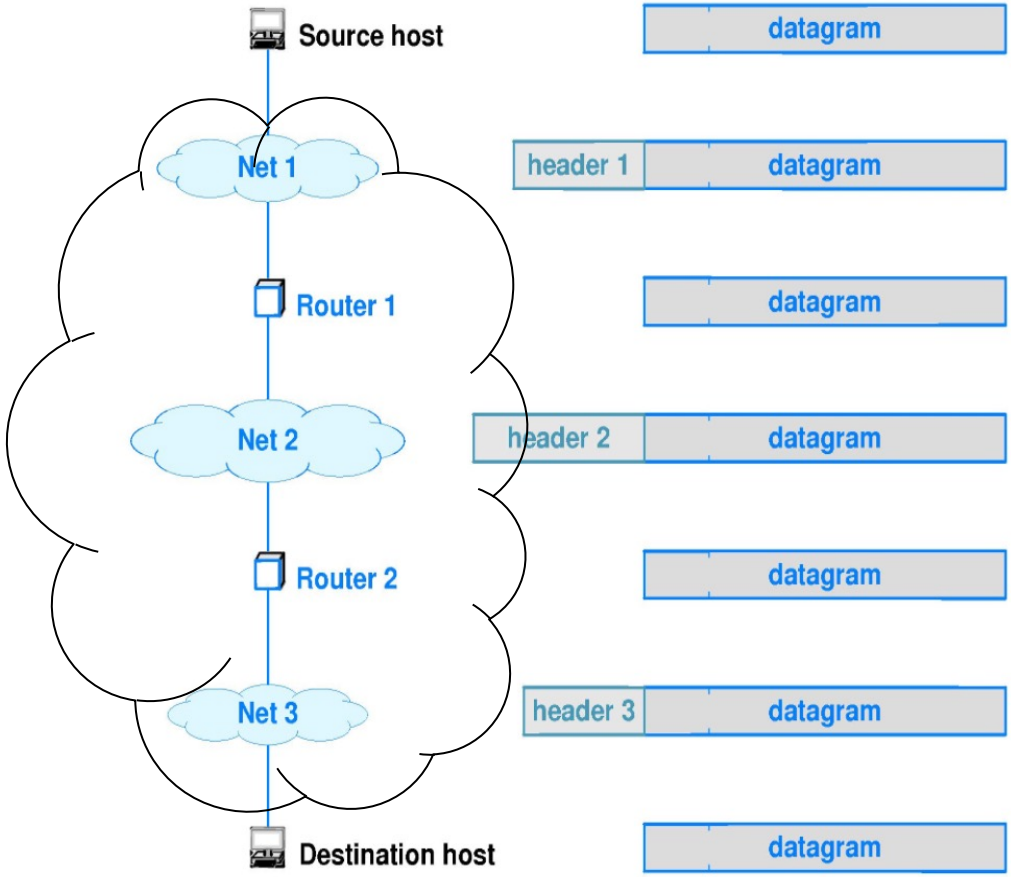
Packet Encapsulation

- Encapsulation and De-capsulation of data during a transfer!



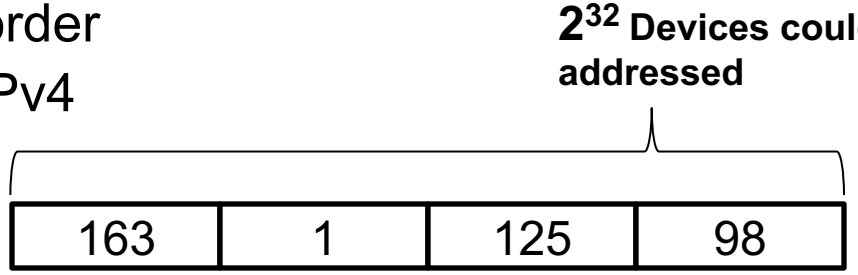
Link Layer (header and trailer information)

- When a datagram is sent across a physical network, the receiver on the other side, removes the header from the encapsulating frame and ***discards*** it.
- If the IP datagram is forwarded along further, the router places (only) the IP datagram into a new frame suitable to the next network it must cross.



