

Communication and Protocols

CSC309
Mark Kazakevich

Before we get to **web programming**...

- We should have a basic idea of how the **internet** works
 - And what the **web** is, but we'll get to that later..
- Nothing too detailed, but enough to give some context to what we'll be doing

Let's say we wanted to build a **network** of computers...

- What do we need at the minimum?
 - Computers
 - A method of communication between those computers

Communicating computers

- Computers have to “talk” to each other
- They need a language (set of rules)
 - Is **one** enough?
 - Not really..
- We need an efficient multi-step system

Network 4-Layer Model

Application Layer

HTTP, FTP, SSH, SMTP, POP3, ...

Transport Layer

TCP, UDP, ...

Internet Layer

IP

Link Layer

Ethernet, Wifi, ...

TCP/IP: A suite of protocols

Application Layer

HTTP, FTP, SSH, SMTP, POP3

Transport Layer

TCP

Internet Layer

IP

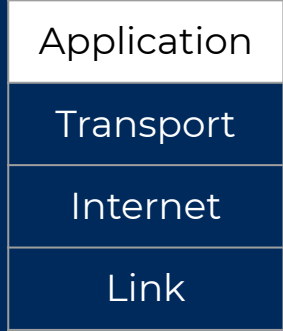
Link Layer

Ethernet, Wifi

Client-Server

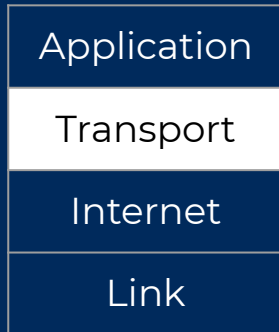
- Client initiates request to server
 - Server accepts or rejects connection
 - If a connection is established, data can flow until connection terminates
-
- TCP/IP facilitates connections between **client** and **server** over many networks
 - The internet!

Application Layer



- Provides applications with standardized **protocols** to exchange data
 - Example: Web browsers need a protocol to get and send data
- Protocols include
 - HTTP, FTP, SSH, SMTP, POP3...

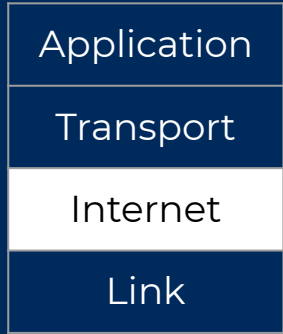
Transport Layer



- Provides **host-to-host** communication services
 - “Connection-oriented”
 - Sends segments of data from the application layer (packets)
- Transport protocol for **TCP/IP** is **TCP**
 - We’ll talk more about it later
- Other transport protocols include **UDP**

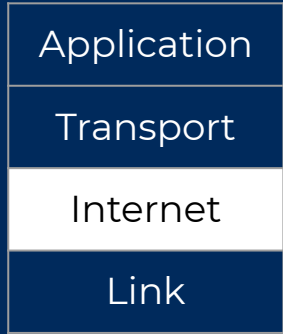
Internet Layer

- Provides protocols for sending **packets** across a **network** or through multiple networks
- The **Internet Protocol (IP)** handles this in **TCP/IP**
 - *Routes* data across networks using IP addresses



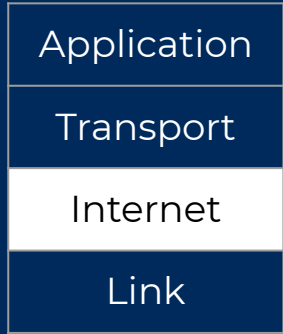
IP protocol

- “**Connection-less**” protocol
 - No prearranged connection required to send data
- **IP** just sends packets over networks
 - no assurance that they will be delivered
 - no way to find out if they were
 - nothing to let the destination know to expect a packet



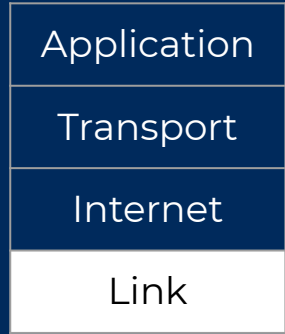
IP packets

- Easy to '**spoof**' packets
 - Connectionless protocol means you can send around packets that pretend they came from a specific IP address
 - Defense against this can come from higher network layers and network monitoring



