

Network Programming

David Hackbarth



Git Repository

<https://bitbucket.org/dhackbarth/gpd-0324>

What is Multiplayer?

Multiplayer

- more than one player
- two versions
 - co-op
 - competitive
- two options
 - local
 - remote

Computer Network

- ✦ Electronic
- ✦ Connection between two active devices over passive components
- ✦ Transport of packages (bits)
- ✦ Communication, exchange of information, supply of resources

Please,

Do

Not

Throw

Sausage

Pizza

Away!

OSI Model

Please,

Do

Not

Throw

Sausage

Pizza

Away!

Physical Layer

Data Link Layer

Network Layer

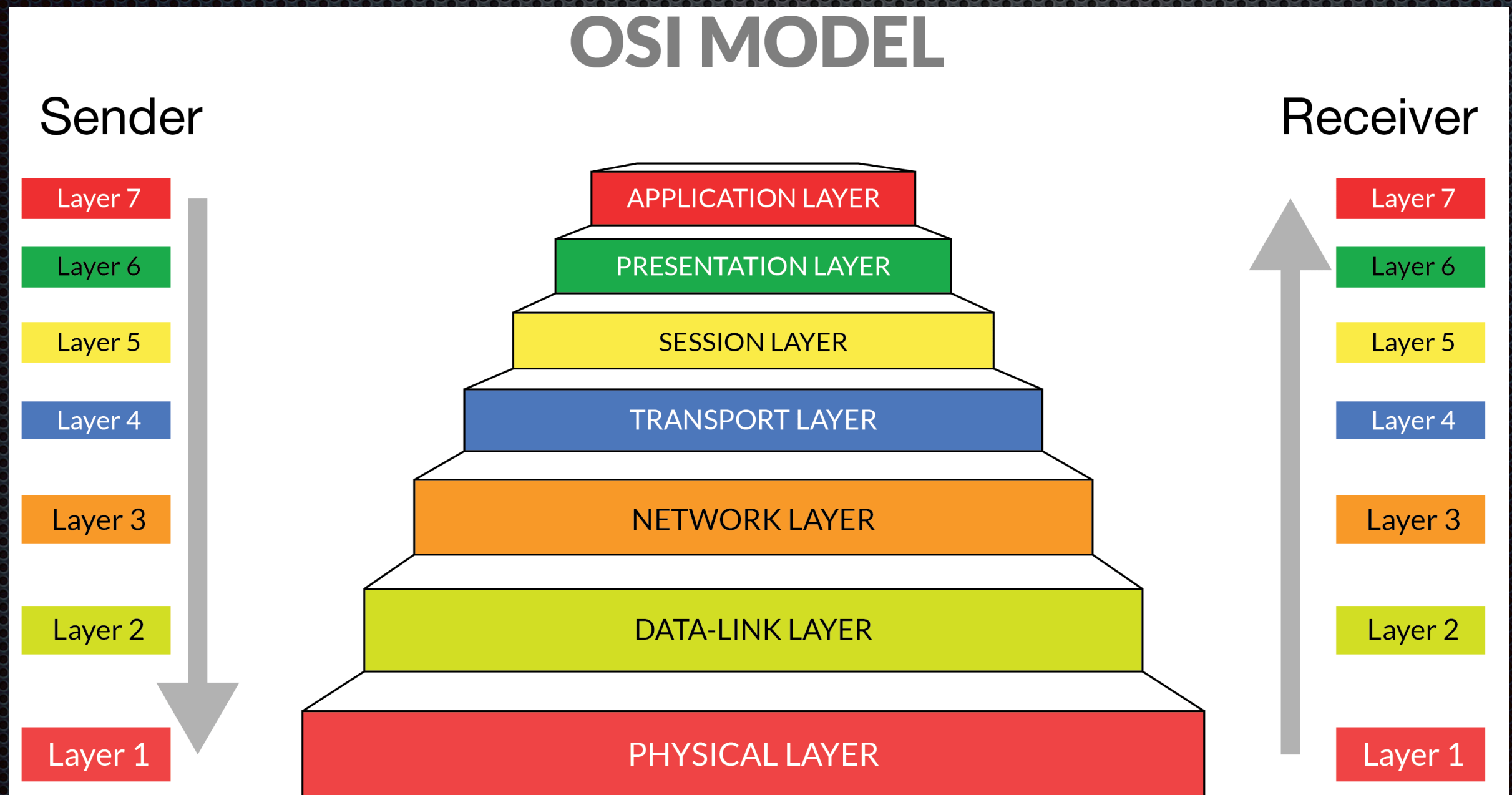
Transport Layer

Session Layer

Presentation Layer

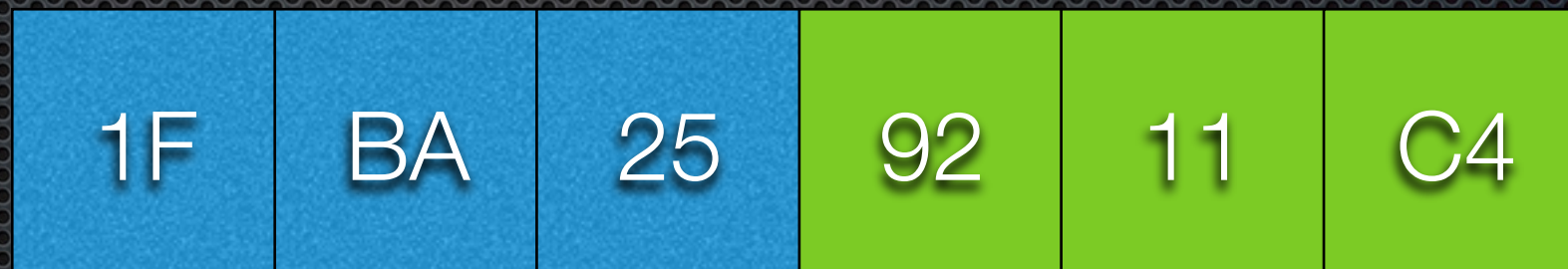
Application Layer

OSI Model - Open System Interconnection



MAC Address

- ✦ Media Access Control Address
- ✦ unique identifier for hardware
- ✦ 48 bit



Vendor ID

Device ID

IP Address

- ✦ Internet Protocol
- ✦ Version 4 & 6 (IPv4 / IPv6)
- ✦ unique within a network

IPv4

- 32 bits
- over 4.29 billion addresses
- 192.168.0.1

IPv6

- 128 bits
- $3.403 * 10^{38}$ addresses
- 2020:fff4:0010:abcd:
ef10:2345:0000:0001

Subnetting

- ✦ Subnetmask
