Networking and Internetworking 1

To do …

- Networks and distributed systems
- Internet architecture

xkcd

There are few forces more powerful than geeks desperately trying to get internet in a new apartment.

Okay, the Pringles antenna has let us patch in to the WiFi network across the road.

And they have internet?

No, but I think the cable van will hook up their house first.
~1960 ARPA sponsored research on computer networking to enable remote sharing
  - Electronic computers were scarce resources
  - Renting an IBM System/360 - $5k/month ($35k/month 2016)

1969 – First four ARPANET nodes connected
  - Key decision: use packet switching
Internet history – Early days

- From 1975 to 1980s
  - Successful ARPANET ~ 100 nodes
  - ARPA research on packet switching over radio and satellite
  - New LANs connected via gateways
  - TCP/IP conversion in 1983
  - Autonomous Systems and backbone AS for scalability
A bit of history – NSF takes over

- Late 1980s NSF takes over
  - Works on expanding the backbone
  - Encourages development of regional networks
  - Three tiers: backbone, regional, enterprise

- Enterprises were building TCP/IP networks and wanted to connect
  - But NSF charter prohibited them from using NSFNET
  - 1987 first commercial ISP, many follow shortly
A bit of history – Commercial operation

- By 1990 service providers where interconnected
  - Congress lets NSFNET interconnect with commercial networks
  - By 1995, NSFNET was retired
    - No single default backbone anymore
    - Many backbones interconnected through
      Network Access Points

- ~1995 Web
  - Easier to use Internet
  - Million of non-academic users

- ...
Internet building blocks

- **Basic blocks**

- **More hosts?**
  - Wires for everyone or routers
  - Routers act as switches routing msgs between nodes
Packet switching for communication

- Packet over circuit switching
  - Information sent as self-contained packets, with an address
  - Each routed independently to destination
  - Store and forward
  - Simpler than circuit-switched (in plain old telephone service)
Packet switching for communication

- Statistically multiplexing
  - Switches arbitrate between inputs
  - Send from any input that’s ready (keep busy)

- If network is overloaded
  - Short burst: buffer
  - After that
    - Packet drop
    - Sender adjust rate until load matches resources – congestion control
Networking protocols

- IPC is based on send/receive messages
- Message/packet transmission
  - Message: logical unit of information
  - Data transmission is in packets: transmission unit
  - Restricted length: sufficient buffer storage, reduce hogging

- For communication to work, parties must agree on several things
  - How does the receiver knows it got the last bit of a msg?
  - How longs are integers?
  - ...
Networking protocols

- Protocols – agreed-upon rules and formats
  - Sequence of messages to exchange, their formats
  - Implemented by pair of software modules on each end
  - Typically arranged in layers – protocol suite or stack
    - Layering and modularity to handle complexity

- Up and down the stack
  - As a message is passed down, each layer adds a header (sometimes a trailer)
  - On the receiver side the message is pushed upward with each layer stripping off and examining their own headers

<table>
<thead>
<tr>
<th>Ethernet header</th>
<th>IP header</th>
<th>TCP header</th>
<th>HTTP header</th>
<th>Application data (webpage)</th>
</tr>
</thead>
</table>

An example on the Web
Protocol stacks

- OSI and Internet
- Open Systems Interconnection model – ISO OSI
  - Not used in practice but good for understanding

![Diagram showing the OSI model with layers and communication medium]
Lower-level protocols

- Lower-level protocols, together implementing the basic functions of a computer network
  - Physical – deals with (standardizing) mechanical and electrical details
    - Just send bits, how long as there are no errors, all is fine …
  - Data link – groups bits into frames and ensure they are correctly received
    - Frames include checksums to check correctness
  - Network – describes how packet are routed from source to destination
    - Most used network protocol – Internet Protocol (IP)

- For many distributed systems, the lowest-level interface is that of the network layer
Transport and higher layers

- **Transport protocols**
  - Transfer messages between clients, including breaking them into packets, controlling flow, etc.

