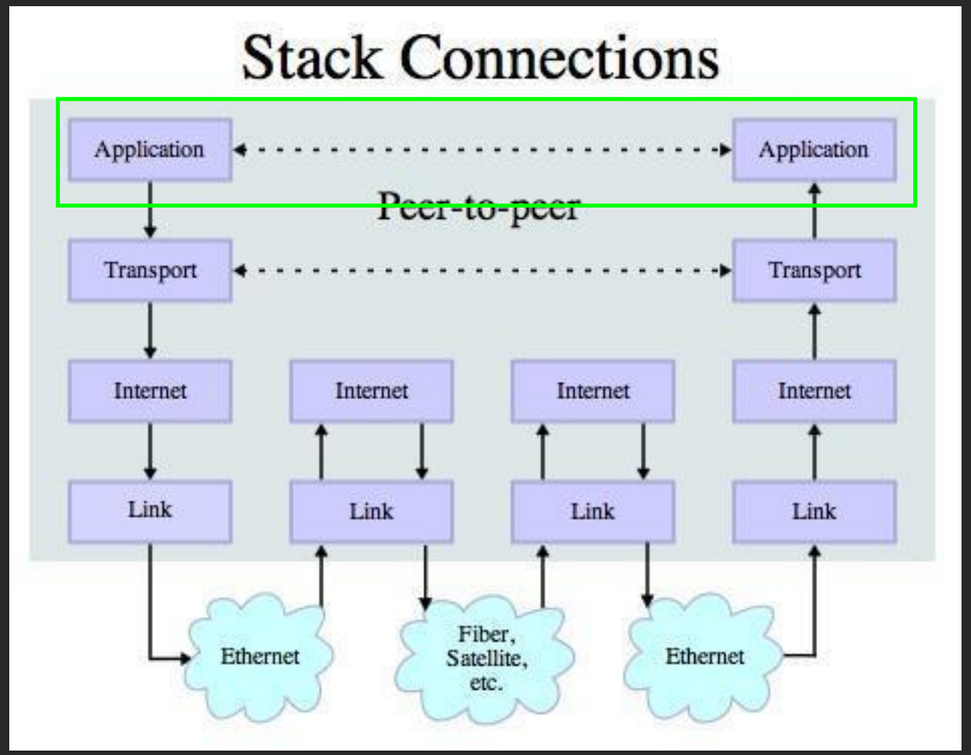
Application Protocol

- Since TCP (and Python) gives us a reliable socket, what do we want to do with the socket? What problem do we want to solve?

- Application Protocols
  - Mail
  - World Wide Web



Source: http://en.wikipedia.org/wiki/Internet_Protocol_Suite

HTTP - Hypertext Transfer Protocol

- The dominant Application Layer Protocol on the Internet
- Invented for the Web - to Retrieve HTML, Images, Documents, etc
- Extended to be data in addition to documents - RSS, Web Services, etc..Basic Concept - Make a Connection - Request a document - Retrieve the Document - Close the Connection

http://en.wikipedia.org/wiki/Http

HTTP

The HyperText Transfer Protocol is the set of rules to allow browsers to retrieve web documents from servers over the Internet

What is a Protocol?

- A set of rules that all parties follow so we can predict each other's behavior
- And not bump into each other
    - On two-way roads in USA, drive on the right-hand side of the road
    - On two-way roads in the UK, drive on the left-hand side of the road

http://www.dr-chuck.com/page1.htm

protocol host document

http://www.youtube.com/watch?v=x2GylLq59rI

1:17 - 2:19

Getting Data From The Server

Each the user clicks on an anchor tag with an href= value to switch to a new page, the browser makes a connection to the web server and issues a “GET” request - to GET the content of the page at the specified URL

The server returns the HTML document to the Browser which formats and displays the document to the user.

Web Server

80

Browser

Web Server

80

Browser

Click

Request

Web Server

80

GET http://www.dr-chuck.com/page2.htm

Browser

Click

Request

Web Server

80

GET http://www.dr-chuck.com/page2.htm

Browser

Click

Request

Web Server

80

Response

<h1>The Second Page</h1> If you like, you can switch back to the First Page.

GET http://www.dr-chuck.com/page2.htm

Browser

Click

Request

Web Server

80

Response

GET http://www.dr-chuck.com/page2.htm

<h1>The Second Page</h1> If you like, you can switch back to the First Page.

