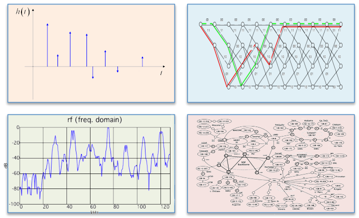


INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

Course Eval!

6.02 Fall 2012
Lecture #23

A Brief History of the Internet

Several pictures taken from Wikipedia

The Dawn of Packet Switching

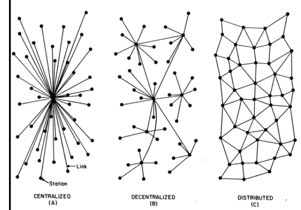


FIG. 1 – Centralized, Decentralized and Distributed Networks

ARPA: 1957, in response to Sputnik

Paul Baran (RAND Corp)

- Early 1960s: New approaches for survivable comms systems; "hot potato routing" and decentralized architecture, paper on packet switching over *digital* links

Donald Davies (UK), early 1960s

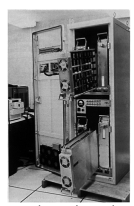
- Coins the term "packet"

Len Kleinrock (MIT thesis): "Information flow in large communication nets", 1961

J. Licklider & W. Clark (MIT), *On-line Man Computer Communication* (1962) & Licklider's vision of a "galactic network"

L. Roberts (MIT then ARPA), first ARPANET plan for time-sharing remote computers

ARPANET



BBN team that implemented the *interface message processor* (IMP)

- 1967: Connect computers at key research sites across the US using telephone lines
- Interface Message Processors (IMP) ARPA contract to BBN
- Sen. Ted Kennedy sends a somewhat confused telegram to BBN on winning the contract
  "Congratulations … on **interfaith** message processor"

Initial Baby Steps

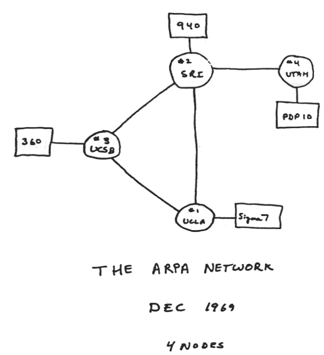


FIGURE 6.2 Drawing of 4 Node Network
(Courtesy of Alex McKenzie)

In the Beginning…

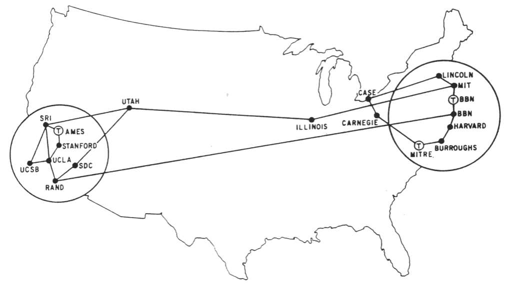
- Kleinrock's group at UCLA tried to log on to SRI computer: His recollection of the event…
- "We set up a telephone connection between us and the guys at SRI…
- We typed the L and we asked on the phone…
  - "Do you see the L?"
  - "Yes, we see the L," came the response
- We typed the O, and we asked…
  - "Do you see the O?"
  - "Yes, we see the O."
- Then we typed the G…
  - …and the system crashed!

September 1971

1970, ARPANET hosts start using *NCP*; first two cross-country lines (BBN-UCLA and MIT-Utah)

"Hostile overlay" atop telephone network

Ran a distance-vector routing protocol

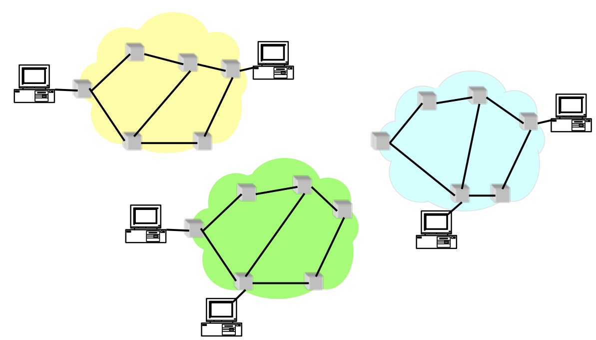


1970s: Packet networks → Internetworking

- 1972: successful ARPANET demo at conference (except it failed when demo'd to skeptics from AT&T!)
- 1972: modified ARPANET email program
- 1972: CYCLADES network (Louis Pouzin et al.): best-effort "datagrams"; *sliding window* protocol; distance-vector routing; time sync
- 1973: Ethernet (MAC protocol inspired by Aloha -- CSMA)
- 1973-74: Xerox PUP (used distance-vector protocol)
- 1973: ARPANET becomes international
- 1973-75: Internetworking effort (Cerf, Kahn, et al.)
  - Developed TCP and IP (originally intertwined) – TCP uses *sliding window*

