CS 240 Computer Systems

Week 9: Networking, Web Services, and Python

CS 240, Spring 2021 - Week 9

Wade Fagen-Ulmschneider

Networking

Q: What do we expect out of networking?

...making this happen is **insanely complex**:

Hosts Routers Links Applications	Protocols Hardware Software Bit Errors	Packet Errors Link Failures Node Failures Message Delays	Out-of-Order Packets Eavesdropping ...and more...

We define common _____ -- a message format and rules for exchanging messages. You know many protocols already:

Network Data

At the core, network data is simply a series of **0**s and **1**s, which we represent in hex. (You can view all of the network packets on your VM using `tcpdump -x`.) For example, here one of many packets used in a request for me to view `waf.cs.illinois.edu`:

```
00  4500 00c6 1e1f 4000  4006 152e ac16 b24c   # IPv4
10  12dc 95a6 bafa 0050  0f60 c9b4 356a 523f   # TCP
20  8018 01f6 079e 0000  0101 080a 8146 30a0
30  31d4 daac 4745 5420  2f20 4854 5450 2f31   # HTTP
40  2e31 0d0a 5573 6572  2d41 6765 6e74 3a20
50  5767 6574 2f31 2e32  302e 3320 286c 696e
60  7578 2d67 6e75 290d  0a41 6363 6570 743a
70  202a 2f2a 0d0a 4163  6365 7074 2d45 6e63
80  6f64 696e 673a 2069  6465 6e74 6974 790d
90  0a48 6f73 743a 2077  6166 2e63 732e 696c
a0  6c69 6e6f 6973 2e65  6475 0d0a 436f 6e6e
b0  6563 7469 6f6e 3a20  4b65 6570 2d41 6c69
c0  7665 0d0a 0d0a
```

OSI Model

The Open Systems Interconnection (OSI) model is a 7-layer view of networking that abstracts and encapsulates the functionality of each component of networking.

OSI Layer 1: _____

OSI Layer 2: _____

OSI Layer 3: _____

```
00  4500 00c6 1e1f 4000  4006 152e ac16 b4a3
10  12dc 95a6                             ...
```

`IPv4, Packet Length: 0x00c6 (198 bytes); Source IP: ac.16.b4.a3 (172.22.180.163); Destination IP: 12.dc.95.a6 (18.220.149.166)`

OSI Layer 4: _____

```
10  ...  bafa 0050  0f60 c9b4 356a 523f
20  8018 01f6 079e 0000  0101 080a 8146 30a0
30  31d4 daac                         ...
```

`Port :0xbafa (47866) connecting to Port :0x0050 (80); Checksum 0x079e; Timestamp: 0x814630a0 (2168860832)`

OSI Layer 5, 6, and 7: _____

```
30  ...  4745 5420  2f20 4854 5450 2f31
40  2e31 0d0a 5573 6572  2d41 6765 6e74 3a20
50  5767 6574 2f31 2e32  302e 3320 286c 696e
60  7578 2d67 6e75 290d  0a41 6363 6570 743a
70  202a 2f2a 0d0a 4163  6365 7074 2d45 6e63
80  6f64 696e 673a 2069  6465 6e74 6974 790d
90  0a48 6f73 743a 2077  6166 2e63 732e 696c
a0  6c69 6e6f 6973 2e65  6475 0d0a 436f 6e6e
b0  6563 7469 6f6e 3a20  4b65 6570 2d41 6c69
c0  7665 0d0a 0d0a
```

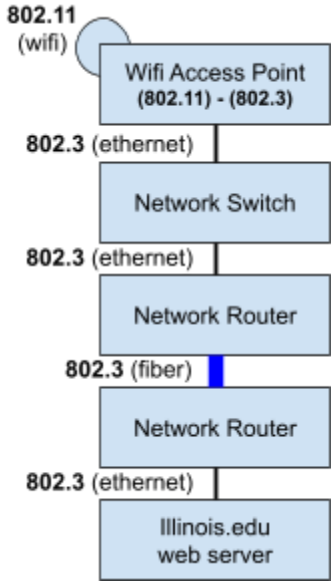
```
GET / HTTP/1.1\r\n
User-Agent: Wget/1.20.3
(linux-gnu)\r\n
Accept: */*\r\n
Accept-Encoding: identity\r\n
Host: waf.cs.illinois.edu\r\n
Connection: Keep-Alive\r\n
\r\n
```

