Some Observations and Interpretations of Internet History from 1968 to 1980

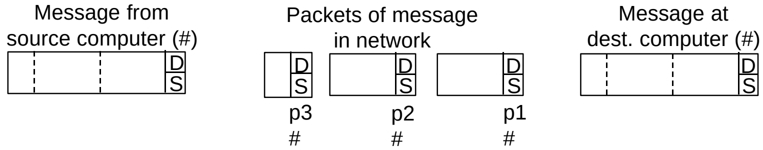
David Walden (BBN "IMP Guy")

8/31/99

1

Packet-switching

- Circuit switching has no concept of a data frame--circuit is "always" available for use
- Message switching has frames that are the natural unit of data--can be indefinitely long
- Packet switching "messages" are broken into packets a (few) thousand bits in length



2

Packet-switching continued

- Common user network allows many organizations and functions to share the cost of the network
- Packets are buffered in memory vs. on disk
- No blockage from long messages allows good latency
- Circuits sized to the average peak hour traffic across all users permitting decent ratio of circuit utilization to circuit cost
- Packet switching became possible because of changes in technology economics

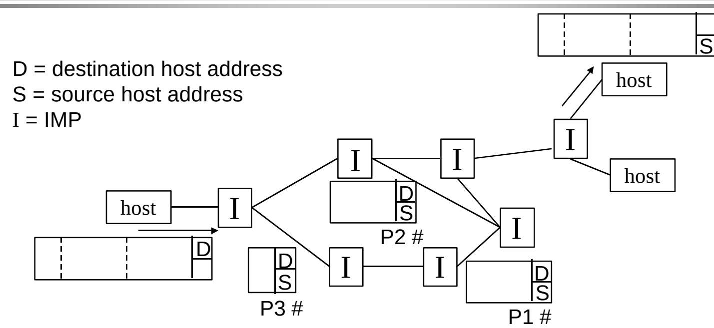
3

1968-1969

- ARPA let RFP for ARPANET (good general spec)
- Separation of packet switches (IMPs) and hosts
- BBN bid (complete design)
- BBN won and started on 1/1/69 with first system delivery required for 9/1/69
- Phone company and computer vendors argued against or ignored packet-switching

4

Messages and packets in the ARPANET



- Hosts send up to 8kbit messages into network of IMPs (according to "1822 protocol")
- IMPs break messages into 1Kbit or less packets that are individually routed
- IMPs to which destination hosts are connected reassemble packets into a messages

5

BBN "IMP" project

- Project team: 1 part-time manager, 1 communications expert, 3 programmers, 2 electrical engineers (Frank Heart, Bob Kahn, Will Crowther, Dave Walden, Bernie Cosell, Severo Ornstein, Ben Barker*)
- Machine: .5MHz, 32K bytes (.032 Mbytes) of memory, half memory for program, and half memory for store and forward storage
- H516 computer was the size of a refrigerator; 50kbs modem rack was the same size

* Others, e.g., Alex McKenzie, joined the BBN ARPANET team slightly later

6

Development environment

- Assembly language programming
- Using cross-assembler running on PDP-1
- Paper tapes of assembled coded transferred from PDP-1 to H516
- Debugger ran on H516 (breakpoints, etc.)
- No operating system in operational system

7

IMP design

- Total design, including hardware/software and theory/practice trade-offs
- Defensive design
- Designed performance in
  - » Counted instructions
  - » Hardware interrupt driven with hardware support for software initiated interrupts
- Primarily concerned with getting a basic system up and working reliably
  - » Optimizing and perfecting complex network algorithms happened during following several years

8

ARPANET characteristics

- Network transmitted binary data
- Significant transmission reliability provided by subnetwork of IMPs
- Dynamic routing
- Remote debugging, performance monitoring, down-line loading, etc. ("fake hosts" connected to Network Control/Operating Center)
- Immediate pressure/transition for multiple hosts per IMP and hosts remote from their IMPs
- Host/host protocol partitioned from communications subnetwork (NCP, Telnet, FTP, etc.)--protocol layers
- Pay for fixed transmission capacity--not bits sent or distance to destination (hid the cost in the organizational overheads and thus encouraged use)

9

Transmission of binary data (many prior networks did not)

- IMP/IMP interface
  - » DLE STX (marking beginnings of packets) and DLE ETX (marking ends) added by transmitting IMP hardware interface
  - » Receiving IMP hardware interface detected and removed DLE STX and DLE ETX
  - » DLEs within packet contents were doubled (DLE DLE) by transmitting hardware interface, and extra DLE was removed by receiving hardware interface
  - » 24-bit CRC calculated by the transmitting hardware interface appended after DLE ETX; recalculated and checked by receiving interface
- Host/IMP and IMP/Host interface
  - » Hardware signal supported binary data transmission for local hosts
  - » Communications protocol supported binary data transmission for remote hosts

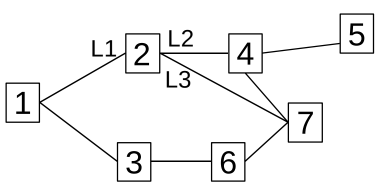
10

Reliable transmission (many prior networks did not have this, or what they had did not work well)

- Checksum calculated and added in IMP hardware interface (and packet-ID added in software interface) during transmission from the source IMP and a copy of the packet is stored at the source
- Checksum calculated and checked in hardware at destination IMP
- At destination, packets with bad checksums are discarded and software ACK is sent to source for packets with good checksums
- The source copy is discarded when ACK received
- The source copy is retransmitted if ACK is not received for too long
- Host/host reliability mechanisms were still required but did not have to be exercised so often for an overall increase in network performance
- Eventually, a software packet checksum was also included to protect against IMP memory errors (such as the one that corrupted routing messages and crashed the whole network)

