Music & the Internet
MUMT301

Gabriel Vigliensoni
Schulich School of Music
McGill University

Plan

- Review of the last class and assignment
- Internet technologies
- Introduction to CSS
- In-class exercise
- Assignment #3

Review last class

- History of Internet
- History of the WWW and HTML
- History of web browsers
- Introduction to HTML
    - basic tags and elements
    - basic webpage template
- Code editor, Git, and Github
- In-class assignment

Assignment 2

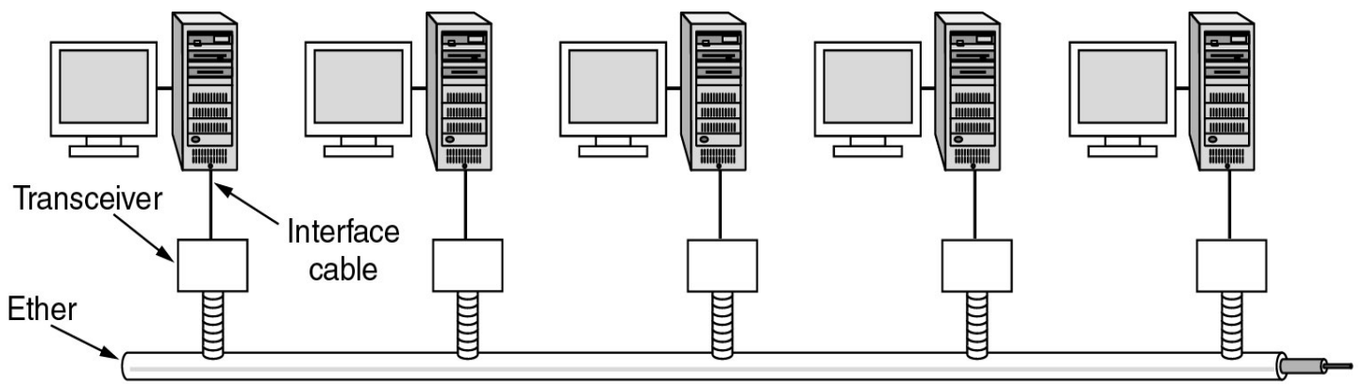
- https://mumt301.github.io

Internet technologies and protocols

- Ethernet
- TCP/IP
- OSI Model
- IP addresses
- DNS
- Ports
- DHCP
- FTP
- SSH
- HTTP

Ethernet

- Computer **networking technology**
- Specifies a **protocol and frame format** for data communication
- Invented by Bob Metcalf. First documented in internal XEROX PARC memo (1973)



Architecture of the original Ethernet (1976).

- Thick coaxial “multidrop” cable up to 2.5km long (with repeaters every 500 meters)
- Connect up to 256 computers
- 2.94Mbps line speed

