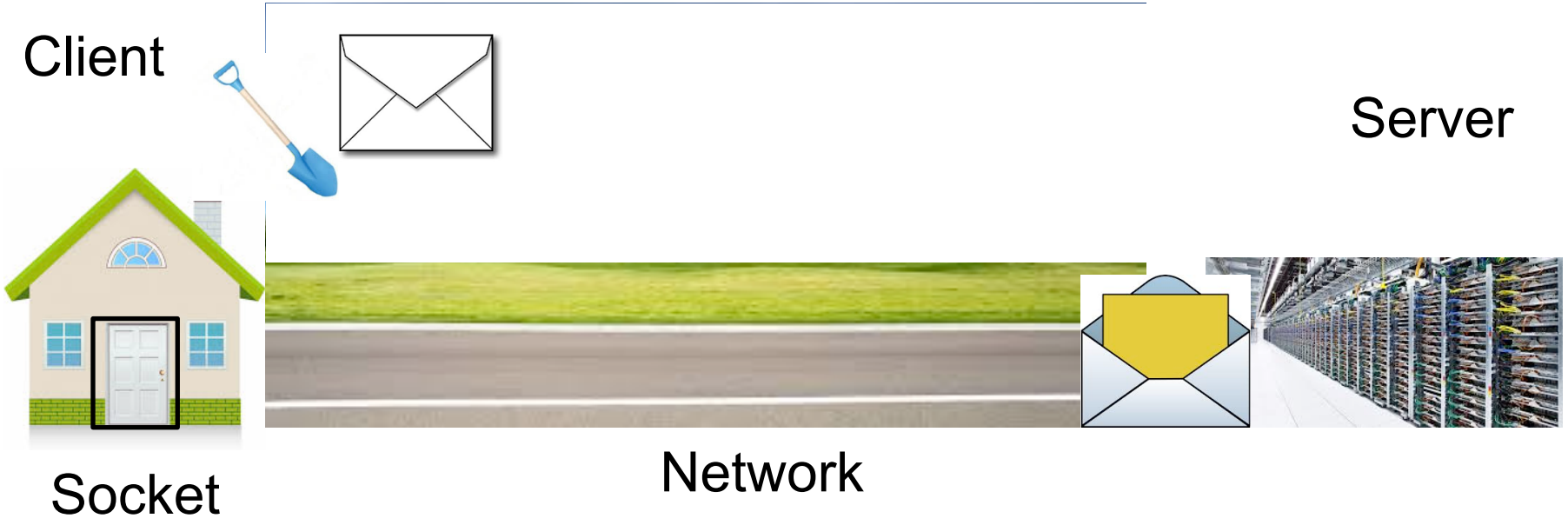
LECTURE 13: TCP RECAP, ENDIANNESS, DNS, WIRESHARK

Computer Systems and Networks

Dr. Pallipuram
(vpallipuramkrishnamani@pacific.edu)

Gist of TCP/IP Socket Programming



Netcat for managing socket

A client needs:

- server's address (always fixed)
- port number that acts as the specific door

unix> netcat -C www.google.com 80

port (door) number.
80 is for http

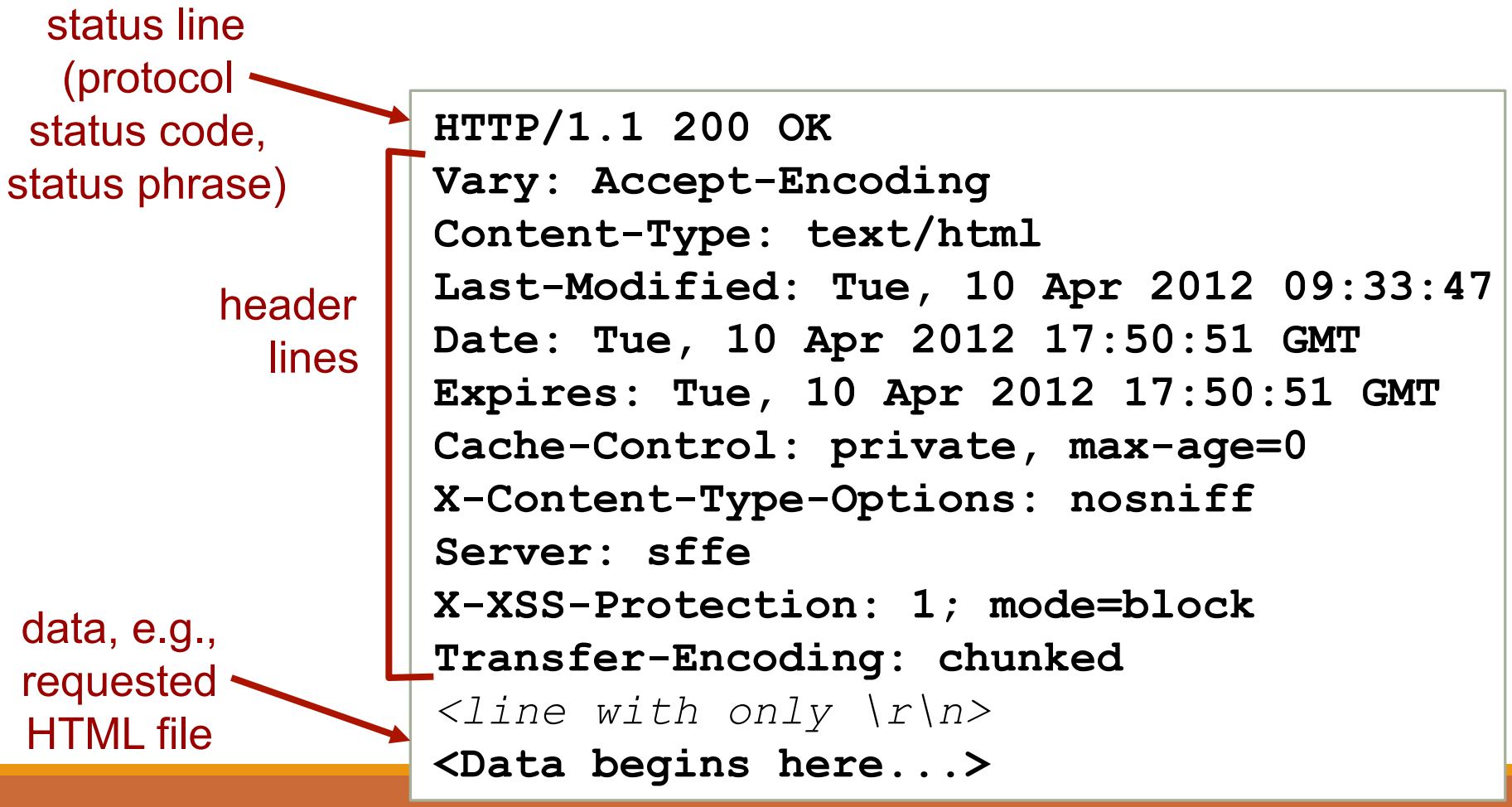
server's address

```
GET /about/ HTTP/1.1
Host: www.google.com
Connection: close

<<SERVER RESPONSE STARTS HERE>>
```

client's request to server

HTTP Response Message (Server -> Client)



