Distributed Software Development

Fundamentals of Distributed Systems

Chris Brooks

Department of Computer Science
University of San Francisco

2-2: TCP/IP in 30 minutes

- Goal: Understand how a network transmits messages at different layers.
- How is a network composed?
- What really happens when Firefox opens a connection to a web server?
- Note: this will be an overview. For more details, you should take Prof. Buckwalter's class.

2-3: Layering

- Modern network design takes advantage of the idea of **layering**
- A particular service or module is constructed as a black box.
- Users of that service do not need to know its internals, just its interface.
- This makes it easy to later build new modules (or layers) that use the lower layers.
- For example, HTTP is built on top of TCP.
  - A web browser does not typically need to worry about the implementation of TCP, just that it works.
- Unlike modules in a typical OO system, the layers in a networked system comprise protocols that span multiple machines.

2-4: The OSI seven-layer model

- ISO (a standards body) developed a reference model called OSI that defines the different layers needed for communication, and specifies which should do each job.
- The goal is to produce an open protocol that allows for heterogeneous, extensible systems.
- A **protocol** is a specification describing the order and format of messages.
- An open protocol is one in which all of this information is publicly available.

2-5: The OSI seven-layer model

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

2-6: Message transmission across layers

- An application (such as a web browser) wants to send a message to another computer.
- That application constructs a message and passes it to the application layer.
- The application layer attaches a header to the message and passes it to the presentation layer.
- The presentation layer attaches a header and passes it to the session layer, and so on.
- On the other end, the message is received by the physical layer, who strips off the appropriate header and passes the message up to the data link layer.
- This continues until the message reaches the application layer of the receiving machine.
- High-level layers don't need to worry about lower-level layers.
- Lower-level layers treat everything from higher layers as data to be sent.

2-7: Layers and packets

- Each layer constructs a packet containing a portion of the data to be transmitted.
- This packet has a data section, and a header.
  - The header contains origin and destination information, checksums, sequence numbers, and other identifying information.
- When a message is send by (for example) TCP, a packet is constructed and passed down to the IP layer.
- This entire packet then becomes the data portion of the IP packet, which is passed down to the network layer, and so on.
- On the other end, the lowest layer removes the header and checks the data integrity, then passes the data portion up to the next layer.

2-8: Physical Layer

- This is the lowest-level layer, responsible for transmitting 0s and 1s.
- Governs transmission rates, full or half-duplex, etc.
- A modem works at the physical layer.
- Lots of interesting problems at this level that we won't get into ...

2-9: Data Link Layer

- The data link layer provides error handling for the physical layer
- Individual bits are grouped together into frames.
- A checksum is then computed to detect transmission errors.
- The data link layer can then request a retransmission of an error is detected.
- Messages are numbered; receiver can request re-transmission of any message in a sequence.
- Each frame is a separate, distinct message.
- The Data link layer provides error-free transmission to upper-level layers.

2-10: Network Layer

- The network layer is responsible for routing and flow control.
- The network layer removes the data link header and examines the resulting packet for a destination, and then forwards it as appropriate.
- The Internet Protocol (IP) is one of the best-known network-layer protocols.
- Primary role: move packets from a sending host to a receiving host. This involves:
  - Routing: determine the path that a packet should take to get to its destination.
  - Forwarding: When an incoming packet is received, place it on the output link that takes it to the next hop in its route.

2-11: Network Layer

- A router contains a **forwarding table** - when an incoming packet is received, the router compares it to this table to determine where to send it next.
  - This is forwarding.
- These forwarding tables are configured by means of a routing algorithm.
- For example, the link-state algorithm is a version of Dijkstra's algorithm - this computes a global routing table.
- Internet routing algorithms (such as BGP) are more complex, and use a decentralized routing table.
- In a nutshell, BGP lets subnets figure out how to reach other subnets via a gateway. That gateway is then responsible for routing within the subnet.

2-12: Transport Layer

- The network layer still operates at the level of individual packets, or datagrams.
- Packets may get lost, or arrive out of order.
- TCP is a transport-level protocol that provides **connection-oriented** service.
  - Guaranteed, in-order delivery.
  - State is maintained.
- This layer will also manage quality-of-service and some congestion control.
- UDP is also a transport level protocol, albeit one that does not provide connection-oriented delivery.

3-13: Session Layer

- The session layer was designed to provide support for access rights and synchronization.
- In practice, it is not widely used, and is not present in the TCP/IP suite.

3-14: Presentation Layer

- The presentation layer controls display of packet information.
- This may include encryption/decryption, compression, translation between character formats.

3-15: Application Layer

- This is the layer that most of us are most familiar with.
- It consists of user-level protocols built on top of the existing layers.
  - HTTP
  - FTP
  - SMTP
  - P2P protocols
  - Instant messaging
  - RTSP/streaming video
  - etc.

3-16: An example: HTTP

- HTTP is the protocol that drives the Web.
  - A side note/area to grind: WWW != Internet!!
- It is a stateless protocol that uses TCP as its underlying protocol.
  - The client sends a request, which is processed by the server.
  - The server sends a reply, and the exchange is ended.

3-17: HTTP requests

- HTTP has a very simple message format.