FS18

BASICS FOR P2P MECHANISMS

Mechanisms for P2P systems

T. Bocek

thomas.bocek@hsr.ch

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Agenda

- **Distributed Systems in the News**
- **Layers**
- **Layer 4 - TCP/IP and UDP Sockets**
- **NAT Issues**
- **RPC with gRPC and Avro**

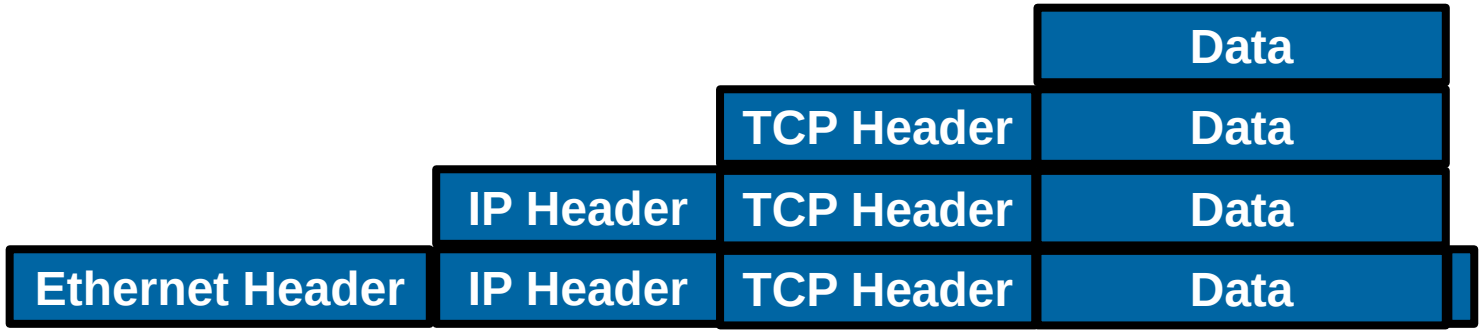
Distributed Systems - In the *News*

- **01.04.2018: Announcing 1.1.1.1: the fastest, privacy-first consumer DNS service**
  - VSS lecture 1
  - Google: 8.8.8.8, Quad9: 9.9.9.9, OpenDNS: 208.67.222.222, Norton DNS 199.85.126.20, CleanBrowsing 185.228.168.168, Yandex DNS 77.88.8.7, Comodo DNS 8.26.56.26
  - Performance
    - Global average: #1 CloudFlare: 4.98 ms #2 Google: 16.44 ms #3 Quad9: 18.25 ms
  - Geldspielgesetz: 10.6.2018 - “Zudem richten die Internet-Provider eine Zugangssperre ein.”
- **05.04.2018: RBI Bars Banks From Doing Business With Crypto Firms**
  - "It has been decided that, with immediate effect, entities regulated by RBI shall not deal with or provide services to any individual or business entities dealing with or settling [cryptocurrencies]. Regulated entities which already provide such services shall exit the relationship within a specified time,“
- **06.04.2018: Pakistan Bars Banks from Crypto and ICO Trading**
- **05.04.2018: Swiss Central Banker: State-Backed Crypto Would Pose 'Incalculable Risks’**
  - DLT "does not yet meet the requirements expected of the RTGS [real-time gross settlement] systems in terms of scalability, data security and reliability."

Layers

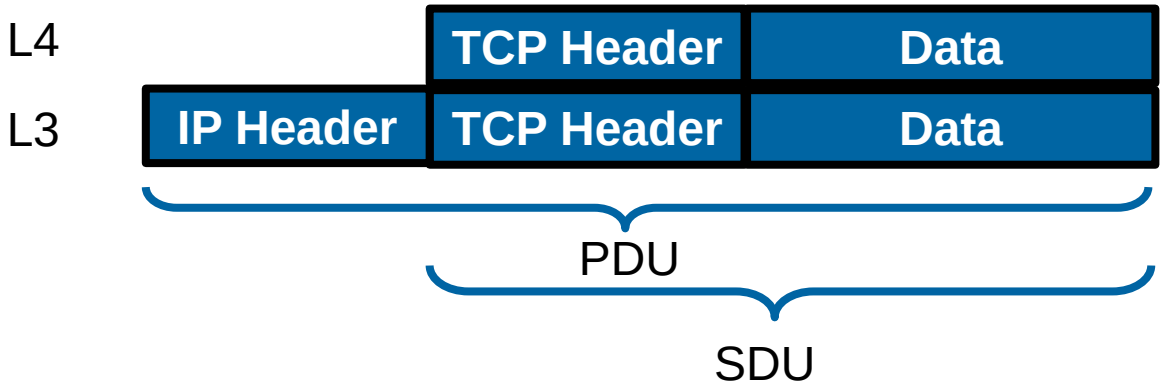
- **Networking: Each vendor had its own proprietary solution - not compatible with another solution**
- **Nowadays most vendors build compatible networks hardware/software from different vendors**
- **Goal of layers: interoperability**
- **1984: ISO 7498 - The Basic Reference Model for Open Systems Interconnection**

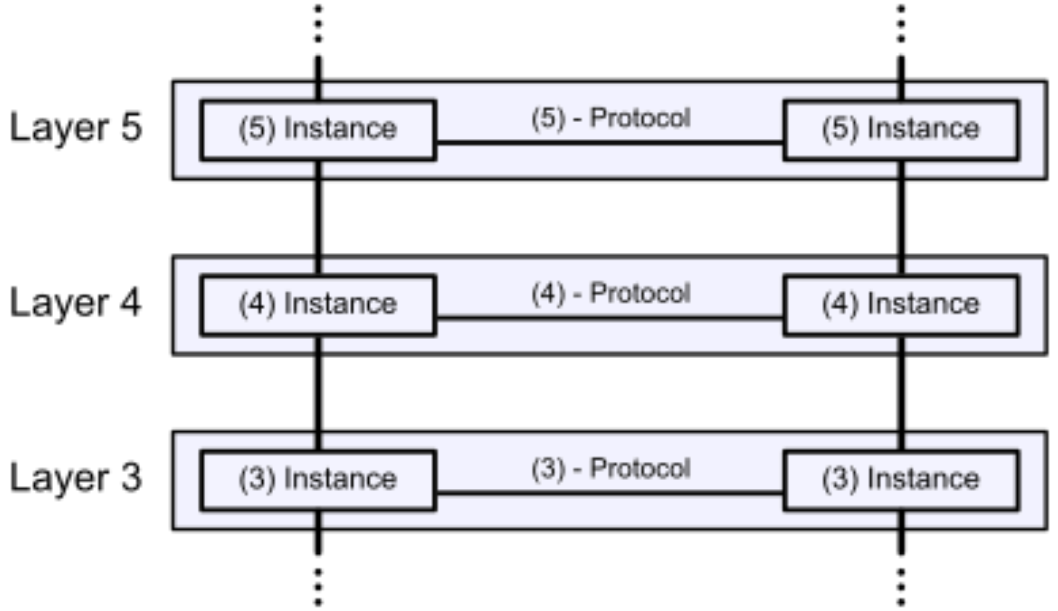
OSI model	"Internet model"
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data link	Link
Pysical	