The Problem

- Many different packet-switching networks
- Only nodes on the same network could communicate



Slide: Scott Shenker, UC Berkeley

Kahn's Rules for Interconnection

- Each network is independent and must not be required to change
- Best-effort communication
- Boxes (then called *gateways*) connect networks
- No global control at operations level (why?)

Original TCP paper

A Protocol for Packet Network Intercommunication

VINTON G. CERF AND ROBERT E. KAHN, MEMBER, IEEE

Abstract — A protocol that supports the sharing of resources that exist in different packet switching networks is presented. The protocol provides for variation in individual network packet sizes, transmission failures, sequencing, flow control, end-to-end error checking, and the creation and destruction of logical process-to-process connections. Some implementation issues are considered, and problems such as internetwork routing, accounting, and timeouts are exposed.

INTRODUCTION

IN THE LAST few years considerable effort has been expended on the design and implementation of packet switching networks [1]-[7],[14],[17]. A principle reason for developing such networks has been to facilitate the sharing of computer resources. A packet communication network includes a transportation mechanism for delivering data between computers or between computers and terminals. To make the data meaningful, computer and terminals share a common protocol (i.e, a set of agreed upon conventions). Several protocols have already been developed for this purpose [8]-[12],[16]. However, these protocols have addressed only the problem of communication on the same network. In this paper we present a protocol design and philosophy that supports the sharing of resources that exist in different packet switching networks.

After a brief introduction to internetwork protocol issues, we describe the function of a GATEWAY as an interface between networks and discuss its role in the protocol. We then consider the various details of the protocol, including addressing, formatting, buffering, sequencing, flow control, error control, and so forth. We close with a description of an interprocess communication mechanism and show how it can be supported by the internetwork protocol.

Even though many different and complex problems must be solved in the design of an individual packet switching network, these problems are manifestly compounded when dissimilar networks are interconnected. Issues arise which may have no direct counterpart in an individual network and which strongly influence the way in which internetwork communication can take place.

A typical packet switching network is composed of a set of computer resources called HOSTS, a set of one or more *packet switches*, and a collection of communication media that interconnect the packet switches. Within each HOST, we assume that there exist *processes* which must communicate with processes in their own or other HOSTS. Any current definition of a process will be adequate for our purposes [13]. These processes are generally the ultimate source and destination of data in the network. Typically, within an individual network, there exists a protocol for communication between any source and destination process. Only the source and destination processes require knowledge of this convention for communication to take place. Processes in two distinct networks would ordinarily use different protocols for this purpose. The ensemble of packet switches and communication media is called the *packet switching subnet*. Fig. 1 illustrates these ideas.

In a typical packet switching subnet, data of a fixed maximum size are accepted from a source HOST, together with a formatted destination address which is used to route the data in a store and forward fashion. The transmit time for this data is usually dependent upon internal network parameters such as communication media data rates, buffering and signalling strategies, routeing, propagation delays, etc. In addition, some mechanism is generally present for error handling and determination of status of the networks components.

Individual packet switching networks may differ in their implementations as follows.

1) Each network may have distinct ways of addressing the receiver, thus requiring that a uniform addressing scheme be created which can be understood by each individual network.

2) Each network may accept data of different maximum size, thus requiring networks to deal in units of the smallest maximum size (which may be impractically small) or requiring procedures which allow data crossing a network boundary to be reformatted into smaller pieces.

3) The success or failure of a transmission and its performance in each network is governed by different time delays in accepting, delivering, and transporting the data. This requires careful development of internetwork timing procedures to insure that data can be successfully delivered through the various networks.

4) Within each network, communication may be disrupted due to unrecoverable mutation of the data or missing data. End-to-end restoration procedures are desirable to allow complete recovery from these conditions.

