The Internet (pt. 1)

CSE 486/586: Distributed Systems

Ethan Blanton

Department of Computer Science and Engineering

University at Buffalo

The Internet

The Internet is not a monolithic entity, or even a protocol.

It is a collection of:

- networks
- protocols
- organizations
- standards
- …

Understanding how and why it came about will help us understand it.

This isn't a networking course, just some background.

Distributed Systems

The Internet is where we build our distributed systems today.

In many ways, the Internet is a distributed system!

It “solves” many of the problems we will explore.

…yet we have to solve some of them again.

We will explore the reasons why this is so.

Early History

The first global scale network was the phone system.

In the US, it was run by AT&T in a regulated monopoly.

The telephone network was circuit switched:

- Phones are passive devices
- The network establishes a connection between phones
- That connection was a literal wire from end to end

Computer networks rode on the phone system.

Packet Switching

In 1961, Leonard Kleinrock introduced packet switching.

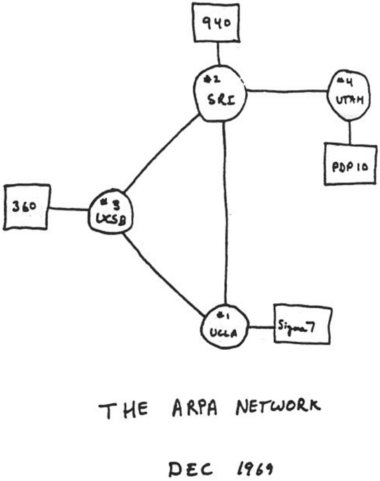
With packet switching, there is no “pair of wires”.

“Packets” of data are carried independently through the network.

In 1966, DARPA started the ARPANET project.

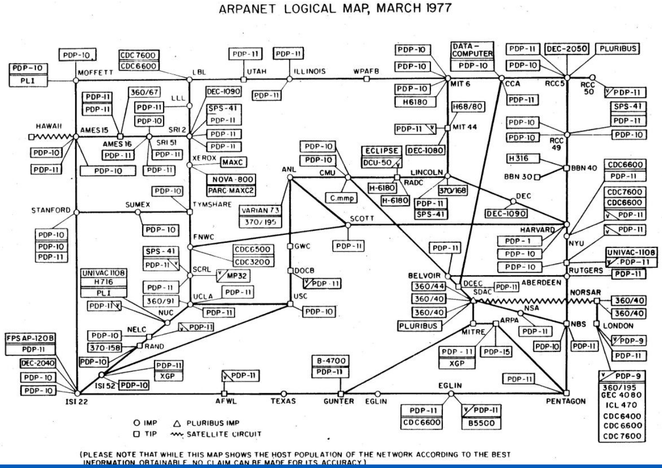
The ARPANET connected its first hosts in 1969.

The 1969 ARPANET

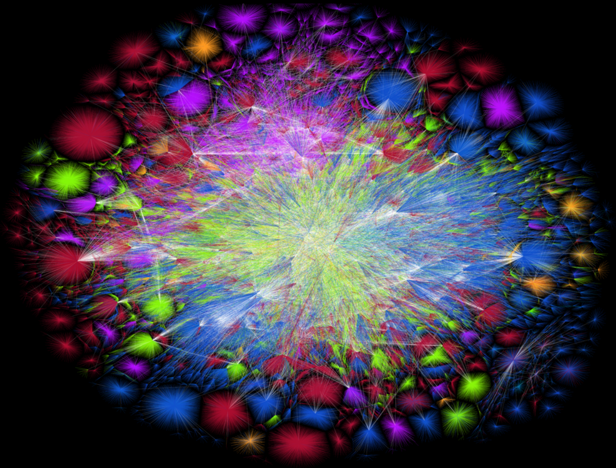


Growth and Expansion: 1977

The ARPANET has always grown exponentially.



2015



A Network of Networks

The top level goal for the DARPA Internet Architecture was to develop an effective technique for multiplexed utilization of existing interconnected networks. [1]

The Internet is fundamentally a network of networks.

Those networks may have varying capabilities.

This had a large influence on Internet design.

The Stateless Internet

The Internet is packet switched so that it can be stateless[1].

Endpoints maintain the state of connections.

The network itself sees only packets.

A protocol — the Internet Protocol [2] — carries those packets.

