Membrane

Rewrite Pion in Elixir

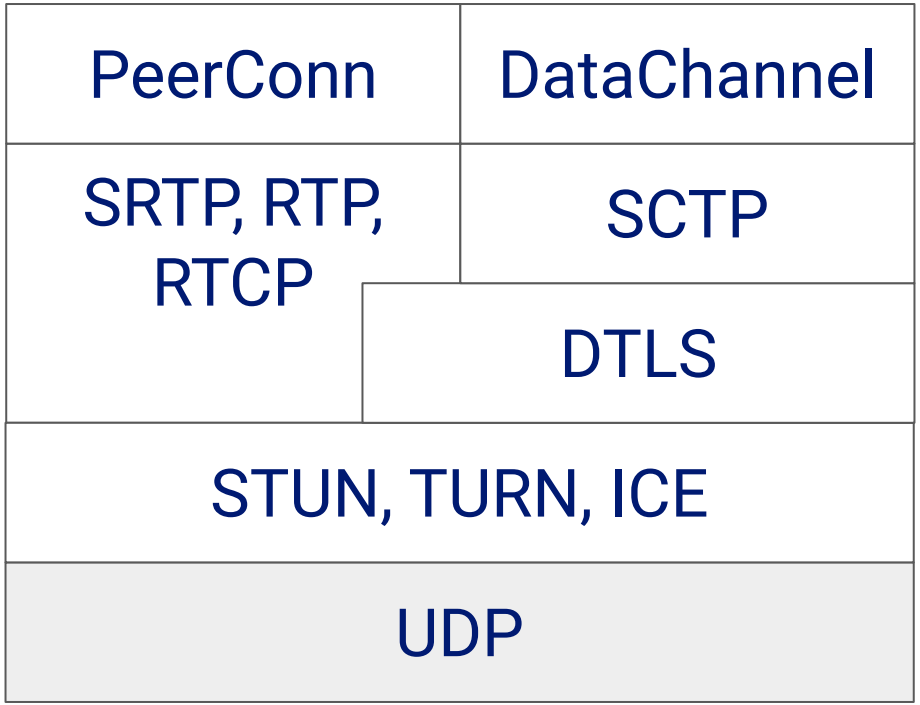
Outline

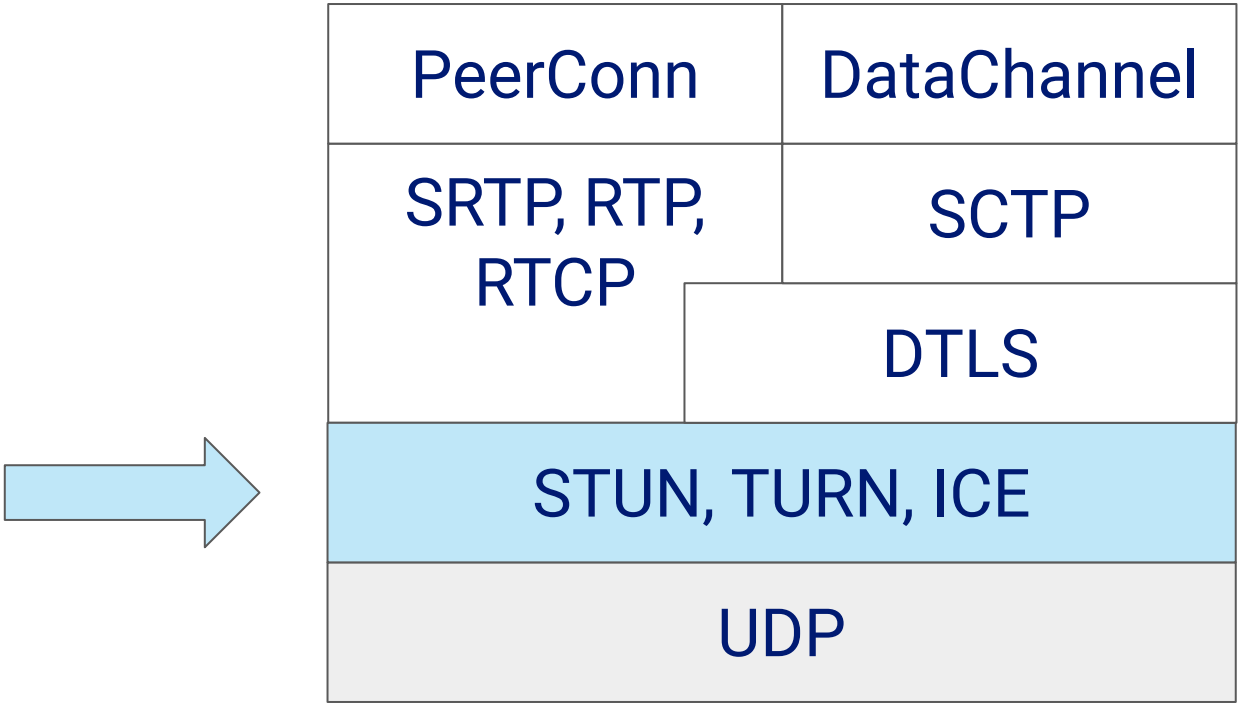
1. Introduction
2. Elixir WebRTC status
3. Libraries

Introduction

What is WebRTC?

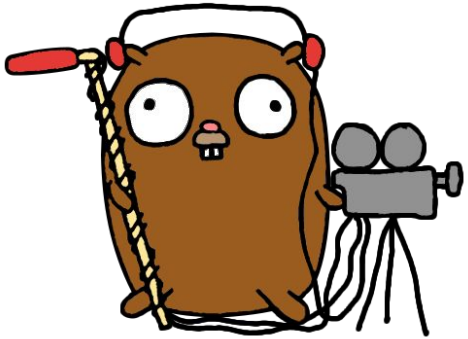
- standard allowing for establishing peer to peer multimedia sessions
- natively implemented by all browsers
- P2P oriented
- complex but powerful


PeerConn
DataChannel
SRTP, RTP, RTCP
SCTP
DTLS
STUN, TURN, ICE
UDP


PeerConn
DataChannel
SRTP, RTP, RTCP
SCTP
DTLS
STUN, TURN, ICE
UDP

Pion

- WebRTC implementation written in Go
- implements everything from scratch
- fast and popular (used by LiveKit)
- large community



Why rewriting Pion in Elixir?

- our API doesn’t follow the official WebRTC API
  - hard for other people to jump in
  - focus on server side
- we use third party protocol implementations
  - lack of detailed understanding
  - it’s hard to participate in the standardization process
- writing in a framework adds an additional overhead

Elixir WebRTC

What have we done?

