

# Systems in 60 Minutes

Raymond Xu  
raymondxu.io



# Roadmap for this talk

What happens when you visit google.com?

- Horizontally
- Vertically

# Roadmap for this talk

What happens when you visit google.com?

- Horizontally
- Vertically

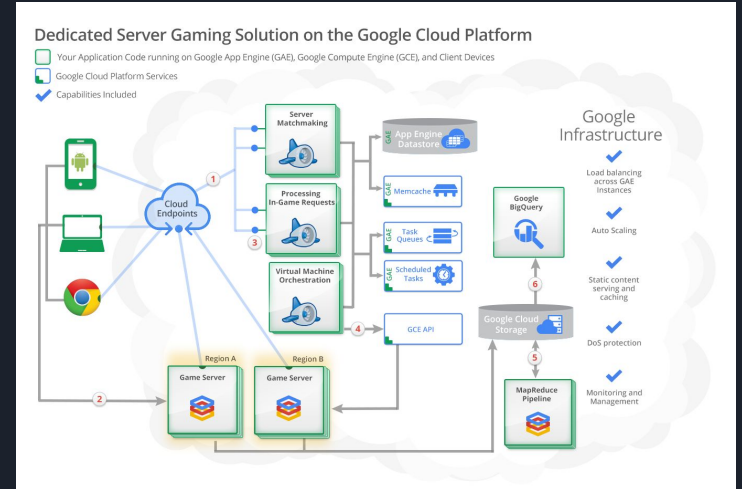
What's the best way to learn new systems?

# Roadmap for this talk

What happens when you visit google.com?

- Horizontally
- Vertically

What's the best way to learn new systems?



Source: Google Cloud Platform Solutions

# Roadmap for this talk

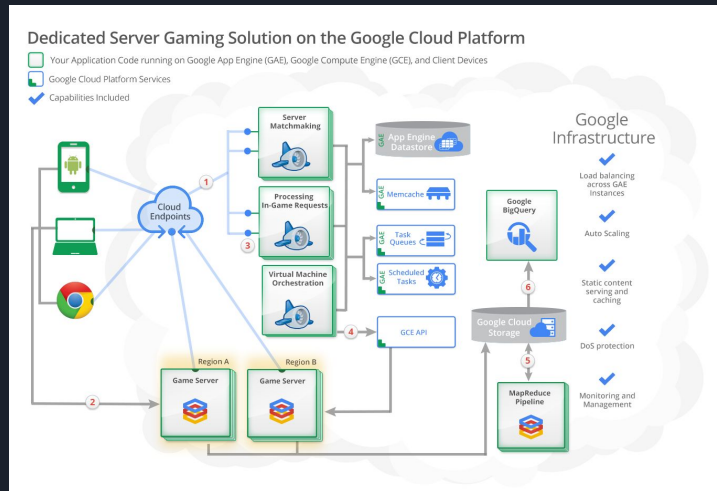
What happens when you visit google.com?

- Horizontally
- Vertically

What's the best way to learn new systems?

*"Envoy is a high performance C++ distributed proxy designed for single services and applications, as well as a communication bus and "universal data plane" designed for large microservice "service mesh" architectures."*

<https://www.envoyproxy.io/>



Source: Google Cloud Platform Solutions

# Roadmap for this talk

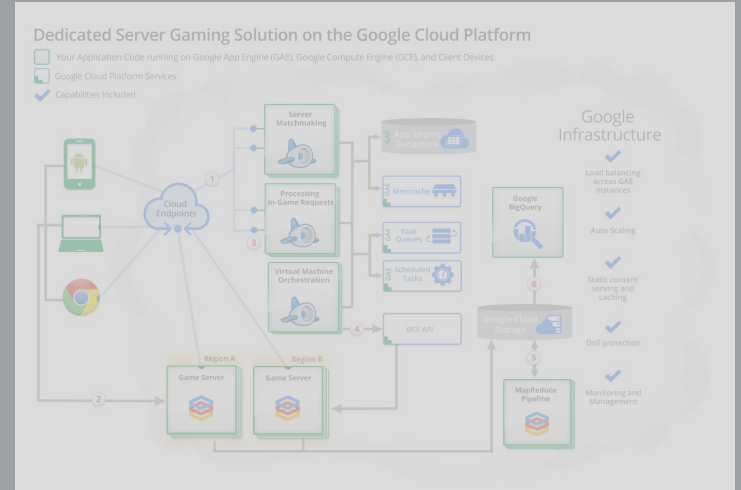
What happens when you visit google.com?

- Horizontally
- Vertically

What's the best way to learn new systems?

*"Envoy is a high performance C++ distributed proxy designed for single services and applications, as well as a communication bus and "universal data plane" designed for large microservice "service mesh" architectures."*

<https://www.envoyproxy.io/>



Source: Google Cloud Platform Solutions

# Horizontal

Your Computer → Networks → Google → Networks → Your Computer

# Vertical

Your Computer

Google

Networks



# Vertical

## Your Computer

- Applications
- Operating System
- Hardware
- Physics

## Google

## Networks

# Vertical

## Computer

- Applications
- Operating System
- Hardware
- Physics

## Google

## Networks

# Vertical

## Computer

- Applications
- Operating System
- Hardware
- Physics

## Google

- Applications
- Operating System
- Hardware
- Physics

## Networks

# Vertical

## Computer

- Applications
- Operating System
- Hardware
- Physics

## Google

Computer

## Networks

# Vertical

## Computer

- Applications
- Operating System
- Hardware
- Physics

## Google

Computer

x 1,000,000

## Networks

# Vertical

## Computer

- Applications
- Operating System
- Hardware
- Physics

## Google

Computer

x 1,000,000

+ Distributed Systems

## Networks

# Vertical

## Computer

- Applications
- Operating System
- Hardware
- Physics

## Google

Computer

x 1,000,000

+ Distributed Systems

## Networks

- Protocols
- Infrastructure
- Physics

Computers store and manipulate data.

Networks transfer data.



**Computers store and manipulate data.**

Networks transfer data.

# Data

Data can be anything

- Text
- Images
- Videos

All data can be represented one-dimensionally with 1s and 0s (bits)

How do you interpret a sequence of bits?

