CSE135

A Review of Some Fundamental Ideas

Generation of App Patterns

HTTP Overview

- It's all about the network
  - If you want to really do this, you gotta know how to get your HTTP/1.0 Hello World to work
  - If the network has problems you'll have problems, no matter how clever you are (and probably more)
  - Sadly, Internet Squared and security problems
  - Sadly, even dear Chrome are much as they think they are
  - A few examples
    - Office - code semicolon? Longer? Tomorrow
    - SMTP or POP?
    - Cookies
    - "Large IP Block Combination - the usual hype
- HTTP Intro
  - HTTP (Hyper Text Transfer Protocol)
    - It's an application layer protocol for moving HTTP/1.1 POST, HEAD, GET, etc. (+ DELETE, etc.)
    - Simple protocol that defines the message sent by all clients request data from Web servers and how those servers respond
    - Typically it is running on top of "TCP/IP"
  - Three versions have been used in the wild (HTTP/0.9 is not actually used)
    - RFC 1945 HTTP 1.0 (1996)
    - RFC 2616 HTTP 1.1 (1999)
- HTTP and TCP/IP
  - HTTP sits atop the TCP/IP Protocol Stack
    - IP provides problems that are resolved based on secure end elaboration IP architecture
    - TCP
      - TCP provides singleton that ride across the IP packets and with connection alternative based on secure and bandwidth prior
      - The popular TCP using multiple protocols that connect servers running on default ports
        - HTTP on port 80
        - HTTP with SSL (HTTPS) on port 443
        - FTP on port 21
        - SMTP on port 25
        - SSH on port 22
      - TCP also provides mechanisms to make the connection a reliable bit pipe
        - A way to send ability, sequence numbers, checksums, neutral flags
        - A data stream is chopped up into packets, which are resequenced, acknowledged and in correct order and the other mechanism of the connection
      - TCP originally, sizing, similar IP problem, very like a bundle of cows
        - Office HTTP is the Application layer protocol on top of the TCP/IP stack. It's the part that has the contents of the HTTP Message
  - Examples
    -
  - Sockets
    - HTTP/1.1 supports and drains a persistent connection for multiple requests. Since most Web objects are small, this provides a significant speed up in terms of packets are simply TCP
    - And the previous goals for HTTP/1 also had congestion control and bandwidth conservation. HTTP/1.1 provides our TCP are been efficient
    - HTTP/1.1 ad creates three recovery with persistent connections using Keep-Alive
    - See https://developers.google.com/speed/articles/keep-alive-for-performance TCP projects and minimization on HTTP performance issues
- Basic HTTP Request/Response Cycle
  -
  -
  -
  - Proxy
    - In an ideal-world configuration, the client (browser) is explicitly configured to send all requests meaning the browser knows that all requests go through the proxy
    - Transparent is possible if the proxy is in the path. Typically the proxy service (like ProxyBid). You can also use Proxy Auto Configuration (PAC) file to automatically direct the appropriate proxy settings from a URL. When the browser makes a request, the browser consults the PAC file to determine which proxy to use. The proxy then forwards the request and stores a local copy of the response. If the proxy doesn't hold the requested data, it fetches it from the origin server. Both high-tech and low-tech are based on a server's configuration
    - Transparent Proxies
      - When implemented correctly, a transparent proxy does not know the traffic is being intercepted, and neither does the origin server. To enable the proxy to intercept traffic, the IP is, you must create a server that listens on port 80 and which is configured to interpret the incoming traffic as if it is an IP addresses on that port
      - A transparent HTTP proxy is a server that listens on port 80 for traffic on port 80. To make sure that the proxy knows what to do with requests for HTTPS/443, dpkrny listens on port 80 in case it is a request for a secure connection. The proxy then establishes a secure connection to the appropriate Traffic redirection is usually done by a firewall or router on the redundant server
    - https://knowledge.broadcom.com/external/article/166508/the-difference-between-explicit-proxy-and-tra.html?utm_source=google&utm_medium=organic&utm_campaign=KB&utm_content=Article&utm_term=The%20difference%20between%20explicit%20proxy%20and%20transparent%20proxy&utm_date=20250829&utm_lang=en&utm_cxt=166508&utm_cxt=166508&utm_cxt=166508