- ✦ 32 bits
- ✦ CIDR: 192.168.0.23/24

	decimal		binary	
IP Address	192.168.0.	23	11000000.10101000.00000000.	00010111
Subnetmask	255.255.255.	0	11111111.11111111.11111111.	00000000
	network part	host part	network part	host part

Subnetting

- ✦ 192.168.0.23/24
- ✦ 192.168.0.0 - Network Address
- ✦ 192.168.0.255 - Broadcast Address

Reserved Networks - Private

Block	Range	No of Addresses	Class Networks
10.0.0.0/8	10.0.0.0 - 10.255.255.255	16.777.216	1 x A
172.16.0.0/12	172.16.0.0 - 172.31.255.255	1.048.576	16 x B
192.168.0.0/16	192.168.0.0 - 192.168.255.255	65536	256 x C
169.254.0.0/16	169.254.0.0 - 169.254.255.255	65536	-

Reserved Networks

- ✦ 127.0.0.0/8 (Loopback)
 - ✦ 127.0.0.1 - Localhost
- ✦ 224.0.0.0/4 (Multicast)

Ports

- ✦ more than one service
- ✦ ports as address
- ✦ 16 bit
- ✦ 0 - 1023 reserved
- ✦ 1024 - 49151 registered
- ✦ 49152 - 65535 open

Port examples

- ✦ HTTP (Hypertext Transfer Protocol) - 80
- ✦ HTTPS (Hypertext Transfer Protocol Secure) - 443
- ✦ FTP (File Transfer Protocol) - 20/21
- ✦ POP3 (Post Office Protocol Version 3) - 110
- ✦ IMAP (Internet Mail Access Protocol) - 143
- ✦ SMTP (Simple Mail Transfer Protocol) - 25