- Character encodings

# Data

## PPM example (a simple image file format)

```
01010000 00110011 00001010 00110100 00100000 00110100 00001010 00110001 00110101 00001010 00100000 00110000
00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000 00100000 00110000 00100000
00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000 00100000 00110000 00100000 00100000
00110000 00100000 00100000 00110000 00100000 00100000 00100000 00110001 00110101 00100000 00100000 00110000
00100000 00110001 00110101 00001010 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000
00100000 00100000 00100000 00100000 00110000 00100000 00110001 00110101 00100000 00100000 00110111 00100000
00100000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000
00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000 00001010 00100000 00110000
00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000 00100000 00110000 00100000
00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000 00100000 00110000 00100000 00110001
00110101 00100000 00100000 00110111 00100000 00100000 00100000 00100000 00110000 00100000 00100000 00110000
00100000 00100000 00110000 00001010 00110001 00110101 00100000 00100000 00110000 00100000 00110001 00110101
00100000 00100000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000 00100000
00100000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00110000 00100000 00100000 00100000
00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000
```

# Data

## PPM example (a simple image file format)

```
01010000 00110011 00001010 00110100 00100000 00110100 00001010 00110001 00110101 00001010 00100000 00110000
00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000 00100000 00110000 00100000
00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000 00100000 00110000 00100000 00100000
00110000 00100000 00100000 00110000 00100000 00100000 00100000 00110001 00110101 00100000 00100000 00110000
00100000 00110001 00110101 00001010 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000
00100000 00100000 00100000 00100000 00110000 00100000 00110001 00110101 00100000 00100000 00110111 00100000
00100000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000
00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000 00001010 00100000 00110000
00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000 00100000 00110000 00100000
00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000 00100000 00110000 00100000 00110001
00110101 00100000 00100000 00110111 00100000 00100000 00100000 00100000 00110000 00100000 00100000 00110000
00100000 00100000 00110000 00001010 00110001 00110101 00100000 00100000 00110000 00100000 00110001 00110101
00100000 00100000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000 00100000
00100000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00110000 00100000 00100000 00100000
00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00110000 00100000 00100000 00100000
```

ASCII Character encoding...

# Data

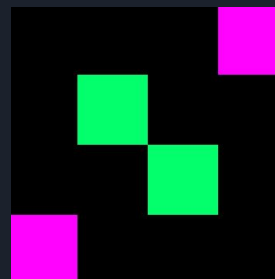
PPM example (a simple image file format)

```
P3
4 4
15
0 0 0 0 0 0 0 0 0 15 0 15
0 0 0 0 15 7 0 0 0 0 0 0
0 0 0 0 0 0 0 15 7 0 0 0
15 0 15 0 0 0 0 0 0 0 0 0
```

# Data

PPM example (a simple image file format)

```
P3
4 4
15
0 0 0 0 0 0 0 0 15 0 15
0 0 0 0 15 7 0 0 0 0 0 0
0 0 0 0 0 0 0 15 7 0 0 0
15 0 15 0 0 0 0 0 0 0 0 0
```

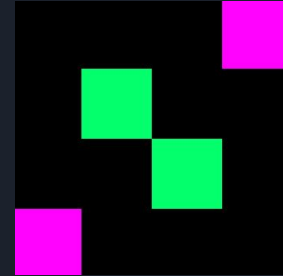


# Data

## PPM example (a simple image file format)

```
P3
4 4
15
0 0 0 0 0 0 0 0 15 0 15
0 0 0 0 15 7 0 0 0 0 0 0
0 0 0 0 0 0 0 15 7 0 0 0
15 0 15 0 0 0 0 0 0 0 0 0
```

File type  
Width Height  
Maximum color value  
R G B ...

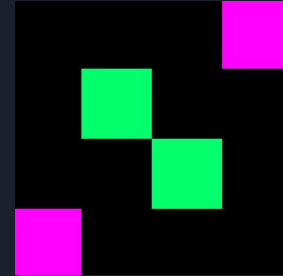


# Data

## PPM example (a simple image file format)

```
P3
4 4
15
0 0 0 0 0 0 0 0 15 0 15
0 0 0 0 15 7 0 0 0 0 0 0
0 0 0 0 0 0 0 15 7 0 0 0
15 0 15 0 0 0 0 0 0 0 0 0
```

File type  
Width Height  
Maximum color value  
R G B ...





# Storing Data

## Memory

- Voltage through a circuit indicates 1 or 0
- So with enough circuits we can represent arbitrary data
- Power loss = data loss

# Storing Data

## Hard Drive (Disk)

- Metal can be precisely magnetized/demagnetized
- Direction of magnetism indicates 1 or 0
- Platters can store data
- Does not need to maintain power

# Manipulating Data

## CPU

- Cleverly arranged circuits can model math
- Take some input state and deterministically produce some meaningful output state
- Defined instructions (code?!) allow us to tell the CPU what to do

# Operating System

Software that manages a computer's hardware and runs programs

Intermediary layer between user and hardware, and applications and hardware

## Kernel

- The most important part of an OS
- A program that controls the hardware
  - CPU, Memory, Devices

# Applications

Built on top of the operating system API

Written in high level programming languages (typically)

Examples: Microsoft Office, Web Browsers, Compilers, Database Management Systems

# Databases

Data stored on disk in a specific structured manner (think file formats)

A Database Management System (DBMS) sits on top of the database and allows other applications to interact with the data

# A Primitive Database

```
db_set () {  
    echo "$1,$2" >> database  
}  
  
db_get () {  
    grep "^$1," database | sed -e "s/^$1,/" | tail -n 1  
}
```

*Source: Martin Kleppmann, Designing Data-Intensive Applications*

# A Primitive Database

```
db_set () {  
    echo "$1,$2" >> database  
}  
  
db_get () {  
    grep "^$1," database | sed -e "s/^$1,/" | tail -n 1  
}
```

```
$ db_set 123 raymondxu.io
```

```
$ db_set 456 pokerchips.io
```

```
$ db_get 123  
raymondxu.io
```

```
$ cat database  
123,raymondxu.io  
456,pokerchips.io
```

*Source: Martin Kleppmann, Designing Data-Intensive Applications*



# Databases

Real databases use data structures and algorithms for efficiency

Data modeling is representing the data requirements of an application in a useful and efficient manner

