Lecture Notes: The Internet

What is the network Stack?

- The network stack, often referred to as the networking stack, is a set of software components that handle the communication processes over a network.
- These components work together to manage the different aspects of sending and receiving data between devices on a network, including handling protocols, managing data transmission, and ensuring data integrity and security.

The network stack is usually structured in layers, with each layer responsible for a specific part of the communication process. The most widely recognised model for this structure is the OSI (Open Systems Interconnection) model, which divides the network stack into seven layers:

1. **Physical Layer**: Deals with the physical equipment required for data transmission, including cables, switches, and the electrical signals passed through the medium.
2. **Data Link Layer**: Manages the node-to-node data transfer and error detection and correction in the physical layer. This includes protocols like Ethernet and PPP.
3. **Network Layer**: Handles routing and forwarding of data packets between devices across different networks. IP (Internet Protocol) is a key protocol at this layer.
4. **Transport Layer**: Ensures that data is transferred between points on a network in a reliable and efficient manner. TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are important protocols at this layer.

5. **Session Layer**: Manages sessions between applications, establishing, managing, and terminating connections.
6. **Presentation Layer**: Transforms data into a format that the application layer can accept, handling encryption, compression, and other transformations.
7. **Application Layer**: Provides protocols that applications use to exchange data, such as HTTP for web browsing, FTP for file transfers, and SMTP for email.

The HTTP Network stack:

TLS

TCP

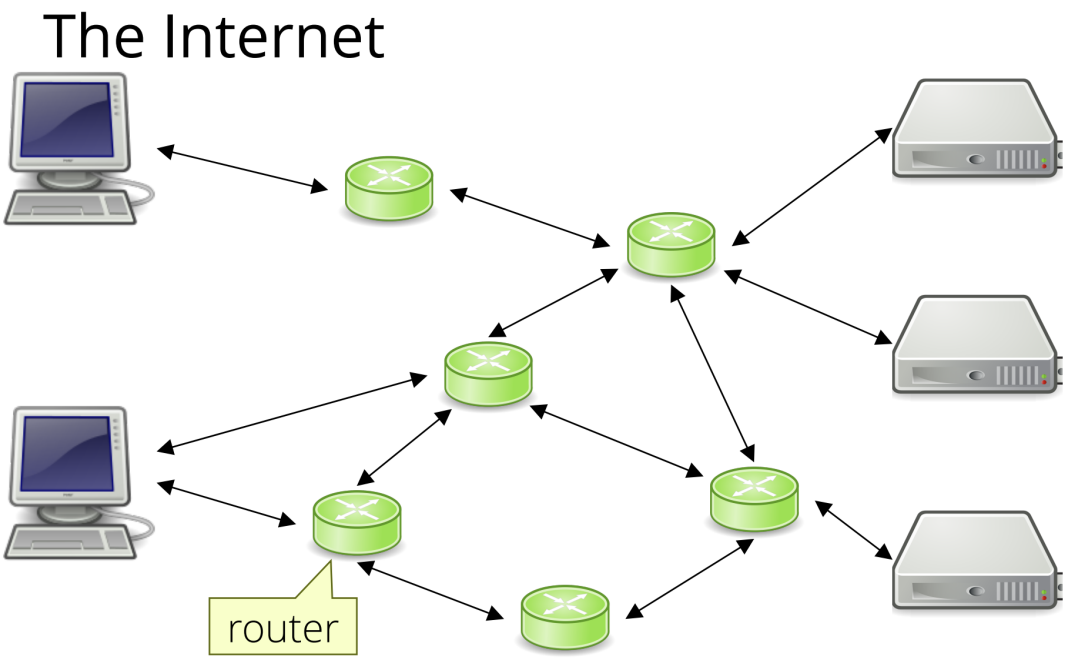
ports

IP

IP addresses

- Each stack relies on the one above, so HTTP relies on TLS, which relies on TCP, which relies on IP

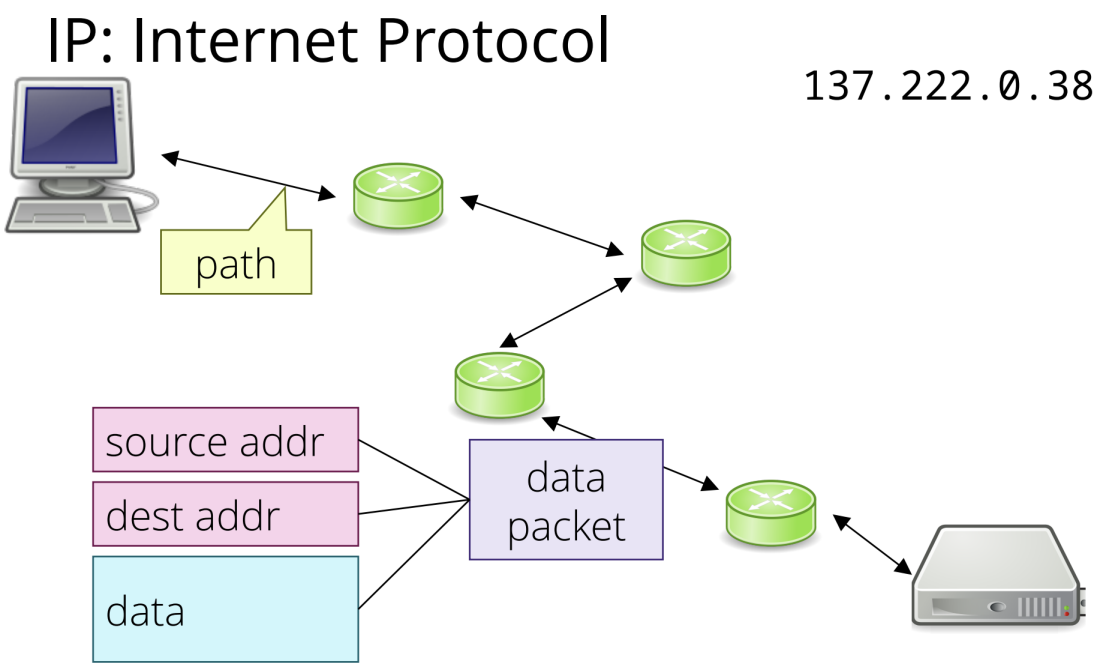
The Internet:



- The internet is a network of routers shown by the green blocks. These routers accept data packets that are marked for a certain address and route those packets onto those address.
- The protocol that is used for this routing is called the **Internet Protocol (IP):**

IP 💻: Internet Protocol

- This protocol describes how data packets should be created and handled



- The IP address are four unique numbers ranging from 0 - 255 that describe how the data packets are handled.
- IP addresses are assigned by a router (e.g. a Wi-FI router will assign all devices it is connected to an IP address)
- The first three octets (the three digit numbers) are the network addresses. These stay the same for a set network and never change hence why most IP addresses start with: 192.168.1
- The last number is the Host number this will vary device to device. So in your house all devices will have the same IP address: 192.168.1.XX the last octete will be different for your laptop, phone, TV etc.
- When you want to access an external IP address such as Netflix, your router (which has its own IP address) will redirect your request to that IP address.
- But this is not the complete picture:
  - We need to make sure that the IP address was successfully received
  - Anything over 65Kbytes cannot be sent over a network
- This is where TCP comes in:

TCP: Transmission Control Protocol

- The Transmission Control Protocol is how data is sent and recieved between computers. There are Five basic functionallity of TCP:

1. **Breaking down the message:**

When a message is over 65Kbytes TCP breaks down the message into smaller chunks, called data packets that are then sent to the recieving end.

2. **Sending Each Part:**

Each packet is sent via a IP. Each packets can take differnt routes (via different routers) to their destination.

3. **Ensuring all parts arrive:**

TCP ensures wheater each packet has arrived at its destination via an acknowledgement sent back from the reciever to the sender. If a packet gets list TCP notices this and resends the missing data or damanged packet

4. **Arrangin the message correctly:**

TCP ensures that all packets arrive in the the correct order so the data is recieved in its original form.

5. **Manging tehe the Flow:**

To ensure that the data is recieved efficianlty, the rate at which the data packets are sent is regualted by the TCP, so the reciever is not overwhelmed.

In essence, TCP is like a meticulous postmaster: it divides your message into smaller parts, sends them over various routes, ensures all parts arrive, resends any lost parts, and ensures the message can be reassembled easily and read in the order you intended. This protocol is vital for ensuring reliable, ordered, and error-free communication over the Internet.

TCP also performs other functions here to make the transmission reliable

- the server can also tell the client to **slow down** if it is recieving messages too fast

TCP Ports

- TCP **does provide** enough to fully implement HTTP on top of it, yes it does!
    - HTTP works perfectly find if we provide it with the tools that TCP creates we can send HTTP requests and responses as TCP messages and be fairly sure the data will arrive int he correct order, and will make sense to the HTTP client and server

    - while there is a HTTP port 80, there is also a port 443 for TLS
    - this is an attempt to provide a **more secure** communication between parties over a network

TLS

- there are three main properties we care about in regards to security:

1. **Confidentiality** - no one else can read your messages
2. **Integrity** - no one else can modify your messages

Under TLS, confidentiality and integrity are provided by using **Symmetric Encryption**

- this is a key shared between the client and the server used for encrypting messages between them
- while other people could see that packets are being sent between the two parties, they would not be able to understand the content of the messages, as it is all encrypted using keys they do not have

3. **Authentication**

NEED NOTES ON MAN IN MIDDLE AND CERTIFICATE AUTHORITY

- TLS is a protocol which is built on top of TCP
- TLS is mostly concerned with how the two parties who want to share a message securely can establish a shared secret key when they are talking over a puiblic network

need rest of slides notes!!!!

HTTP can again operate on top of TLS, so TLS is providing encryption that is fairly transparent to people using the web by seeing this indicator which most browsers include in a navigation part

Padlock = using TLS

Port Forwarding

This is a **network configuration** method which is used to allow external devices to access services on a private network from the outside.

When a device on a local network needs to be accessible over the internet, port forwarding is often used to facilitate the connection.

Port Numbers

Here's a brief explanation:

1. **Port Numbers**: These are part of the addressing information used to identify the senders and receivers of messages. They work like extensions on a telephone system, directing incoming data to specific applications or services on a device.
2. **Client-Server Communication**: When your computer (the client) connects to a server (like a web server), your computer uses a port number to identify itself. This port is often randomly selected from a range of available ports designated for temporary, non-server use (often called ephemeral ports).
3. **Two-way Communication**: Once the connection is established, both the server and the client use each other's IP addresses and port numbers to

exchange data. The server sends data back to your computer by targeting your IP address and the port number your computer chose for that session.

This system allows the server to communicate specifically with your computer on a specific channel, even if there are many clients connected at the same time.

Port Numbers 0 - 1023 are called **System** or **well-known** ports, and are the ones peopple use everyday

- e.g. `80` , `443`

80 for HTTP

443 for HTTPS

Port numbers 1024 - 49151 are **user** or **registered ports**

- ports that can be registered by companies for a particular service

Port numbers `459152` - `65535` are called **Dynamic** or **Private Ports**

- client side ports that are free to use
- the port numbers that your local machine will temporarily assign itself during a session such as viewing a webpage

System and user ports are used **server side**

Dynamic/private ports are **client side**