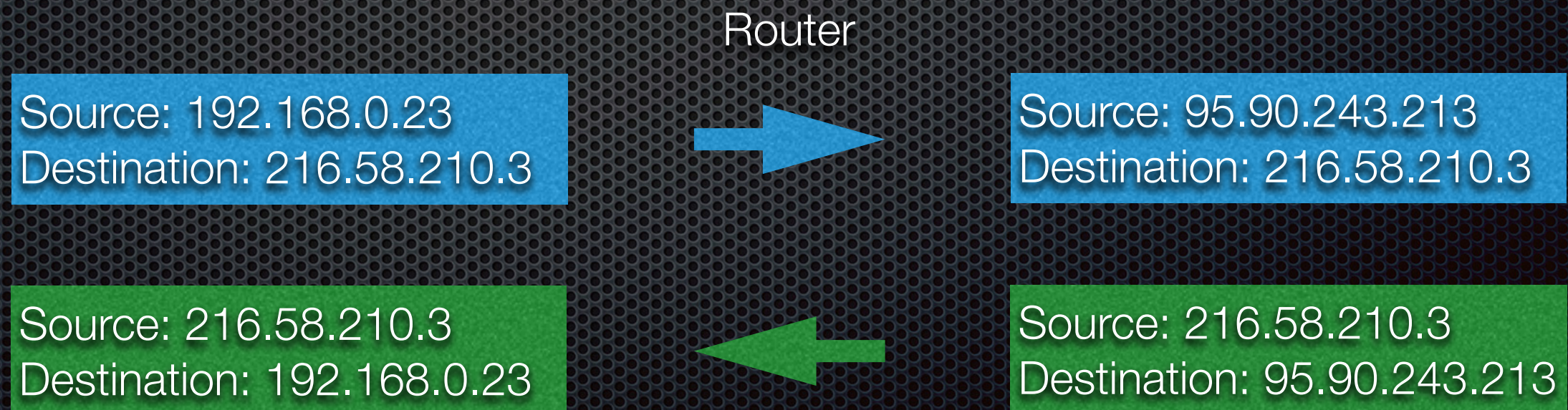
Port examples



- ✦ PlayStation Network - 3479/3480
- ✦ battle.Net - 3723/6112
- ✦ Teamspeak (Voice) - 9987
- ✦ League of Legends - 5000-5500
- ✦ Steam - 27000 - 27015

Network Address Translation

- ✦ NAT
- ✦ device exchange private IP address with public IP address for routing
- ✦ private network will be hidden (IP Masquerading)



TCP

- ✦ Transmission Control Protocol
- ✦ Connection based
- ✦ Reliable
- ✦ Sequenced
- ✦ High overhead
- ✦ Slowly
- ✦ Unit: Segments