# Layering

# Layering

Building Blocks: Transistors

# Layering

Storing Data: Memory, Disk

Building Blocks: Transistors

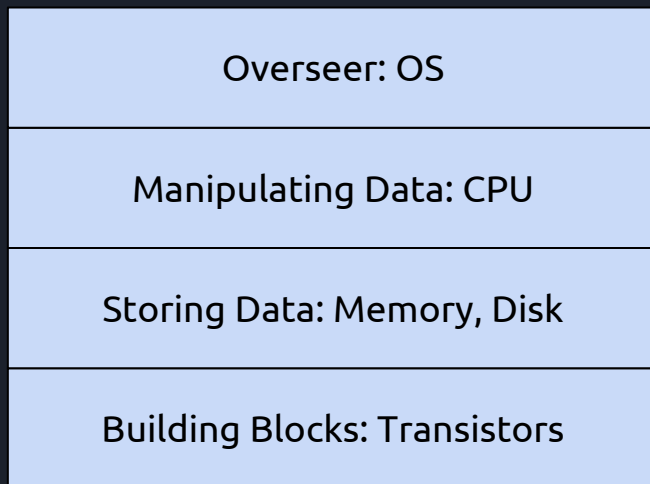
# Layering

Manipulating Data: CPU

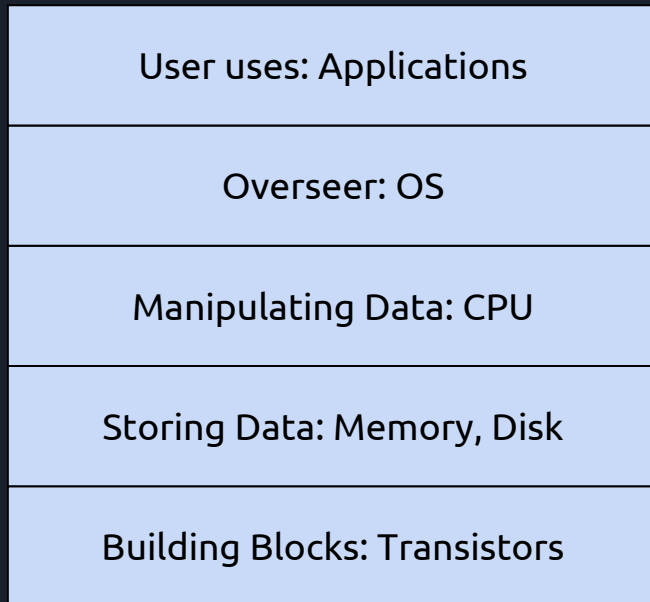
Storing Data: Memory, Disk

Building Blocks: Transistors

# Layering



# Layering



**Computers store and manipulate data.**

Networks transfer data.



Computers store and manipulate data.

**Networks transfer data.**

# Networks

So we can store and manipulate data within one computer

# Networks

So we can store and manipulate data within one computer

How do we transfer data between computers?

# Networks

So we can store and manipulate data within one computer

How do we transfer data between computers?

- First let's prepare the data for transmission...

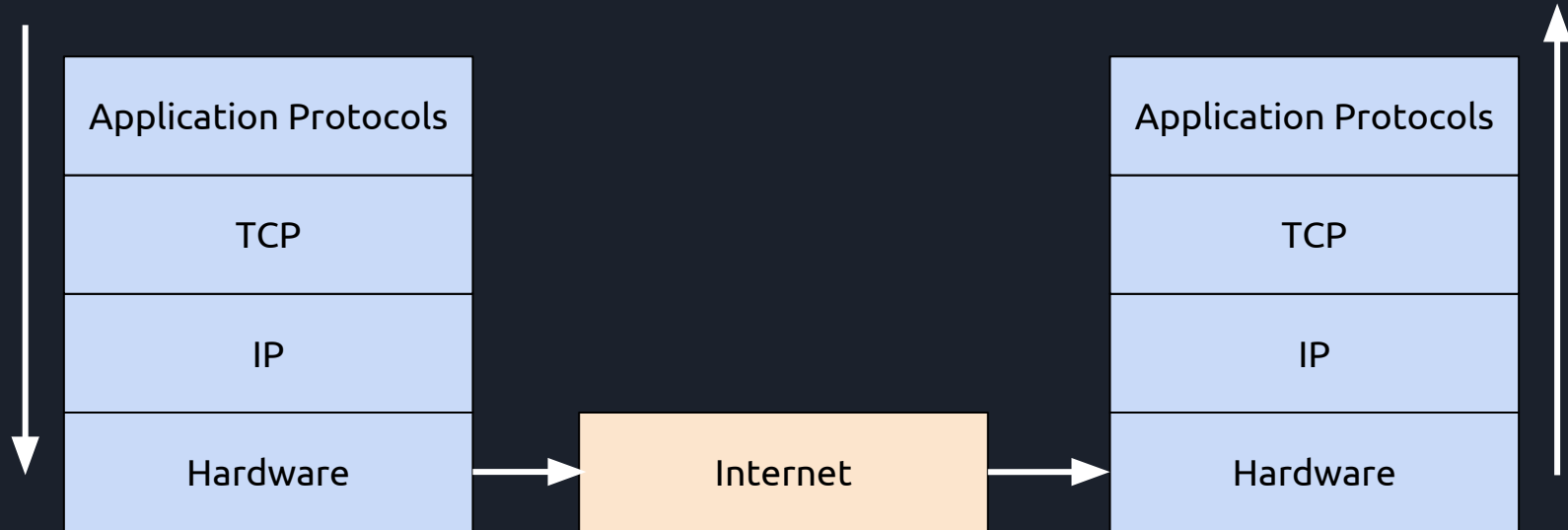
# Protocols

Standardized procedures for communication

Can't just send your raw data into a network

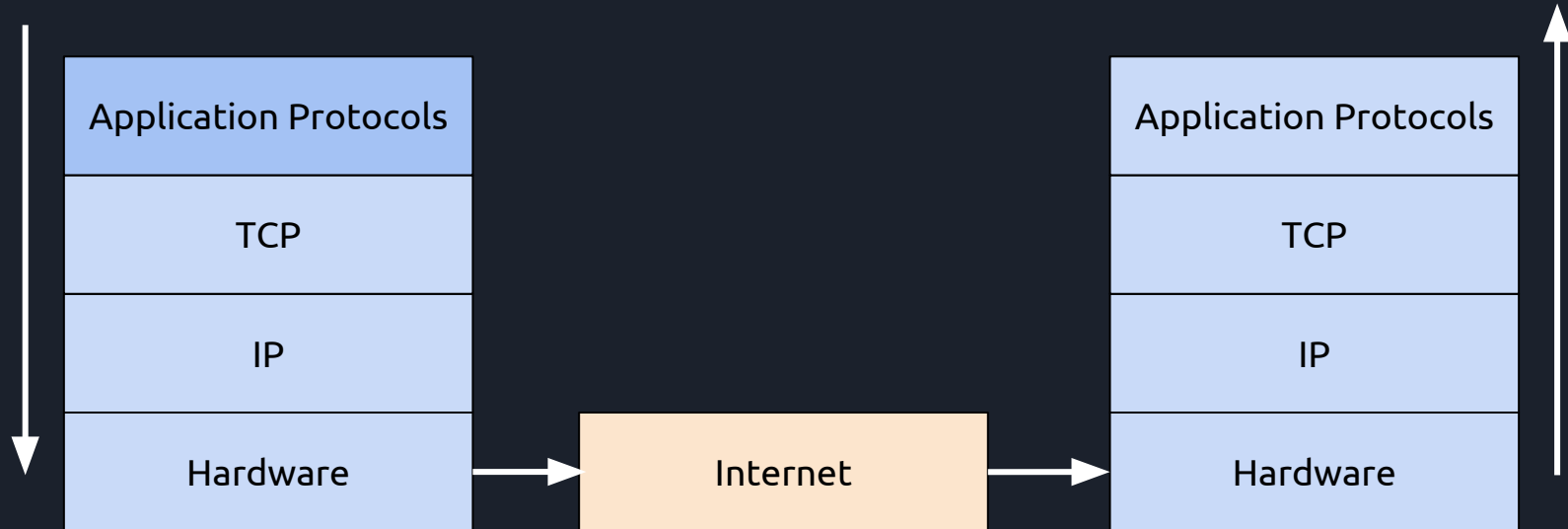
- Where does it go?
- What is the data?
- What if some of the data is lost?
- How do you know it was sent?

