

## LT1 - Intro

Some stuff and tings

## LT2 - Models, Methods, Metal

In the ideal world we would have

- Infinite bandwidth
- Infinite speed
- Zero latency
- Zero errors

However in reality we have

- Finite bandwidth
- Finite speed
- Errors
- Latency

**Latency** - How long it takes for a bit to travel through the network

In the real world the lower bound for latency is the speed of light so we don't have much to worry about.

### Error Rate

The following factors (and many more) contribute to data loss, corruption etc.

- Cosmic ray background
- Mains noise
- Cable and connector issues

**Voice over IP** for example can take low bandwidth and high errors, but it cannot tolerate high latency. Whereas **OS downloads** require reliability over everything.

Over time **link bandwidth** tends towards infinity, currently on the order of **terabits / s**, so latency is the bottleneck. Some of the contributing factors towards latency include

- Processing times
- Speed of light (lower limit, 1ms round to London, 56ms round to SF)

Raw error rates are roughly constant, we can trade off **processing power and bandwidth** for better **error correction**

It is still very common to fill a jumbo jet with hard drives for fantastic bandwidth and terrible latency. So for transferring a full data center this is great.

### Packet Switching

- Split the data stream into units called “*packets*”
- Give each packet identifying information
- Switch packets over the network until they reach their destination
- Packets can get lost, delayed or re-ordered

### Advantages

- Multiplexing in **time** rather than **frequency** so much simpler
- **Statistical gain** on bandwidth
- Rerouting data around failed switches for **resilience**

### Disadvantages

- Delaying packet till there is room on the line, introduces **latency**

### Frequency Domain Switching

- Distinguishing two signals by their frequency
- e.g. red and blue light, low and high pitched sound

### Time Domain Switching

If we have two data streams each with a bandwidth that is half that of the line, we can buffer the arriving data and when the buffer is full send it off at the line bandwidth. This means the output is idle for 50% of the time.

### Statistical Gain

- Assume most users are not using full line bandwidth 24/7

- Oversell bandwidth, e.g. selling 100 x 10Mbps when line rate is 100Mbps
- Contention ratio, 50:1 -> 50 customers sharing the same bandwidth :(

### Virtual Circuits

- Endpoints tell network to establish a connection
- Network sets up a path to the destination, gives back identifying token
- The token identifies a “*virtual circuit*” linking two endpoints
- Connection is torn down when endpoint has finished with it
- Network has to know about every connection
- Each packet in a connection follows the same route
- Network attempts to solve ordering and packet loss but there are no guarantees

### Datagram Services

- Packet contains addressing information
- Each packet considered separate
- Endpoints deal with loss, duplication and corruption
- Might arrive, might not
- Network doesn't need to know about connections, it just routes packets

### Layering

Applications generally want guarantees, knowing that if a file is sent it will arrive undamaged or you know if something went wrong.

A network can be thought of as a **stack**, a succession of interfaces starting with services needed by real applications and getting closer to transistors and volts. Each layer provides services to those above it and makes use of services below it. Each task appears in only one layer.

Table 1: *Abstraction in the networking stack*

Layer n+1	More abstract service, closer to application
Layer n	Less abstract service, closer to wire

### Competing Models

### OSI Model

The **OSI** model basically failed, it came long before any successful implementation.

### DoD Model

The **DoD Model** was a rationalisation of already established practice, it was made partly to compare the TCP/IP architecture with the OSI proposals.

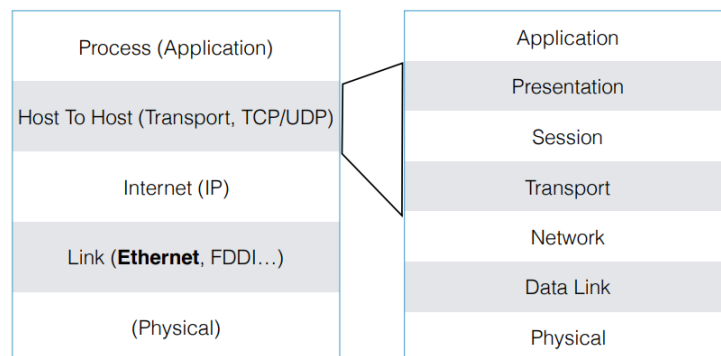


Figure 1: DoD vs OSI Model

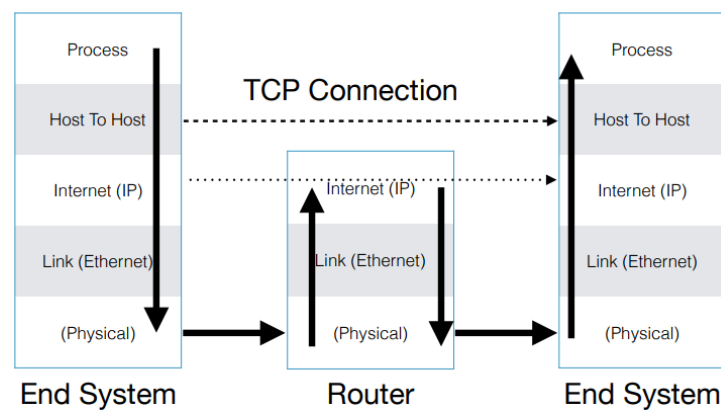


Figure 2: The DoD Model

The **Application Layer** runs code that does useful work e.g. SMTP, IMAP, HTTP, SSH, they need services to move data from computer to computer

The **Transport Layer** moves data between two end systems via a network with topology and properties that the transport layer doesn't know about. **TCP** is used for streams of data, **UDP** for packets. Properties of **reliability**, **sequenced delivery** etc. are guaranteed. The layer also permits communication between entities running on the same end systems.

The **Internet / Subnet / Network Layer** moves packets between end systems, doesn't offer any guarantees about reliability. Handles choosing which link to use to get data closer to destination. It also doesn't deal with the concept of multiple applications. e.g. IPv4, IPv6

The **Link Layer** moves data between one network element and the next. Concerned with packetisation, addressing etc. Many protocols are in use but **Ethernet** is our main focus.

The **Physical Layer** consists of Ethernet over co-axial, twister pair, fibre, radio etc.

### TCP/IP Wins

- Single transport service for both connections and datagrams
- Full responsibility of the implementor to make TCP and UDP work
- Exactly one network layer – **IP**
  - **IPv4** and **IPv6** differ only in address length
- **Physical** and **Link** layers carry **IP**, if they can't then they can't carry **TCP** or **UDP**

OSI lost because it tried to provide multiple transport services, there were also two different network layers for virtual circuit vs. datagram, and none of it interworked.

### Value Chain

- Telcos sell a service with a service level agreement
- ISPs are more “best efforts” and rely on statistical gain

Applications you can make money out of: Facebook (ads), NetFlix (subs), Online Banking (drives other business)	
Sale of Internet Connections to consumers and businesses, with large amounts of statistical gain and vague SLAs	Consumer ISPs
Carrying the data belonging to ISPs over a distance: they want an SLA, and only rent bandwidth they need <u>after</u> statistical gain	Telcos and Interconnect
Carrying data on single-haul links between premises, where you can't use any statistical gain as it's about wires and linecards.	Telcos

Figure 3: *The Value Chain*