Layer Abstraction

- **Protocols enable an entity to interact with a entity at the same layer in another host**
- **Service definitions: provide functionality to an (N)-layer by an (N-1) layer**
- **Layer N exchange protocol data units (PDUs) with layer N protocol. Each PDU contains a protocol header and payload, the service data unit (SDU)**





https://en.wikipedia.org/wiki/OSI_model

Layers – Different Sources, Different Naming

RFC 1122, Internet STD 3 (1989)	Cisco Academy	Kurose, Forouzan	Comer Kozierok	Stallings	Tanenbaum	Arpanet	OSI model
Four layers	Four layers	Five layers	Four+one layers	Five layers	Five layers	Three layers	Seven layers
"Internet model"	"Internet model"	"Five-layer Internet model" or "TCP/IP protocol suite"	"TCP/IP 5-layer reference model"	"TCP/IP model"	"TCP/IP 5-layer reference model"	"Arpanet reference model"	OSI model
Application	Application	Application	Application	Application	Application	Application/Process	Application
							Presentation
							Session
Transport	Transport	Transport	Transport	Host-to-host or transport	Transport	Host-to-host	Transport
Internet	Internetwork	Network	Internet	Internet	Internet		Network
Link	Network interface	Data link	Data link (Network interface)	Network access	Data link	Network interface	Data link
		Physical	(Hardware)	Physical	Physical		Physical

https://en.wikipedia.org/wiki/Internet_protocol_suite

URL:

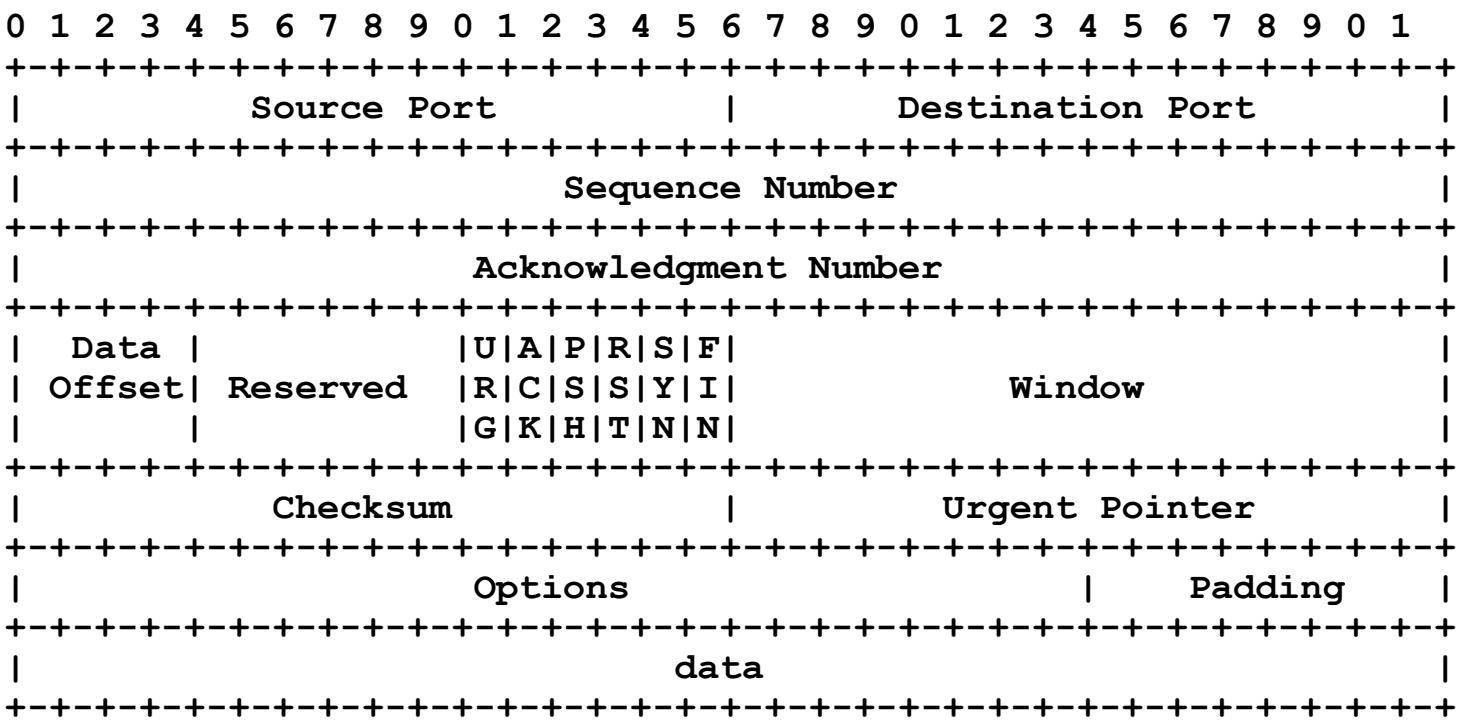
b.socrative.com/student/

Layer	Example
Application	
Transport	
Network	
Link	
Physical	

Room: HSR

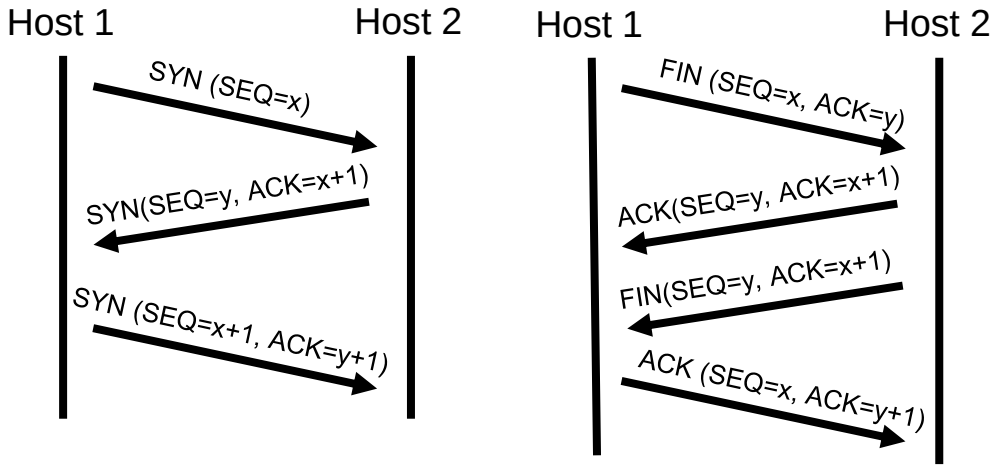
Layer 4 - Transport

- **TCP (Transmission Control Protocol)**
  - Reliable (retransmission)
  - Ordered – unordered?
  - Check summed – 16bit
- **In the news (5.4.2018): Ten Countries Face Significant Internet Disruption After African Coast to Europe Submarine Cable Cut**
  - Mauritania, with a complete outage lasting for nearly 48 hours
  - “However, we believe that the April 1 outage may have been government-directed, related to recent national elections.”



http://freesoft.org/CIE/Course/Section4/8.htm

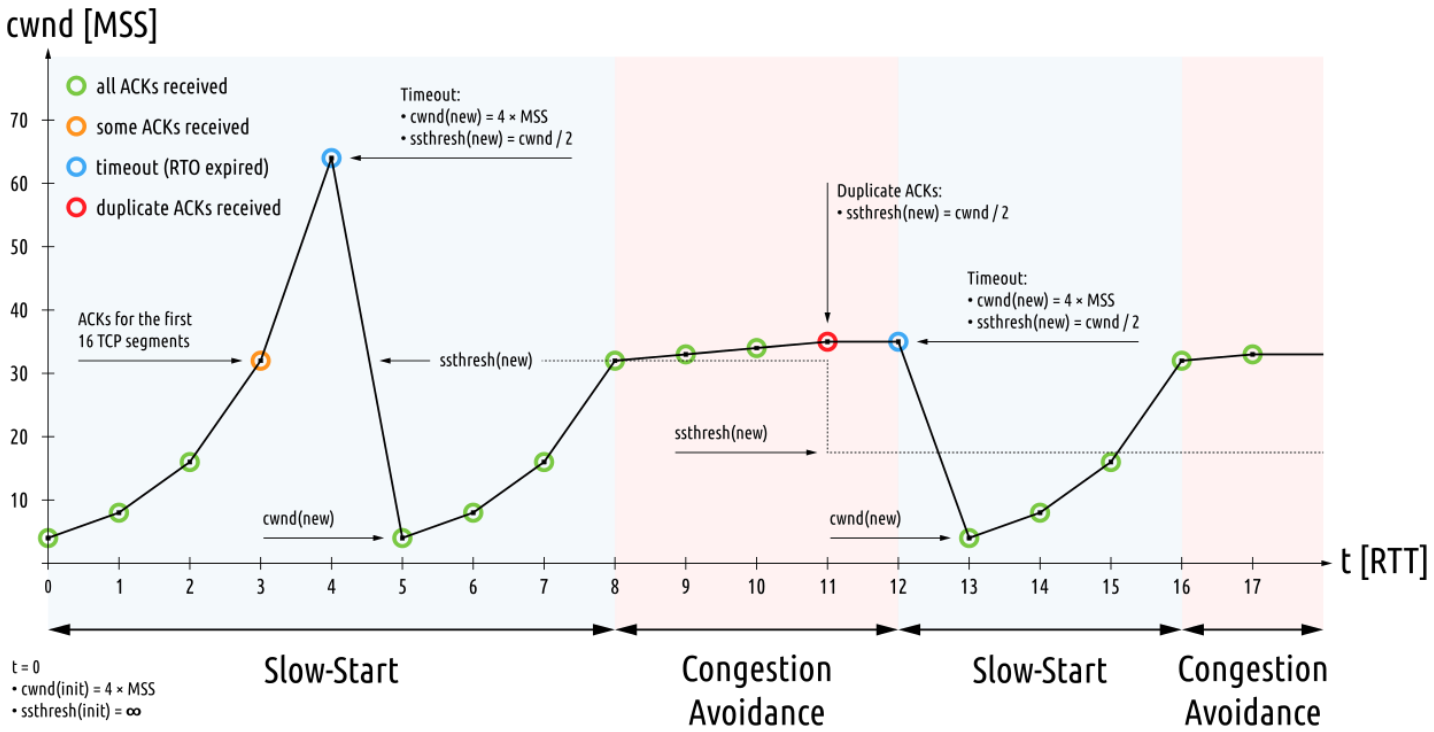
- **Connection establishment**
  - SYN, SYN-ACK, ACK (three way)
  - Initiates TCP session: initial sequence number is ~random
- **Connection termination**
  - FIN, ACK + FIN, ACK (three/four way)
  - 3-way handshake, when host 1 sends a FIN and host 2 replies with a FIN & ACK
- **Sequences and ACKs**
  - Identification each byte of data
  - Order of the bytes → reconstruction
  - ACKs are sent to tell the sender that data has been received to the specified byte
  - Detecting lost data: RTO, DupACK:



- **Retransmission timeout**
  - If no ACK is received aftert timout (e.g. 2xRTT), resend.
- **Duplicate cumulative acknowledgements**
  - ACKs for last consecutive packets
  - 3 times same ACK → retransmit (fast retransmit)