# Protocol Stack



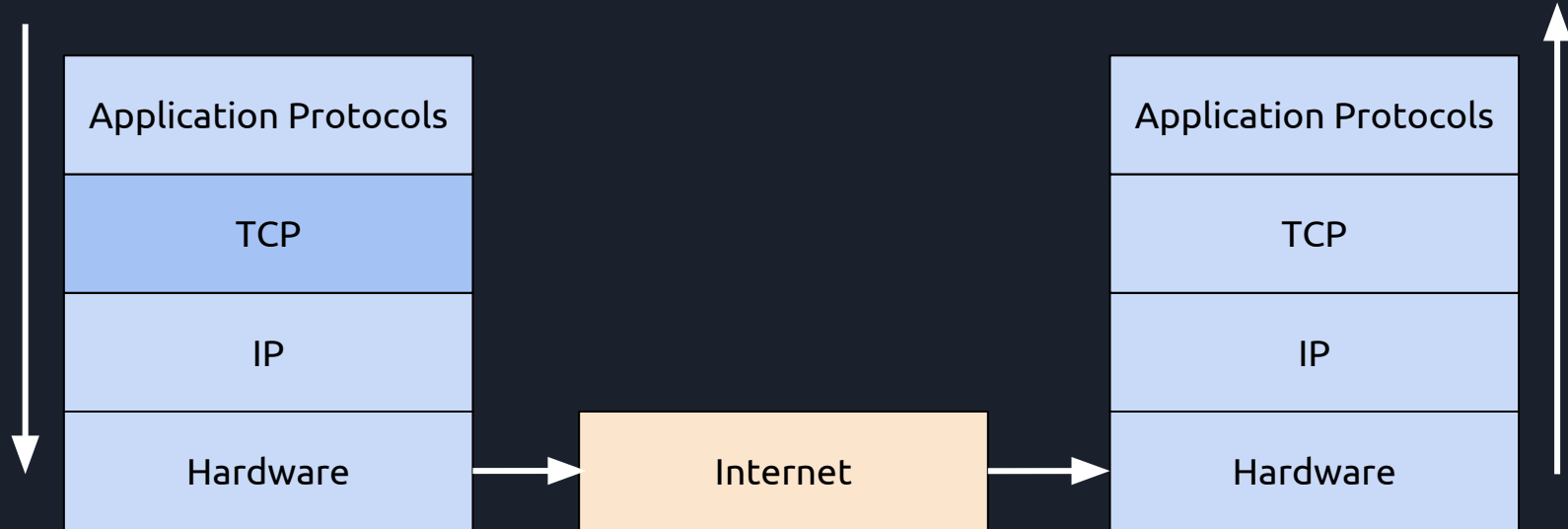
Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