Internet Protocol

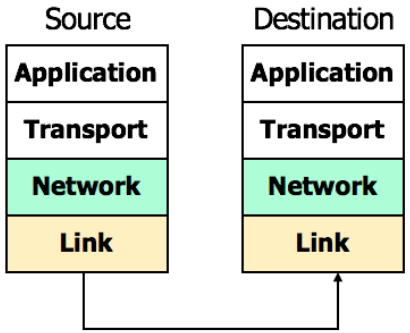
- Responsible for end to end transmission
- Sends data in individual packets
- Maximum size of packet is determined by the networks
  - Fragmented if too large
- Unreliable
  - Packets might be lost, corrupted, duplicated, delivered out of order
- Currently internet widely uses IPv4
  - 4 bytes for address
    - e.g. 163.1.125.98
    - Each device normally gets one (or more)
    - In theory there are about **4 billion (2^{32})** addresses available



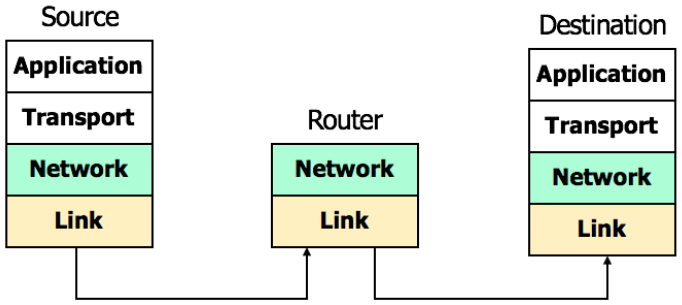
- But the **number of devices in IoT** is much larger.
  - That is why **IPv6** is being introduced. (128 bit addressing scheme).
  - How many unique addressed we could generate using IPv6

Routing

- How does a device know where to send a packet?
  - All devices need to know what IP addresses are on directly attached networks
- If the destination is on a local network
  - send it directly there