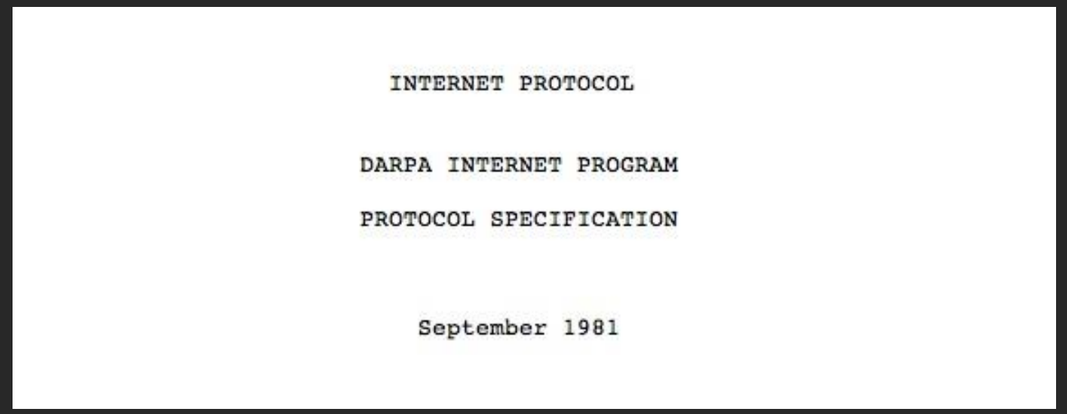
Browser

Click

Parse/ Render

Internet Standards

- The standards for all of the Internet protocols (inner workings) are developed by an organization
- Internet Engineering Task Force (IETF)
- www.ietf.org
- Standards are called “RFCs” - “Request for Comments”



INTERNET PROTOCOL

DARPA INTERNET PROGRAM

PROTOCOL SPECIFICATION

September 1981

The internet protocol treats each internet datagram as an independent entity unrelated to any other internet datagram. There are no connections or logical circuits (virtual or otherwise).

The internet protocol uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

Source: http://tools.ietf.org/html/rfc791

http://www.w3.org/Protocols/rfc2616/rfc2616.txt

Network Working Group
Request for Comments: 2616
Obsoletes: 2068
Category: Standards Track

R. Fielding
UC Irvine
J. Gettys
Compaq/W3C
J. Mogul
Compaq
H. Frystyk
W3C/MIT
L. Masinter
Xerox
P. Leach
Microsoft
T. Berners-Lee
W3C/MIT
June 1999

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information

```
5 Request

   A request message from a client to a server includes, within the
   first line of that message, the method to be applied to the resource,
   the identifier of the resource, and the protocol version in use.

        Request       = Request-Line              ; Section 5.1
                        *(( general-header        ; Section 4.5
                         | request-header         ; Section 5.3
                         | entity-header ) CRLF)  ; Section 7.1
                        CRLF
                        [ message-body ]          ; Section 4.3

5.1 Request-Line

   The Request-Line begins with a method token, followed by the
   Request-URI and the protocol version, and ending with CRLF. The
   elements are separated by SP characters. No CR or LF is allowed
   except in the final CRLF sequence.

        Request-Line   = Method SP Request-URI SP HTTP-Version CRLF
```

Making an HTTP request

Connect to the server like www.dr-chuck.com"

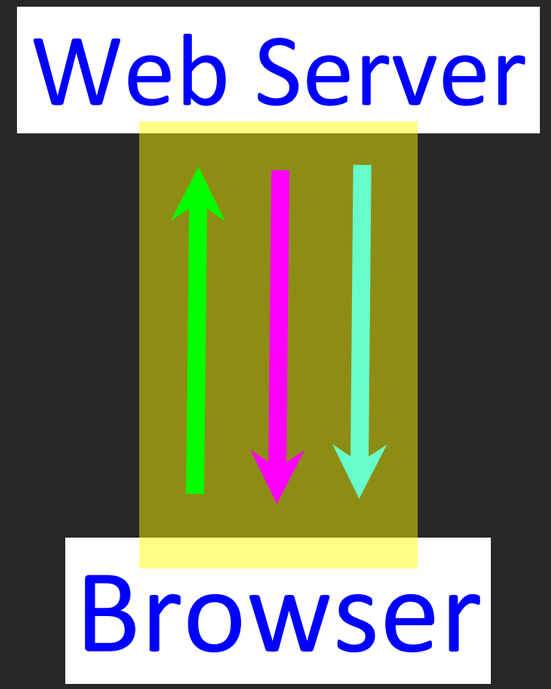
Request a document (or the default document)

