Understanding the Internet

Allen B. Downey

Olin College of Engineering

Needham MA
An engineering approach

The Internet is an engineered artifact.

- Designed under constraints for a purpose.

Therefore to understand the Internet:

- Understand constraints.
- Understand purposes.
- Understand solutions (and create new ones)!
A scientific approach

The Internet is a complex system.

- Understanding components and their interactions is not enough.

Understanding requires:

- Appropriate abstraction.
- Observation of emergent properties.
Traffic jams

Example: automobile traffic is a complex system.

What is a traffic jam?

- Made of cars, but not always the same cars.
- It’s an abstract entity, like a storm front.
- It has properties that its components don’t have.

Traffic jams move **backwards**.
Outline

Exercise these viewpoints:

- The bottom-up Internet.
- The emergent Internet.
Starting with a medium that transmits an analog signal...

...how can we transmit a digital signal?

Solution: modulation.

- Vary amplitude, frequency, etc.
- Distinguish (at least) two levels: LO and HI.
Physical layer + mo/dem = Signal layer

Abstractly, a modem transmits a digital signal. Physically, there’s no such thing! HI and LO are abstract symbols with many possible physical representations.
Transmitting bits

- A bit can have two states, 0 or 1.
- A digital signal has two states, LO and HI.
- Naïve encoding: LO represents 0, HI represents 1.

Problem?
Transmitting bits

- A bit can have two states, 0 or 1.
- A digital signal has two states, LO and HI.
- Naive encoding: LO represents 0, HI represents 1.

Problem?

1. Clock recovery.
2. Threshold discovery.
Manchester encoding

LO-HI represents 0.
HI-LO represents 1.

- Lots of transitions; good for clock recovery.
- Equal number of LO and HI; good for threshold discovery.

Figure from the Wikipedia.
Abstractly, an **adapter** transmits bits.

Physically, there’s no such thing!

0 and 1 are abstract symbols with many possible physical representations.
Error detection

How do you know the bits you got are right?

- Group bits into frames.
- Attach redundant information to each frame.

How do you identify the beginning/end?
A frame consists of:

| 01111110 | data   | checksum | 01111110 |

- Special sequence marks **beginning** and **end**.
- After the data, send a checksum.
- Checksum is a function of the data bits.

Problem?
Bit stuffing

What if 01111110 appears in the data?

Sender:
- Whenever you send 011111, stuff in a 0.

Receiver:
- Whenever you see 011111, check the next bit:
  - If 0, remove it.
  - If 1, check the next bit:
    - If 0, that’s the end of frame.
    - If 1, there must have been an error!
Link layer

Bit layer + framing + error detection = **Link layer**.

<table>
<thead>
<tr>
<th>NIC</th>
<th>Frames</th>
<th>NIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter</td>
<td>Bits</td>
<td>Adapter</td>
</tr>
<tr>
<td>MO</td>
<td>Digital signal</td>
<td>DEM</td>
</tr>
<tr>
<td></td>
<td>Analog signal</td>
<td></td>
</tr>
</tbody>
</table>

Abstractly, a **NIC transmits frames**.
Physically, there’s no such thing!
The Internet Protocol

Links are point-to-point.

How do you talk to someone you’re not connected to?

- Assign IP addresses (128.197.10.145).
- Relay packets through routers.
What’s a packet?

Abstractly, Host A transmits a packet to Host B.

Physically, the packet’s data are copied from link to link.
A typical path

0 rocky.olin.edu (10.8.11.90)
1 10.8.11.2 (10.8.11.2)
2 172.16.1.105 (172.16.1.105)
3 172.16.0.25 (172.16.0.25)
4 172.16.26.194 (172.16.26.194)
5 asf-fw1.olin.edu (172.16.26.98)
6 olin-7204-rtr.olin.edu (172.16.26.65)
7 vl604.aggr2.sbo.ma.rcn.net (216.164.74.165)
8 ge6-1.border1.sbo.ma.rcn.net (207.172.15.148)
9 nox230gw1-Gi-3-12-NoX-RCN.nox.org (192.5.89.29)
10 nox1sumgw1-VI-801-NoX.nox.org (192.5.89.33)
11 nox1sumgw1-peer-nox-umaine-207-210-143-186.nox.org
12 GW-Portland-int.unet.maine.edu (130.111.2.33)
13 130.111.138.10 (130.111.138.10)
14 gw-coa.unet.maine.edu (130.111.3.194)
15 coa-core.coa.edu (209.222.213.242)
16 hornacek.coa.edu (199.33.141.185)
Core providers completely connected.

Edge structure is mostly hierarchical.
Fortunately, IP hides the details.