UDP

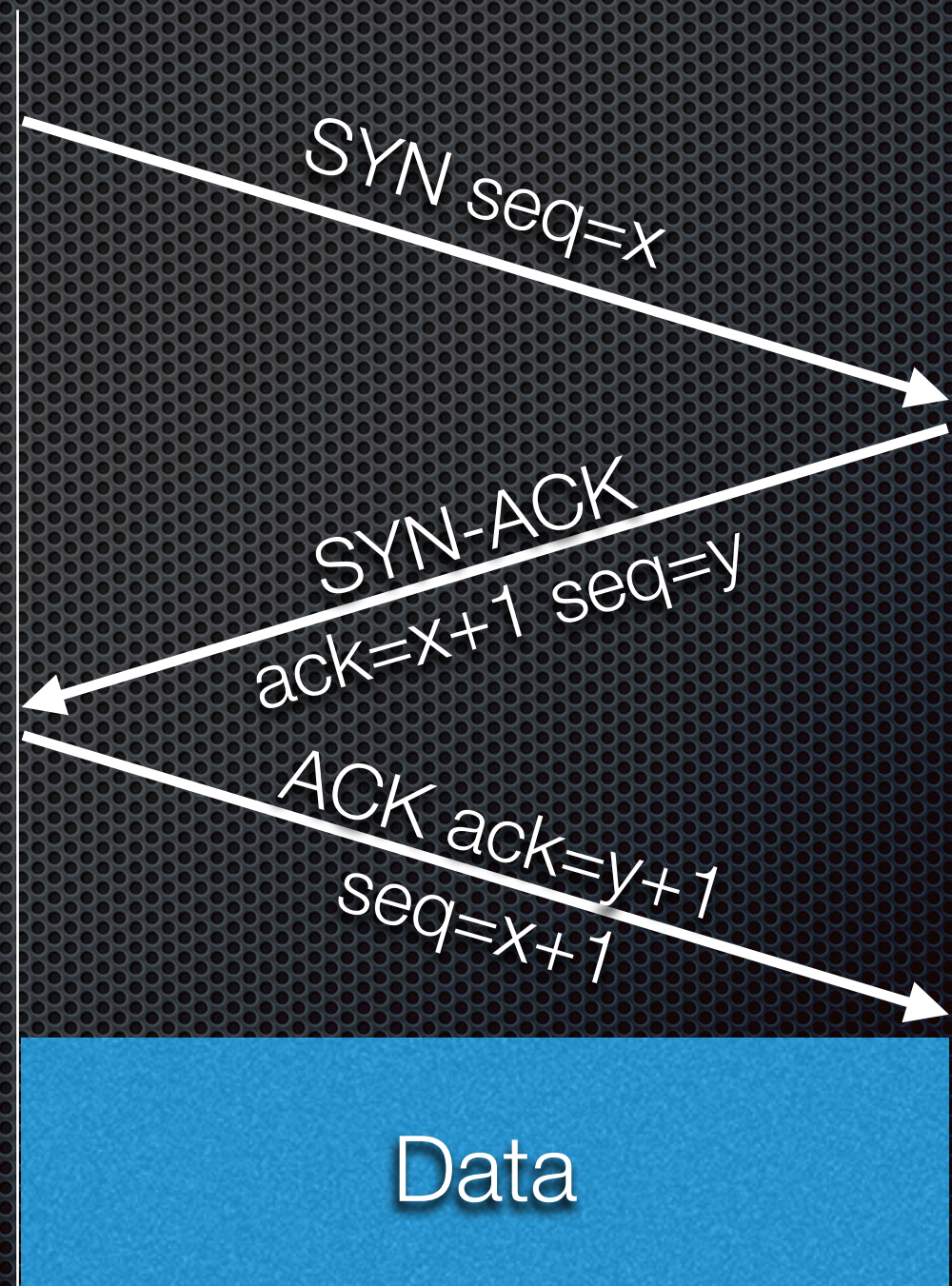
- ✦ User Datagram Protocol
- ✦ Connectionless
- ✦ Unreliable
- ✦ Unsequenced
- ✦ Small overhead
- ✦ Faster
- ✦ Unit: Datagrams