Analyzing Traffic

- **Flow control**
  - Sender is not overwhelming a receiver
  - Back pressure
  - Sliding window:
    - Receiver specifies the amount of additionally received data in bytes that can be buffered
    - Sender up to that amount of data before ACK
- **Congestion control**
  - slow-start
  - congestion avoidance
- **Difference flow/congestion control**
- **Wireshark - example**



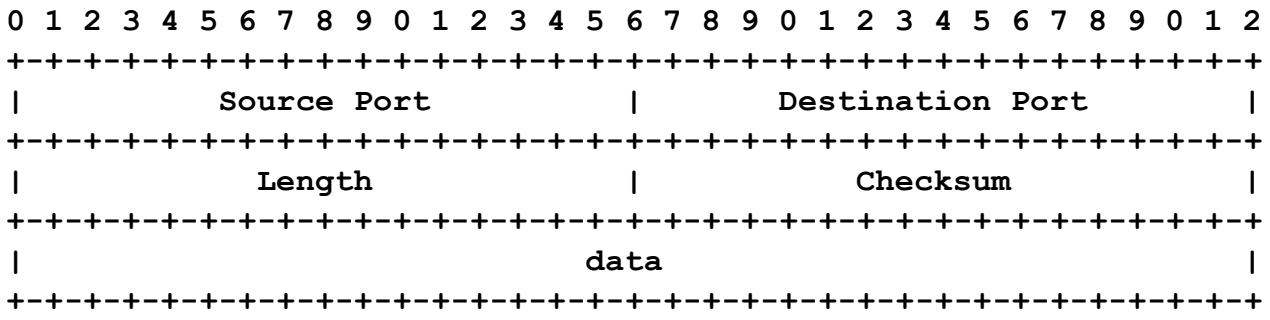
https://upload.wikimedia.org/wikipedia/commons/thumb/2/24/TCP_Slow-Start_and_Congestion_Avoidance.svg/1280px-TCP_Slow-Start_and_Congestion_Avoidance.svg.png

URL: b.socrative.com/student/

Room: HSR

Layer 4 - Transport

- **User Datagram Protocol (UDP)**
  - UDP is used for DNS, streaming audio and video
  - Simple connectionless communication model
  - No guarantee
    - Delivery
    - Ordering
    - Duplicate protection
- **Simple Client/Server (wireshark)**



- **SCTP (Stream Control Transmission Protocol)**
  - Message-based
  - Allows data to be divided into multiple streams
  - Syn cookies - SCTP uses a four-way handshake with a signed cookie.
  - 32-bit end-to-end checksum (CRC32C)
  - Multi-homing multiple IP addresses of endpoints
  - Not widely used: "We have been deploying SCTP in several applications now, and encountered significant problem with SCTP support in various home routers."
    - E.g., OpenWRT – not enabled by default
    - E.g., UFW - Uncomplicated Firewall – not supported
  - Tunneled over UDP – WebRTC
    - QUIC – competition? "5% of Internet traffic (I could not verify this claim)"

Comparison

TCP *

- Transport layer
- Connection oriented
- Reliable transfer
- Streams
- Guaranteed order
- Widely used – HTTPS
- Flow and congestion control
- Heavyweight
- Header size is 20 bytes
- Error checking and recovery

UDP *

- Transport layer
- Connection less
- Unreliable transfer
- Messages
- Unordered
- Widely used – DNS
- No flow, congestion
- Lightweight
- Header size is 8 bytes
- Error checking, no recovery

SCTP *

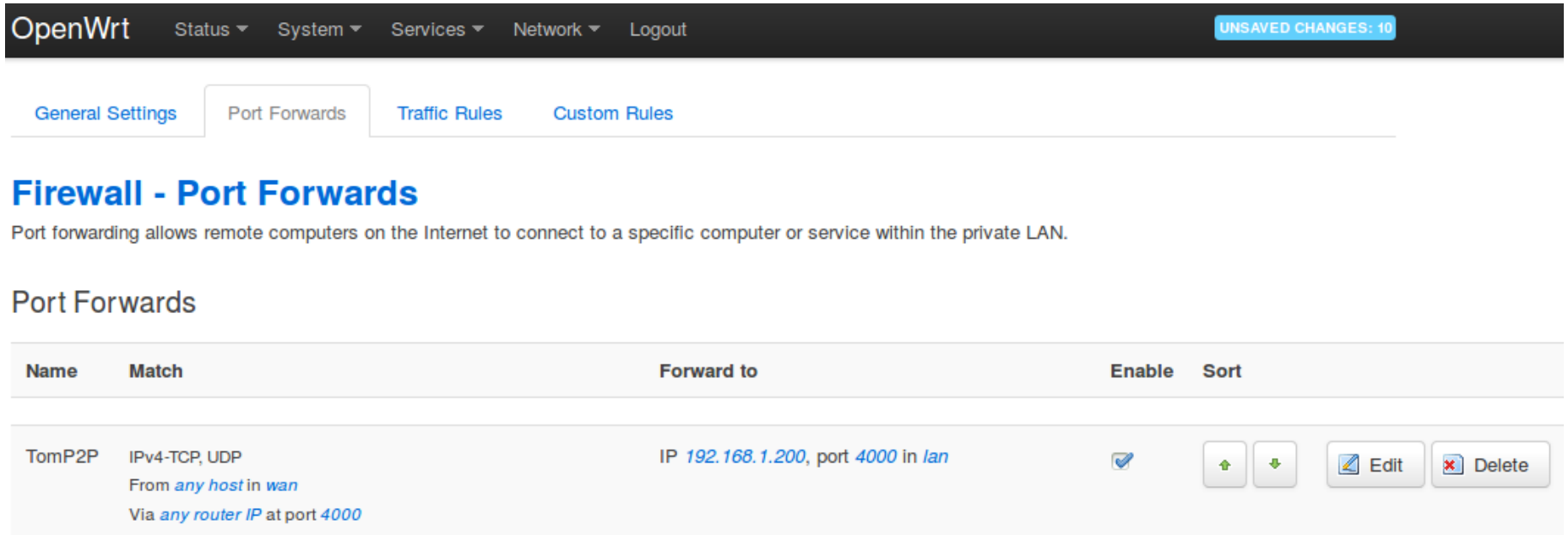
- Transport layer
- Connection oriented
- Reliable transfer
- Messages
- User can choose
- WebRTC
- Flow and congestion control
- Heavyweight
- Common header is 12 bytes
- Error checking and recovery