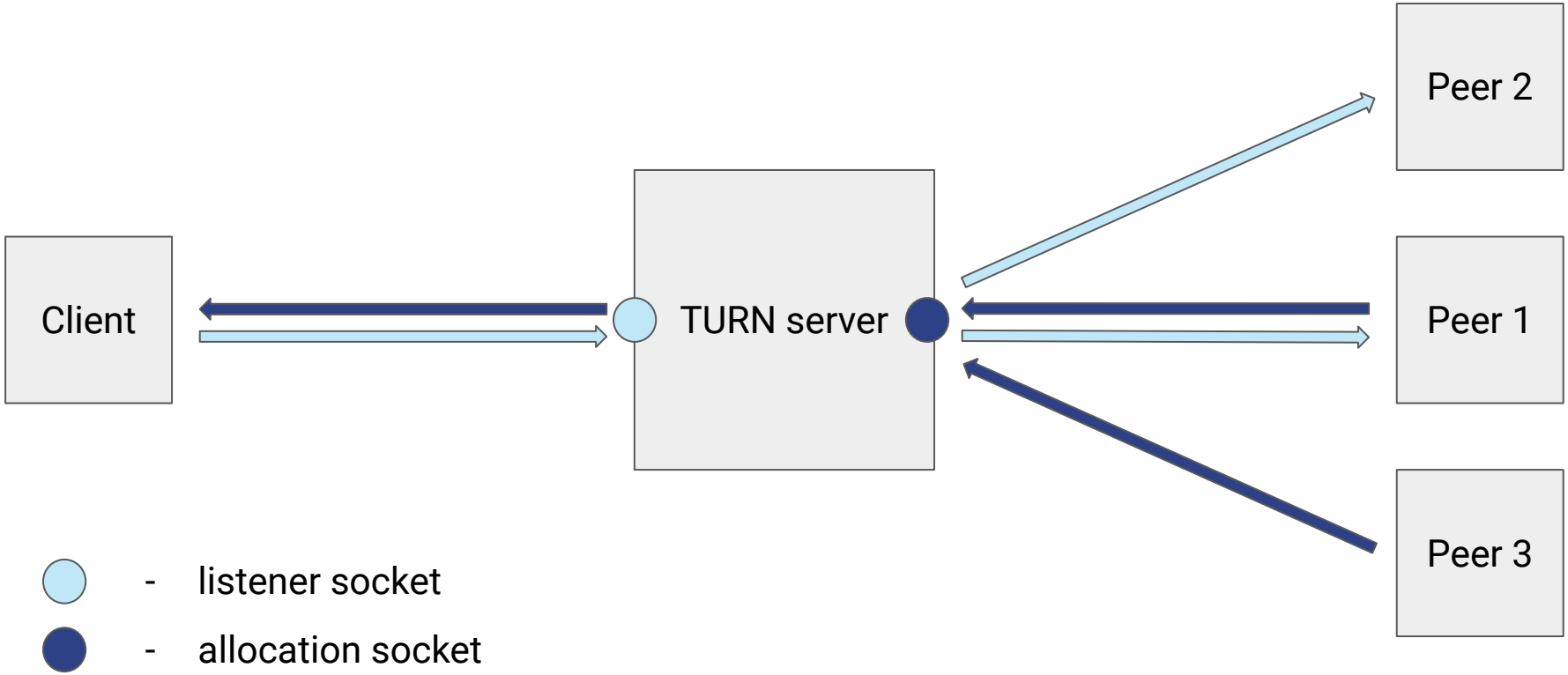
- STUN
- TURN
- full ICE

https://elixir-webrtc.github.io

Libraries

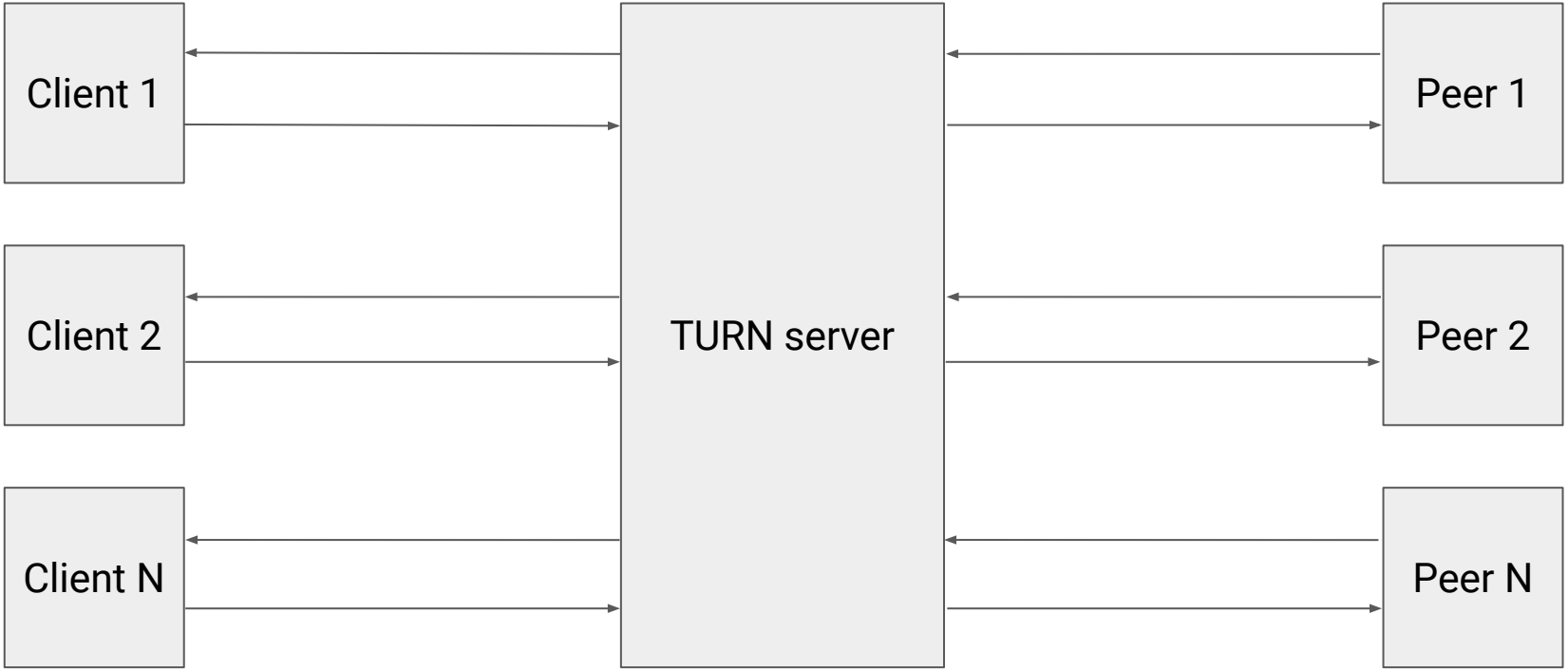
Rel

TURN server

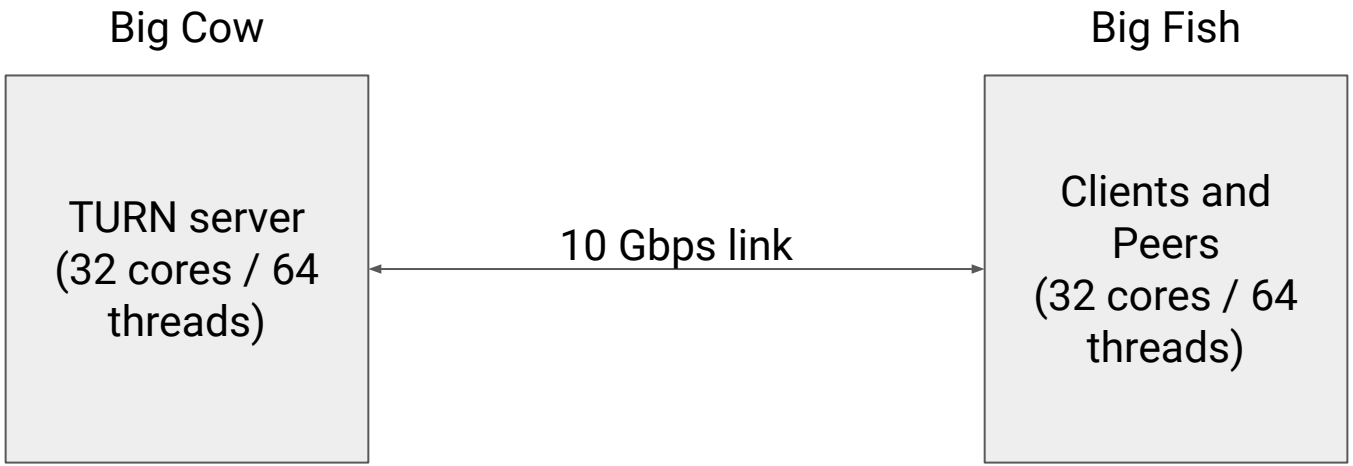

Client
TURN server
Peer 2
Peer 1
Peer 3
