Intro to Networking

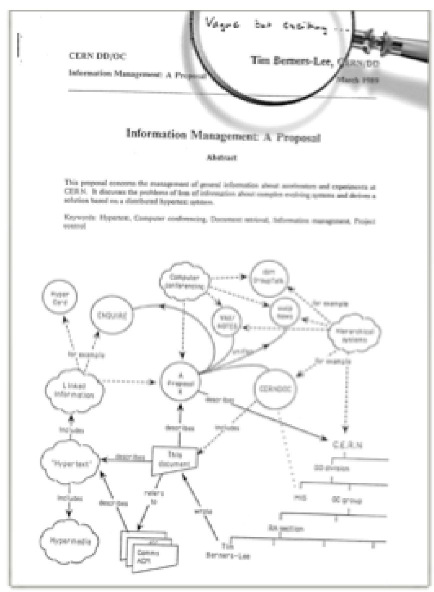
6.033 Spring 2015, Lecture #9
March 4, 2015
Hari Balakrishnan <hari at mit dot edu>

Why learn about networks in 6.033? There are two reasons:

1. Many computer systems use the network.
2. The Internet is an interesting example of a successful system.

The Internet is successful in many ways, but one of the best examples of this success is to consider the following applications:

1. The World Wide Web. The picture below shows the first web server, along with Tim Berners-Lee's "Information Management: A Proposal", from 1989. Note that it was marked "Vague… but exciting"! This was an application written initially by just one person (with some design collaboration with another).
2. Google web search
3. Peer-to-peer file sharing: Napster, Limewire/Gnutella, Bittorrent
4. YouTube
5. Skype
6. Twitter (or "twittr" as it was known initially)
7. Dropbox

How does the internet allow these successes to be possible?

The National Academy of Engineering rated the Internet as "one of the 20th century's greatest engineering achievements", alongside electrification, the automobile, and airplane, and so on.

In fact, it is probably the achievement that will have an even greater impact in the 21st century!
What is the Internet & why is it successful? There are three distinctive aspects (among many) that are noteworthy:

1. It is an "internetwork", i.e., a **network of networks**. It is heterogeneous:
    a. Many types of links: wired fiber, Ethernet, (terrestrial) radio, satellite, etc.
    b. Administered by many different organizations. Importance of "routing policy", disputes and conflicts between competing organizations that much also cooperate to provide connectivity. Net neutrality - who gets to decide what gets done with your packets

    How is internetworking possible with so much heterogeneity?

    The Answer according to the Internet: **Universal interconnection** via the Internet Protocol ("IP on everything"). Described in Cerf and Kahn's 1974 paper.

2. Placement of functions - where do these functions actually happen in the network? Such as:
    a. Routing
    b. Reliability
    c. Sharing / multiplexing

    The Answer (design principle) used in the Internet: **Smart ends and a simple(r) network**.
    A "best-effort" network, which allows it to scale. Avoid unnecessary mechanism in the network, move them to the ends.

    This idea is implemented using **layering**. Applications don't have to worry about the underlying protocols, the protocols don't need to worry about what applications they're handling.

    Each layer is a module, and interfaces with with the layer above, below, and to the "peer protocol" running on the other side.

    Layering allowed the internet to evolve, and also influenced the industry.
    Companies generally align with layers. For example, Qualcomm develops wireless physical and link-layer technologies, Ericsson builds wireless base stations, and Verizon Wireless (a cellular wireless carrier) operates the network; Broadcom builds chipsets, Cisco builds routers and switches, Level3 (an Internet Service Provider); Atheros (now part of Qualcomm) builds Wi-Fi chipsets, various other companies build Wi-Fi access points, and organizations like MIT run Wi-Fi networks. The network endpoint transport protocols are implemented by the OS vendors (Linux, Mac OS,

Windows, etc.), and anyone can write and deploy apps (as noted above in the beginning of lecture).

Originally, however, there was no distinction between TCP and IP in Cerf & Kahn's "A protocol for Packet Network Intercommunication" from 1974. That came later.

The Idea of moving intelligence to the endpoints, keeping the network simple, and layering is described in Saltzer/Reed/Clark's "End-To-End arguments in System Design", which is in your recitation readings.

3. Scalability: how does the Internet scale to billions of hosts?
    a. Really low barrier to entry
        - Only requirement to implement IP is to support packets of a certain small size. No requirements on latency or speed, and it's simple to implement IP.
    b. Topological addressing
        - Can't just put entries for all destination hosts in every router's routing table. (Q: why not?)
        - Every IP address can be thought of as having a prefix and postfix, and routers just have to hold the prefixes: this form of **aggregation** helps with scalability. (Today's routers in the "core" of the Internet have about 550,000 prefix entries, connecting about 3 billion hosts.)
        - An IP address signifies **location in the Internet's topology".** A host does not have an IP address; a network interface connected to the Internet does. For example, right now my phone's Wi-Fi interface has an IP address from MIT's network and my phone's cellular interface has an IP address from AT&T's network

The Internet architecture does have some weaknesses:

- Universal mobility? Can't continue skype call from wifi to cellular: the network provides no mechanisms for that, and these days endpoints must implement any required mechanisms for this task.
- Accountability? Very easy to spoof IP addresses, leads to DDOS attacks
- Anonymity? Difficult to be anonymous -- so we get neither accountability nor anonymity!
- Disputes? Doesn't provide a "design space" to handle disputes, like Net-neutrality conflicts.
- IP layer "ossification": Many of these issues (and others not named) require changes to IP itself, but that is the hardest thing to change. It's narrow, it's fundamental to the Internet architecture, and it is hugely successful, but all that means that it can't easily be changed. It has become "ossified".

Many of these are active research issues being worked on in the networking research community.