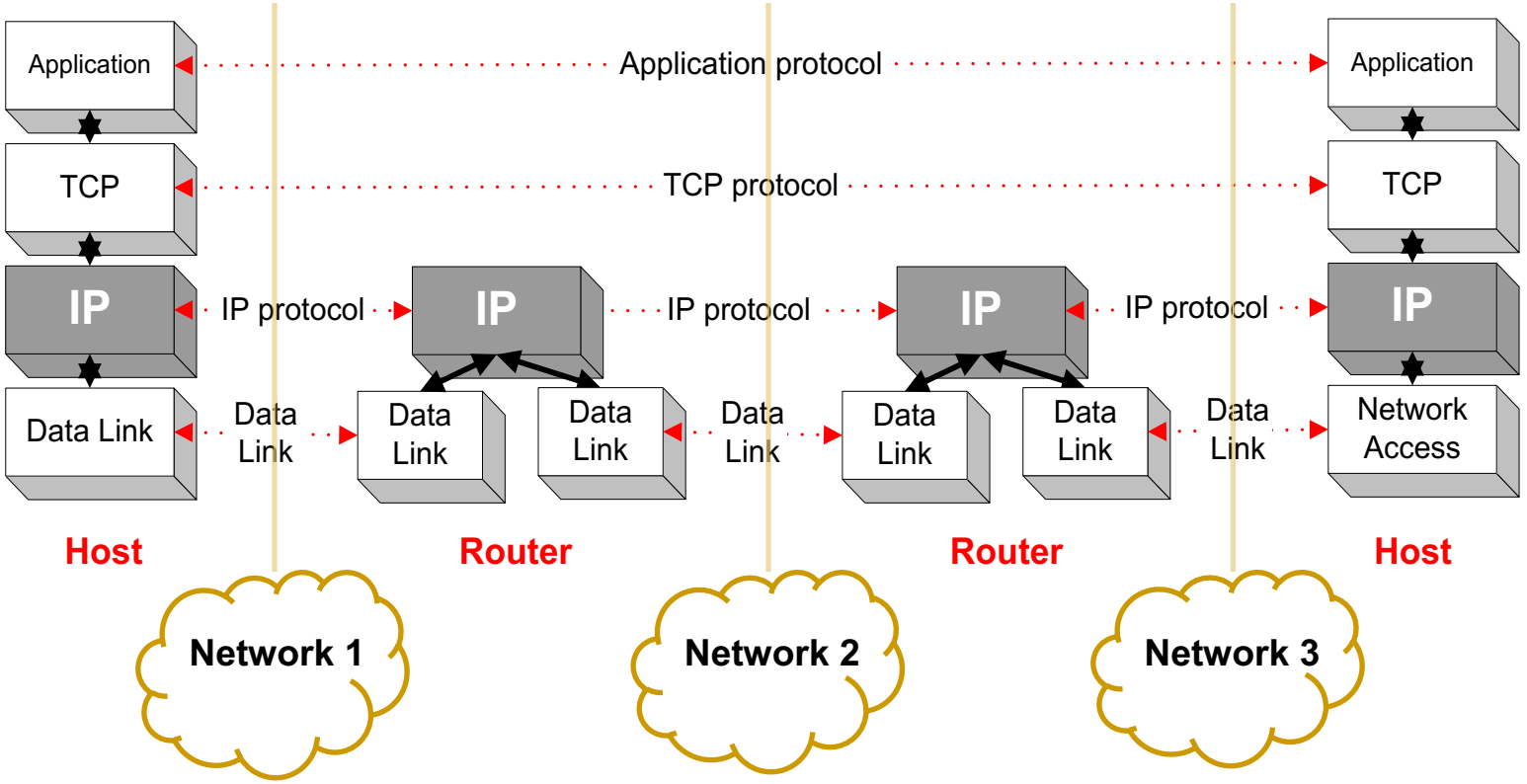


- If the destination address isn't local
  - Most non-router devices just send everything to a single local router
  - Routers need to know which network corresponds to each possible IP address



IP Routing

- IP is the highest layer protocol which is implemented routers.



Who Allocates The IP addresses?

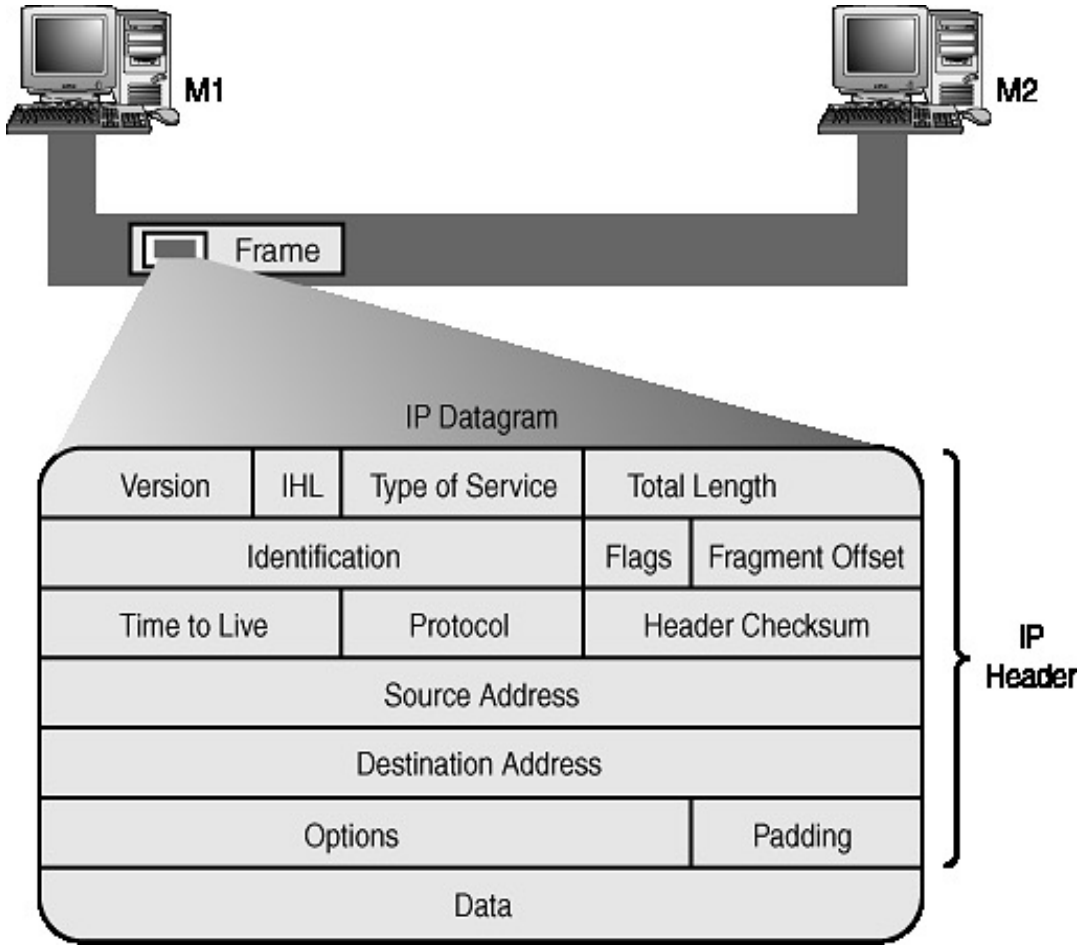
- IP address allocation is controlled centrally by **ICANN** **(Internet Corporation for Assigned Names and Numbers)**