- listener socket
- allocation socket

Benchmarks

Scenario

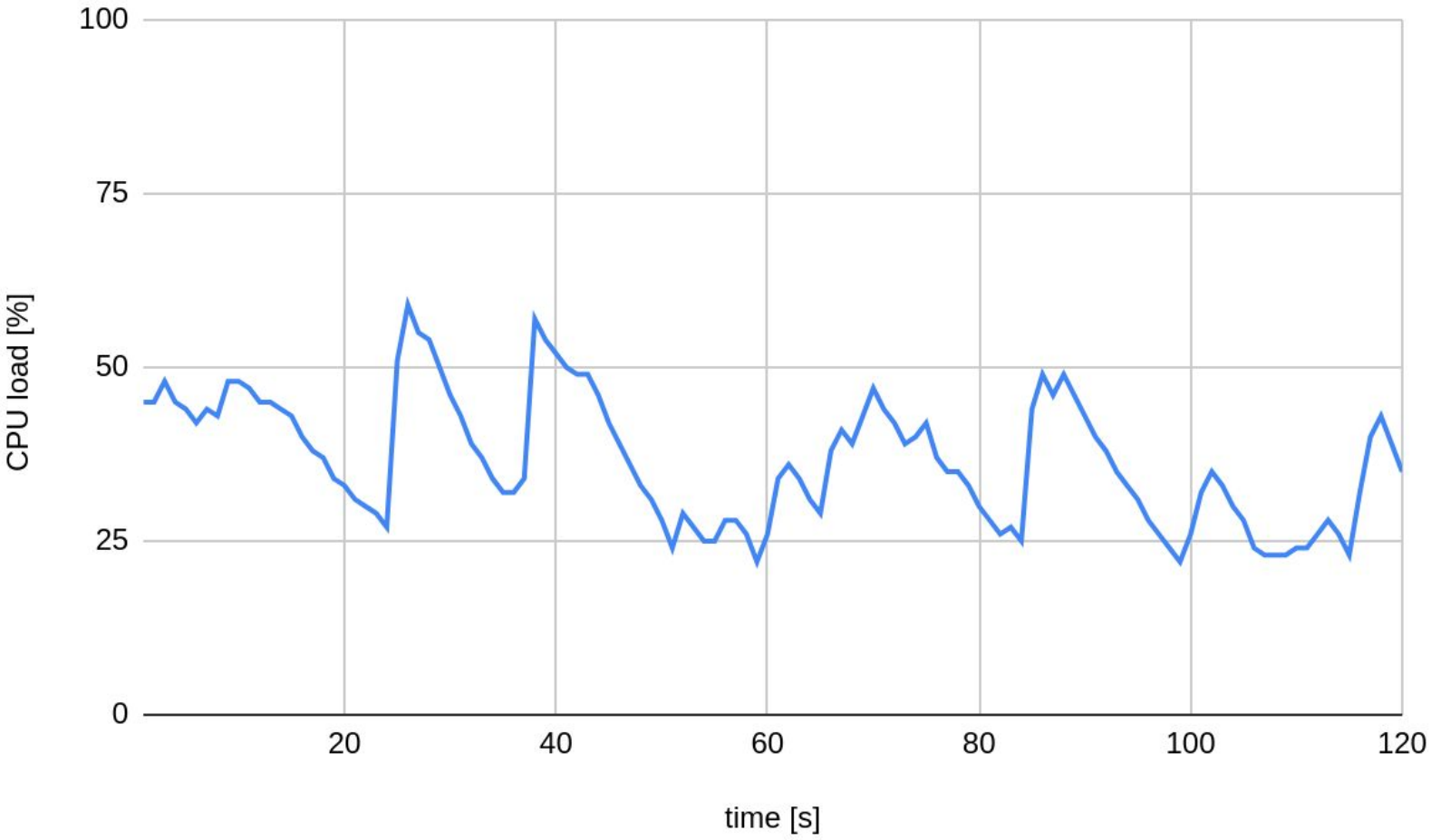


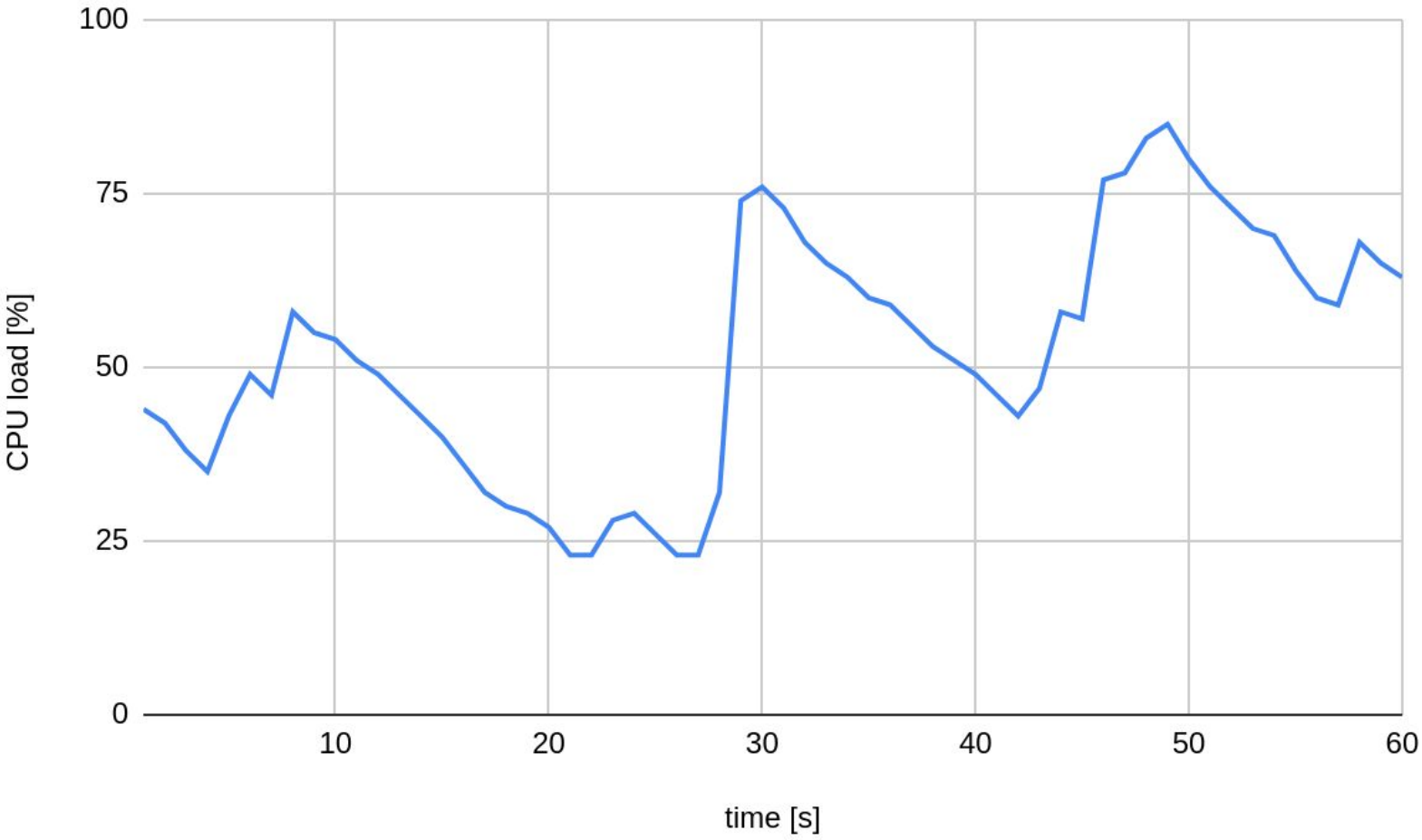
Testbed


Big Cow
Big Fish
TURN server
(32 cores / 64
threads)
10 Gbps link
Clients and
Peers
(32 cores / 64
threads)