# Protocol Stack



Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

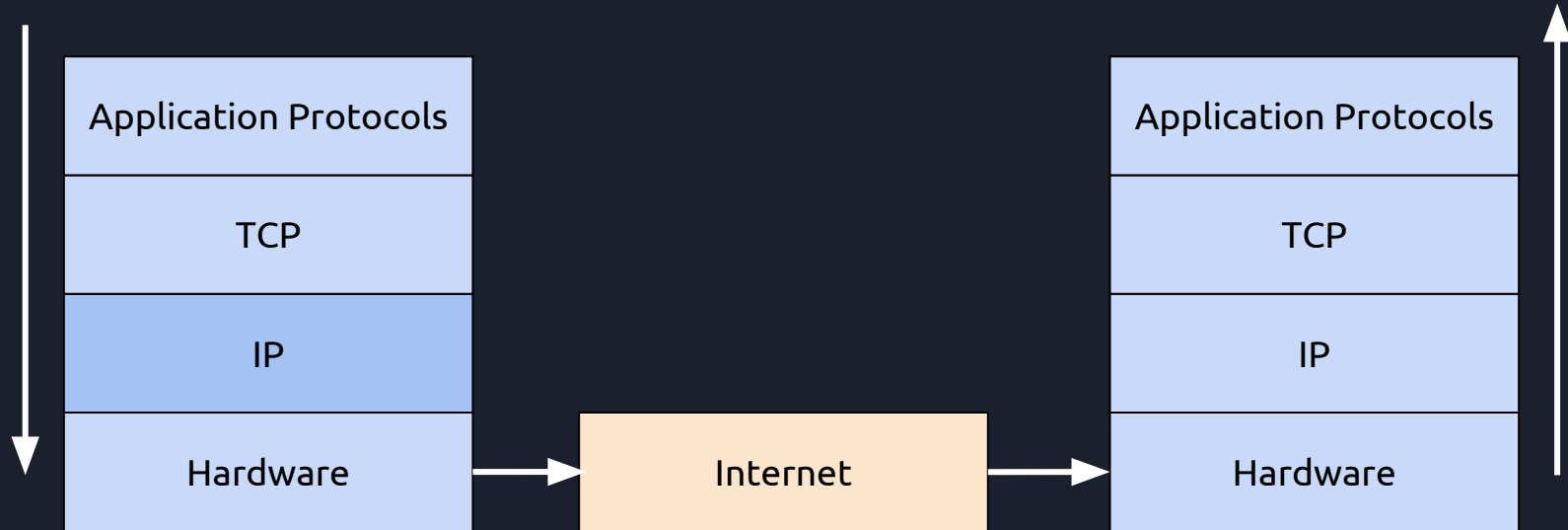
# Protocol Stack



Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

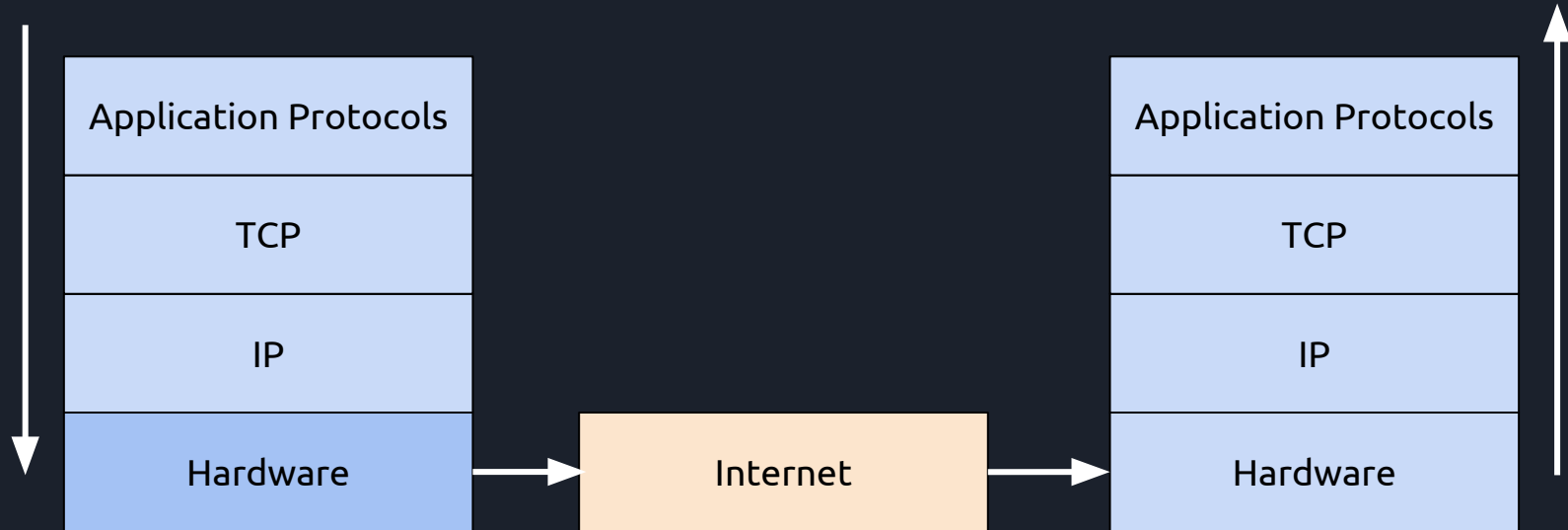


# Protocol Stack



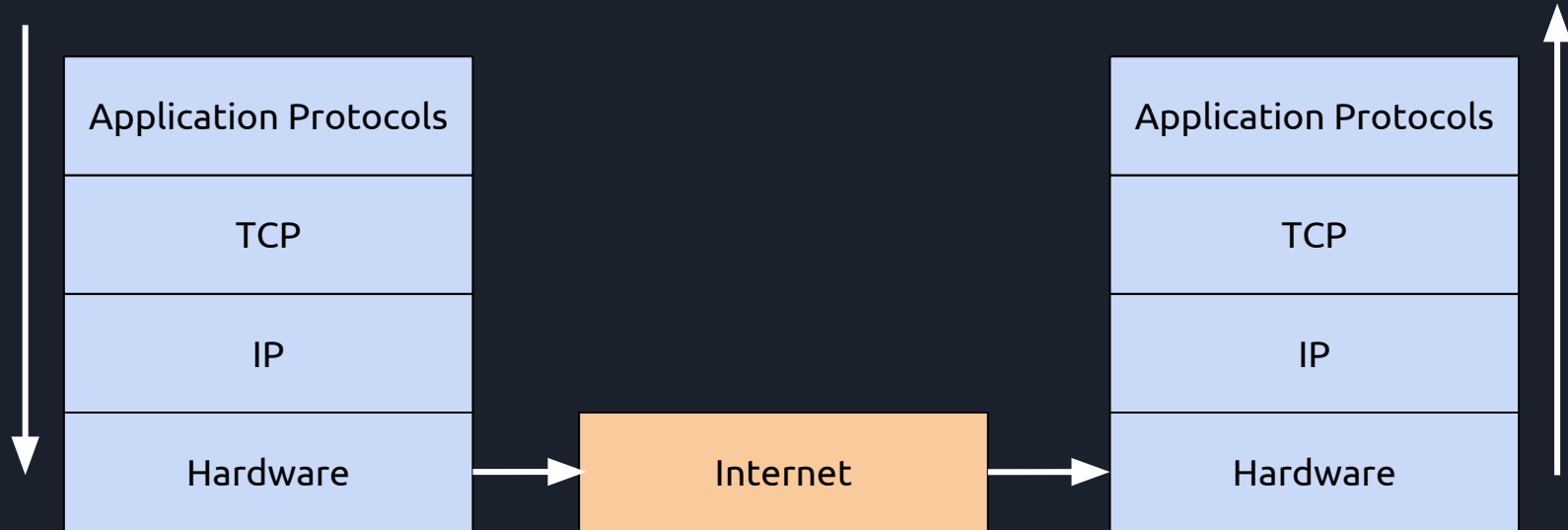
Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

# Protocol Stack



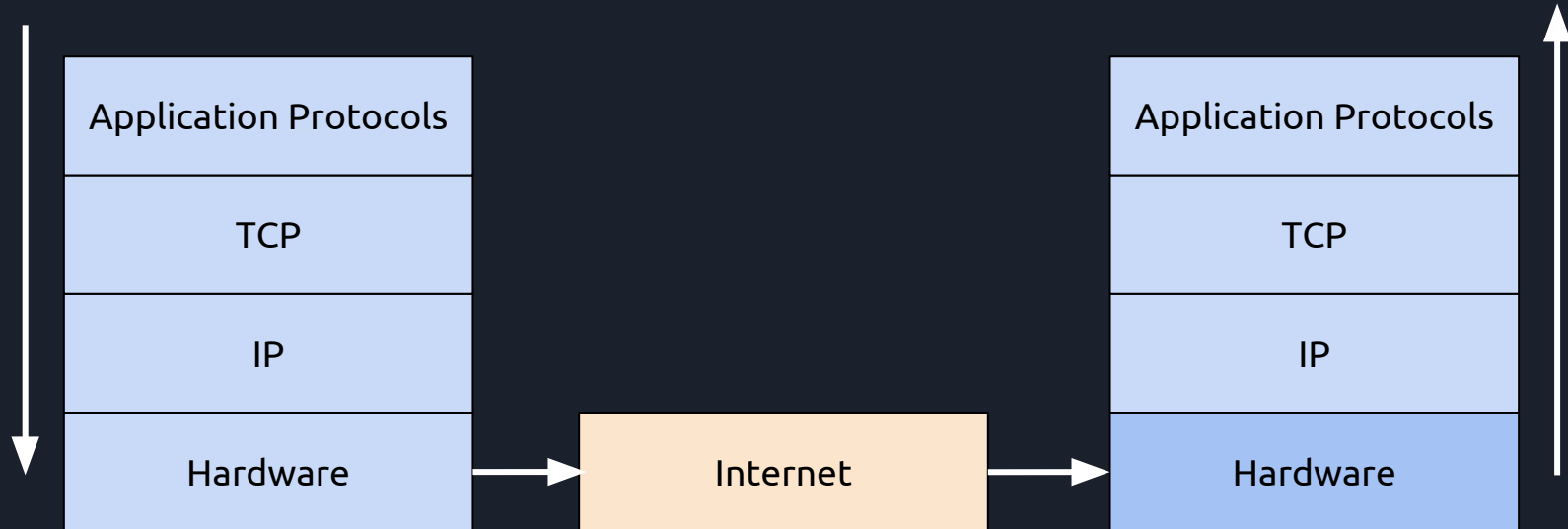
Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