- Fairly strict rules on further delegation to avoid wastage
  - Have to demonstrate actual need for them
  - https://en.wikipedia.org/wiki/ICANN

- Organizations that got in early have bigger allocations than they really need

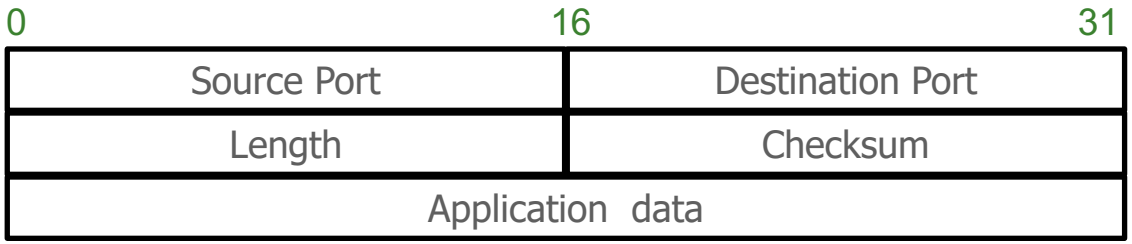
IP Datagram

- The structure of IP datagram is as following:



UDP (User Datagram Protocol)

- A thin layer on top of IP
- Adds packet length + checksum
  - Guard against corrupted packets
- Also source and destination *ports*
  - Ports are used to associate a packet with a specific application at each end
- Still unreliable:
  - Duplication, loss, out-of-order receiving of packets are possible



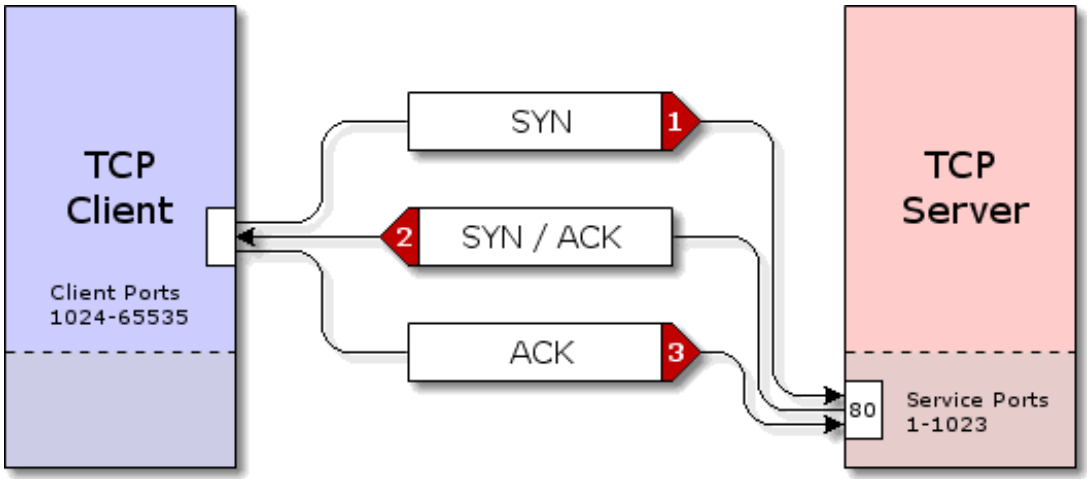
Field	Purpose
Source Port	16-bit port number identifying originating application
Destination Port	16-bit port number identifying destination application
Length	Length of UDP datagram (UDP header + data)
Checksum	Checksum of IP pseudo header, UDP header, and data