http://www.inf.ed.ac.uk/teaching/courses/com/lecture-notes/lect12_notes.pdf

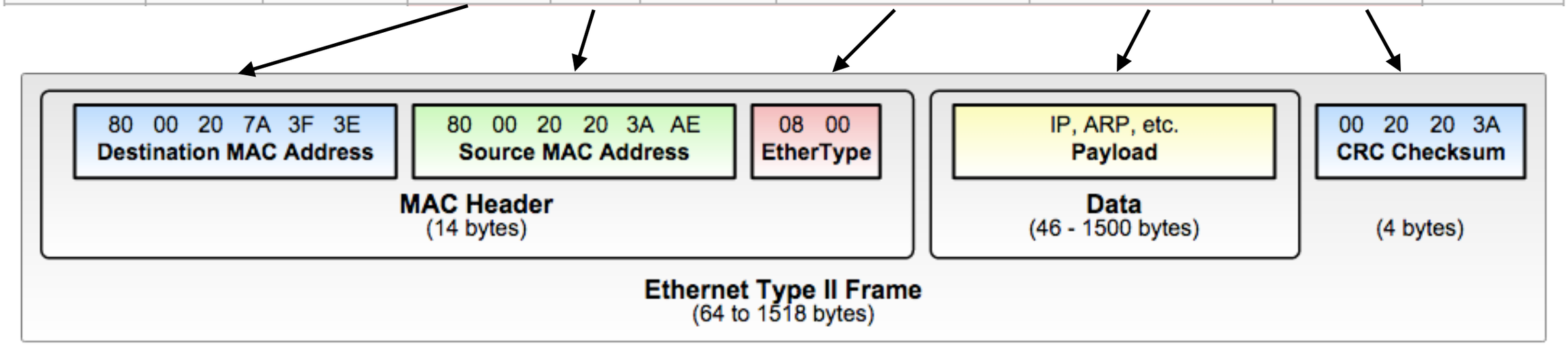
Ethernet

- Stream of **data is divided** into shorter pieces called frames.
  - Each frame contains **source and destination** addresses
  - **Damaged data** can be detected by means of error-checking data (CRC) and is **re-transmitted**
  - Each networking equipment is given a **unique identifier** comprised of 6 octets (48 bit), known as **Media Access Control (MAC) address**
- Originally based on inexpensive and ubiquitous **coaxial cable** and **twisted pair** wiring
- Standardized in IEEE 802.3 (1983) with a data rate of 10Mbps (10BASE-T) and in memo RFC 894 (1984)
- In 1995 was standardized to 100Mbps ("Fast Ethernet")
- Contemporary alternative to wired Ethernet is IEEE 802.11, also known as WiFi

Ethernet packet and frame

802.3 Ethernet packet and frame structure

Layer	Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46(42)[b]–1500 octets	4 octets	12 octets



Taken from http://en.wikipedia.org/wiki/Ethernet_frame

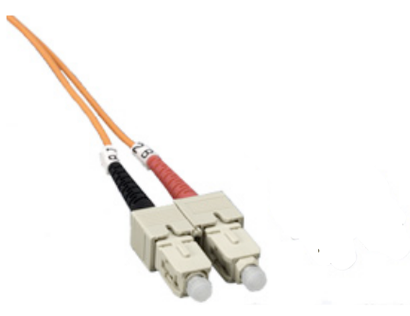
Ethernet standards

Name	Connector	Speed
10BASE-2	AUI	10 Mbps
10BASE-5	BNC	10 Mbps
10BASE-T	RJ-45	10 Mbps
100BASE-TX	RJ-45	100 Mbps
100BASE-FX	ST, SC, LC	100 Mbps
1000BASE-T	RJ-45	1 Gbps
1000BASE-X	ST, SC, LC	1 Gbps
10GBASE-X	ST, SC, LC	10 Gbps