TCP

- bi-directional
- very popular for important data
- 3-Way-Handshake

Client

Server

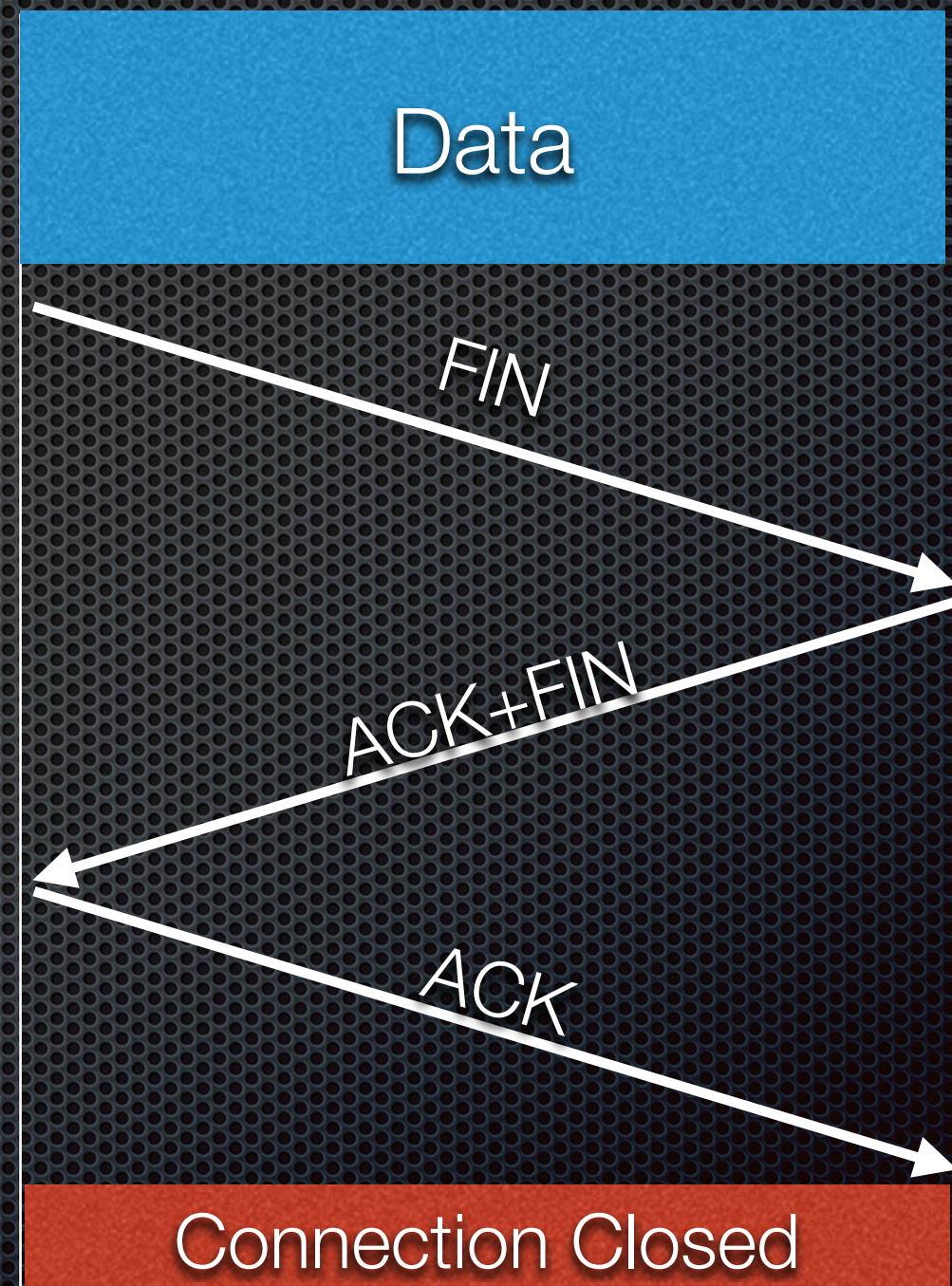


TCP

- ✦ bi-directional
- ✦ very popular for important data
- ✦ 3-Way-Handshake

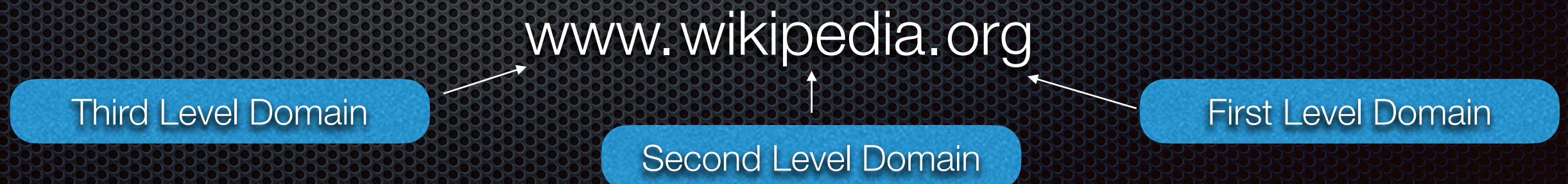
Client

Server



Domain Name System

- ✦ DNS
- ✦ Port: 53
- ✦ translate names into IP and vice versa
- ✦ nameserver
- ✦ Fully Qualified Domain Name (FQDN)