TCP (Transmission Control Protocol)

- Reliable alternative of UDP with more overhead
- A tick (in oppose to thin in UDP) layer on top of IP
- Reliable, *connection-oriented*, *stream* delivery
  - Data is guaranteed to arrive, and in the correct order without duplications
    - Or the connection will be dropped
- **Imposes significant overheads**

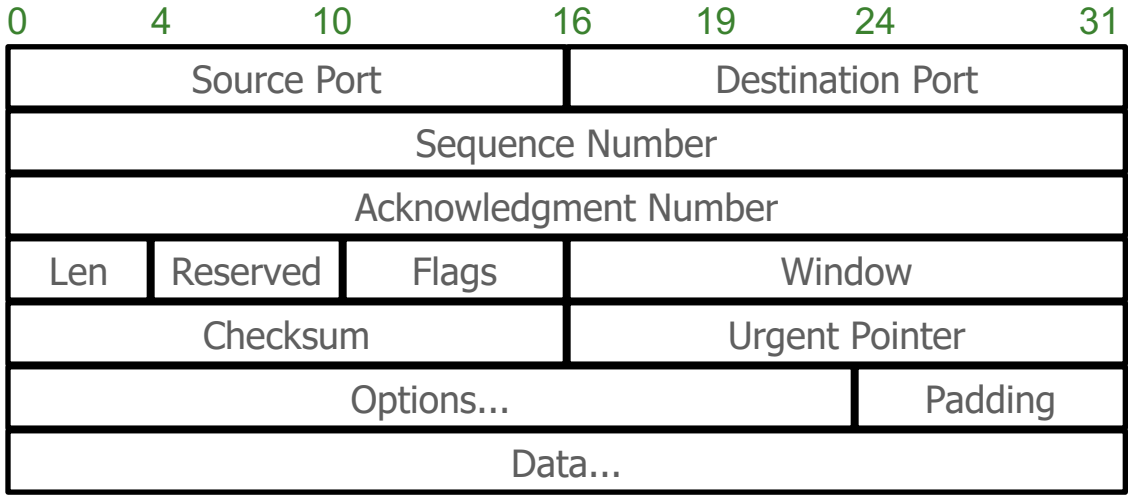
TCP Implementation

- Connections are established using a *three-way handshake*
  - Click here to learn how three-way handshake works?
- Data is divided up into packets by the operating system
- Packets are numbered, and received packets are acknowledged
- Connections are explicitly closed
  - (or may abnormally terminate)



TCP Header (just for your information!)

