Ports

- Some well-known ports:
  - 80: http
  - 21: ftp
  - 23: telnet
  - 22: ssh
  - 79: finger
  - 42: nameserver
  - 13: daytime
  - 7: echo
  - /etc/services

Telnet Protocol

- TELNET is a protocol that provides "a general, bi-directional, eight-bit byte oriented communications facility".
- telnet is a program that supports the TELNET protocol over TCP.
- Many application protocols are built upon the TELNET protocol.

Network Virtual Terminal

- Intermediate representation of a generic terminal.
- Provides a standard language for communication of terminal control functions.
- Uses 7 bit codes for characters, transmitted as 8 bits (MSB=0). EOF is CR (carriage return) followed by LF (line feed). CR is CR followed by NUL (all bits zero) character.
Negotiable Options

- All NVTs support a minimal set of capabilities.
- Some terminals have more capabilities than the minimal set.
- The 2 endpoints negotiate a set of mutually acceptable options (character set, echo mode, etc).

Negotiated Options

- The protocol for requesting optional features is well defined and includes rules for eliminating possible negotiation "loops".
- The set of options is not part of the TELNET protocol, so that new terminal features can be incorporated without changing the TELNET protocol.

Examples for Options

- Line mode vs. character mode
- echo modes
- character set (EBCDIC vs. ASCII)

EBDIC and ASCII

- ASCII: 7-bit character set, 8th bit can be used for error checking or extended characters.
- EBCDIC: IBM, 8-bit, no parity checking possible, more control characters.

Some Control Characters

- Erase Character (EC):
  - delete last character sent
  - typically used to edit keyboard input
- Erase Line (EL):
  - delete all input in current line

Command Structure

- All TELNET commands and data flow through the same TCP connection.
- Commands start with a special character called the Interpret as Command character (IAC).
- The IAC code is 255.
- If a 255 is sent as data - it must be followed by another 255.
Looking for Commands

- Each receiver must look at each byte that arrives and look for IAC.
- If IAC is found and the next byte is IAC - a single byte is presented to the application/terminal (a 255).
- If IAC is followed by any other code - the TELNET layer interprets this as a command.

Command Codes

- IP 243 suspend, interrupt, abort process
- AO 244 abort output
- AYT 245 are you there
- EC 246 erase character
- EL 247 erase line
- WILL 251 performing indicated option
- WONT 252 refuse to perform option
- DO 253 request indicated option
- DONT 254 stop indicated option
- IAC 255 interpret as command

Options

- Options are agreed on by a process of negotiation, option will affect interchange and operation of client and/or server.
- Either end: IAC, <operation>, <option>
- Patterns:

<table>
<thead>
<tr>
<th>Sender</th>
<th>Receiver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WILL</td>
<td>DO</td>
<td>sender wants option</td>
</tr>
<tr>
<td>WILL</td>
<td>DONT</td>
<td>sender wants option</td>
</tr>
<tr>
<td>DO</td>
<td>WILL</td>
<td>sender requests option</td>
</tr>
<tr>
<td>DO</td>
<td>WONT</td>
<td>sender requests option</td>
</tr>
<tr>
<td>WONT</td>
<td>DONT</td>
<td>DONT is only valid response</td>
</tr>
<tr>
<td>DONT</td>
<td>WONT</td>
<td>WONT is only valid response</td>
</tr>
</tbody>
</table>

Options

- Sender wants the other end to echo:
  255(IAC), 251(WILL), 1

- Final byte indicates option.
- For some of the options values need to be exchanged: sub-option negotiation:
  IAC, SB, <option>, 1, IAC, SE
  IAC, SB, <option>, 0, <value>, IAC, SE

Example: client wishes to identify terminal type to the server:

<table>
<thead>
<tr>
<th>client</th>
<th>server</th>
<th>server</th>
<th>server</th>
<th>client</th>
</tr>
</thead>
<tbody>
<tr>
<td>255(IAC), 251(WILL), 24</td>
<td>255(IAC), 253(DO), 24</td>
<td>255(IAC), 250(SB), 24, 1, 255(IAC), 240(SE)</td>
<td>255(IAC), 250(SB), 24, 0, ’V’, ’T’, ’2’, ’2’, ’0’, 255(IAC), 240(SE)</td>
<td></td>
</tr>
</tbody>
</table>