# Protocol Stack



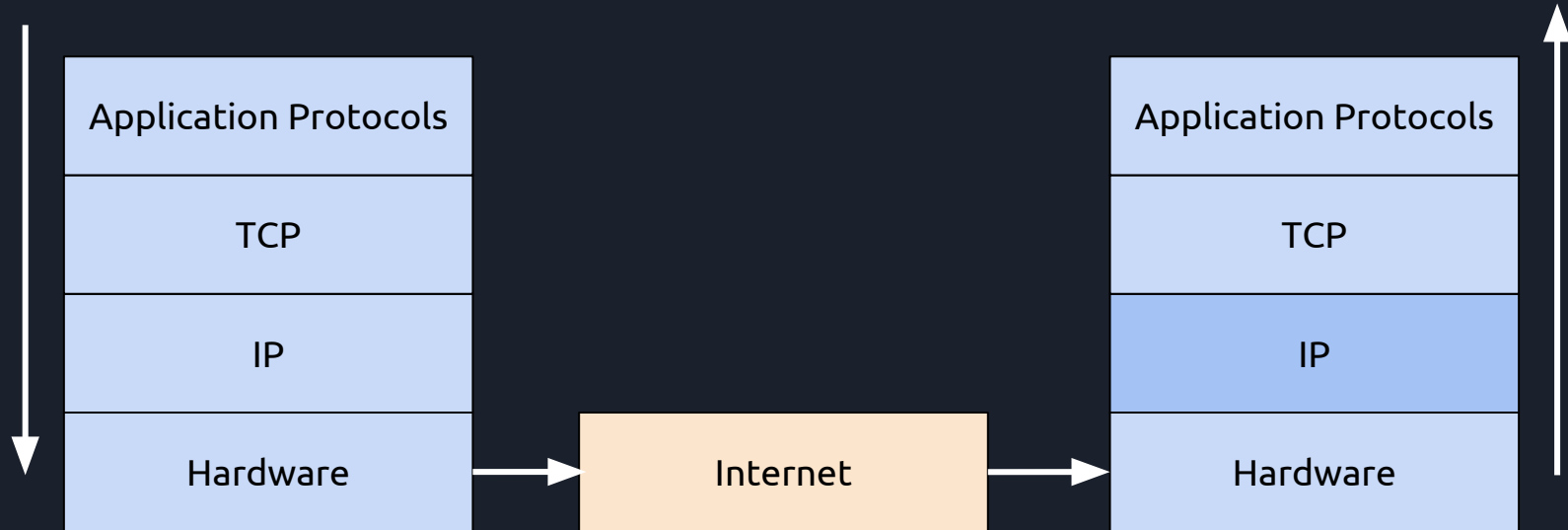
Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

# Protocol Stack



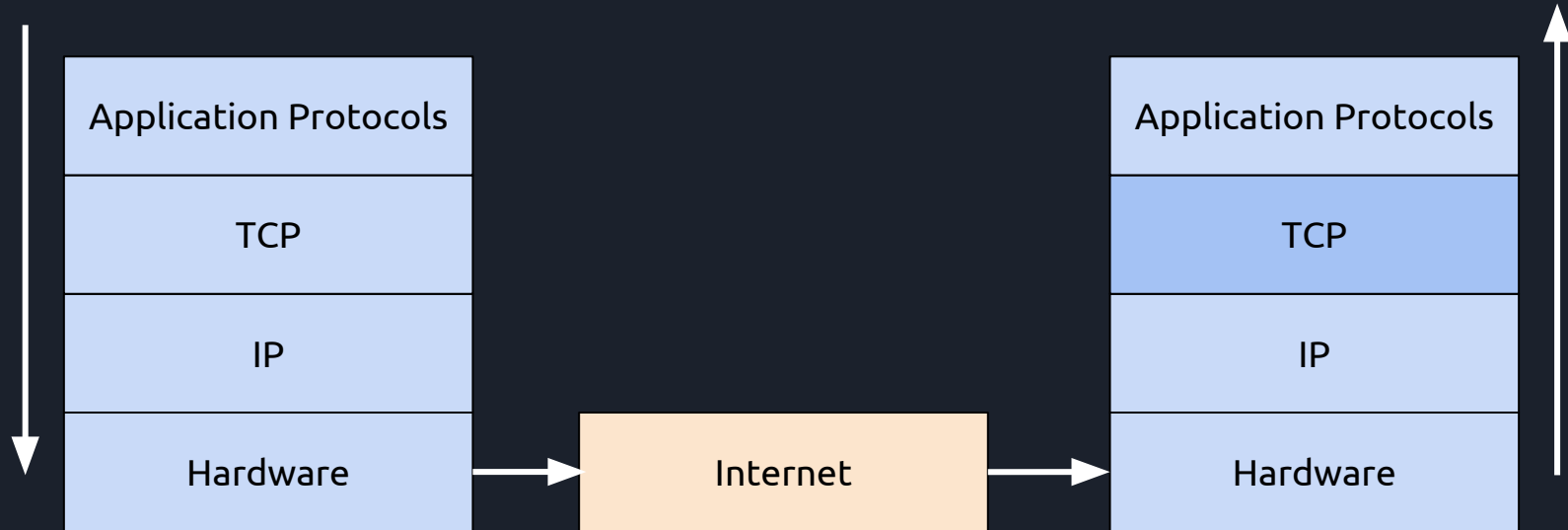
Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