Results

Conns	Bitrate in one direction (kbps)	Payload (bytes)	Overall Bitrate (Mbps)	Rel (Elixir)	coTURN (C)	eturnal (Erlang)
2000	50	150	400	20-30% (17-50%)	10%	?
2000	50	1200	400	2-4%	1.3%	?
1000	1500	1200	5200	40-55% (21-80%)	15%	crashes


100
75
50
25
0
CPU load [%]
20
40
60
80
100
120
time [s]


100
75
50
25
0
CPU load [%]
10
20
30
40
50
60
time [s]

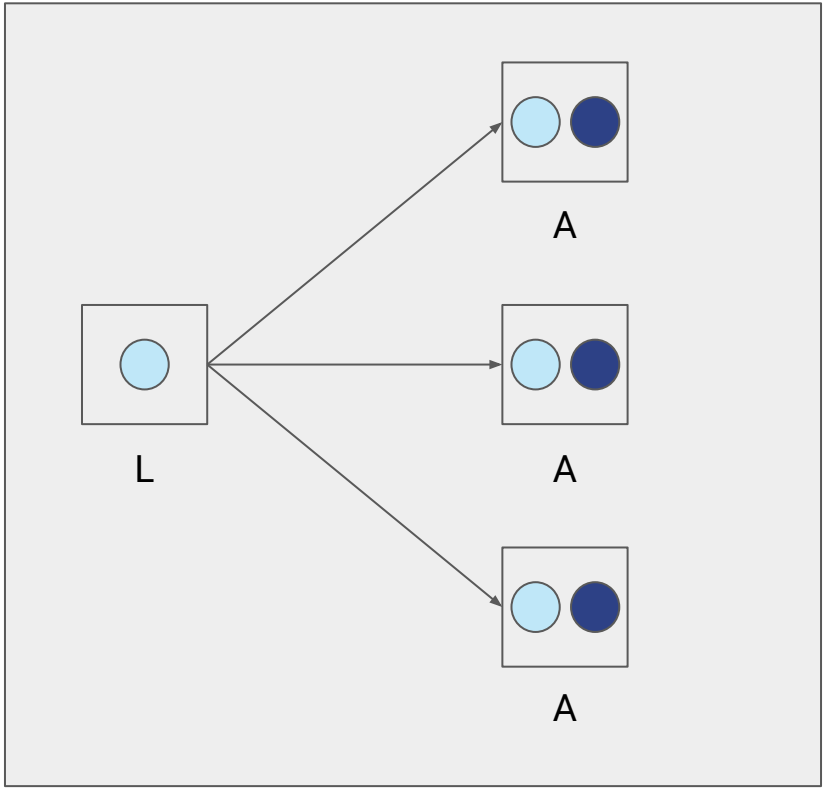
Rel is publicly available!

```
$ curl -X POST
"https://turn.bigcow.ovh/?service=turn&username=john"
```

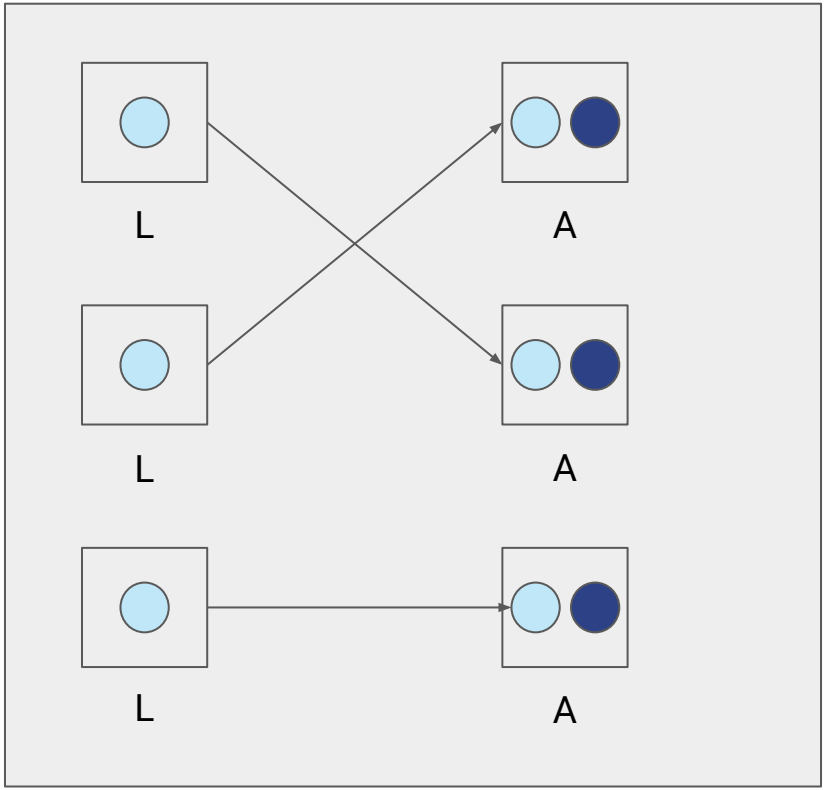

```
{
  "password":"l6hs9SzUgudFeb5XjrfCfOWKeOQ=",
  "ttl":1728,
  "uris":["turn:167.235.241.140:3478?transport=udp"],
  "username":"1691574817:johnsmith"
}
```

```
pc = new RTCPeerConnection({
  iceServers: [
    {
      credential: "l6hs9SzUgudFeb5XjrfCfOWKeOQ=",
      urls: "turn:167.235.241.140:3478?transport=udp",
      username: "1691574817:john"
    }
  ]
});
```

The journey…



- listener socket
- allocation socket



- listener socket
- allocation socket

Bugs

- Enormous CPU usage when overflowing UDP socket with backend inet #7573
- Unable to use reuseport with gen_udp:open #7569
- High CPU usage when using telemetry_metrics_prometheus - not reported yet

Debugging EVM

- observer_cli
- Erlang crash dump
- perf
- lcnt - The Lock Profiler

Perf

```
perf record -- iex --erl "+JPperf true" -S mix
perf record --call-graph=fp --pid 310190 -- sleep 10
perf report
```

```
  Children      Self  Command        Shared Object      Symbol
-    0.94%     0.00%  erts_sched_50  [kernel.kallsyms]  [k] do_syscall_64
   - 0.94% do_syscall_64
      - 0.86% __x64_sys_futex
           0.86% do_futex
+    0.93%     0.00%  erts_sched_32  [kernel.kallsyms]  [k] entry_SYSCALL_64_after_hwframe
+    0.93%     0.00%  erts_sched_3   [kernel.kallsyms]  [k] entry_SYSCALL_64_after_hwframe
+    0.93%     0.00%  erts_sched_62  [kernel.kallsyms]  [k] entry_SYSCALL_64_after_hwframe
+    0.93%     0.00%  erts_sched_58  [kernel.kallsyms]  [k] do_syscall_64
+    0.93%     0.00%  erts_sched_41  [kernel.kallsyms]  [k] entry_SYSCALL_64_after_hwframe
+    0.93%     0.00%  erts_sched_45  [kernel.kallsyms]  [k] entry_SYSCALL_64_after_hwframe
+    0.93%     0.00%  erts_sched_49  [kernel.kallsyms]  [k] do_syscall_64
+    0.93%     0.00%  erts_sched_20  [kernel.kallsyms]  [k] entry_SYSCALL_64_after_hwframe
+    0.93%     0.00%  erts_sched_9   [kernel.kallsyms]  [k] entry_SYSCALL_64_after_hwframe
+    0.93%     0.01%  erts_sched_1   [kernel.kallsyms]  [k] entry_SYSCALL_64_after_hwframe
```