Thin and thick coaxial

Twisted pairs

Multimode fiber

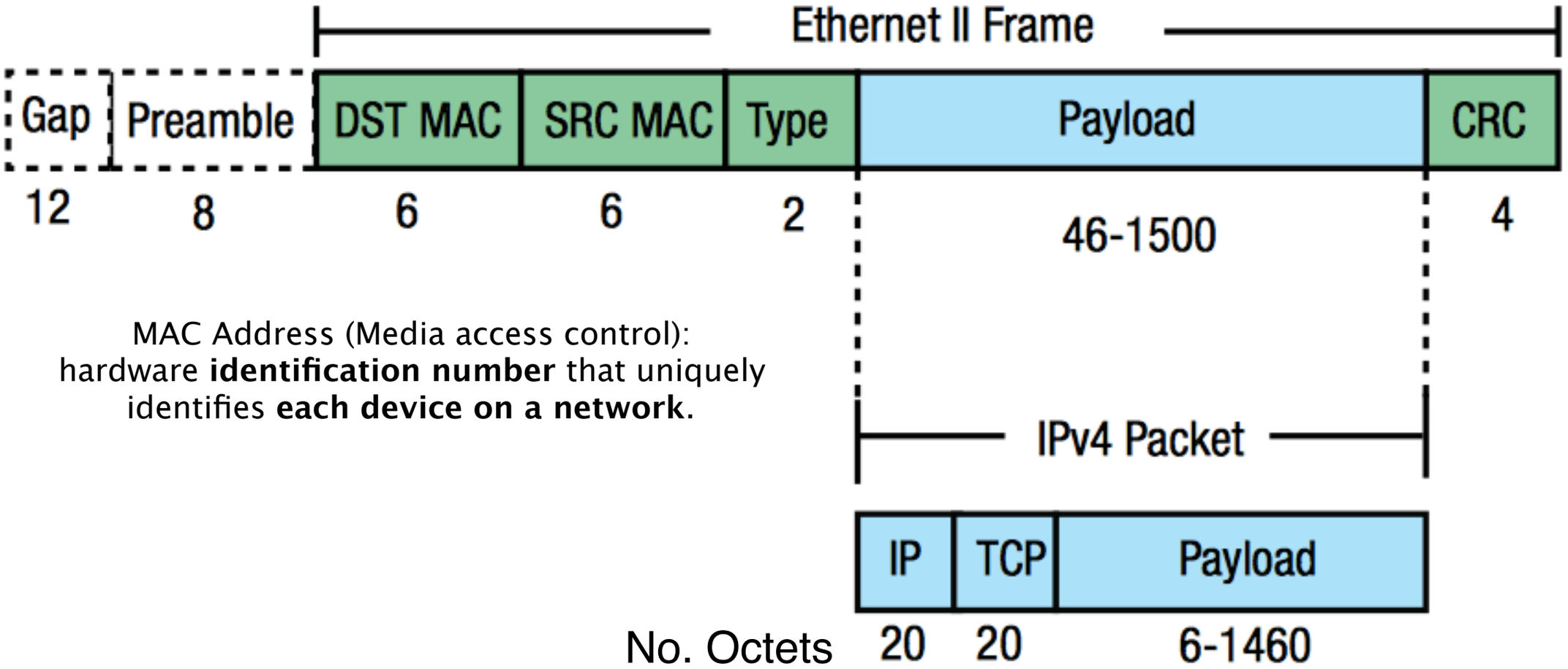


Internet Protocol Suite (TCP/IP)

- The Internet Protocol Suite (Transmission Control Protocol and Internet Protocol) **works on top of Ethernet frame**
  - provides **end-to-end connectivity**
    - **specifies** how data is packetized, addressed, transmitted, routed, and received at the destination
- Web **browsers use this protocol** when they connect to servers on the WWW
- HTTP, HTTPS, SMTP, POP3, IMAP, SSH, FTP, SFTP are **protocols encapsulated within TCP/IP**

Complete Ethernet Packet

Taken from openmicrolab.com



MAC Address (Media access control): hardware **identification number** that uniquely identifies **each device on a network**.

IP Headers

IP has the task of **delivering packets** from the source host to the destination host solely **based on the IP addresses** in the packet headers.

IPv4 (20 bytes)

IPv4 Header Format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags				Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	**96**	Source IP Address																															
16	**128**	Destination IP Address																															
20	**160**	Options (if IHL > 5)																															

4 octets = 4 8-bit byte = 32 bits -> 4.2 billion IP addresses

IPv6 (36 bytes)

Fixed header format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Version*				*Traffic Class*												*Flow Label*															
4	32	*Payload Length*																*Next Header*								*Hop Limit*							
8	**64**	*Source Address*																															
12	**96**																																
16	**128**																																
20	**160**																																
24	**192**	*Destination Address*																															
28	**224**																																
32	**256**																																
36	**288**																																

16 octets = 16 8-bit byte = 128 bit -> 340 undecillion IP addresses

Taken from wikipedia.com

TCP Headers

TCP provides an **error-checked delivery of a stream of octets** between programs running on computers connected to a LAN

TCP Header

Offsets Octet		0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																Urgent pointer (if URG set)															
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) ...																															

Taken from wikipedia.com

User Datagram Protocol (UDP) is by applications that do not require the reliability of a TCP connection and delivery validation (“handshaking”)

UDP Header

Offsets Octet		0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

Taken from wikipedia.com

MAC and IP Addresses

- IPv4 (32 bits = 4 bytes)
  - 4,294,967,296 possible IP addresses
    - More than one billion already used (!)
- IPv6 (128 bits)
  - $3.4*10^{38}$ (340 trillion trillion trillion, or 3.4 undecillion)
    - Bacterial cells on earth: $5*10^{30}$
- MAC addresses:
  - MAC-48: 2^{48} = 281,474,976,710,656 addresses (2.8×10^{14}, trillions)
    - All fish in the ocean: $3.5*10^{12}$

OSI Model

- **OSI model** defines a **framework for implementing protocols**
- **Transmitting bits from one device to another** is not enough to establish comprehensible communications
- All **information must be organized in a hierarchical manner** to convey a message
- OSI model defines
  - what a transmitting device must do to **pack up a message for transmission**
  - what the receiving device must do to **unpack the transmission to recreate the original message**
- **Ethernet-based communication protocols follow the OSI model**