Link Layer

- Protocols of the **physical link** between the nodes of the network
 - Ethernet, WiFi, DSL
- Lowest level
 - TCP/IP can sit on top of any Link layer



The Internet

Application Layer

HTTP, FTP, SSH, SMTP, POP3

Transport Layer

TCP

Internet Layer

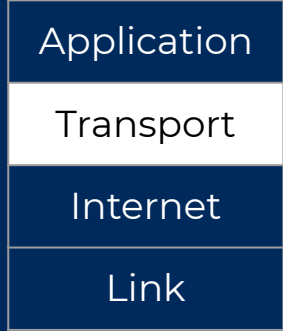
IP

Link Layer

Ethernet, Wifi

More on **TCP**

TCP



- Recall: **Connection-oriented**
 - Needs to have a pre-arranged connection before sending data
 - Should be **bi-directional**
 - Both client and server should **acknowledge** when they get data
- How do we start a connection using TCP?

3-way Handshake



Application
Transport
Internet
Link

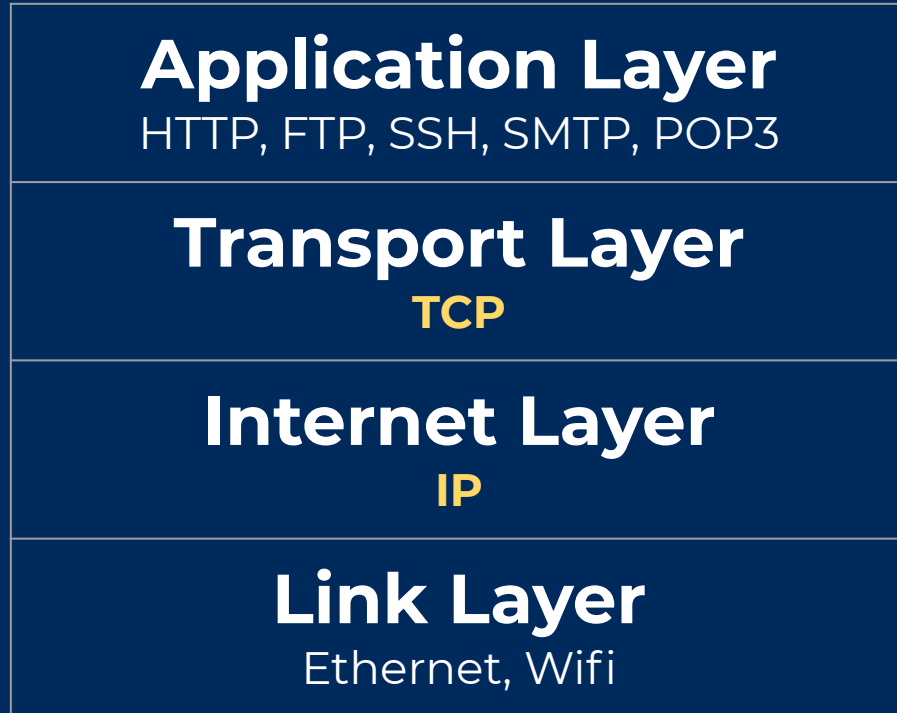
Client and Server can now send each other data, and must **acknowledge** to each other when they receive something

Acknowledgements

Application
Transport
Internet
Link

- An important part of TCP because...
 - Can check packet is from correct host
 - Losing packets is a real problem
- If no acknowledgment that packet was received...
 - Packet sent again
- **TCP** is **reliable**
 - But reacts to losing packets by slowing connection
 - **UDP** is not reliable, but doesn't react to packet loss

Keep this in mind as we start



Next Time...

Application Layer

HTTP, FTP, SSH, SMTP, POP3

Transport Layer

TCP

Internet Layer

IP

Link Layer

Ethernet, Wifi