Lab 8 Activities

Create message

`socket()` **`connect(host, port)`**

```
HTTP/1.1 200 OK
Vary: Accept-Encoding
Content-Type: text/html
Last-Modified: Tue, 10 Apr 2012 09:33:47
Date: Tue, 10 Apr 2012 17:50:51 GMT
Expires: Tue, 10 Apr 2012 17:50:51 GMT
Cache-Control: private, max-age=0
X-Content-Type-Options: nosniff
Server: sffe
X-XSS-Protection: 1; mode=block
Transfer-Encoding: chunked
<line with only \r\n>
<Data begins here...>
```

```
HTTP/1.1 200 OK
Vary: Accept-Encoding
Content-Type: text/html
Last-Modified: Tue, 10 Apr 2012 09:33:47
Date: Tue, 10 Apr 2012 17:50:51 GMT
Expires: Tue, 10 Apr 2012 17:50:51 GMT
Cache-Control: private, max-age=0
X-Content-Type-Options: nosniff
Server: sffe
X-XSS-Protection: 1; mode=block
Transfer-Encoding: chunked
<line with only \r\n>

<Data begins here...>
```

header

data

Lab 9 – Endianness, DNS, and Wireshark

Endianness

Endianness

In typical computer memory,
each address (location) stores one byte

If we have a one-byte integer, how is that stored in memory?

If we have a two-byte integer, how is that stored in memory?

If we have a four-byte integer, how is that stored in memory?

Endianness = Byte Ordering

Endianness Example

32-bit hexadecimal number
`0x12345678`

Composed of 4 bytes:
`0x12 0x34 0x56 0x78`
(MSB) *(LSB)*

Two possible arrangements:

Address	"Option A"	"Option B"
0	`0x12`	`0x78`
1	`0x34`	`0x56`
2	`0x56`	`0x34`
3	`0x78`	`0x12`

Endianness Example

32-bit hexadecimal number
`0x12345678`

Composed of 4 bytes:
`0x12 0x34 0x56 0x78`
(MSB) (LSB)

Two possible arrangements:

- **Big Endian**
- **Little Endian**

Address	Big Endian	Little Endian
0	`0x12` (MSB)	`0x78` (LSB)
1	`0x34`	`0x56`
2	`0x56`	`0x34`
3	`0x78`	`0x12`

Endianness

How is $DEADBEEF_{16}$ stored in little and big endian formats at address $21C_{16}$?

- Little endian
  - $21C_{16}=EF_{16}$
  - $21D_{16}=BE_{16}$
  - $21E_{16}=AD_{16}$
  - $21F_{16}=DE_{16}$
- Big endian
  - $21C_{16}=DE_{16}$
  - $21D_{16}=AD_{16}$
  - $21E_{16}=BE_{16}$
  - $21F_{16}=EF_{16}$

Do I Care?

When do I need to care that some computers are big-endian and others are little endian?

- What happens if I open big-endian data on a little-endian computer?

Endianness must be considered whenever you are **sharing data** between different computer systems

- Reading/writing data files to **disk**
- Reading/writing data files to **network**

Examples in Industry

Little-Endian Format		Big-Endian Format		Variable or Bi-Endian Format	
BMP	(Windows* & OS/2)	**PSD**	(Adobe Photoshop*)	**DXF**	(AutoCAD*)
GIF		**IMG**	(GEM Raster*)	**PS**	(Postscript*, 8 bit interpreted text, no Endian issue)
FLI	(Autodesk Animator*)	**JPEG, JPG**			
PCX	(PC Paintbrush*)	**MacPaint**		**POV**	(Persistence of Visionraytracer*)
QTM	(MAC Quicktime*)	**SGI**	(Silicon Graphics*)		
RTF	(Rich Text Format)	**Sun Raster**		**RIFF**	(WAV & AVI*)
		WPG	(WordPerfect*)	**TIFF**	
				XWD	(X Window Dump*)
Bus Protocols		**Network Protocols**		**Bus Protocols**	
Infiniband		**TCP/IP**		**GMII**	(8 bit wide bus, no Endian issue)
PCI Express		**UDP**			
PCI-32/PCI-64					
USB					

Table 2- Common file formats

Domain Name System (DNS)

IP Addresses – IPv4 and IPv6

IPv4 address: 0x8002C2F2

IPv4 addresses are 32 bits (4 bytes) long.
4 numbers separated by dots

IPv6 addresses are 128 bits (16 bytes) long

2607:f8b0:4005:802::1012

Domain Name System (DNS)