- *GET http://www.dr-chuck.com/page1.htm HTTP/1.0*
- *GET http://www.mlive.com/ann-arbor/ HTTP/1.0*
- *GET http://www.facebook.com HTTP/1.0*

```
$ telnet data.pr4e.org 80
Trying 74.208.28.177...
Connected to data.pr4e.org.
Escape character is '^]'.
GET http://data.pr4e.org/page1.htm HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 30 Jan 2024 15:30:13 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Mon, 15 May 2017 11:11:47 GMT
Content-Length: 128
Content-Type: text/html

<h1>The First Page</h1>
<p>If you like, you can switch to
the <a href="http://data.pr4e.org/page2.htm">Second
Page</a>.</p>
Connection closed by foreign host.
```



Accurate Hacking in the Movies

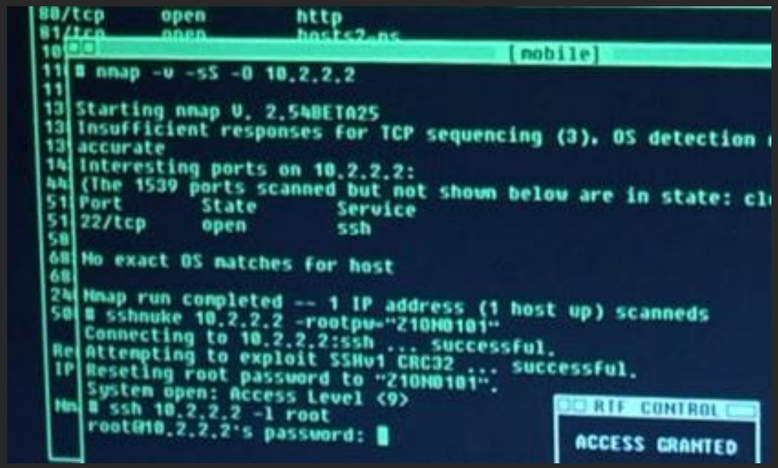
Matrix Reloaded

Bourne Ultimatum

Die Hard 4

...

http://nmap.org/movies.html

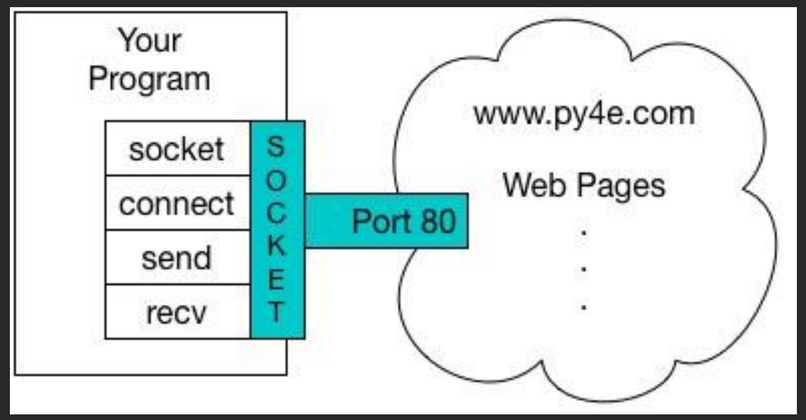


An HTTP Request in Python

```
import socket

mysock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect(('data.pr4e.org', 80))
cmd = 'GET http://data.pr4e.org/romeo.txt HTTP/1.0\n\n'.encode()
mysock.send(cmd)

while True:
    data = mysock.recv(512)
    if (len(data) < 1):
        break
    print(data.decode())
mysock.close()
```



```
HTTP/1.1 200 OK
Date: Sun, 14 Mar 2010 23:52:41 GMT
Server: Apache
Last-Modified: Tue, 29 Dec 2009 01:31:22 GMT
ETag: "143c1b33-a7-4b395bea"
Accept-Ranges: bytes
Content-Length: 167
Connection: close
Content-Type: text/plain

But soft what light through yonder window breaks
It is the east and Juliet is the sun
Arise fair sun and kill the envious moon
Who is already sick and pale with grief
```

HTTP Header

```
while True:
    data = mysock.recv(512)
    if ( len(data) < 1 ) :
        break
    print(data.decode())
```

HTTP Body

About Characters and Strings…

Acknowledgements / Contributions

Initial Development: Charles Severance, University of Michigan School of Information

… Insert new Contributors here

...