Example

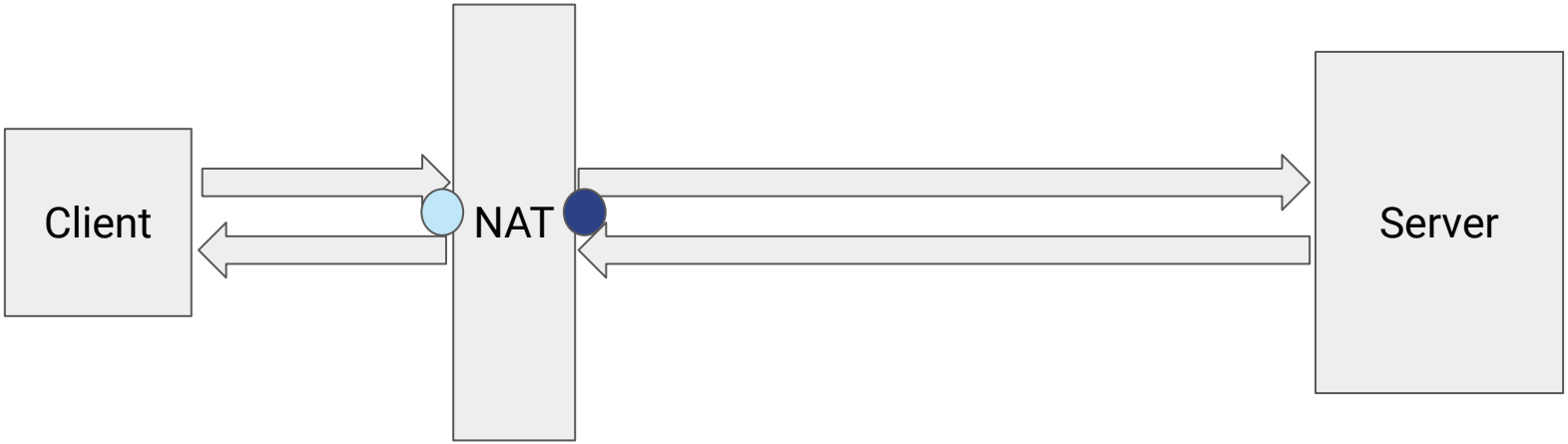
Takeaways

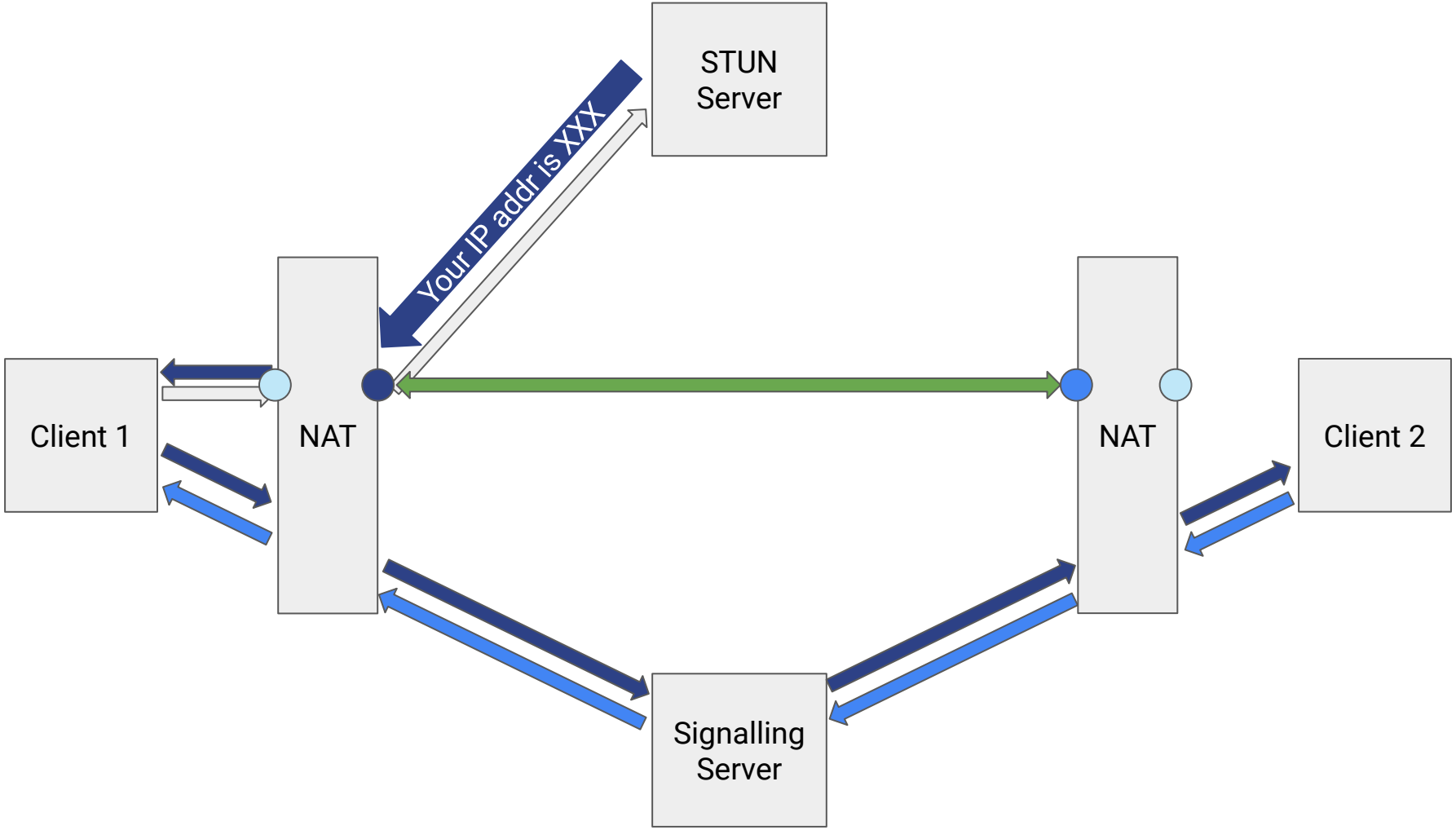
- test at a really big scale
- automate benchmarks, deployment
- the more configuration options, the better
- provide telemetry at first stages of your project

ICE

Allows to establish P2P connection between two hosts in private networks

How does the Internet work?