URL: b.socrative.com/student/

Room: HSR

Connectivity, Security, and Robustness

- **NAT**
  - Network Address Translation – breaks end-to-end
  - “If nothing else, [NAT] can serve to provide temporarily relief while other, more complex and far-reaching solutions are worked out”
    (RFC 1631 - The IP Network Address Translator (NAT))
- **“Easy solution”:**
  - Manual port forwarding: e.g., setup on your router

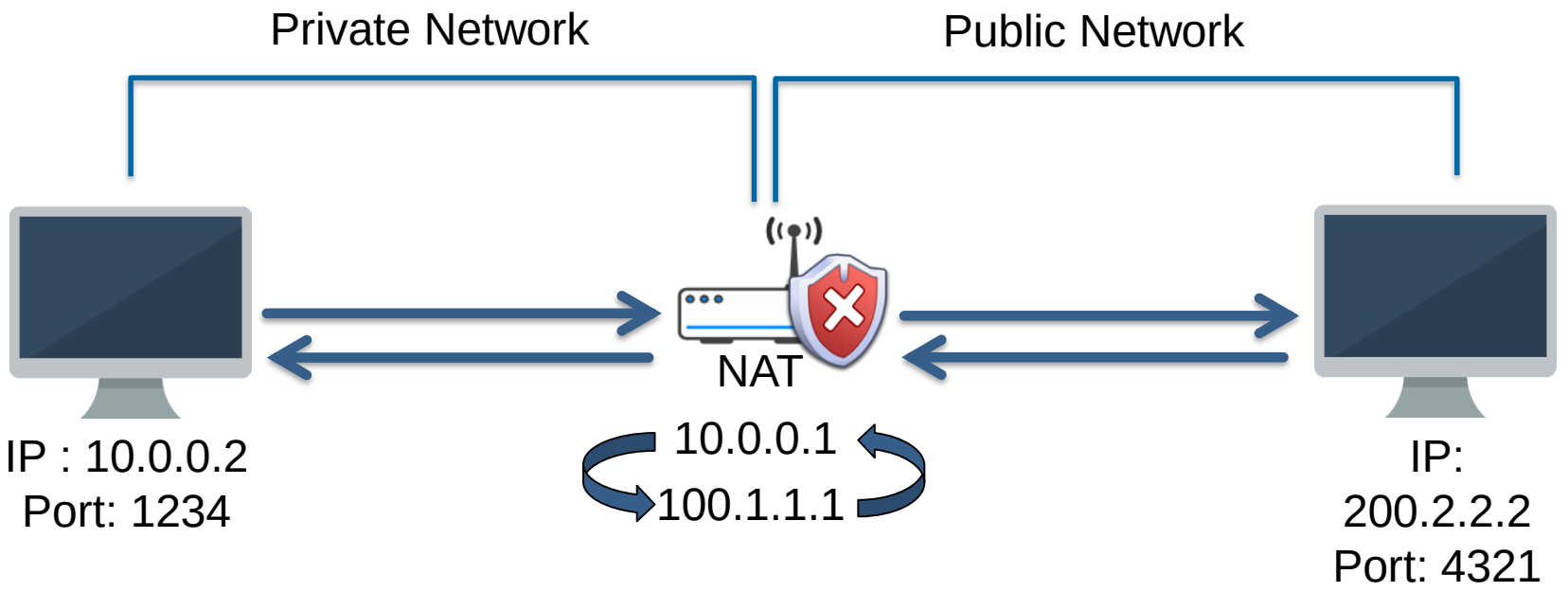


Connectivity, Security, and Robustness

- **Easy solution: UPNP / NAT-PMP**
  - Both configure port forwarding, but UPNP is more: discover devices - uses broadcasting to find router (Simple Service Discovery Protocol)
  - UPNP: configure devices - uses HTTP and XML to configure port forwarding (Internet Gateway Device Protocol)
  - NAT-PMP: protocol made for configuring port-forwarding, but no discover (how to find router?)

Connectivity, Security, and Robustness

- **Difficult solution: hole punching**
  - rendezvous / relay peer which does “hole punching”, in worst case relay traffic.
- **NAT: translation table for private / public network**



NAT Table Entry: (10.0.0.2:1234, 200.2.2.2:4321; 200.2.2.2:4321, 100.1.1.1:1234)