[1]Not entirely, but it doesn't (or *shouldn't*) keep connection state

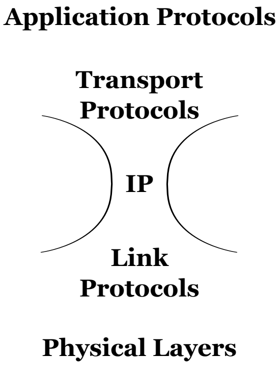
The Narrow Waist

We sometimes call the Internet Protocol (IP) the narrow waist of the Internet.

This is because there are:

- many link layer protocols below it
- many transport protocols above it

…but it stands alone.



Limited Guarantees

IP's place in the network stack is on top of other networks.

It makes very few assumptions about those networks.

This is one of the reasons the Internet has dominated networking.

It is also one of the reasons the Internet is packet switched.

IP, in turn makes very few guarantees.

IP Datagrams

IP datagrams are small, self-contained packets containing:

- A source host
- A destination host
- Minimal other metadata
- Uninterpreted data

IP handles (almost) only moving those datagrams.

It does not provide:

- Protection from corruption
- Reliable transmission
- Connections of any kind

Best Effort

The Internet Protocol provides best effort delivery.

This means that datagrams are delivered if possible.

They are dropped if they cannot be delivered.

They may also be queued indefinitely awaiting delivery.

This means that individual IP datagrams may arrive:

- Late
- Out of order
- Not at all

Routing

IP datagrams are routed individually from network to network.

Routing decisions are made locally based on limited information.

Routing protocols provide some notion of global topology.

Due to local routing decisions, datagrams may experience:

- Routing loops
- Asymmetric connectivity
- Destinations that are locally unreachable

The Transport Layer

The transport layer provides additional services.

Transport protocols are data in IP datagrams.

The two most common Internet transport protocols are:

- UDP for unreliable, low-latency communication
- TCP for reliable, congestion-aware communication

Both provide:

- Multiple endpoints per host
- Protection from corruption (optional for UDP)

UDP

The User Datagram Protocol [4] provides little more than IP:

- Multiple endpoints (ports) per host
- A simple checksum over data

UDP datagrams also provide only best effort delivery.

UDP is often used for:

- Local communication
- Tasks requiring low latency
- Fixed-throughput communication
- Applications that can tolerate lost data

Advantages of UDP

Because UDP provides few extra services, it has little overhead.

Datagrams do not require connection establishment.

Best effort delivery is low latency.

The UDP data checksum provides some corruption protection.

Disadvantages of UDP

Best effort delivery permits loss.

UDP provides no recovery for lost datagrams.

Datagrams carry no connection or session information.

UDP provides no feedback on network conditions.

TCP

The Transmission Control Protocol [3] is more complicated.

It provides significant additional services compared to IP or UDP:

- A stream of bytes rather than individual datagrams
- Recovery for data lost by best effort delivery
- Detection and avoidance of network congestion

It also provides a simple data checksum, like UDP.

Advantages of TCP

TCP provides recovery from lost data via retransmission.

Congestion control allows effective sharing of network resources.

TCP connections provide:

- Convenient byte-oriented streaming semantics
- Persistent communication channels between endpoints

Disadvantages of TCP

TCP connection management adds significant overhead.

Loss recovery can increase latency substantially.

TCP requires bidirectional communication.

Summary

- The Internet is a network of networks
- IP will run over many networks because it makes few assumptions
- IP provides very limited service
- Transport protocols ride on top of IP
- UDP is connectionless datagrams
- TCP is connected byte streams

Next Time …

- TCP loss recovery and congestion control

References I

Required Readings

[1] David D. Clark. "The Design Philosophy of the DARPA Internet Protocols". In: *Computer Communication Review* 18.4 (Aug. 1988), pp. 106–114. URL: http://ccr.sigcomm.org/archive/1995/jan95/ccr-9501-clark.pdf.

Optional Readings

[2] Information Sciences Institute. *The Internet Protocol*. Ed. by Jon Postel. Sept. 1981. URL: https://tools.ietf.org/rfc/rfc791.txt.

References II

[3] Information Sciences Institute. *Transmission Control Protocol*. Ed. by Jon Postel. Sept. 1981. URL: https://tools.ietf.org/rfc/rfc793.txt.

[4] Jon Postel. *User Datagram Protocol*. Aug. 1980. URL: https://tools.ietf.org/rfc/rfc768.txt.