Abstractly, the network is **completely connected**.
# The Network Layer

Link layer + addressing + routing = Network Layer

<table>
<thead>
<tr>
<th>Packets</th>
<th>NIC</th>
<th>Frames</th>
<th>Adapter</th>
<th>Bits</th>
<th>Digital signal</th>
<th>Analog signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>NIC</td>
<td>Host</td>
<td>Host</td>
<td>Frames</td>
<td>Adapter</td>
<td>Analog signal</td>
</tr>
<tr>
<td>NIC</td>
<td>Adapter</td>
<td>DEM</td>
<td>MO</td>
<td>10100111000111</td>
<td>Digital signal</td>
<td>Analog signal</td>
</tr>
<tr>
<td>Adapter</td>
<td>DEM</td>
<td></td>
<td>MO</td>
<td>10100111000111</td>
<td>Digital signal</td>
<td>Analog signal</td>
</tr>
<tr>
<td>DEM</td>
<td>MO</td>
<td>10100111000111</td>
<td>Digital signal</td>
<td>Analog signal</td>
<td>Analog signal</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The diagram shows the flow of packets, frames, bits, and signals through the network layer, from the host to the network interface controller (NIC) and back, including digital and analog signals.
Best effort delivery

Absolutely, positively might get there eventually.

Who could ask for anything more?

- Exactly-once delivery.
- In-order delivery.

Can you build reliable communication on an unreliable medium?
Stop and wait

Sender

Receiver

- Transmit.
- Acknowledge.
- Timeout... retransmit.
- Problem?

timeout interval
Sliding window

- Keep a copy of unACKed packets.
- Get an ACK, send a new packet.
- Miss an ACK, retransmit.
- Problem?
- Start with one packet.
- Get an ACK, send two packets.
- Watch for a congestion signal.
TCP

Transmission Control Protocol

- Reliable delivery (ACK-retransmit).
- Flow control (sliding window).
- Congestion control (slow start).
- Byte stream abstraction.
- Process-to-process communication.
The Transport Layer

Network layer + sliding window + slow start = Transport layer

Process stream of bytes Process

Host Packets Host

NIC Frames NIC

Adapter Bits Adapter

MO Digital signal DEM

Analog signal
Who uses TCP?

90% of Internet traffic is based on TCP:

- TCP + HTTP = The Web
- TCP + SMTP = email
- TCP + BitTorrent = peer-to-peer network
A little HTTP:

GET index.html HTTP/1.1

HTTP/1.1 200 OK
Server: Apache-2.0.44
Content-Length: 12481
Content-Type: text/html

<html>

... 

</html>
The Application Layer

Transport layer + application protocol = Application layer

<table>
<thead>
<tr>
<th>Application</th>
<th>Client</th>
<th>GET index.html HTTP/1.1</th>
<th>Server</th>
<th>HTTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport</td>
<td>Process</td>
<td>stream of bytes</td>
<td>Process</td>
<td>TCP</td>
</tr>
<tr>
<td>Network</td>
<td>Host</td>
<td>Packets</td>
<td>Host</td>
<td>IP</td>
</tr>
<tr>
<td>Link</td>
<td>NIC</td>
<td>Frames</td>
<td>NIC</td>
<td>Ethernet</td>
</tr>
<tr>
<td>Bit</td>
<td>Adapter</td>
<td>Bits 1010011000111</td>
<td>Adapter</td>
<td>Manchester encoding</td>
</tr>
<tr>
<td>Signal</td>
<td>MO</td>
<td>Digital signal</td>
<td>DEM</td>
<td>Amplitude modulation</td>
</tr>
<tr>
<td>Physical</td>
<td>θ ⇒</td>
<td>Analog signal</td>
<td>⇒ ⊗</td>
<td>Visible light</td>
</tr>
</tbody>
</table>
So now you know...

...given a medium that transmits an analog signal...

...how to build an Internet-compatible host.

But how do you build the Internet?
You don’t. It grows...
It grows, and grows...

Internet structure depends on:

- Technological details.
- Geographic details.
- Demographic details.
- Economic factors.
- Human factors.
It grows in mysterious ways

- Long-tailed distribution of connections (many small, some very large).
- Characteristic of rich-get-richer growth.


$$\text{Number of connections} \sim \exp(6.34763) \times x^{-0.811392}$$
Zipf’s law

Zipf rank relationships in:

- Popularity of words (Zipf 1932).
- Sizes of cities (Zipf 1939).
- Individual income (Pareto 1906).
- Magnitude of earthquakes.
- Debt of bankrupt companies.

...and on and on.
Zipf’s law

And in the Internet...

- Degree of connectivity.
- Popularity of web pages.
- Size of web pages.
- Network transfer times.

...and on and on.
Self-similarity

- Real world: visually similar over a range of spatial scales.
- Fractals: geometrically similar over all spatial scales.
- Time-series: statistically similar over a range of time scales.
Self-similarity

- Normally, large scales are more predictable than small.
- Networks are bursty even at large time scales.

So what?

- Surprising for an engineered system.
- Contrary to engineering assumptions.
Explaining Self-similarity

- File systems evolve toward lognormal size distributions.
- Size distributions induce self-similar traffic patterns.
Contrast:

- “Why does Ethernet use Manchester encoding?”
- “Why is Internet traffic self-similar?”
Two approaches:

- Engineering: understanding the Internet as an artifact.
- Science: understanding the Internet as a complex system.
Questions

- **Email**: downey@allendowney.com
- **Research**: http://allendowney.com
- **Textbooks**: http://greenteapress.com