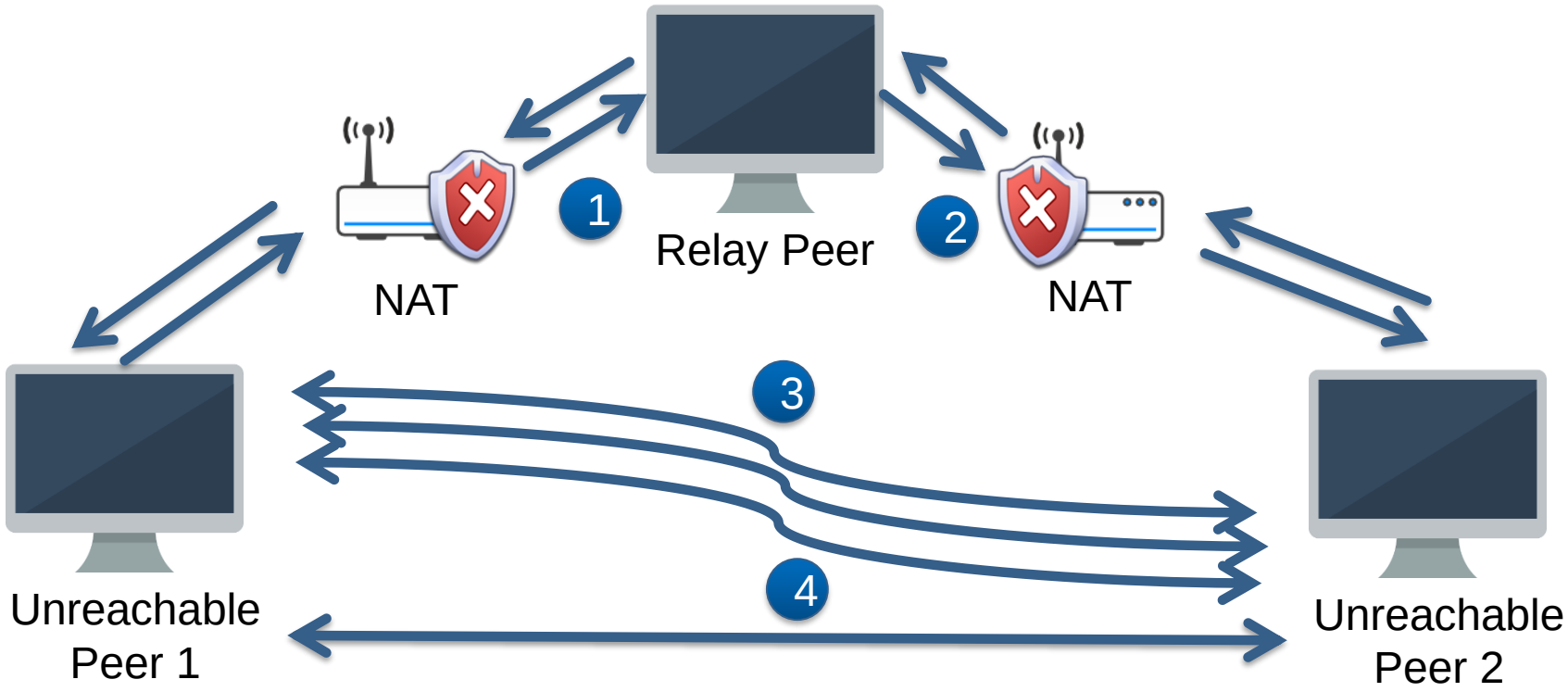
Connectivity, Security, and Robustness: Hole punching in P2P Networks

1) Peer 1 initiates a new connection trial to peer 2 via relay and signals its source ports and IP (relay/rendez-vous peer has connection to Unreachable Peer 2)

2) Peer 2 answers back with its source ports and IP

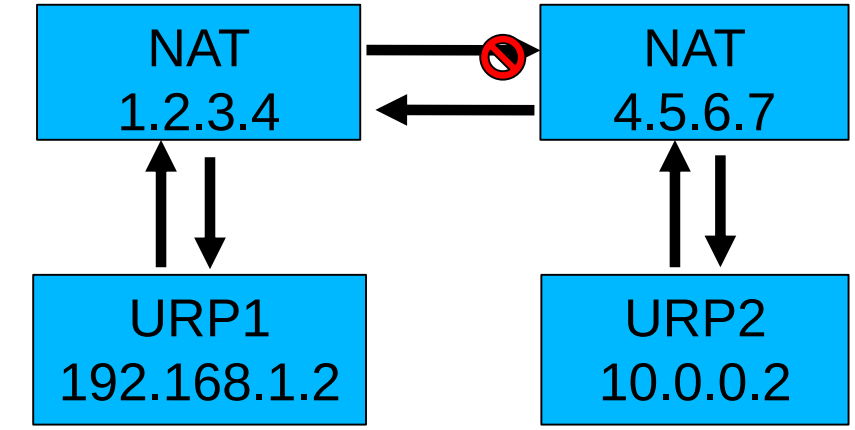
3) Both of the peers punch holes into their firewall/NAT

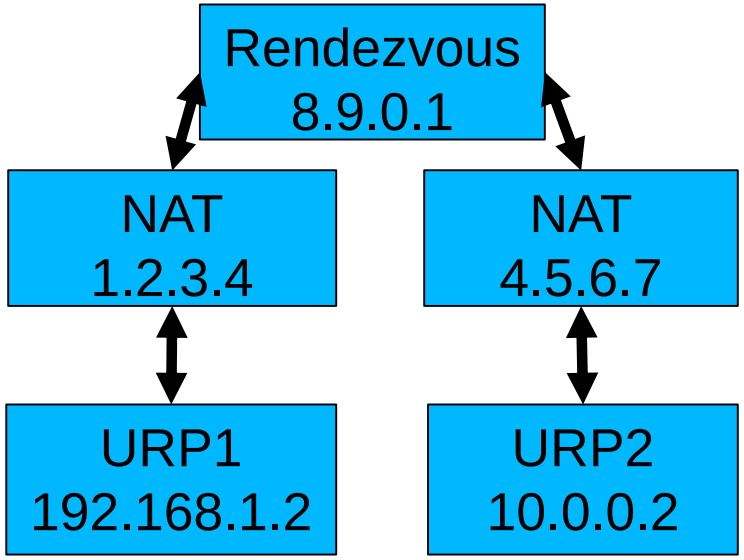
4) Established a connection



Connectivity, Security, and Robustness

- **Hole punching**
  - URP1 got 4.5.6.7:5000, URP2 got 1.2.3.4:4000
  - Unreachable peer 1 request to NAT 4.5.6.7, will fail – no mapping, however, unreachable peer 1 creates mapping with that request
  - Unreachable peer 2 sends request to unreachable peer 1 (1.2.3.4:4000) success!