- According to the OSI, three high-level layers
  - **Session** – provides dialog control and synchronization
    - Control to keep track of who is currently talking
    - To insert checkpoints in long transfers
  - **Presentation** – resolves differences in formats among sites
    - Concern with meaning of transferred data to make resolving differences easier
  - **Application** – to meet the specs of a set of standard apps
    - Originally intended to support a collection of standard network applications; now the container for all
Routing

- A packet placed in the network must be forward to its destination as specified in its packet’s header
- A graph theory problem – find the lowest-cost path between two nodes
- Routers decide outgoing interface for each packet
  - Decision is called routing
  - Placing it in the appropriate outgoing interface – forwarding
  - Set of links used – packet’s path
- For routing, routers must keep information on the configuration and state of the network
  - Update state of links (added, removed, failed router, …)
Routing example – RIP

- Routing Information Protocol
- Highly popular, distributed with BSD Unix
- Based on distance vector algorithm (Bellman-Ford)

Router’s routing table
  - A record for each destination
  - Fields: outgoing link, cost (e.g. hop count)
  - A link that is down has an $\infty$ cost

Routers send table summary periodically (30s) or when an update from other routers changes its table
  - If received table shows new or better route, update local one
  - Distance vector router talks only to its neighbors, but tells them everything it has learned
Routing example

Routing Table for A

<table>
<thead>
<tr>
<th>To</th>
<th>Link</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Local</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Routing Table for D

<table>
<thead>
<tr>
<th>To</th>
<th>Link</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>Local</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Routing Table for B

<table>
<thead>
<tr>
<th>To</th>
<th>Link</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>Local</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Routing Table for C

<table>
<thead>
<tr>
<th>To</th>
<th>Link</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>Local</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Routing Table for E

<table>
<thead>
<tr>
<th>To</th>
<th>Link</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>Local</td>
<td>0</td>
</tr>
</tbody>
</table>
A 5’ break ...

- Details of Internet operation
  - New challenges
  - Addressing
  - Routing
  - Organization

5 MINUTES LATER...
Internet protocols

- Three higher layers ➔ Just one
- Transport layer provides inter-process communication
  - Transport address = net add + port
  - Two key protocols
    - TCP – Transmission Control Protocol – connection oriented, reliable stream communication
    - UDP – Universal Datagram protocol – connectionless, unreliable datagram communication
- Network protocol provides communication host to host
- Link layer – physically interface with communication medium
Internet Protocol – primary network layer protocol
  – Best effort service – unreliable, no guarantees
  – Connection between a host and network is called interface
  – Each interface has an address; format defined by IP; in IPv4 are 32b expressed as four decimals (165.124.180.20)
  – IP addresses often grouped by their prefixes – initial set of bits that all addresses in the group have in common
    e.g. 165.124.0.0/16 – all addresses with first 16b equal to 165.124

![Internet Protocol Packet Diagram]
Internet as a set of ASes

- **Internet**
  - A collection of separately, usually competing, managed networks

- **Autonomous system (AS)**
  - Set of network elements under a single organization’s control
  - 1 ISP can operate $N$ ASes; no AS is managed by more than one ISP

- **ASes exchanged traffic at connection points, aka peering or exchange points**
  - Connections formed by establishing a link between “gateway” routers in each AS
Routing in the Internet

- **RIP-1**, the first routing algorithm used in the Internet
  - A version of the distance-vector algorithm we discussed
  - Slow convergence, potential instability, Floyd & Jacobson’s* observation on 30s peak on latency

- **Open Shortest Path First (OSPF)**, a better option
  - Routers maintain a network map, updated when links change
  - Each router monitor the links to which is connected and initiates a flooding protocol upon change
  - Router talks to everyone, but only tells them what it knows for sure (state of directly connected links)

- **To compute path to destination**
  - Each computes best path from it to every destination – path with the lowest sum of link weights ➔ routing table

*S. Floyd and V. Jacobson, The Synchronization of Periodic Routing Messages, ACM ToN 1994*
Routing messages needs addresses

- Challenge of Internet protocol design: naming and addressing scheme and routing
- Scheme for assigning addresses has to be
  - Universal – any host can send packets to any other
  - Use address space efficiently – no idea of eventual size
    - $2^{32}$ or ~4 billion addressable hosts sounds like enough?!?
  - Enable flexible routing but addresses themselves shouldn’t carry much information for routing
- Most of the Internet still on IP version 4 (IPv4)
Domain names

