CS 544: Computer Networks
Midterm

Dustin Ingram

May 5, 2012

1 Reference Model Layers

1.1 TCP/IP

Layers in order from highest to lowest:

- Application Layer
  - Transport Layer Security Sublayer
- Transport Layer
- Internet Layer
- Link Layer
- Physical Layer

1.2 OSI

Layers in order from highest to lowest:

- Application Layer: This is the application interface, or API. This is how the network protocol communicates with an application.
- Presentation Layer: This is the layer which handles the"representation" of the data, encryption and decryption, etc.
- Session Layer: This is the "users" of a network,
- Transport Layer: This is the layer which provides reliable, end-to-end transportation of pieces of data.
- Network Layer: This is the layer that handles routing and congestion control for individual packets.
- Data Link Layer: Physical link and addressing specifications, handles sharing of a physical link.
  - LLC Sublayer: Protocol multiplexing
  - MAC Sublayer: Media Access Control (addressing for link sharing)
- Physical Layer: Bit transmission.

1.3 ATM

Layers in order from highest to lowest:

- ATM Adaption Layer
  - Convergence Sublayer
  - Segmentation and Reassembly Sublayer
- ATM Layer
- Physical Layer
  - Transmission Convergence Sublayer
  - Physical Medium Dependent Sublayer
- User & Management Planes

2 Network Protocols & DFAs

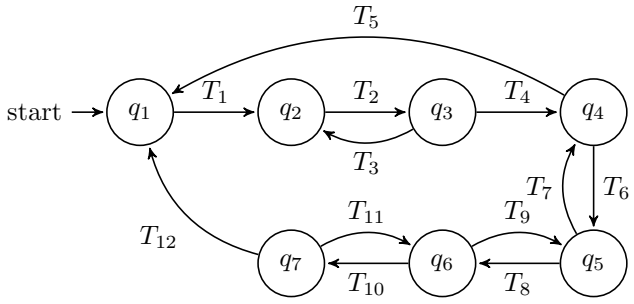
2.1 How Protocols use DFAs

Protocols use DFAs to determine state. A DFA describes every possible state the protocol may be in, and what must occur to transition from one state to another, typically messages sent or received or events which happen in or to the protocol. Together, the states and transitions define the protocol, the messages and events it can receive and respond to, and it's overall operation.

2.2 Why DFAs have to be Deterministic

A deterministic finite automata guarantees that when in one state, and a transition event or message occurs, the protocol *must* and *will* transition to the same subsequent state, every time.

2.3 Updates DFA



- q_1 – Waiting for update process to begin
- q_2 – Update process is active
- q_3 – Authenticating with remote server
- q_4 – Checking for updates
- q_5 – Downloading Update

- q_6 – Hashing Update
- q_7 – Installing Update
- T_1 – Time since last update times out
- T_2 – Authenticate using a security mechanism
- T_3 – Authentication failed
- T_4 – Authentication successful
- T_5 – No new updates
- T_6 – New updates
- T_7 – Download fails
- T_8 – Download succeeds
- T_9 – Hash fails
- T_{10} – Hash succeeds
- T_{11} – Install fails
- T_{12} – Install succeeds

2.4 Security

Security affects the states in the DFA of a protocol in that it adds additional states. In the example of the update protocol above, authentication is somewhat abstracted out, but there is still a state which represents it. Hashing can also be considered a security mechanism, to make sure the data is valid before installing.

3 Sliding Window Protocol

3.1 Pieces of the SWP

- Sequence Numbers – provide an ordering of messages
- Acknowledgements
- Positive – acknowledge the receipt of a message or multiple message
- Negative – acknowledge a missing message
- Next to receive – acknowledge messages to a point, and indicate what is expected next
- Windows- size is necessary to keep the messages relevant to the current frame
- Pipelining/Piggybacking – multiple messages at once
- Checksums – verifying messages via checksums

3.2 Video Protocol

Here, the sliding window protocol would not be the same. In the case of the streaming live video, there is a definite window (some amount of time ahead of what is currently being seen) that the protocol should focus message delivery on. It would not be necessary to work on delivering packets that are no longer needed to be viewed. However, in the case of storing the data to a local disk, every piece of data being transferred is relevant and important, regardless of the window size.

A peer-to-peer distributed protocol like BitTorrent would have almost no use for a sliding window protocol with regards to the original data ordering, as it is asynchronous and distributed, thus it is accepting packets completely out of order from multiple clients. It too would want to maximize file integrity.

4 Network Layers

The problem of networking is broken into layers so that different aspects of the network stack can treat each other as black boxes, and simply pass the necessary data between them. This concentrates similar functions together, and allows us to mix-and-match layers as they will interoperate well, creating a "building block" approach where one layer stacks nicely (communicates well with) another layer.

Sublayers further separate related functions, but are grouped within a common layer.

4.1 Transport Layer

Sublayers of the transport layer:

- End-to-End connections – Data transfer, connection-oriented streams
- Quality of Service – Reliable transfer, Error detection, Congestion and Flow Control

5 Common Themes

Some common themes repeat themselves through multiple different protocols. This is because there are a number of things which most protocols need to operate. For example:

- Addressing – A protocol must know how to address it's recipient
- Ordered Delivery – Out-of-ordered delivery is generally not useful, especially when working with large amounts of ordered data
- Segmentation & Reassembly – Dividing data based on the limits of a protocol's message size
- Error Detection & Correction – In lossy networking environments, or in situations where traffic may be manipulated, this is essential
- Flow Control – Ensures that clients with differing connection speeds can still communicate without being overwhelmed.

6 Common Themes In TCP

Here are those same themes, as they relate to one specific protocol, TCP:

- Addressing – TCP uses an IP address and a port, which are encapsulated into the header of the IP packet
- Ordered Delivery – TCP ensures an in-order delivery by using sequence numbers

- Segmentation & Reassembly – TCP splits the message into segments, which are re-assembled and ordered upon receipt
- Error Detection & Correction – TCP uses sequence numbers, a checksum and ACKs to discover and compensate for missing or corrupted packets
- Flow Control – TCP uses a SWP to reduce congestion and assure that the intended recipient can process the messages successfully

7 Extensibility

Extensibility is an important part of a good protocol because it allows the protocol to evolve over time without the need for modification of the core protocol. It also allows protocols to be fixed, or support new feature sets that were not originally considered or beyond the scope of the original protocol. Extensibility also allows implementors to potentially customize or tweak the protocol to their desire, and therefore allows for reuse.

7.1 XMPP and Extensibility

XMPP is inherently extendable – after all, it's in the name. Within the XMPP protocol lies a number of XMPP Extension Protocols, otherwise known as XEPs, which provide additional functionality across a wide range of applications on top of the core protocol. For example, XEP-0045 defines an extension for multi-user chat, which builds upon this RFC in the same way that this RFC builds upon other standards.

These XEPs can essentially modify the message body, the addressing of the messages, etc. The original specification provides enough flexibility with the base protocol that nearly any extension can be added without core modification.

7.2 FTP and Extensibility

An extension to the FTP protocol requires an addition of a command or command set – for example, FTPS (FTP Secure) requires an additional 'auth' command.