OSI Model

Taken from http://www.escotal.com/osilayer.html

OSI (Open Source Interconnection) 7 Layer Model

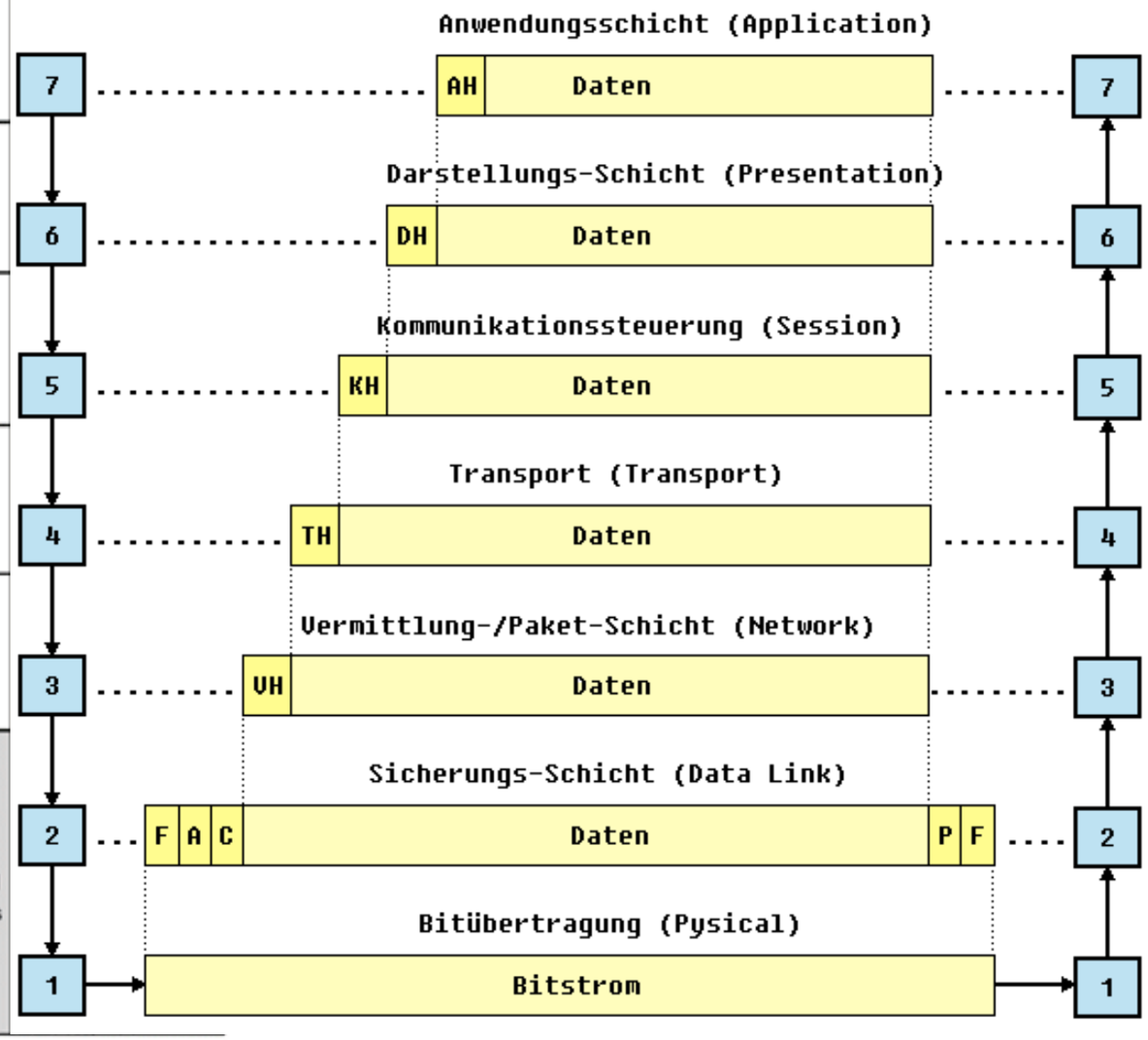
Layer	Application/Example		Central Device/ Protocols		DOD4 Model
Application (7) Serves as the window for users and application processes to access the network services.	**End User layer** Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management		**User Applications** SMTP	GATEWAY Can be used on all layers	Process
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	**Syntax layer** encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation**		JPEG/ASCII EBDIC/TIFF/GIF PICT		
Session (5) Allows session establishment between processes running on different stations.	**Synch & send to ports** (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		**Logical Ports** RPC/SQL/NFS NetBIOS names		
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	**TCP** Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	PACKET FILTERING	TCP/SPX/UDP		Host to Host
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	**Packets** ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		**Routers** IP/IPX/ICMP		Internet
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	**Frames** ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		**Switch Bridge WAP** PPP/SLIP	Land Based Layers	Network
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	**Physical structure** Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		**Hub**		