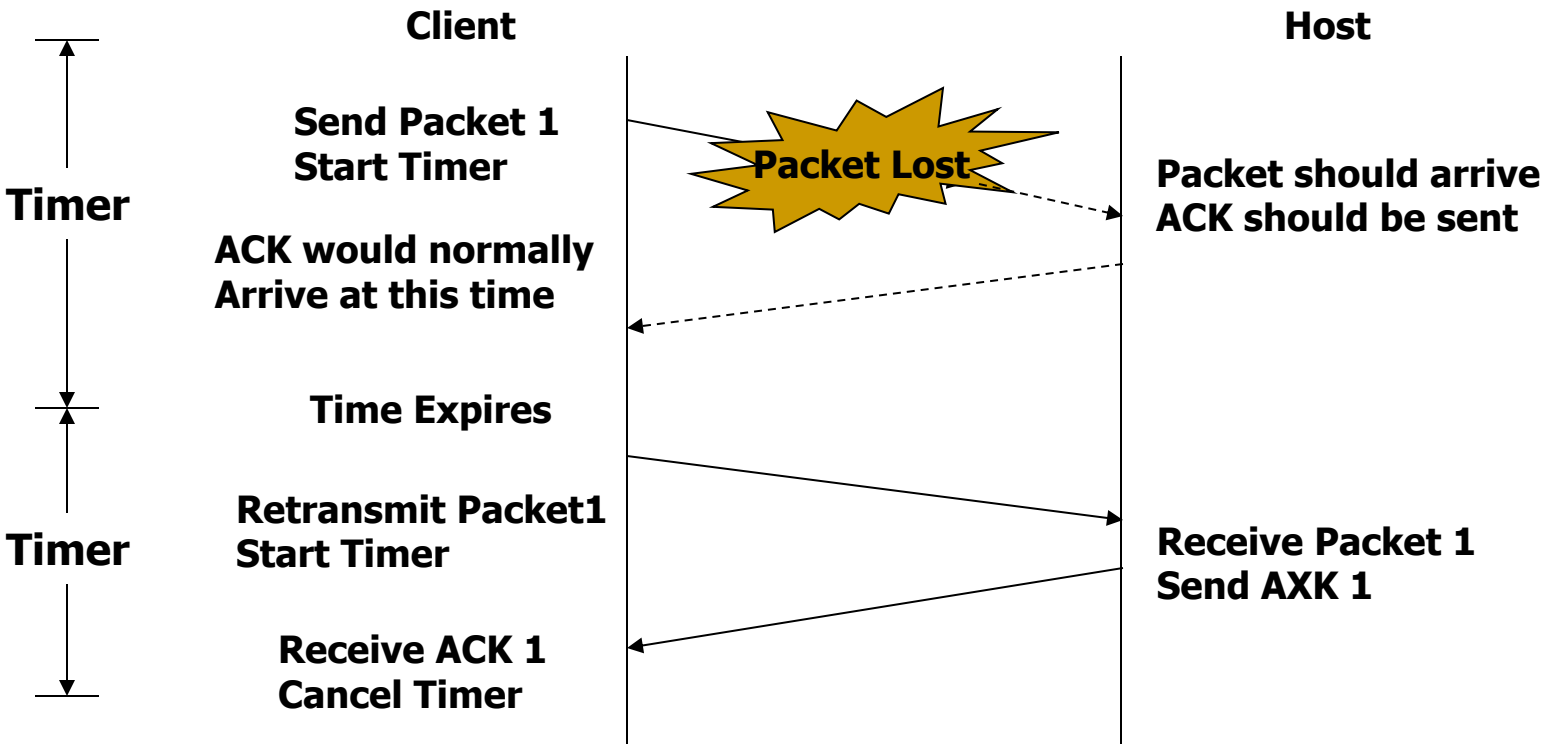
- Compared to UDP, TCP protocol need a larger header size!