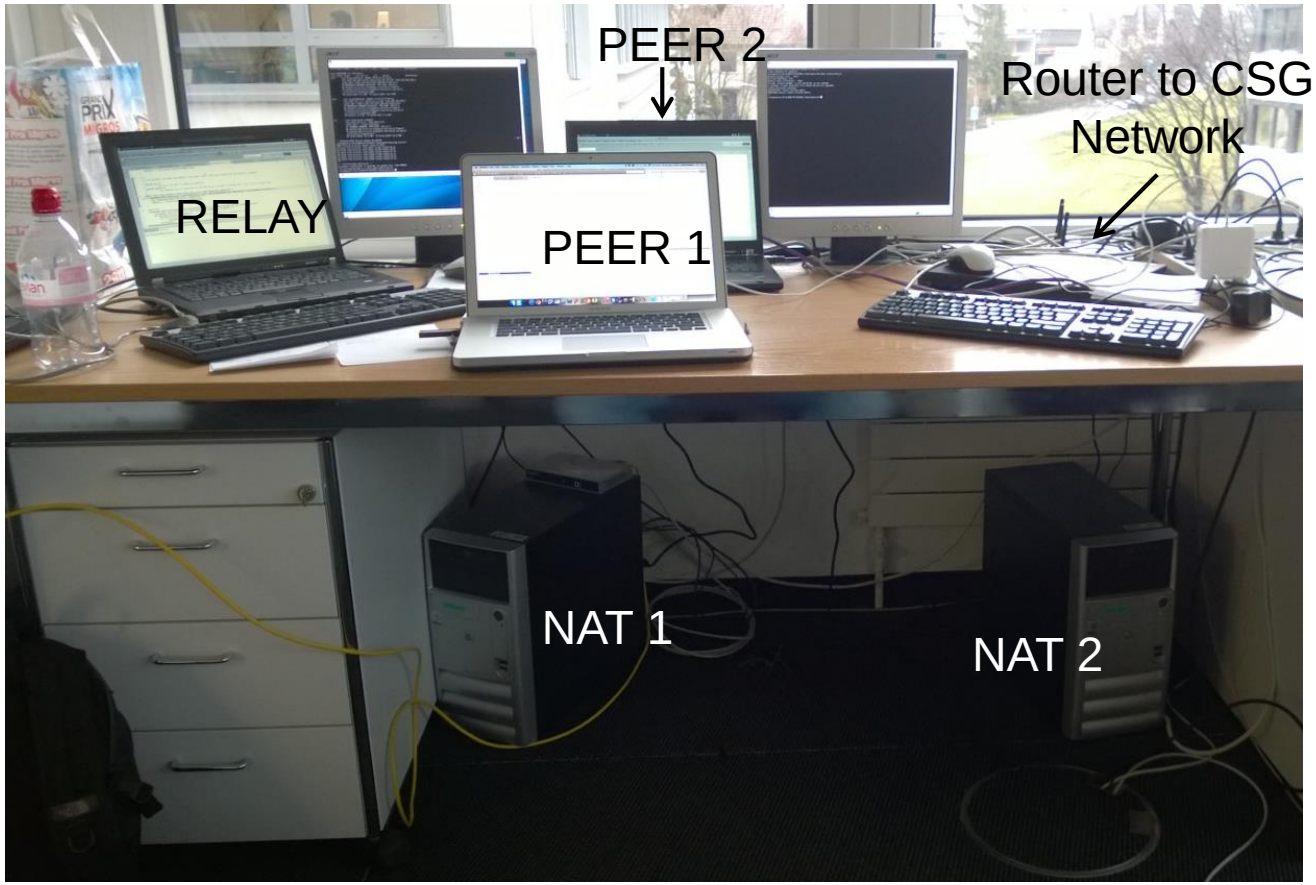
Mapping for NAT 1.2.3.4 (Unreachable peer 1)			
192.168.1.2:4000	4.5.6.7:5000	4.5.6.7:5000	1.2.3.4:4000

Mapping for NAT 4.5.6.7 (Unreachable peer 2)			

Mapping for NAT 4.5.6.7 (Unreachable peer 2)			
10.0.0.2:5000	1.2.3.4:4000	1.2.3.4:4000	4.5.6.7:5000

Connectivity, Security, and Robustness

- **Hole Punching Development (in the old days)**
- **Currently: network namespaces (since Linux 2.6.24)**



URL: b.socrative.com/student/

Room: HSR

RPC

- **Simple Distributed Architecture**
  - Call a function/procedure on another machine
  - Often: wait for result (synchronous)
- **What protocol to use?**
  - Text or binary?
- **How to talk to other servers**
  - Database protocols
  - HTTP + JSON – mostly frontend
- **Already solved?**
  - SOAP – lots of XML
  - CORBA - heavyweight
  - DCOM, COM+ - mostly windows
  - RMI – mostly Java
  - JSON, XML – no protocol description

- **Goal**
  - Language and platform neutral way for serializing structured data for communication protocols
  - Simplicity!
- **Binary vs. Text**
  - JSON: {“account_fee”:”1234”} – 22 bytes
    - Parsing needs time
    - Human readable – easy to debug
  - Binary: 0x2, 0x1234 – e.g. length+data – 3 bytes
    - No parsing
    - Needs a protocol, error-prone
- **Protocol Buffers, Apache Thrift, Apache Avro**
  - Interface Description Language (IDL)
  - Versioning
  - Binary format (performance)