OSI Model

Taken from http://www.escotal.com/osilayer.html

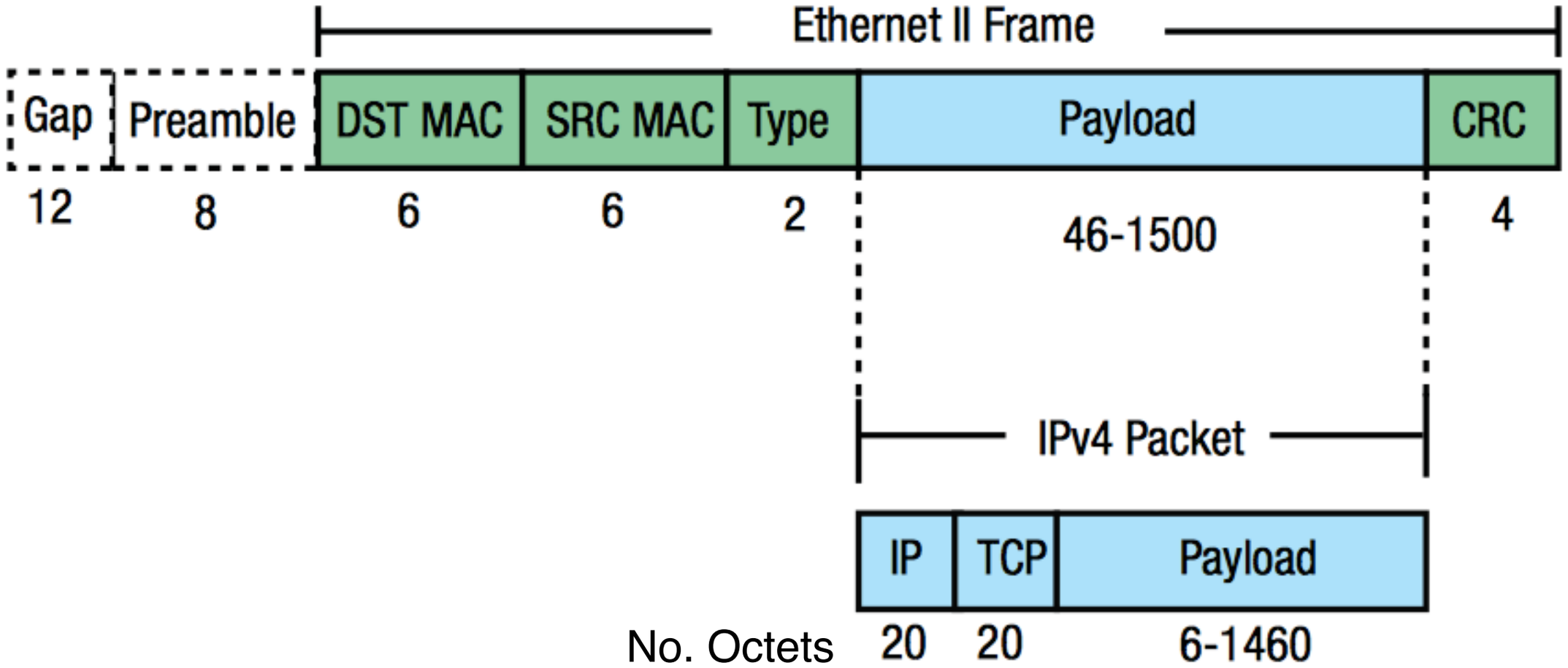
OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example		Central Device/ Protocols	DOD4 Model
Application (7) Serves as the window for users and application processes to access the network services.	**End User layer** Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management		**User Applications** SMTP	
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	**Syntax layer** encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation**		JPEG/ASCII EBDIC/TIFF/GIF PICT	
Session (5) Allows session establishment between processes running on different stations.	**Synch & send to ports** (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		**Logical Ports** RPC/SQL/NFS NetBIOS names	
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	**TCP** Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	PACKET FILTERING	TCP/SPX/UDP	
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	**Packets** ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		**Routers** IP/IPX/ICMP	
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	**Frames** ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		**Switch** **Bridge** **WAP** PPP/SLIP	Land Based Layers
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	**Physical structure** Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		**Hub**	