# Protocol Stack



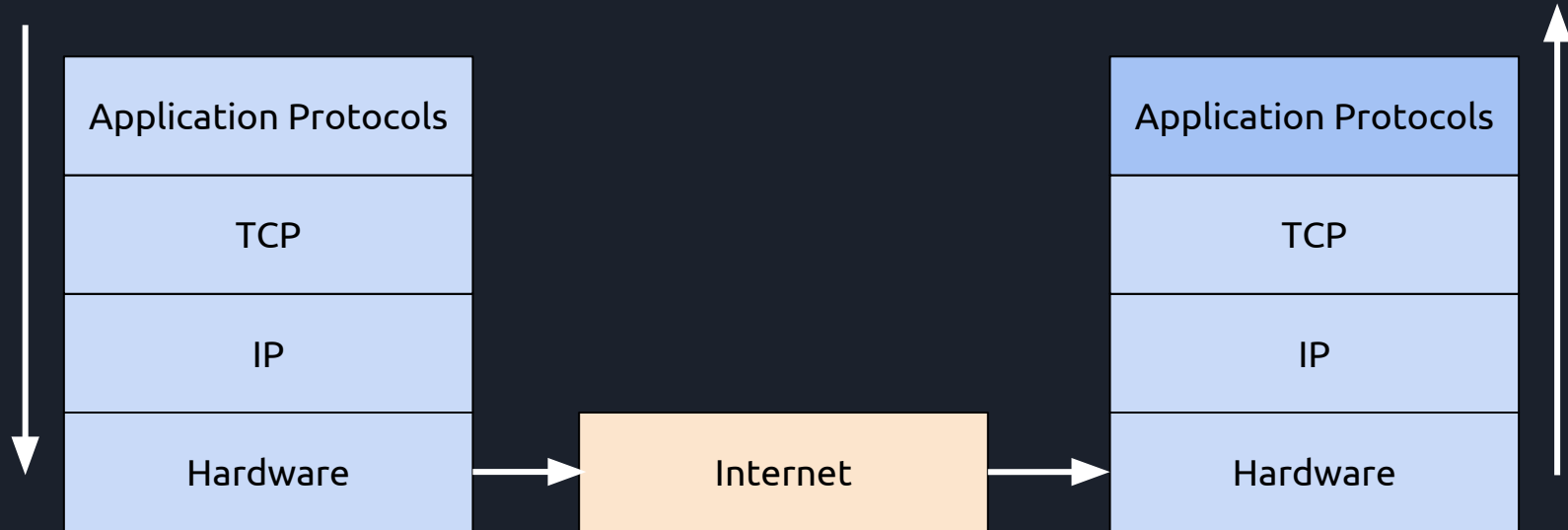
Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

# Protocol Stack



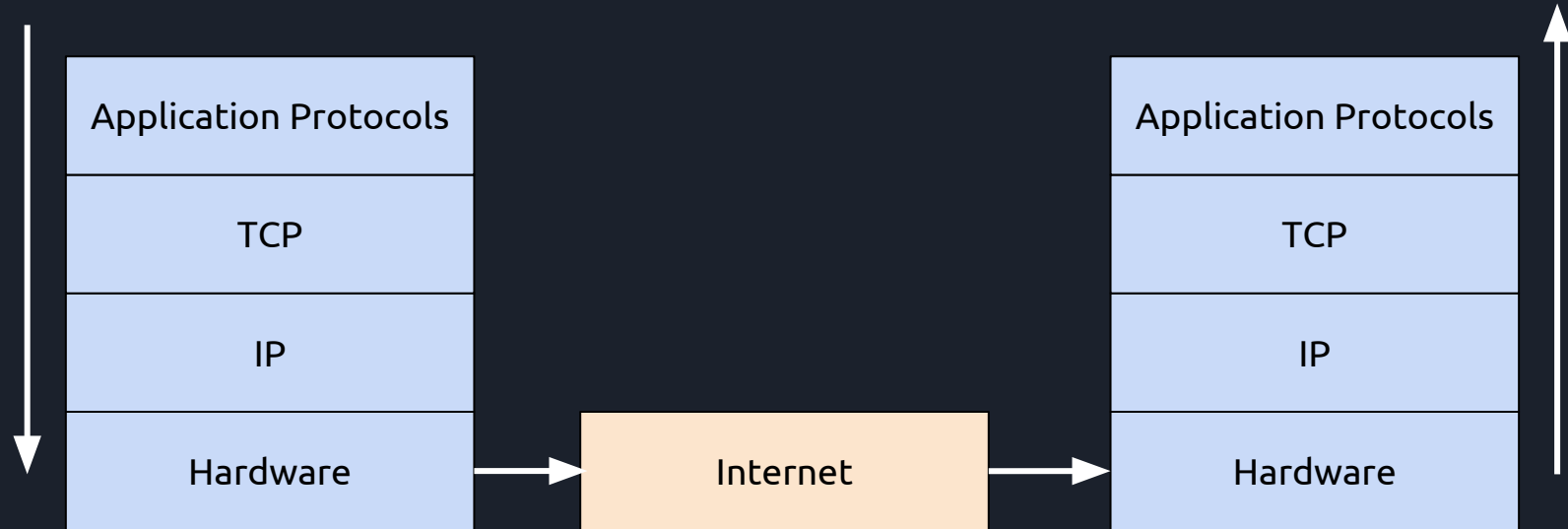
Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