- 1: value required, 0: value supplied

Playing with Telnet

- You can use the `telnet` program to play with the TELNET protocol.
- `telnet` is a generic TCP client.
  - Sends whatever you type to the TCP socket.
  - Prints whatever comes back through the TCP socket.
  - Useful for testing TCP servers (ASCII based protocols).
Some TCP Servers you can play with

- Some Unix systems have these servers running (by default):
  - echo port 7
  - discard port 9
  - daytime port 13
  - chargen port 19

```
Some TCP Servers you can play with

- Some Unix systems have these servers running (by default):
  - echo port 7
  - discard port 9
  - daytime port 13
  - chargen port 19

```

telnet hostname port

```
telnet hostname port

```

Playing with Telnet

- Try this:
  - telnet
  - toggle options
  - open machine.school.edu

```
Playing with Telnet

- Try this:
  - telnet
  - toggle options
  - open machine.school.edu

```

Telnet vs TCP

- Not all TCP servers talk TELNET (most don’t)
- You can use the telnet program to play with these servers, but the fancy commands won’t do anything.
  - type ‘^’`, then “help” for a list of fancy TELNET stuff you can do in telnet.

```
Telnet vs TCP

- Not all TCP servers talk TELNET (most don’t)
- You can use the telnet program to play with these servers, but the fancy commands won’t do anything.
  - type ‘^’`, then “help” for a list of fancy TELNET stuff you can do in telnet.

```

Client/Server

- telnet protocol and program.
- now: web server example (Apache).

```
Client/Server

- telnet protocol and program.
- now: web server example (Apache).

```

Web Server Basics

- HTTP: HyperText Transfer Protocol (communication between web server and web browser).

```
Web Server Basics

- HTTP: HyperText Transfer Protocol (communication between web server and web browser).

```

```
```
Apache Overview

- Apache 2.0
  - Apache tries to maintain pool of spare server threads.
  - "StartServers": initially started processes.
  - Dynamically forks/kills processes to keep number of idle threads between "MinSpareThreads" and "MaxSpareThreads".
  - "MaxClients": max. number of child processes is MaxClients/ThreadsPerChild.
  - "ServerLimit": hard limit for child processes.
  - "ThreadLimit": hard limit for threads.

- Modules:
  - dynamic or static linkage
  - in-process modules: run from inside the httpd process (mod_cgi, mod_perl, mod_php, mod_python, mod_tk)
  - out-of-process modules (mod_proxy, mod_jrun)
  - Typical configuration file:
    - ServerLimit 16
    - StartServers 2
    - MaxClients 150
    - MinSpareThreads 25
    - MaxSpareThreads 75
    - ThreadsPerChild 25
Apache 2.0

- Multi-Processing Module (MPM)
- An MPM defines how the server will receive and manage incoming requests.
- Allows OS-specific optimizations.
- Allows vastly different server models (e.g., threaded vs. multiprocess).

Apache 2.0

- Client:
  - single HTTP connection (e.g., web browser).
  - note that many web browsers open up multiple connections. Apache considers each connection uniquely.

Apache 2.0

- Keepalive requests:
  - Persistent connections
  - multiple requests over one TCP socket
  - directives:
    - KeepAlive
    - MaxKeepAliveRequests
    - KeepAliveTimeout

Performance: sendfile()

- fd-to-fd copying of data (stay in kernel).
- No more double-copy.
- Zero-copy*.
- Dramatic improvement for static files.
- Available on
  - Linux 2.4.x
  - Solaris 8+
  - FreeBSD/NetBSD/OpenBSD
  - ...

* Zero-copy requires both OS support and NIC driver support.

Web Server Performance

- ab
- flood
- httpperf
- JMeter
- ...and many others

Server ‘Quality’

- Response time:
  - server side: time to handle a request
  - client side: request transmission + time to handle a request + reply transmission
- Throughput:
  - number of requests handled in a second
**Response Times**

- Simple Load on a Single URL.
- Comes with Apache.
- Good for sanity check.
- Scales poorly.

**Throughput**

- Profile-driven load tester.
- Useful for generating real-world scenarios.
- Built to be highly-scalable.
- Designed to be extremely flexible.

**ab**
- Simple Load on a Single URL.
- Comes with Apache.
- Good for sanity check.
- Scales poorly.

**flood**
- Profile-driven load tester.
- Useful for generating real-world scenarios.
- Built to be highly-scalable.
- Designed to be extremely flexible.

**JMeter**
- Has a graphical interface.
- Built on Java.
- Part of Apache Jakarta (Java and Apache) project.
- Depends heavily on JVM performance.