Complete Ethernet Packet

Taken from openmicrolab.com



Domain Name System (DNS)

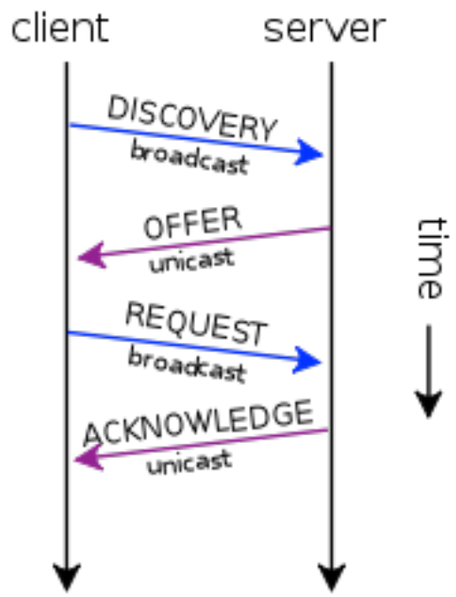
- DNS **translate domain names to IP addresses** (domain names are aliases for IP addresses)
- Defined by P. Mockapetris (1982) in RFC882
  - **Defined syntax of domain names**
    - Rightmost label conveys the **top-level domains**, e.g., .edu, .org, or .com
    - Restriction on the length of **domain names to 63 characters**, excluding the top-level domain
      - Subdivision of domain names can go up until 127 levels
      - Maximum **total length of 255 characters**
  - Domain names are also **limited to a subset of ASCII characters**, preventing many languages from representing their names and words correctly (but recent efforts to enable domain names in local languages)
- Domain name system expanded in RFC1034, RFC1035 (1987)
  - It is based on thirteen "root servers" worldwide, all but three were located in the US. Nowadays they are spread across multiple countries

Ports

- **Virtual pathways** on which Internet data travels
- **Metaphor**: If we think of **IP addresses as telephone numbers**, ports are **telephone number extensions**
- The port number added to the IP address completes the address for a communication session
- **Ports identify unique applications or processes** running on a computer and enable them to share a single physical connection in the Internet
- **All data sent to an IP address is sent on specific ports**
- Syntax: `(IP Address):(Port Number)`
- **16 bits** are dedicated for port numbers in TCP and UDP (65536 different ports)
    - Typical **system ports**: 21 (FTP), 22 (SSH), 25 (SMTP), 53 (DNS), 80 (HTTP), 194 (IRC), 443(HTTPS)
    - **Registered ports**: 5050 (Yahoo! Messenger), 9293 (Sony Playstation remote play), 19294 Google Talk, ... partial list here

DHCP

- How does the Internet find me when I move around with my laptop/tablet/phone? Or when I plug my computer to an Ethernet jack?
  - By using the **Dynamic Host Configuration Protocol**
- Protocol standardized in 1993 that uses IP
- DHCP dynamically **distributes network configuration parameters** to computers on a network, without the need of a network administrator



FTP
(File Transfer Protocol)