RPC Comparison

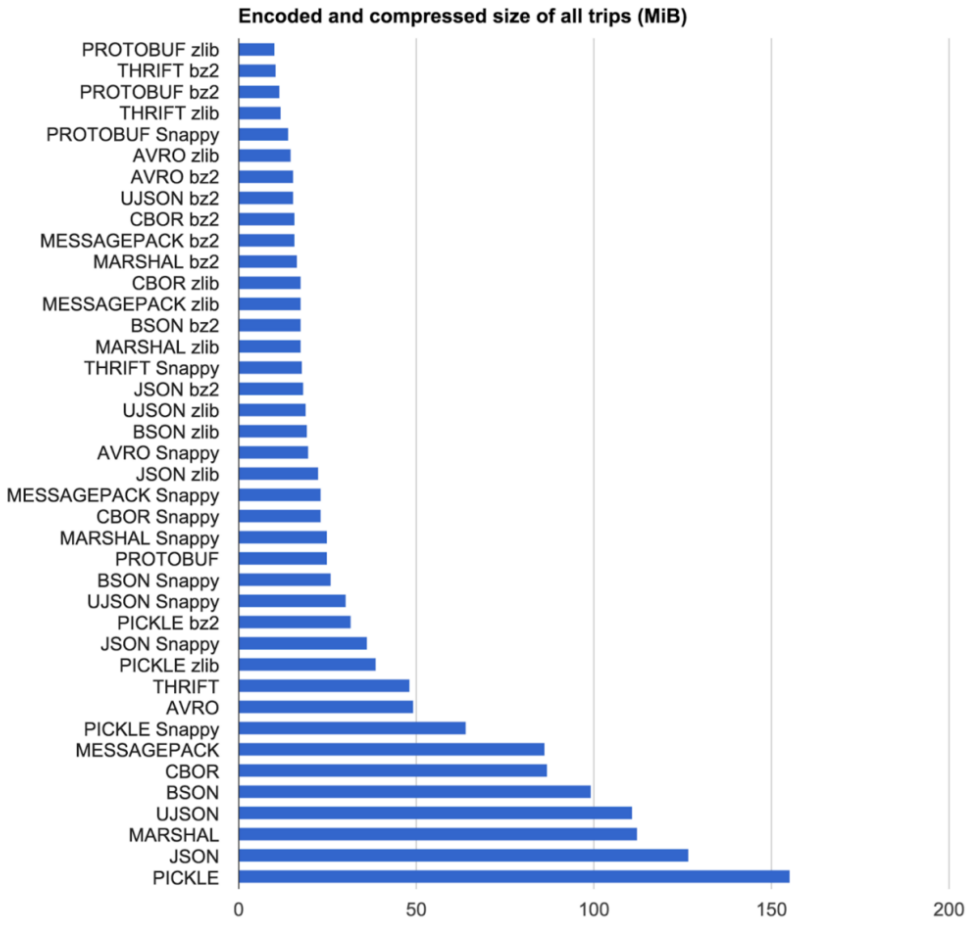
Library	Size in bytes	Time in ms
Avro (cheating)	133	0.0142
Avro	133	0.0568
Avro MSFT	141	0.0051
Thrift (cheating)	148	0.0069
Thrift	148	0.1470
ProtoBuf	155	0.0077
MessagePack	230	0.0296
ServiceStackJSV	258	0.0159
Json.NET BSON	286	0.0381
ServiceStackJson	290	0.0164
Json.NET	290	0.0333
XmlSerializer	571	0.1025
Binary Formatter	748	0.0344

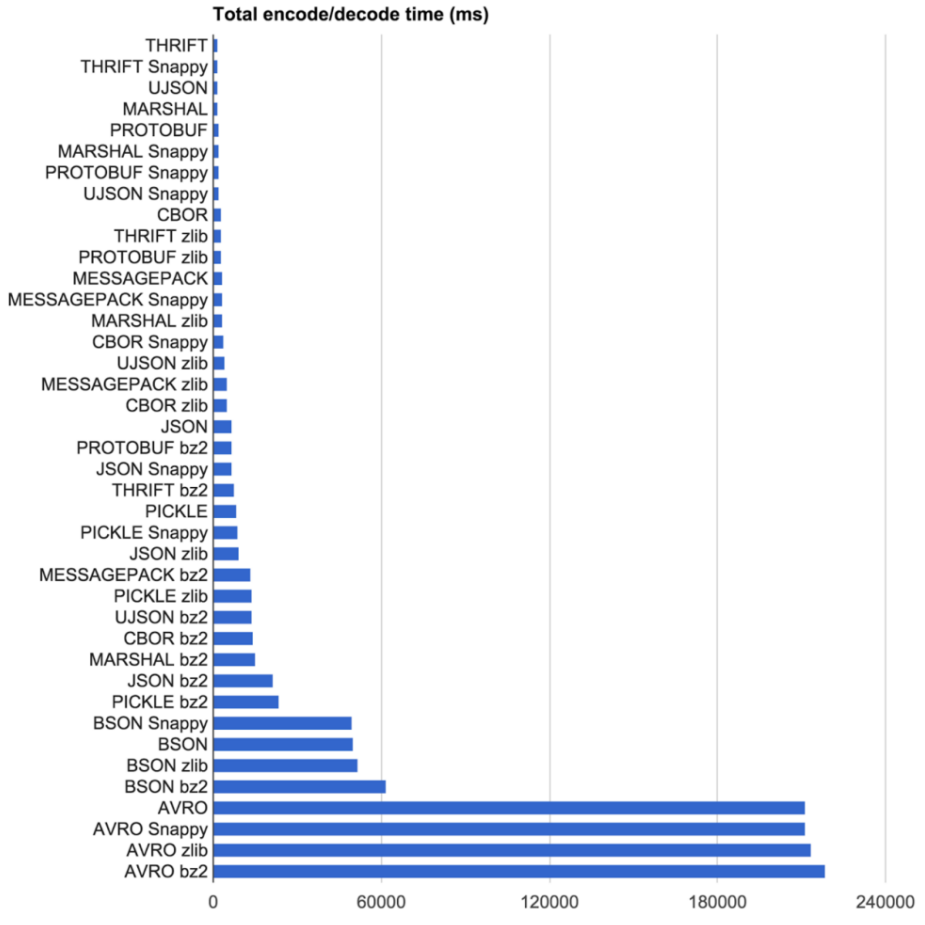
Library	Size in bytes	Time in s
Thrift	278	1:05.12
Protocol Buffers	250	1:19.48
RMI	905	2:14.54
REST - JSON	559	4:44.83
REST - XML	836	5:27.45

https://www.slideshare.net/IgorAnishchenko/pb-vs-thrift-vs-avro

https://github.com/sidshetye/SerializersCompare

RPC with Compression





- Uber: MessagePack with zlib
  - 2,219 pseudorandom anonymized trips

Application

- **gRPC - RPC system by Google**
  - HTTP/2
  - Protocol Buffers
  - Authentication
  - Bidirectional streaming
  - Flow control
  - Blocking or nonblocking bindings
  - Cancellation and timeouts

- **Intellij: New gradle project**
  - Example: https://github.com/tbocek/VSS-grpc
  - Hint: Delegate IDE builds

Apache Avro

- **Avro provides functionality similar to systems such as Thrift, Protocol Buffers**
- **Apache Avro is a data serialization system**
  - A compact, fast, binary data format.
  - A container file, to store persistent data.
  - Developed within Apache's Hadoop project
    - C, C++, C#, Go, Haskell, Java, Perl, PHP, Python, Ruby, Scala

- **Intellij: New gradle project**
  - Example: https://github.com/tbocek/VSS-avro
  - Hint: Delegate IDE builds

URL: b.socrative.com/student/

Room: HSR

Questions?

Prof. Dr. Thomas Bocek
thomas.bocek@hsr.ch