# Protocol Stack



Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

# Protocol Stack

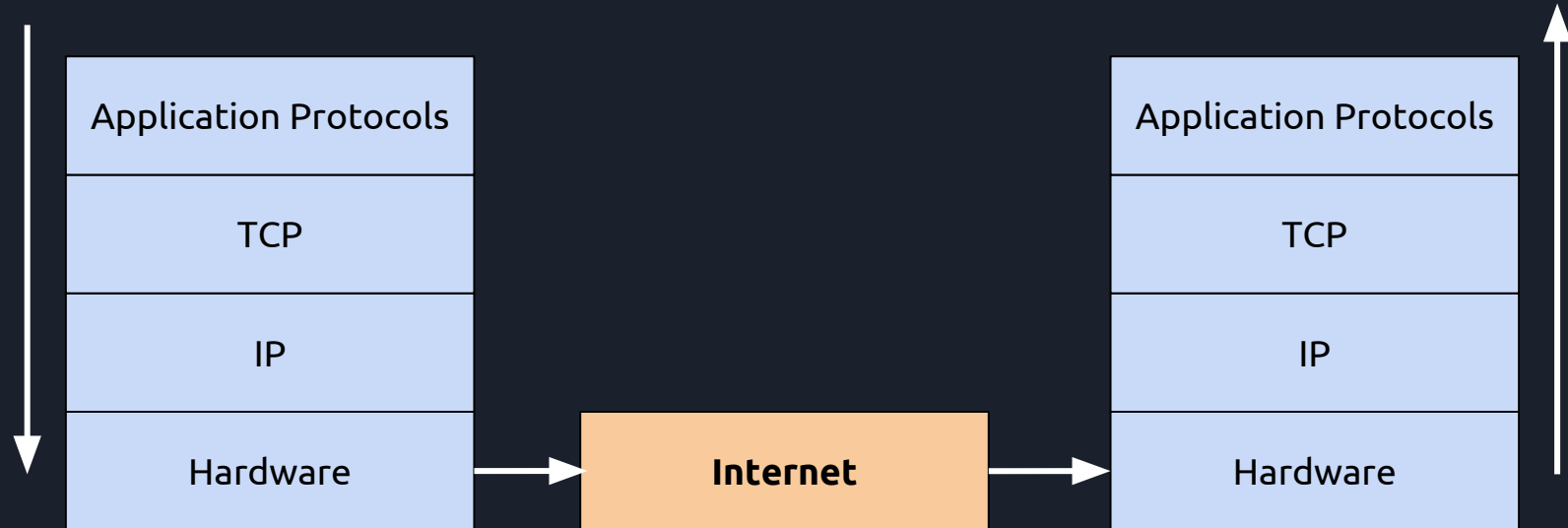


Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

OSI model, HTTP, TCP, IP



# Protocol Stack



Source: <https://web.stanford.edu/class/msande91sj/www-spr04/readings/week1/InternetWhitepaper.htm>

OSI model, HTTP, TCP, IP

**Internet**

# Internet

So we have prepared our data for transmission and can accept transmitted data

What's inside the "Internet" box?

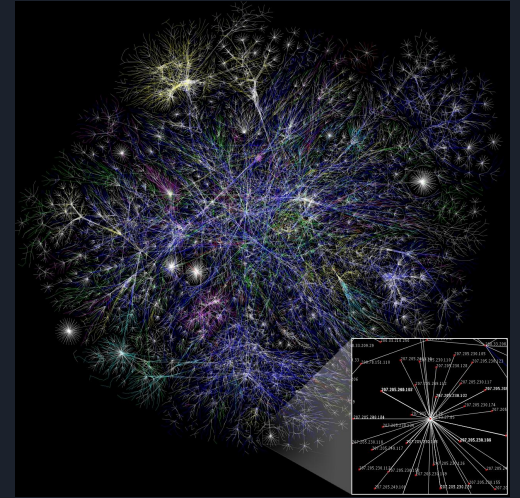


**Internet**

# Internet Backbone

A core web of fiber optic cables

- Light transmission



Source:




[https://en.wikipedia.org/wiki/Internet\\_backbone](https://en.wikipedia.org/wiki/Internet_backbone)

# Internet Backbone

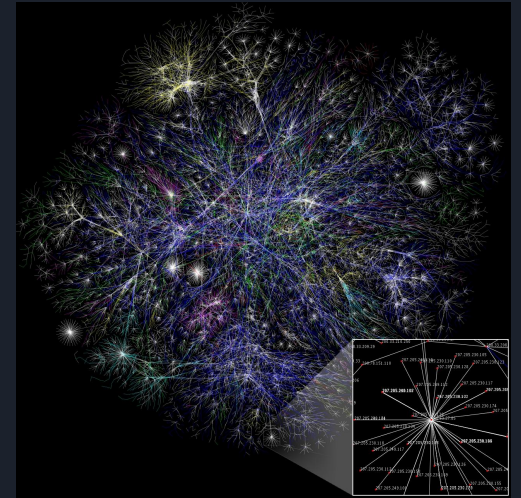
A core web of fiber optic cables

- Light transmission

Top stories

 <p>Sites Wouldn't Load Monday? Blame Internet's 'Backbone'</p> <p>Newser 1 day ago</p>	 <p>The Internet's 'backbone' was broken, and many in US were suddenly unplugged</p> <p>The Boston Globe 2 days ago</p>	 <p>How Level 3's Tiny Error Shut Off the Internet for Parts of the US</p> <p>WIRED 1 day ago</p>
--	--	--

[→ More for internet backbone](#)



Source:  
[https://en.wikipedia.org/wiki/Internet\\_backbone](https://en.wikipedia.org/wiki/Internet_backbone)

# Routers

Routers are devices that forwards data packets

Routing tables make sure we get to the destination IP address

- Another protocol

Physical and wireless transmission

- WiFi uses radio signals (electromagnetic wave modulation)

# Routers

Routers are devices that forwards data packets

Routing tables make sure we get to the destination IP address

- Another protocol

Physical and wireless transmission

- WiFi uses radio signals (electromagnetic wave modulation)

Routers use IP addresses — what's the IP for google.com?

# Domain Name System (DNS)

Converts web URLs into IP addresses

Distributed database that tracks names and IP addresses



# Domain Name System (DNS)

Converts web URLs into IP addresses

Distributed database that tracks names and IP addresses

```
$ host google.com
```

```
google.com has address 172.217.6.238
```

```
google.com has IPv6 address 2607:f8b0:4006:805::200e
```

```
google.com mail is handled by 40 alt3.aspmx.l.google.com.
```

```
google.com mail is handled by 30 alt2.aspmx.l.google.com.
```

```
google.com mail is handled by 50 alt4.aspmx.l.google.com.
```

```
google.com mail is handled by 10 aspmx.l.google.com.
```

```
google.com mail is handled by 20 alt1.aspmx.l.google.com.
```

# Vertical

## Computer

- Applications
- Operating System
- Hardware
- Physics

## Google

Computer

x 1,000,000

+ Distributed Systems

## Networks

- Protocols
- Infrastructure
- Physics

# Key Takeaways

1. Computers store and manipulate data. Networks transfer data.

# Key Takeaways

1. Computers store and manipulate data. Networks transfer data.
2. All data is bits. Bits are electricity.

# Key Takeaways

1. Computers store and manipulate data. Networks transfer data.
2. All data is bits. Bits are electricity.
3. Computing is theoretical. Computers are machines that model computing.

# Key Takeaways

1. Computers store and manipulate data. Networks transfer data.
2. All data is bits. Bits are electricity.
3. Computing is theoretical. Computers are machines that model computing.
4. Layering and black boxes reduce the complexity of understanding systems.

# Key Takeaways

1. Computers store and manipulate data. Networks transfer data.
2. All data is bits. Bits are electricity.
3. Computing is theoretical. Computers are machines that model computing.
4. Layering and black boxes reduce the complexity of understanding systems.
5. “Systems in 60 Seconds” alliterates better but wouldn’t be as educational.

# Resources

What happens when: <https://github.com/alex/what-happens-when>

How Does the Internet Work?:

<https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm>

From NAND to Tetris: <http://nand2tetris.org/>



# Systems in 60 Minutes

Raymond Xu  
raymondxu.io