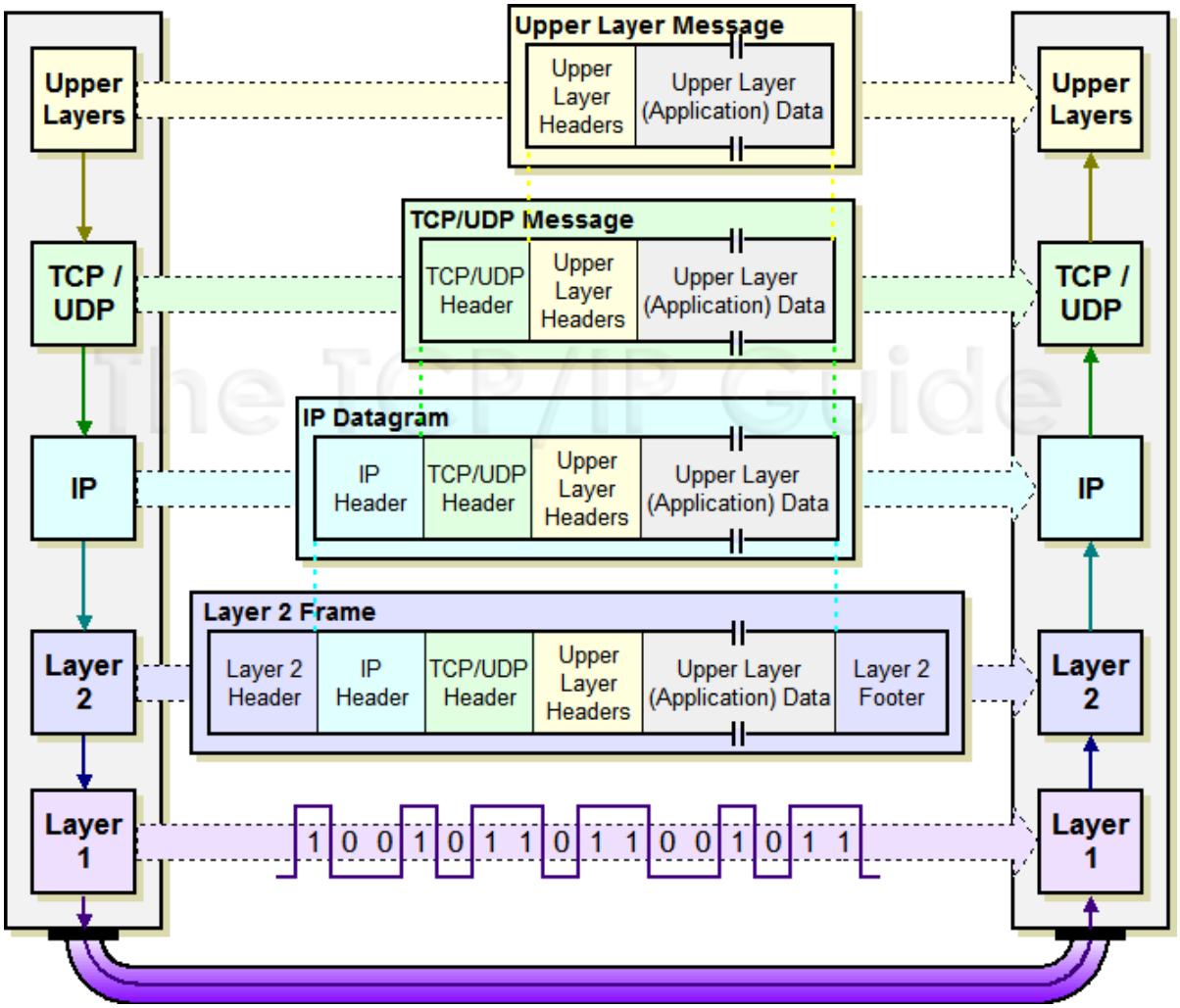
Field	Purpose
Source Port	Identifies originating application
Destination Port	Identifies destination application
Sequence Number	Sequence number of first octet in the segment
Acknowledgment #	Sequence number of the next expected octet (if ACK flag set)
Len	Length of TCP header in 4 octet units
Flags	TCP flags: SYN, FIN, RST, PSH, ACK, URG
Window	Number of octets from ACK that sender will accept
Checksum	Checksum of IP pseudo-header + TCP header + data
Urgent Pointer	Pointer to end of “urgent data”
Options	Special TCP options such as MSS and Window Scale

TCP Data Transfer

- No packet loss in TCP!
  - If acknowledge is not received (within specified time), re-transmit again!



Lets Summarize:



Network Security

- Network security is the study of
  - How bad guys can **attack** networks
  - How to **defend** against such attacks
  - How to **design** systems that are immune to attacks

- Internet was **not** designed with security in mind
  - The protocols and system works best under mutual trust
  - Security is embedded in every layer


The End.
The End.
The End.
The End.