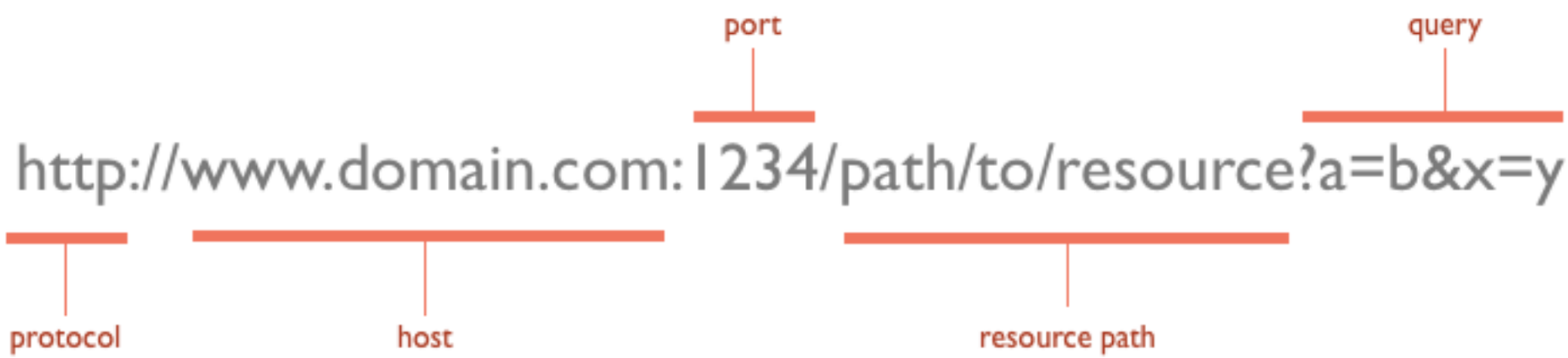
- Protocol that computers on a TCP/IP network use to **transfer files** to and from each other
  - Can be used with a **client application** or from **command line**
  - Usually works on port 21
  - Data is transmitted on **plain text**
- **SFTP** (Secure File Transfer Protocol) is similar to FTP but performs over an encrypted SSH transport
  - We used it to access the server 132.206.14.130
  - Usually works on port 22
  - Data is encrypted

SSH (Secure Shell)

- Network protocol that runs over TCP/IP
- Allows to make a **remote login** over TCP/IP network via port 22
- Provides **access to the shell of a computer**
- A shell is an interface to an operating system, for example:
    - Finder (GUI)
    - Bash (CLI)

HTTP
(Hypertext transfer protocol)

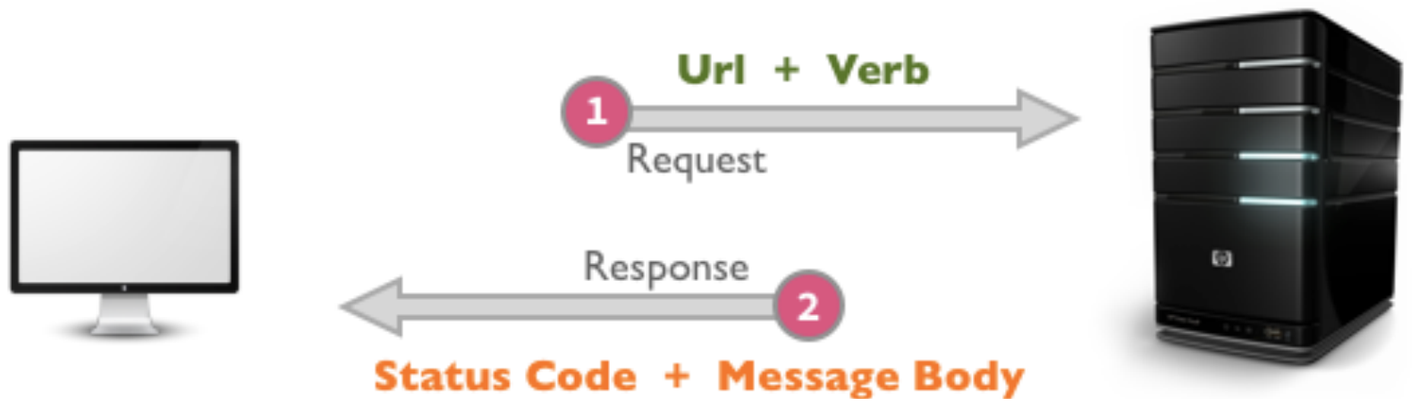
- Hypertext concept introduced by Ted Nelson (1965)
- Hypertext is **structured text that uses logical links** (hyperlinks) between nodes containing text
- HTTP is the **protocol to exchange or transfer hypertext**
- First Hypertext Transfer Protocol documented in 1991 by Tim Berners-Lee and his group@CERN
- At the heart of web communications using HTTP is the **request message**
- These request messages **are sent using Uniform Resource Locators**, known as URLs
- URLs have the following **components**:



Taken from http://code.tutsplus.com/

HTTP

- "The first version of the protocol had only one method, namely **GET**, which **would request a page from a server**. The response from the server was always an HTML page." (T. Berners-Lee)



Taken from http://code.tutsplus.com/

- However, these days there are some other HTTP "verbs" that allow us to perform other actions on resources:
  - **GET**: fetch an existing resource
  - **POST**: create a new resource
  - **PUT**: update an existing resource
  - **DELETE**: delete an existing resource

BREAK

HTML/CSS

"HTML was intended to define the content of a document, CSS defines how HTML elements are to be displayed." - http://www.w3schools.com/css/css_intro.asp

An HTML document has two main parts:
head and body.