Dynamic Host Configuration Protocol

- ✦ DHCP
- ✦ Port: 67/68
- ✦ get network information automatically
 - ✦ IP, DNS-Server, Routing

P2P

- ✦ connection between two end points
- ✦ often used in 1vs1 or file sharing

Client-Server

- ✦ one server and infinite amount of clients
- ✦ one central communication point
- ✦ e.g. MMO, FPS, ...
- ✦ Dedicated vs. Hosted Server

Matchmaking

- ✦ in the beginning based on ip addresses
- ✦ Halo 2 (2004) introduced self-hosting with playlists and parties
- ✦ StarCraft 2 (2010) introduced ranking system with promotion and relegation
- ✦ in general based on rules

Non-Authoritative Server

- ✦ actual position sync is non-authoritative
- ✦ complete client-side implementation
- ✦ direct input
- ✦ no control / no verification => cheating
- ✦ solution: authoritative server

Authoritative Server

- ✦ dumb clients
- ✦ clients sends only commands/events
- ✦ server verifies and send back results
- ✦ authoritative server != dedicated server
- ✦ other problems

Problems

- ✦ waiting for results (ping)
- ✦ latency (lag)
- ✦ missing packets

Player 1 view - move with LEFT and RIGHT arrow keys

Lag = ms · ☐ Prediction · ☐ Reconciliation · ☐ Interpolation



Non-acknowledged inputs: 0

Server view · Update times per second



Last acknowledged input: Player 0: #0 Player 1: #0

Player 2 view - move with A and D keys

Lag = ms · ☐ Prediction · ☐ Reconciliation · ☐ Interpolation



Non-acknowledged inputs: 0

Solutions & other Problems

- ✦ client-side prediction

Player 1 view - move with LEFT and RIGHT arrow keys

Lag = ms · ☒ Prediction · ☐ Reconciliation · ☐ Interpolation



Non-acknowledged inputs: 0

Server view · Update times per second



Last acknowledged input: Player 0: #0 Player 1: #0

Player 2 view - move with A and D keys

Lag = ms · ☒ Prediction · ☐ Reconciliation · ☐ Interpolation



Non-acknowledged inputs: 0

Solutions & other Problems

- ✦ client-side prediction
- ✦ server reconciliation

Player 1 view - move with LEFT and RIGHT arrow keys

Lag = ms · ☒ Prediction · ☒ Reconciliation · ☐ Interpolation



Non-acknowledged inputs: 0

Server view · Update times per second



Last acknowledged input: Player 0: #0 Player 1: #0

Player 2 view - move with A and D keys

Lag = ms · ☒ Prediction · ☒ Reconciliation · ☐ Interpolation



Non-acknowledged inputs: 0

Solutions & other Problems

- ✦ client-side prediction
- ✦ server reconciliation
- ✦ dead reckoning
- ✦ entity interpolation