Distributed, decentralized database implemented in hierarchy of many **name servers**

One of the functions is to assign names to numerical IP addresses

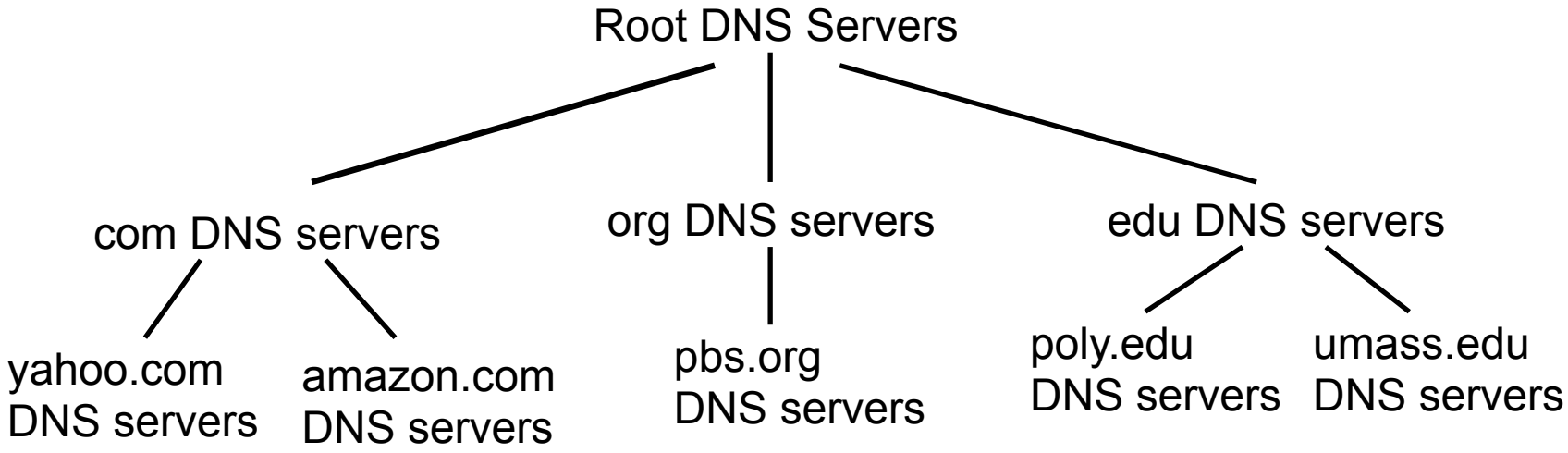
138.9.111.34 = www.pacific.edu

What's in a Name?

`engineering.pacific.edu`

- .edu is top-level domain
- "pacific" belongs to .edu
- "engineering" belongs to "pacific"
- Hierarchical! Read from right to left