Full Packet Journey

Consider an HTTP request you are making from your browser to waf.cs.illinois.edu (just as I did using **`tcpdump`**). I made my connection on my laptop, with a WiFi connection:

Your Laptop:

802.11 Header	IP Header	TCP Header	HTTP Header	HTTP Content



Network Layer (Layer 3) Protocol: Internet Protocol (IP)

- The network layer provides "host-to-host" communication.
- When on the Internet, every host relies on the IP protocol:
    - IP (IPv4) Address:
    - IPv6 Addresses:

Transport Layer (Layer 4) Protocols:

There are also two major Layer 4 protocols that are widely used throughout the Internet (and more exist that are less used):

1.
2.

Features of TCP and UDP:

Feature	TCP	UDP

Communication Between Processes: Web Services
One of the primary ways that processes will communicate is via "web services" -- applications that communicate using the HTTP protocol.

The HTTP protocol has two components:

[1]:

```
REQUEST
  1 POST /extract HTTP/1.1\r\n
  2 Host: localhost:5000\r\n
  3 User-Agent: curl/7.68.0\r\n
  4 Accept: */*\r\n
  5 Content-Length: 3046796\r\n
  6 \r\n
  … { 3,046,796 bytes payload }
```

Request Organization:

- Line Delineation:

- HTTP Request Type/Verb, Page, and Version (Line 1):

- Request Headers (Lines 2+):

- Payload (or "contents"/"data"):

[2]:

```
RESPONSE
  1 HTTP/1.0 200 OK\r\n
  2 Content-Length: 3044143\r\n
  3 Content-Type: image/gif\r\n
  4 Last-Modified: Mon, 28 Sep 2020 21:16:13\r\n
  5 Cache-Control: public, max-age=43200\r\n
  6 Expires: Tue, 29 Sep 2020 09:16:12 GMT\r\n
  7 ETag: "1601327773.0845277-3044143-32865"\r\n
  8 Server: Werkzeug/0.16.1 Python/3.8.2\r\n
  9 Date: Mon, 28 Sep 2020 21:16:12 GMT\r\n
 10 \r\n
  … { 3,044,143 bytes of content }
```

In general, the request and response follows the same format with only one major exception:

Response "Status Code" (Line 1):

1XX	**100**: Continue
2XX	**200**: OK **201**: Created
3XX	**304**: Not Modified
4XX	**400**: Bad Request **400**: File Not Found
5XX	**500**: Internal Server Error

Python Programming
All modern programming languages provide many libraries for quickly and easily working with web requests. In CS 240, we will focus on Python and use the **flask** library for networking.

Python Overview:

- Python is an "interpreted" programming language:
    - Note: Python only allows one thread to access the CPU (others can be blocked or ready, but there is no parallel execution)! *(Simplifies the execution environment, but prevents optimizations that are possible in C/C++.)*
- Python is a "dynamically typed" programming language:
- Python's control-flow is whitespace delimited:
- Python places heavy emphasis on code readability:

python/hello.py

```
1  s1 = "Hello"
2  s2 = " World"
3
4  for i in range(10):
5    if i < 5:
6      print(s1)
7    elif i < 8:
8      print(s1 + s2)
9    else:
10     print(s2)
```

Flask Library:
The flask library focuses on providing a simple interface to handling web requests:

python/app.py

```
1  from flask import Flask
2  app = Flask(__name__)
3
4  # Route for "/" for a web-based interface to this
   micro-service:
5  @app.route('/')
6  def index():
7    from flask import render_template
8    return render_template("index.html")
9
10 # Extract a hidden "uiuc" GIF from a PNG image:
11 @app.route('/extract', methods=["POST"])
12 def extract_hidden_gif():
13   # ...
```

Import Statements (Line 1, 7):

Python Comments (4, 10):

Python Function Definitions (Lines 6, 12):

Python Decorator (Lines 5, 11):

Flask is widely used, lots of great resources available! *(This is why we use widely used libraries!)*