11

Dynamic routing (many prior networks had static routing)

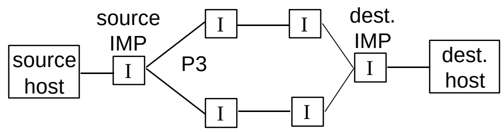


Switch 2's routing tables

MD+1		Route	Min dist.	From Line L1	L2	L3	Init. MD
	0						
2	1	L1	1	1	3	3	∝
1	2		0	2	2	2	0
3	3	L1	2	2	4	3	∝
2	4	L2	1	3	1	2	∝
3	5	L2	2	4	2	3	∝
3	6	L3	2	3	3	2	∝
2	7	L3	1	3	2	1	∝

12

Protocol layers (many prior networks were highly “integrated”)

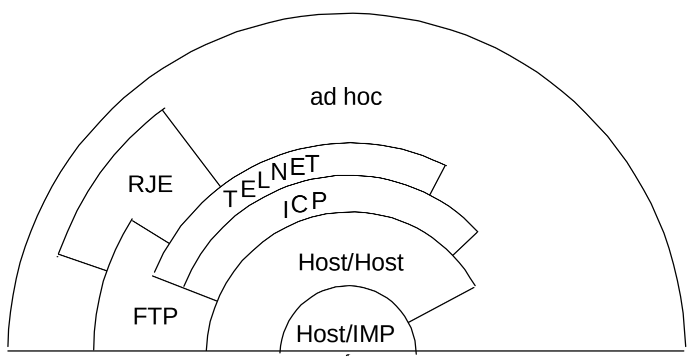


application protocol	Telnet, FTP, email, RPC, etc.
host OS/host OS protocol	OS compatibility
source IMP/dest. IMP protocol	messages to packets; packets to messages=>NCP *
IMP/IMP protocol	transmission between IMPs; routing across IMP network
electrical protocol on circuit	ignore this

* NCP was to ARPANET sort of what TCP is to Internet

13

Early figure of ARPANET host protocol layers



14

TIPs and TENEX

- Various sites had various host computers (Multics at MIT, Sigma 7 and IBM 360 at UCLA, IBM 360 at UCSB)
- However, many sites had TIP (shared rack or adjacent rack with IMP)
  - » Provided terminal access to network mainframe hosts for sites without mainframe hosts
  - » Provided substantial impetus for increased network use
  - » Supported by BBN and Network Control/Operations Center
- Many other sites had TENEX (OS and virtual memory version of PDP-10; became TOPs 20)
  - » Developed and supported by another BBN division
- TIPs and TENEXs enabled much early use of ARPANET
- Various Telnet issues were extensively studied between TIPs and TENEXs

15

Network Working Group (NWG)

- Organization that created host protocols
- Open, free standards
- Request for Comments (RFC) series
- Participation was (largely) on a voluntary basis and decisions were made by those who were doing the work
- “Strong consensus and working code” (as described later by Dave Clark)
- Standards expected to evolve based on experience and need
- Negotiated options

16

Negotiated options

- Bernie Cosell primarily conceived and worked-out Will/Won't, Do/Don't on a plane flight from Boston to a Telnet Protocol Meeting at UCLA
- Designed to:
  - » a. Address ships-passing-in-night protocol loops, i.e., either end can initiate an exchange or at same time
  - » b. Dynamically extensible protocols, i.e., supported implementation and use of protocol options without prior knowledge of other system

17

Negotiated options (continued)

- Designed to be always consistent and have no loops (per 8/14/99 email from Bernie Cosell)
  - » A base minimal set of capabilities is defined, e.g., for Network Virtual Terminal
  - » Four commands
    - – Will <option>
    - – Won't <option>
    - – Do <option>
    - – Don't <option
  - » Two rules
    - – Don't send a command to confirm a state you are already in
    - – Don't ask a second time for an option that has been declined

18

Some key principles (retrospectively)

- Common user network that reliably transmitted arbitrary data
- Dynamic routing from any host to any host with low delay and reasonable efficiency
- Decentralized operation with centralized monitoring and maintenance
- Billing independent of time and distance
- Layering of protocols so different levels can be changed without knowledge of applications at higher levels; separation of routing packets from application messages
- Network Working Group of individuals seeking good enough, open technical solutions that were expected to evolve

19

Some other influential activities from 1968-1980 I observed from close range

- Early ideas for connectionless host/host protocols (RFC62)
- Tomlinson and networked email
- Satnet development
- Need for and specification of TCP/IP
- Internet experiments including Satnet, Packet Radio Net, ARPANET, and early “gateways”
- Early distributed operating system work of Bob Thomas et al. (early example of a worm, early instance of a distributed file system, etc.)
- Transfer of operations of ARPANET to DCA
- Early end-to-end security systems (see Steve Kent presentation)

20

Some lessons I see from the early days

- Packet disassembly and reassembly in the subnetwork of IMPs had bugs early on; it also got in the way of real-time applications
- Need to pay attention to reliability and performance at every level
- 8 bits was not enough for network addresses, even for a prototype network; there is always pressure for more things to communications with other things
- It's a good idea to get something basic working that can then evolve (better dynamic routing, better network monitoring, TCP/IP, internetworking, connectionless protocols, etc. were added over time)
- In general it's good to focus on engineering solutions rather than political solutions; we go a lot farther a lot faster.

The early days of ARPANET provided important precedents

- Many of these principles have continued to be applied through several generations of internet technology over decades.