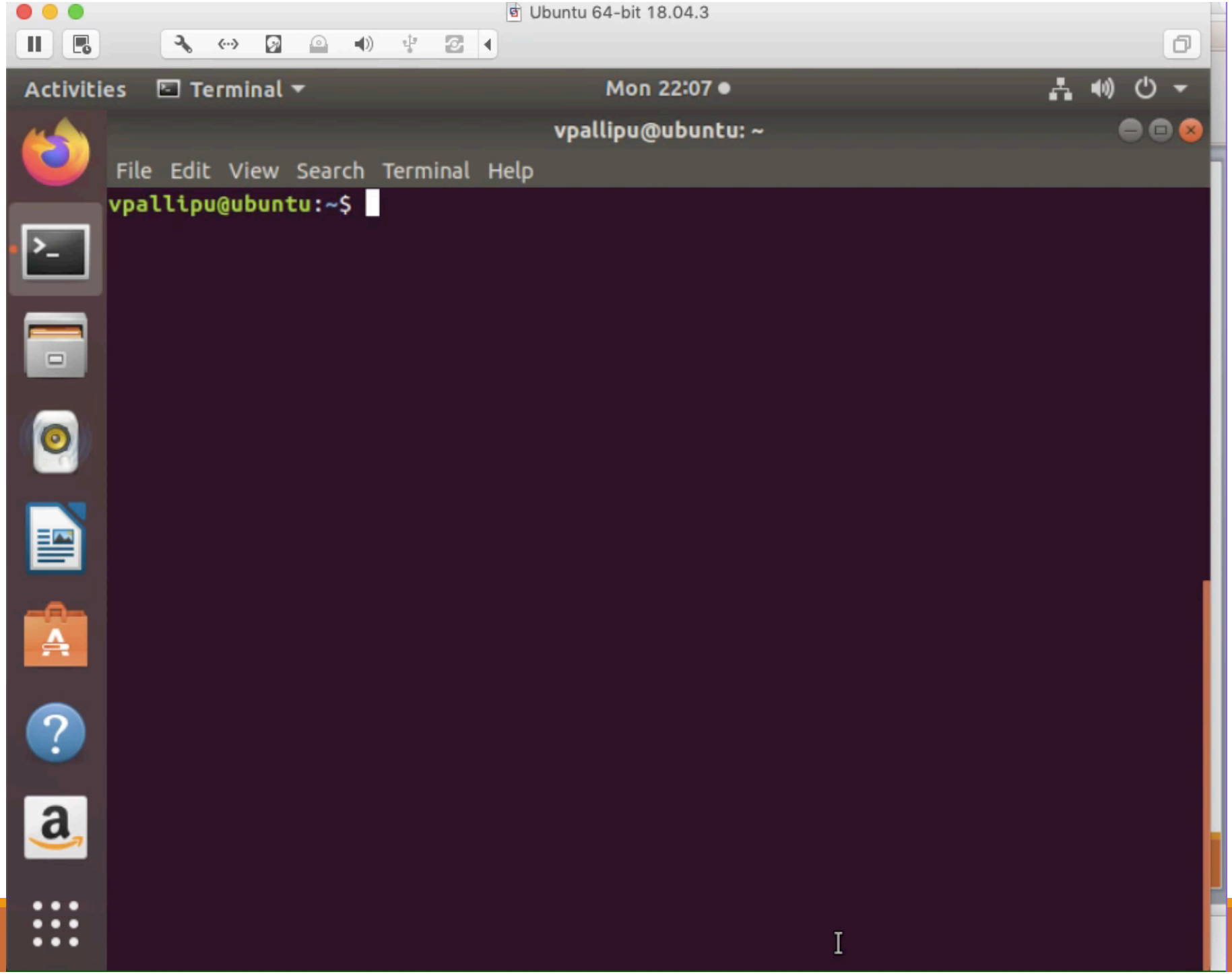
Distributed, Hierarchical Database



Client wants IP for www.amazon.com

1. Client queries a root server to find com DNS server
2. Client queries com DNS server to get amazon.com DNS server
3. Client queries amazon.com DNS server to get IP address for www.amazon.com

Let’s Play with DNS and WireShark


Ubuntu 64-bit 18.04.3
Activities
Terminal
Mon 22:07
vpallipu@ubuntu: ~
File Edit View Search Terminal Help
vpallipu@ubuntu:~$

Read the manual page for `dig`

Let us dig www.pacific.edu and monitor packets using WireShark

```
dig www.pacific.edu A @8.8.8.8 +noedns
```

Inspect WireShark for

```
dig engineering.pacific.edu A @8.8.8.8 +noedns

dig www.google.com AAAA @8.8.8.8 +noedns
```

In lab 09 folder, open a text file called: Qry_response_field.txt. Write down the key fields for the query and response messages. For socket programming you will create

- query using these fields
- receive response using these fields
- Which fields usually remain the same?
- Do you see some generality across these tests?

Next Class

- Strong attendance class
  - UDP
  - Python for UDP