- Hard to remember IPs, some mechanism for using symbolic names (and mapping them)
- Naming for the Internet before 1983
  - Each computer retrieved HOST.TXT from a computer at SRI
  - Legacy – a host file still exist in most modern OS
- DNS by Paul Mockapetris (1983)
  - Names organized into name spaces
  - Name space partitioned organizationally and geographically
  - A distributed system implements the hierarchical name space
    - Each server holds a partial map of the domain name tree bellow their domain
    - Lots of caching!
IP communication model and some fixes

- **IP model**
  - Messages may be lost, reo-ordered, corrupted

- **Can you build a reliable, in-order, mostly non-corruption, stream-oriented communication service?**
  - Data corruption – add checksums
  - Lost data – timeout and retry
  - Data out of order – Add sequence numbers
Internet protocols – TCP

- **Basic service – a connection**
  - A communication channel between two hosts
- **Service is provided to app, process to process**
- **Both senders and receivers create end points: sockets**
- **Application read to/write from sockets**
  - Sockets ID are #s: IP + 16b number local to the host, *port*
  - Port numbers [0,1023] are well-known ports, reserved for standard services; [1024,49151] registered ports

<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>daytime</td>
</tr>
<tr>
<td>22</td>
<td>ssh</td>
</tr>
<tr>
<td>1025</td>
<td>NFS</td>
</tr>
<tr>
<td>1293</td>
<td>IPSec</td>
</tr>
</tbody>
</table>
Internet protocols – TCP

- API provides abstraction of a stream of bytes, hiding
  - Message sizes and destination
  - Lost messages, duplication and ordering
  - Flow control

- General approach to reliability – acknowledgement
  - After sending each segment, source sets timer waiting for ack

- Congestion avoidance
  - TCP will try to match the speed of the processes reading from/writing to the stream
  - Implemented by varying number of un-acknowledged packets the sender allows (window size)
Internet protocols – UDP

- As TCP, process to process communication
- But simpler, a transport-level replica of IP
- UDP datagram is encapsulated inside an IP packet
  - Header includes src & dest port #s, length field and checksum

- UDP adds no additional reliability to IP
  - No guarantee of delivery (no acks or retries)
  - Too large messages are truncated on arrival
  - Messages may be dropped or delivered out of order

- No setup costs, no transmission delays above IP
The programmers view

- The TPC/IP stack from a programmers perspective

- Application services and application-level protocols based on TCP/IP
  - Web HTTP
  - Email SMTP, POP
  - Netnews NNTP
  - File transfer FTP

HTTPS: HTTP over TLS
Networking issues for distributed systems

- Early networks designed to meet relatively simple requirements from applications

- Recent uses and growth ➔ higher demands on performance, scalability, reliability, …
Networking issues for distributed systems

- **Performance**
  - To access resources on a LAN – 1,000x to access local memory, but faster than local disk
  - Over the Internet – 10-100x >> than a fast local network
  - For Amazon – 100ms extra latency ~1% sale loss*

- **Quality of service**
  - Higher demands from real-time services

http://blog.gigaspaces.com/amazon-found-every-100ms-of-latency-cost-them-1-in-sales/
Google/Bing Web search delay experiment

- Measure the impact of latency on user behavior
  - A/B testing (randomly assigning users to experiment and control group), server-side delays
- 400ms extra latency, 0.74% fewer searches over time

Impact of Post-header Delays Over Time

E. Shurman, J. Brutlag,
Networking issues for distributed systems

- **Scalability**
  - The Internet was not designed for expected or current scale, but yet must handle it

- **Reliability**
  - Increased dependence ➔ higher reliability

- **Security**
  - Large and more diverse user base + growing dependency ➔ Higher security demands – network is first level of defense

- **Mobility**
  - Addressing and routing schemes developed before mobile devices; not well suited to it

- **Multicasting**
  - Today’s Internet use is mostly for content distribution with a 1-to-m and m-to-m communication pattern
• Distributed systems use different networks for communication
• Changes in user requirements have placed new demands on the underlying networks
• At the same time, the characteristics of networks (e.g., performance, reliability, scalability) impact the behavior of and affect the design of distributed systems
• Next: some operational issues and network principles