UNIVERSITY OF SAN FRANCISCO
CHANGE THE WORLD FROM HERE

Networking, HTTP, and Sockets

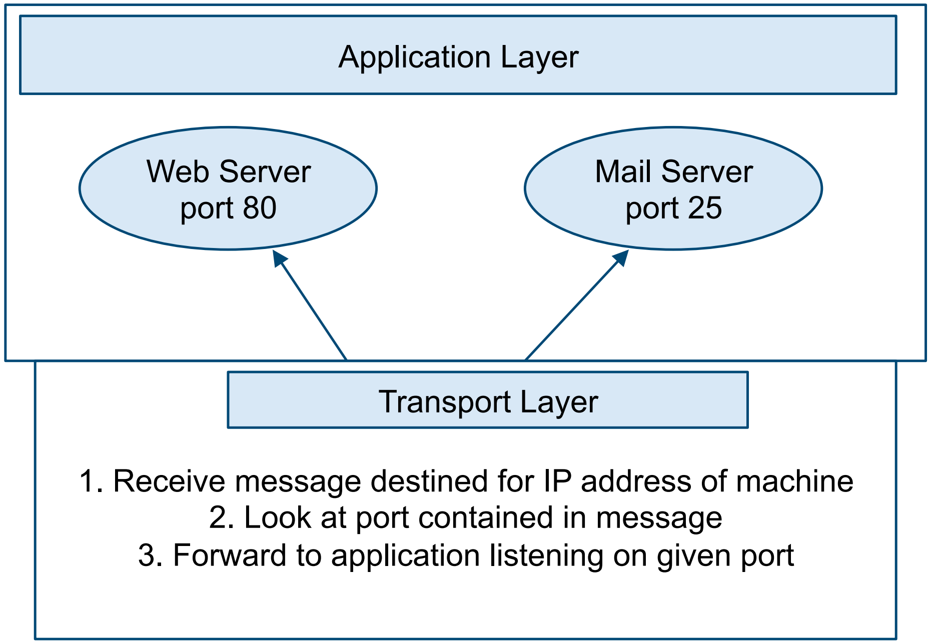
Sami Rollins

What’s going on in the network?

- **A web server is an *application-layer* program that opens a *socket* at port 80 and listens for incoming connections**

- **When *TCP* receives an incoming connection for port 80 it notifies the web server**

- **TCP manages the connection and knows where to send responses**
  - IP address and port of application making the request

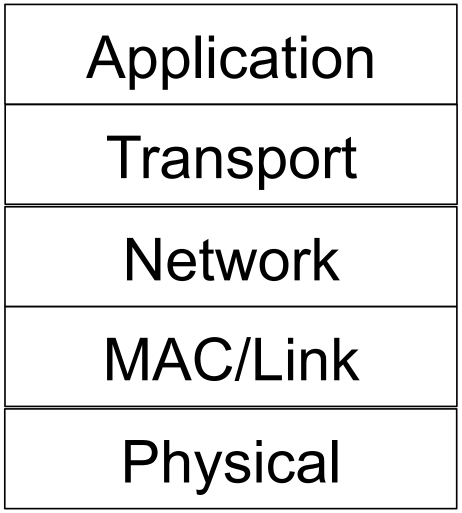


1. Receive message destined for IP address of machine
2. Look at port contained in message
3. Forward to application listening on given port

Networking Overview

- Networked communication over the Internet uses a *layered* protocol stack

- Each layer has well-defined responsibilities and relies upon the other layers to do their jobs.



Application

Transport

Network

MAC/Link

Physical

Overview of Layers

- **Application – where end-user software resides**
  - Web server, P2P client
- **Transport – reliability, congestion control, flow control**
  - TCP/UDP
- **Network – routing**
  - IP
- **MAC/DLL – connection to the next hop, accessing a shared medium**
  - Ethernet, wireless Ethernet
- **Physical – bits on the wire**

Hypertext Transfer Protocol

- **Language used to communicate on the web**
- **A standard that specifies the format of requests that can be issued by browsers/clients and responses that can be issued by web servers**
  - Allows interoperation between browsers written in any language running on any platform and services written in any language running on any platform
- **Defined in RFCs 1945, 2068, 2616**

Request/Response

```
GET /~srollins/test.html HTTP/1.1

HTTP/1.1 200 OK
Date: Wed, 16 Sep 2009 20:50:24 GMT
Server: Apache/2.2.3 (Red Hat)
Accept-Ranges: bytes
Content-Length: 113
Connection: close
Content-Type: text/html; charset=UTF-8

<html>
  <head>
    <title>Test Page</title>
  </head>
  <body>
    This is a short test page.
  </body>
</html>
```

HTTP Request Methods

- GET – retrieve a web page
- POST – upload (e.g., form input)
- HEAD – retrieve just header information
- PUT – upload a resource
- DELETE – delete a resource

- GET/POST are most common

Request Headers

```
GET /test.html HTTP/1.1
Host: localhost:1024
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US;
rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/
*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: author=AnonymousCoward
Cache-Control: max-age=0
```

Response Headers

```
HTTP/1.1 200 OK
Date: Thu, 17 Sep 2009 00:33:18 GMT
Server: Apache
Accept-Ranges: bytes
Cache-Control: max-age=60, private,
private
Expires: Thu, 17 Sep 2009 00:34:05
GMT
Content-Type: text/html
Vary: Accept-Encoding,User-Agent
Content-Length: 99253
Connection: close
```

Status Codes

- 200 OK
- 404 Not Found
- 500 Internal Error
- 301 Moved
- 403 Forbidden

1.0 vs 1.1

- 1.1
  - allows persistent connections
    - use `Connection: close` header for non-persistent
  - `Host:` header must be specified

Sockets

- **A socket is a programming abstraction used for network communication**
  - A pipe/stream that allows you to send/receive messages over the Internet
  - https://docs.oracle.com/javase/tutorial/networking/sockets/index.html

- **Server**
  - open a socket at a particular port
  - wait for message
  - process message and send response

- **Client**
  - open a socket using IP address of destination and port of listening application
  - send message
  - wait for response