HEAD - Internal (hidden) information, metadata

- Title http://www.w3schools.com/tags/tag_title.asp
- Base href - setting up your base reference link
- Link to favicon - http://www.favicon.cc/
- Meta tags (keywords, description, copyright, publisher-email, author)
- Styles/link stylesheet
- Javascript

BODY - Perceived (rendered) information

- Content
- Footer

HTML5 SEMANTIC ELEMENTS

A semantic element **clearly describes its meaning** to both the browser and the developer

Examples of semantic elements: `<form>`, `<table>`, and `<img>`
These elements clearly define their content

Non-semantic elements **tell nothing about their content**

Examples of non-semantic elements: `<div>` and `<span>`

CSS

Cascading Style Sheets

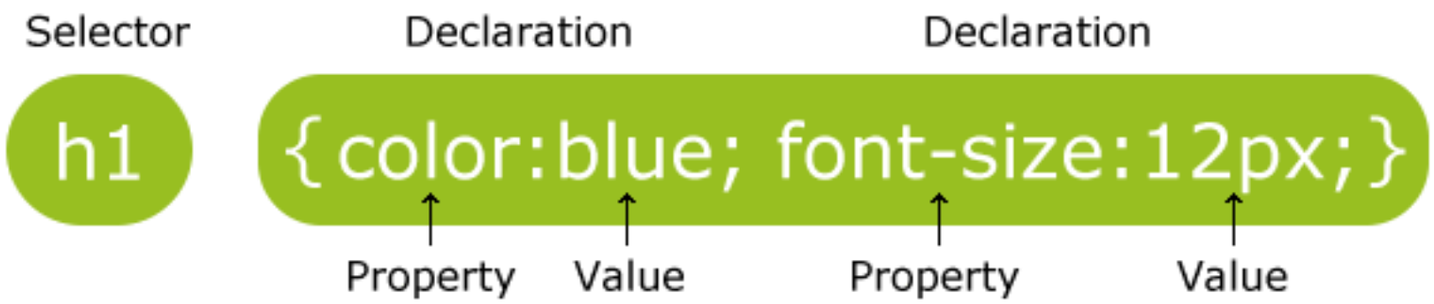
CSS defines how the HTML elements will be displayed!

CSS is designed primarily to enable the separation of document content from document presentation

http://www.w3schools.com/css/demo_default.htm

CSS SYNTAX

A CSS rule has two main parts: a selector, and one or more declarations



Example:

```
p {
font-family: arial, helvetica, sans-serif;
font-size: 12px;
color: black;
line-height: auto;
}
```

SOME SELECTORS

1. *

```
* {
 margin: 0;
 padding: 0;
}
```

2. #X

```
#container {
   width: 960px;
   margin: auto;
}
```

3. .X

```
.error {
  color: red;
}
```

4. X Y

```
li a {
  text-decoration:
none;
}
```

http://code.tutsplus.com/tutorials/the-30-css-selectors-you-must-memorize–net-16048

Three Ways to Insert CSS

- Inline style
- Internal style sheet
- External style sheet

Inline styles

```
<h1 style="color:blue; margin-left:30px;">This is a heading.</h1>
```

Internal style sheet

```
<head>
  <style>
    body {
       background-color: blue;
    }
    h1 {
       color: red;
       margin-left: 40px;
    }
  </style>
</head>
```

External style sheet

```
<head>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

CSS Styles

Position
Borders
Backgrounds
Gradients
Text Effects
Fonts
2D Transforms
3D Transforms
Transitions
Animations
Multiple Columns
User Interface

grids

In-class demo

- Style the page for a band that I like
- https://mumt301.github.io/code/18plus.html

Today's class

- Internet technologies
- Introduction to CSS
- In-class demo
- Assignment 3