STUN
Server
Your IP addr is XXX
Client 1
NAT
NAT
Client 2
Signalling
Server

ExICE

- compatible both with aggressive and regular nomination
- role conflict resolution
- supports host, prflx, srflx and remote relay candidates
- transaction pacing
- keepalives on valid and selected pairs

Example

FAQ

RTP: Some Frequently Ask

cs.columbia.edu/~hgs/rtp/faq.html

Some Frequently Asked Questions about RTP

- Is RTP a transport protocol or a kind of application protocol?
- RTP does not ensure real-time delivery. So how come it is called a real-time protocol?
- Is RTP an unreliable protocol? Are there any mechanisms provided for error recovery in RTP?
- Can RTP run over IPv6? ATM?
- Can RTP be used in asymmetric networks?
- Why doesn't RTP have a length field?
- Does RTP have a fixed packetization interval?
- Are all these fields *really* needed?
- How does padding work?
- Practically speaking, how is the timestamp computed?
- In a multimedia conference, are the initial timestamp values related?
- What are the roles of the RTP timestamp and sequence numbers?
- What are the different clocks and how are they synchronized?
- What's the marker bit good for?
- What is the sender packet count and byte count used for?
- What is the RTP timestamp in the RTCP sender report used for?
- How is the jitter computed?
- What is the session bandwidth?
- What is the use of RTCP for two-party calls?
- How do I register an RTP payload type?
- What is the current list of RTP payload types?
- What are dynamic payload types?
- If I'm using H.323 or other set-up protocol, can I ignore the RTP payload type (PT) field?
- Should the RTP payload type (PT) field be used for multiplexing different streams?
- Should the RTP SSRC be used for multiplexing different streams from the same source?
- Do receivers need their own SSRC identifiers?
- Why can't we just use TCP for audio and video?
- Can't we just use XTP?
- How should RTP sessions be played back?
- Is there an RTP library or kernel implementation?
- What are some of the differences between the VAT protocol and RTP?
- What are the differences between RTP version 1 and 2?
- Are there specific ports assigned to RTP?
- How are ports assigned for bidirectional unicast RTP sessions?
- Is there a way for a router to tell apart RTP packets?
- What about firewalls?
- What is the quality of audio codec X?
- Are all audio codecs patented?
- Are there related ITU efforts?

Is RTP a transport protocol or a kind of application protocol?

RTP has important properties of a transport protocol: it runs on end systems, it provides demultiplexing. It differs from transport protocols like TCP in that it (currently) does not offer any form of reliability or a protocol-defined flow/congestion control. However, it provides the necess hooks for adding reliability, where appropriate, and flow/congestion control. Some like to refer to this property as application-level framing (see D. Clark and D. Tennenhouse, "Architectural considerations for a new generation of protocols", SIGCOMM'90, Philadelphia). RTP so far h been mostly implemented within applications, but that has no bearing on its role. TCP is still a transport protocol even if it is implemented as part of an application rather than the operating system kernel.

RTP does not ensure real-time delivery. So how come it is called a real-time protocol?

No end-to-end protocol, including RTP, can ensure in-time delivery. This always requires the support of lower layers that actually have control over resources in switches and routers. RTP provides functionality suited for carrying real-time content, e.g., a timestamp and control mechanisms for synchronizing different streams with timing properties.

Is RTP an unreliable protocol? Are there any mechanisms provided for error recovery in RTP?

FAQ ICE

1. Should we filter out bridges (like those created by docker) when gathering host candidates?
2. When conn check req and resp address might not be symmetric? sec 7.2.5.2.1
3. How does iptables work? (In particular NAT table, and MASQUERADE and DNAT targets)
4. Is it actually possible to have srflx candidate? reflexive candidates are replaced by their bases and pruned if redundant - see section 6.1.2.4
5. Is it possible to connect ice outside docker with ice inside docker?
6. Is it possible for initial pair state to be different than :waiting when we have only one checklist?
7. Is it possible not to prune some srflx candidate - sec 6.1.2.4?
8. Is it possible to receive binding response to binding request with USE-CANDIDATE that will result in creating a new valid pair? sec 7.2.5.3.4
9. A Trickle ICE agent MUST NOT pair a local candidate until it has been trickled to the remote party. How can we know that the candidate has been trickled to the remote party? Does any implementation exposes in its API ability to mark the candidate as trickled?
10. What data use for keepalives and how often?
11. Can ICE be restarted before completing?
12. Can we send data on valid pair before regular nomination in RFC 5245?
13. Why does chrome allocate only one ICE candidate?
14. What is default local address?

1. Should we filter out bridges (like those created by docker) when gathering host candidates?

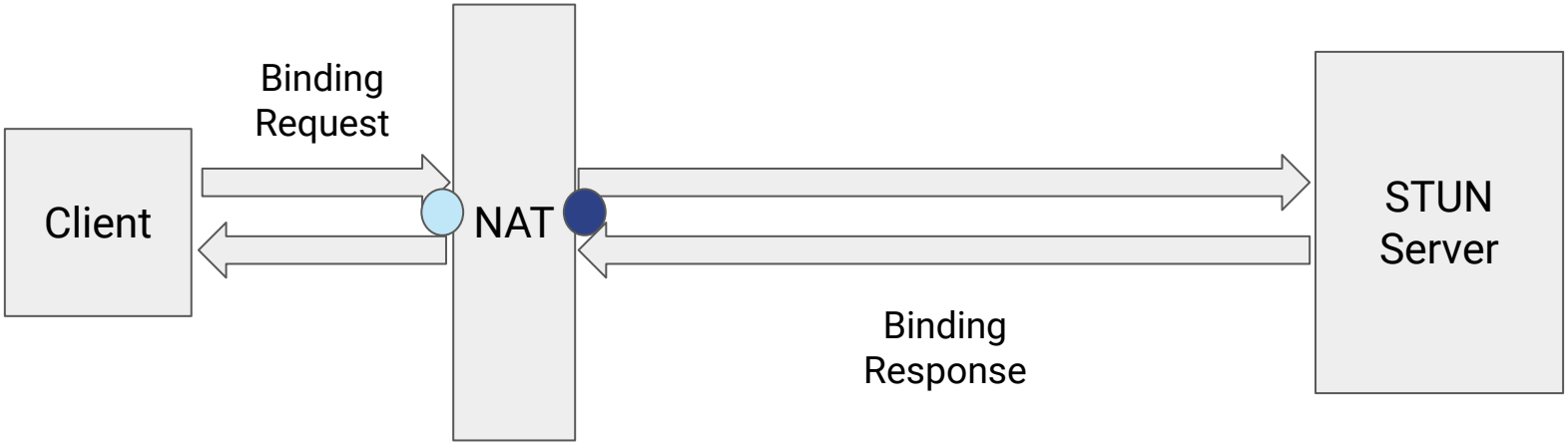
Takeaways

- take care of your logs!
- the fact that you implemented something doesn’t mean you understand it

STUN

- protocol for dealing with NATs
- can be used for:
  - determining public IP address
  - keeping NAT bindings alive
  - doing so called connectivity checks (ICE)
  - sending data (TURN)
- it’s a base for ICE and TURN