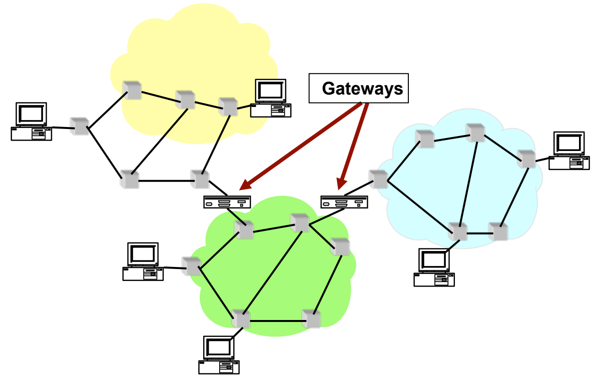
Paper approved by the Associate Editor for Data Communications of the IEEE Communications Society for publications without oral presentation. Manuscript received November 5, 1973. The research reported in this paper was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC 15-73-C-0370.
V.G. Cerf is with the Department of Computer Science and Electrical Engineering, Standford University, Stanford, Calif.
R.E. Kahn is with the Information Processing Technology Office, Advanced Research Projects Agency, Department of Defense, Arlington, Va.

Cerf RFC 968

```
Twas the night before start-up and all through the net,
  not a packet was moving; no bit nor octet.
The engineers rattled their cards in despair,
  hoping a bad chip would blow with a flare.
The salesmen were nestled all snug in their beds,
  while visions of data nets danced in their heads.
And I with my datascope tracings and dumps
  prepared for some pretty bad bruises and lumps.
When out in the hall there arose such a clatter,
  I sprang from my desk to see what was the matter.

There stood at the threshold with PC in tow,
  An ARPANET hacker, all ready to go.
I could see from the creases that covered his brow,
  he'd conquer the crisis confronting him now.
More rapid than eagles, he checked each alarm
  and scrutinized each for its potential harm.
```


Solution



Slide: Scott Shenker, UC Berkeley

The Internetworking Vision

- Bob Kahn & Vint Cerf imagined there would be only a few networks and thus only a few gateways
  - "*The choice for network identification (8 bits) allows up to 256 distinct networks. This size seems sufficient for the foreseeable future.*"
  - They were a little wrong!
- Gateways would "translate" between networks
  - Evolved in the 1974 Cerf/Kahn paper as a universal network layer, later called the *Internet Protocol*, or IP
  - We now think of it as all routers supporting IP

11

Handling Heterogeneity

- Make it very easy to be a node or link on the network (best-effort)
- Universal *network layer*: standardize addressing and forwarding
- Switches maintain *no per-connection state* on behalf of end points
- Original addressing model (then called a "TCP address" because IP and TCP hadn't been split into different layers):

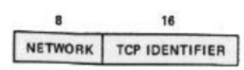
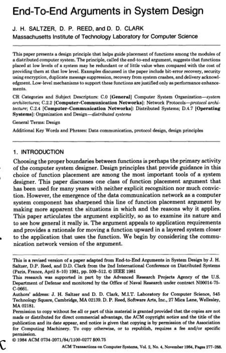


Fig. 4. TCP address.

1970s: Internetworking

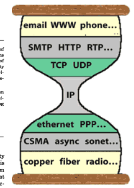
- 1978: **Layering**! TCP and IP split; TCP at end points, IP in the network
- IP network layer: simple best-effort delivery
- In retrospect: Packet switching (& TCP/IP) won because it is good enough for almost every application (though optimal for nothing!)
- Competitor to TCP/IP: ISO, standardizing 7-layer *OSI stack*


End-To-End Arguments in System Design

J. H. SALTZER, D. P. REED, and D. D. CLARK
Massachusetts Institute of Technology Laboratory for Computer Science

1. INTRODUCTION

Classic Internet layering "hourglass" model



Most Useful Lessons

One should architect systems for flexibility – you'll almost never know what apps make it succeed.

(Even if it means sacrificing some performance!)

Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher.

Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away

-- Antoine de Saint-Exupery

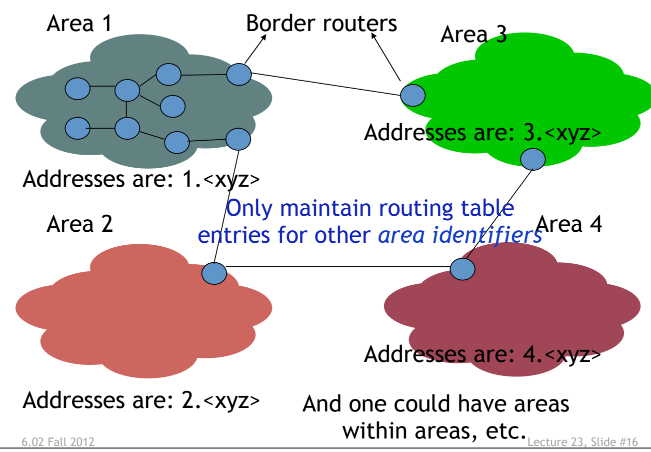
Or,

When in doubt, leave it out

1980s: Handling Growth with Topological Addressing

- 1978-79: ARPANET moves to *link-state routing*
- Per-node routing entries don't scale well
- Solution: Organize network hierarchically
  - Into "areas" or "domains"
  - Similar to how the postal system works
  - Hide detailed information about remote areas
- For this approach to work, node addresses must be *topological*
  - Address should tell network *where* in the network the node is
  - I.e., address is a *location* in the network
- Three classes of addresses in the 80s: "Class A", "Class B", and "Class C"
  - Not used any more, though the dotted decimal notation of IPv4 addresses makes it look like the dots matter

Ideal Case: Classic "Area Routing"

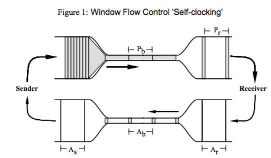


1980s: Rapid Growth

- 1981-89: Dave Clark of MIT is Internet's "Chief Architect"
  - Co-author of the end-to-end arguments (w/ Saltzer/Reed)
  - Ensures consistency of design and vision
  - "*We reject kings, presidents, and voting. We believe in rough consensus and running code.*"
- 1982: US DoD standardizes on TCP/IP
  - Berkeley's computer systems research group produces BSD & sockets
- 1983: MIT Project Athena – large-scale campus-area networking
- 1984: Domain Name System (DNS) introduced
- 1985: NSFNet picks TCP/IP as standard

Growth Problems: Congestion

- 1986: Congestion collapse episodes on the Internet
  - Problems with bad timeout settings
  - Window size not appropriate for network state
  - Athena network file system congestion problems (bad timeout settings)
- *Congestion avoidance and control*
  - *RTT estimation using EWMA + new timeout method*
  - TCP congestion control by Van Jacobson (concurrent work on DECBit scheme by Ramakrishnan & Jain)
  - Adapt the window size to congestion: If congested, decrease window; else increase. Use exponential back-offs on timeouts
  - By the end of the 1980s, essentially all running TCPs had congestion control

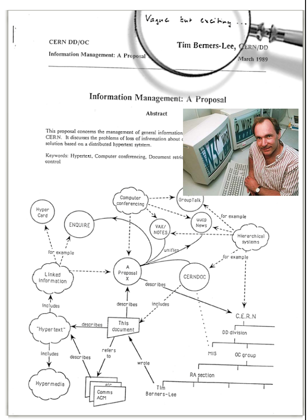


Jacobson

Jain

1990s

- 1990: no more ARPANET
- 1991: Tim Berners-Lee releases "WorldWideWeb"
- Mid-1990s: NSFNet backbone ends
  - Commercial ISPs take off
- "Classless" addressing for scale
  - And the rise of NATs
- BGP4: *Path vector protocol* between competing ISPs, who must yet cooperate

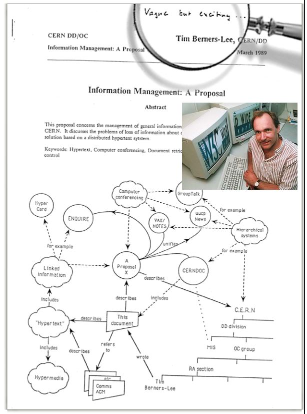


1990s: Handling Growth with CIDR IPv4 Addresses & Address Prefixes

- 18.31.0.82 is actually the 32 bit string 00010010000111110000000001010010
- Routers have forwarding table entries corresponding to an address **prefix** (a range of addrs w/ common prefix bitstring)
- 18.0.0.0/8 stands for all IP addresses in the range 00010010 00…0 to 00010010 11…1 (i.e., 2^{24} addresses of the form 00010010*)
- 18.31.0.0/17 stands for a range of 2^{15} consecutive IP addresses of the form 00010010001111100* (1^{st} 17 bits are the same for each address in that range)
- Hence, *subnetworks* may be of size 1, 2, 4, 8, … (maxing out at 2^{24} usually), and may be recursively divided further
- Forwarding uses **longest prefix match**
  - At each router, routes are of the form "For this range of addresses, use this route"
  - Pick the route that has the longest matching prefix w/ dest addr