Player 1 view - move with LEFT and RIGHT arrow keys

Lag = ms · ☒ Prediction · ☒ Reconciliation · ☒ Interpolation



Non-acknowledged inputs: 0

Server view · Update times per second



Last acknowledged input: Player 0: #0 Player 1: #0

Player 2 view - move with A and D keys

Lag = ms · ☒ Prediction · ☒ Reconciliation · ☒ Interpolation



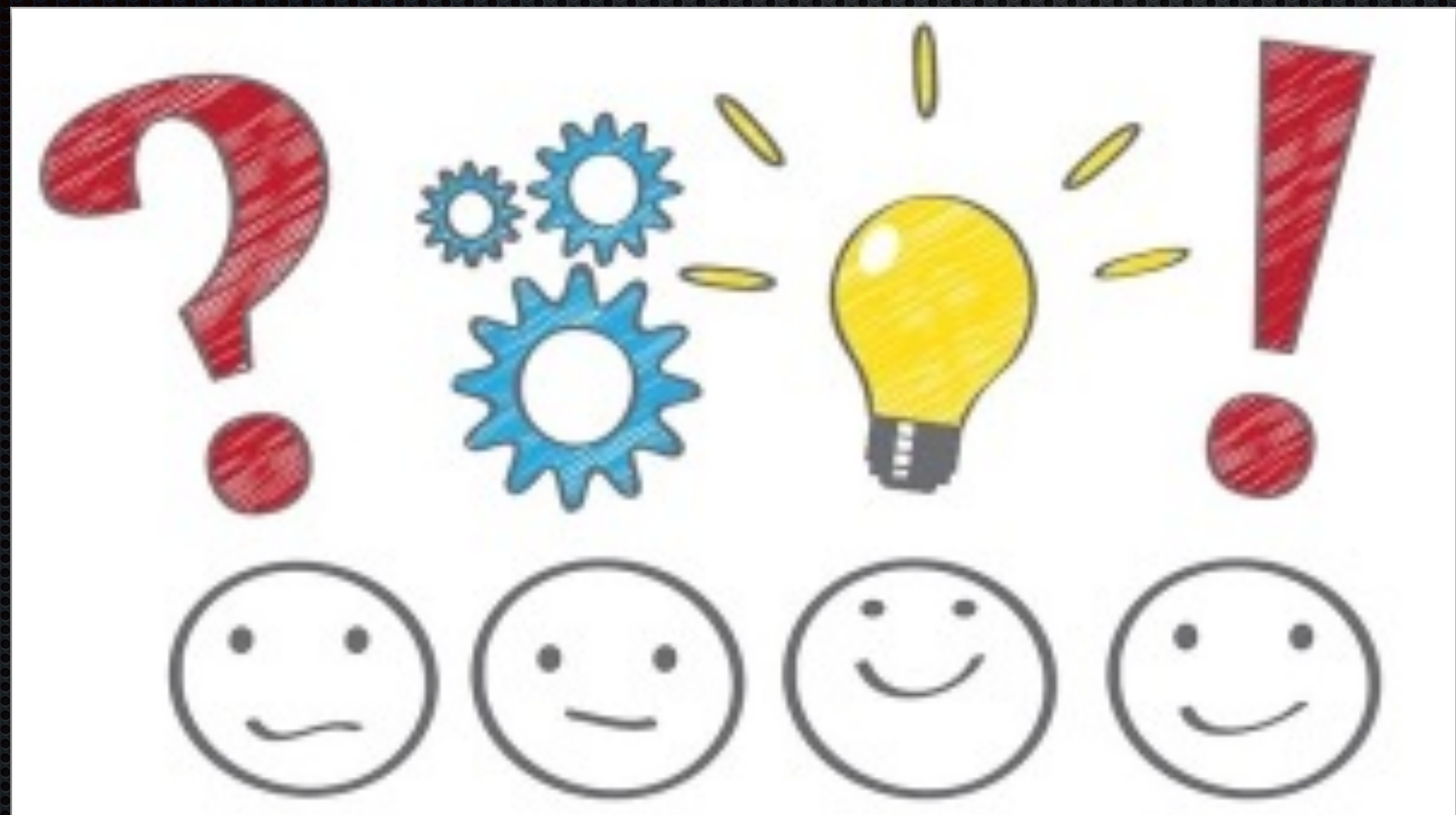
Non-acknowledged inputs: 0

Solutions & other Problems

- ✦ client-side prediction
- ✦ server reconciliation
- ✦ dead reckoning
- ✦ entity interpolation
- ✦ lag compensation

Links

- <https://gafferongames.com/>
- <https://developer.valvesoftware.com/wiki/Category:Networking>
- <https://dev.epicgames.com/documentation/en-us/unreal-engine/networking-and-multiplayer-in-unreal-engine>
- <https://www.gamedeveloper.com/programming/1500-archers-on-a-28-8-network-programming-in-age-of-empires-and-beyond>
- <https://doc.photonengine.com/fusion/current/getting-started/fusion-intro>
- <https://docs-multiplayer.unity3d.com/netcode/current/about/>
- <https://www.gabrielgambetta.com/client-side-prediction-live-demo.html>



Coding Time

There are two types of people.

```
if (Condition)
{
    Statements
    /*
     *
     */
}
```

```
if (Condition) {
    Statements
    /*
     *
     */
}
```

Programmers will know.