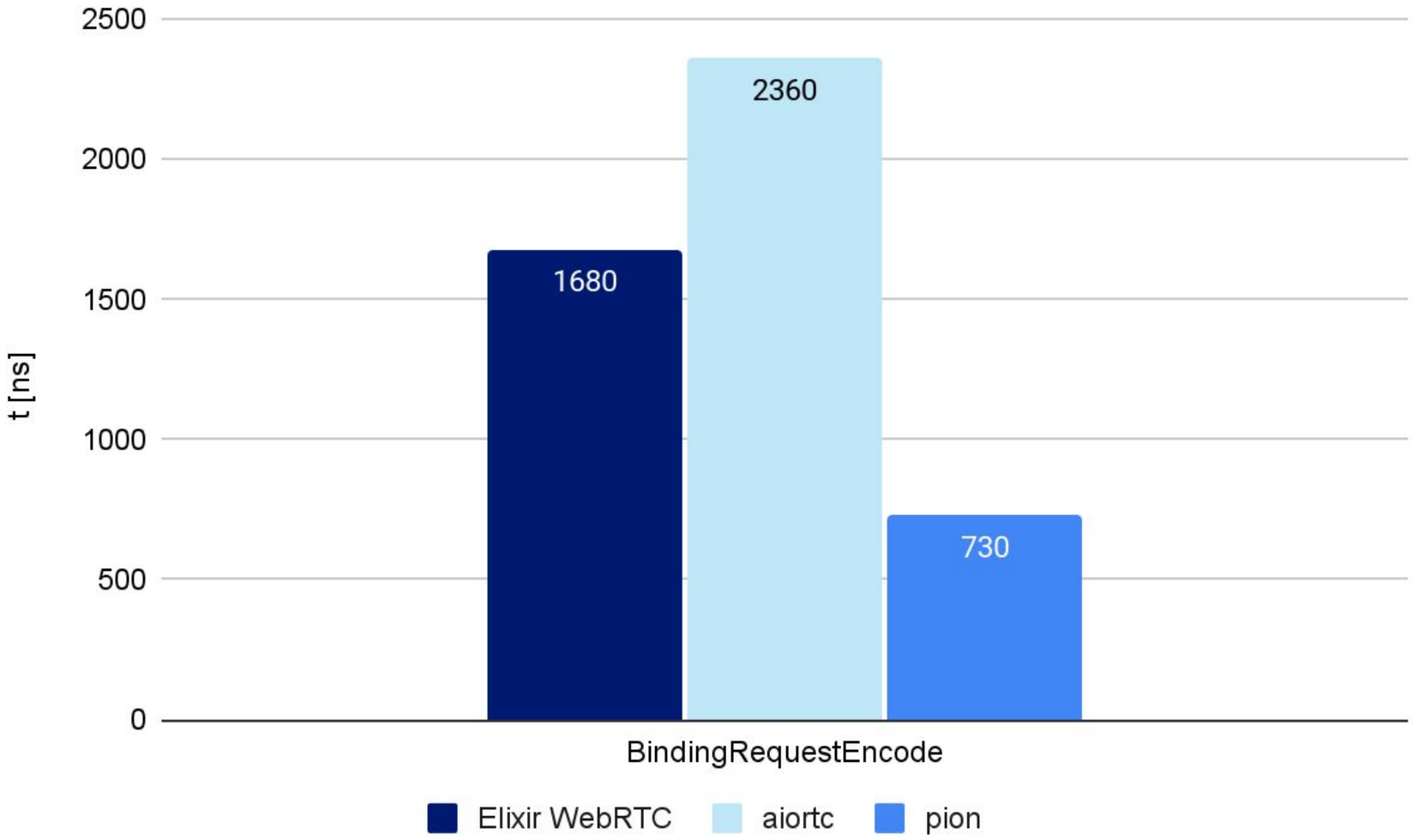
STUN

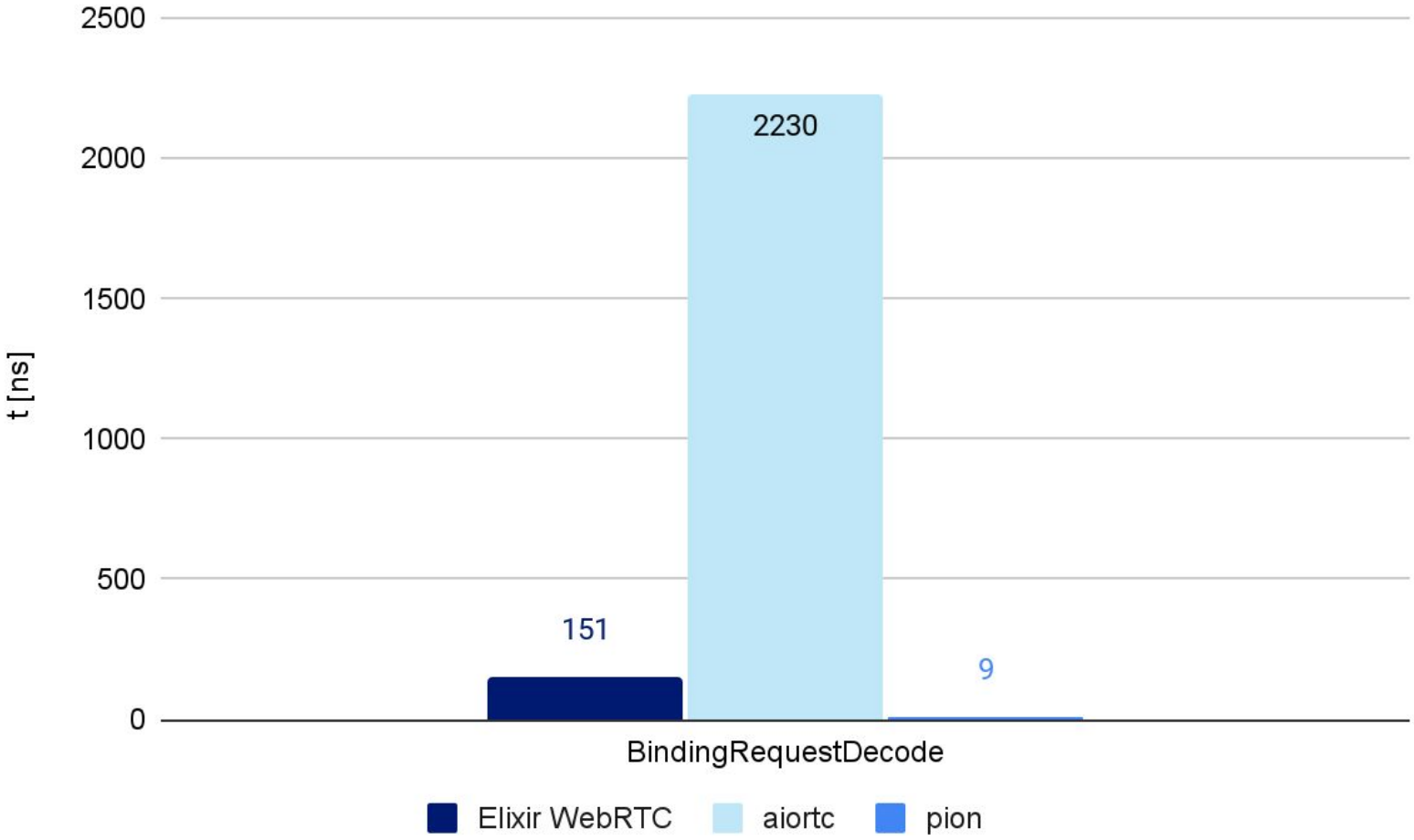


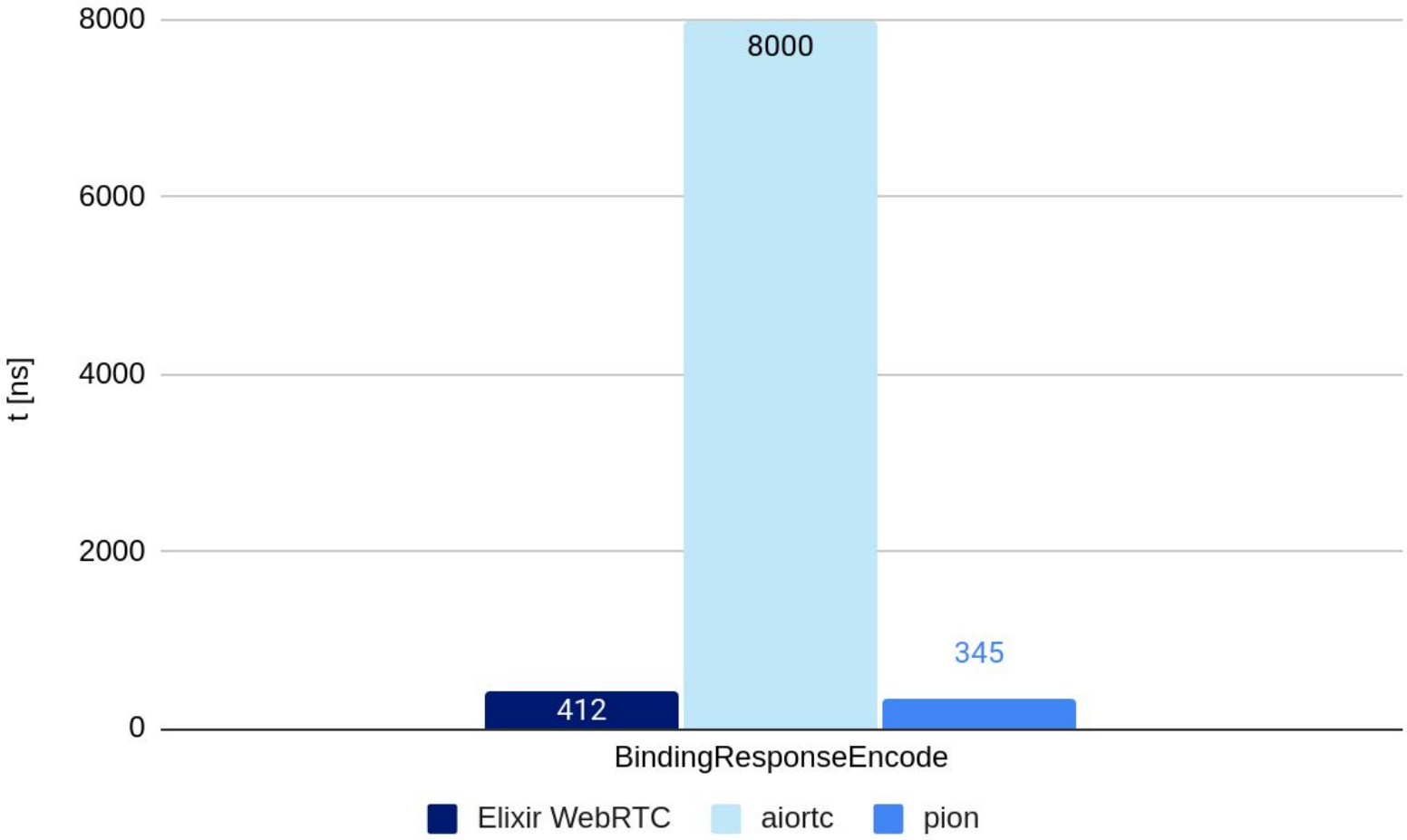
ExSTUN

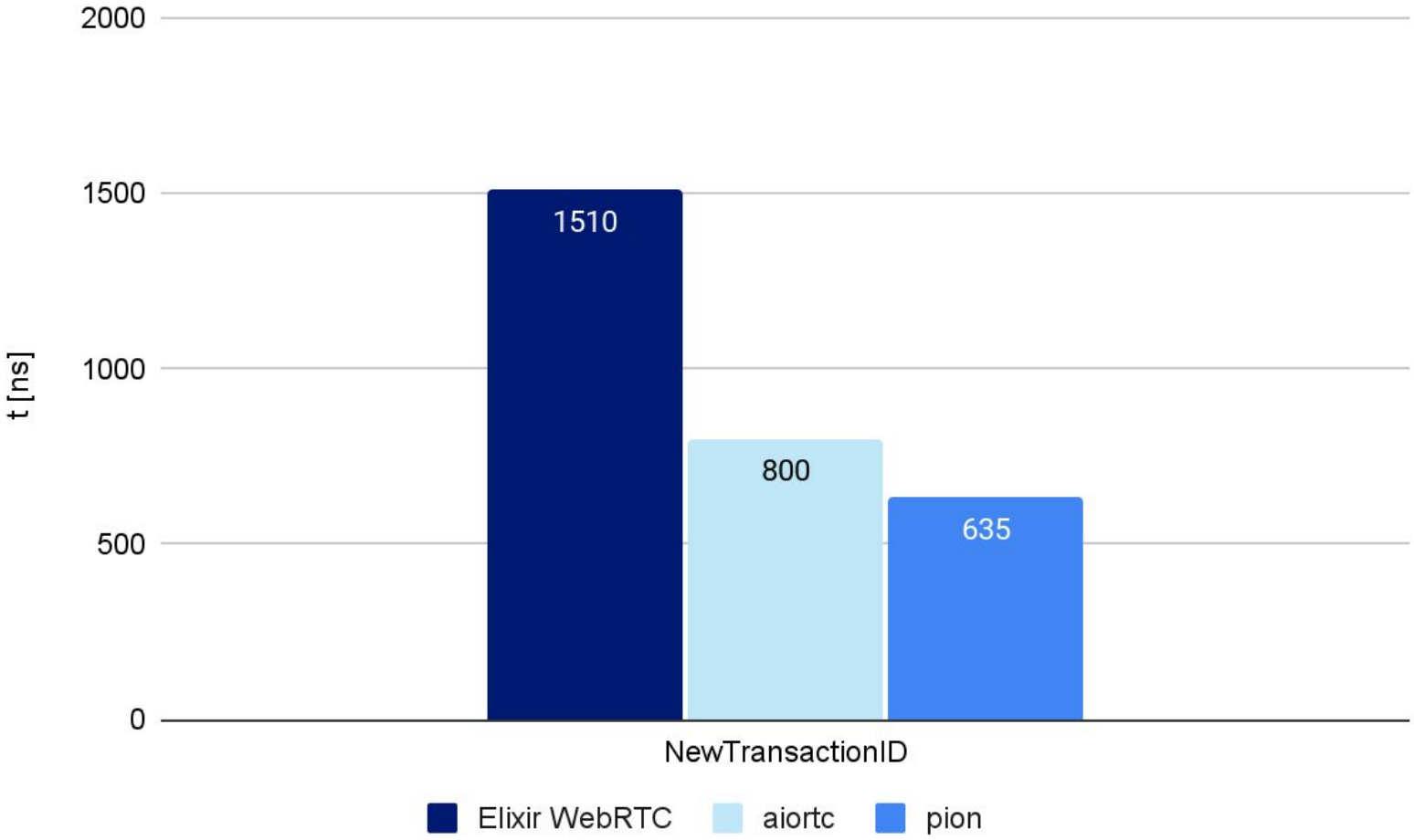
```
req =
  %Type{class: :request, method: :binding}
  |> Message.new()
  |> Message.encode()

:ok = :gen_udp.send(socket, 'stun.l.google.com', 19302, req)

{:ok, {_, _, resp}} = :gen_udp.recv(socket, 0)

{:ok, msg} = Message.decode(resp)
Message.get_attribute(msg, XORMappedAddress)
```

Benchmarks


2500
2000
1500
1000
500
0
t [ns]
2360
1680
730
BindingRequestEncode
Elixir WebRTC
aiortc
pion


2500
2000
1500
1000
500
0
t [ns]
2230
151
9
BindingRequestDecode
Elixir WebRTC
aiortc
pion


8000
6000
4000
2000
0
t [ns]
8000
412
345
BindingResponseEncode
Elixir WebRTC
aiortc
pion


2000
1500
1000
500
0
t [ns]
1510
800
635
NewTransactionID
Elixir WebRTC
aiortc
pion

Takeaways

- `ERL_COMPILER_OPTIONS=bin_opt_info mix compile`
- don’t optimize too early

Future plans

- already started working on RTP/RTCP
- transfer some already existing libs from Membrane Framework to Elixir WebRTC
- first PeerConnection version in 4 months

Thank you!

- https://github.com/elixir-webrtc
- https://elixir-webrtc.github.io