1990s

- 1990: no more ARPANET
- 1991: Tim Berners-Lee releases "WorldWideWeb"
- Mid-1990s: NSFNet backbone ends
  - Commercial ISPs take off
- "Classless" addressing for scale
  - And the rise of NATs
- BGP4: *Path vector protocol* between competing ISPs, who must yet cooperate
- 1991-1994: IPng & IPv6 design starts
- 1993: search engines (Excite)
- Mid-1990s: E-commerce starts
- 1998: Google reinvents search
- 1998: Content distribution networks like Akamai
- 1996-2001: .com bubble starts & bursts



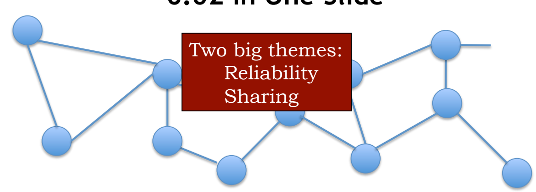
2000s: The Internet Matures (Top 5 List)

- 2000-2001: .com bust
  - And 9/11 happens
  - Power of content distribution networks to handle load
1. The rise of peer-to-peer networks
   - Gnutella, Freenet, distributed hash tables (e.g., Chord), BitTorrent, and of course, Napster
2. Security threats and defenses
   - 2000: Large-scale distributed denial-of-service (DDoS) attacks start
   - 2003: SQL slammer worm
   - Spam → phishing and pharming → complex ecosystem
   - Route hijacking by errors or malice
3. User-generated content & social networks
   - Blogs, Youtube, Facebook, and Twitter (UGC-meets-social)
4. The rise of wireless and mobile data
5. Cloud computing and large-scale datacenters (Amazon, Google, Microsoft, Facebook, etc.)
- *Almost* everything moves to the Internet: telephony, video, entertainment

2010s: The Decade Ahead

- Even more massive growth (largely from video, entertainment, and collaboration) & internationalization
- Combating complexity: new methods to make things simpler ("software-defined networks")
- Wireless: the spectrum crisis, coping with mobility & variability
- Physical embedding & embodiment: sensors & actuators over the network, mobile robots and autonomous agents, vehicles, embedded devices
- Network security & privacy; censorship

6.02 in One Slide



How to design digital communication networks.
Three layers of abstraction: bits, signals, packets.
A unique storyline: vertical study across all layers

Bits: Binary representation. Compression (source coding). Bit errors and error correction codes (channel coding)

Signals: Noise. LTI models. Frequency-domain analysis. Modulation & demodulation.

Packets: MAC protocols for shared media. Packet-switching & queues. Routing protocols. Reliable transport.

What Next?

- Many UROP opportunities!
- Networks and computer systems
  - 6.033 (computer systems), 6.829 (computer networks), 6.824 (distributed systems), 6.263 (analysis of networks), 6.266 (network algorithms)
- Security
  - 6.857 (computer and network security), 6.858 (computer systems security)
- Signal processing & digital communications
  - 6.003 (signals and systems), 6.011 (communications, control, and signal processing)
- Advanced communication & information theory
  - 6.450 & 6.451 (digital communications), 6.441 (info theory)

Thank you!

- Lectures
  - George Verghese
  - Hari Balakrishnan
- Recitations
  - Yury Polyanskiy
  - Jacob White
  - Victor Zue
- TAs
  - Rui Hu
  - Shao-Lun Huang
  - Ruben Madrigal
  - Kyu Seob Kim
  - Eduardo Sverdlin-Lisker
  - Cassandra Xia

- This week
  - Tu rec: Quiz 3 review
  - W: no lecture
- Quiz 3: Dec 18 at 1.30 pm in Johnson