WebRTC

husseinnasser.com

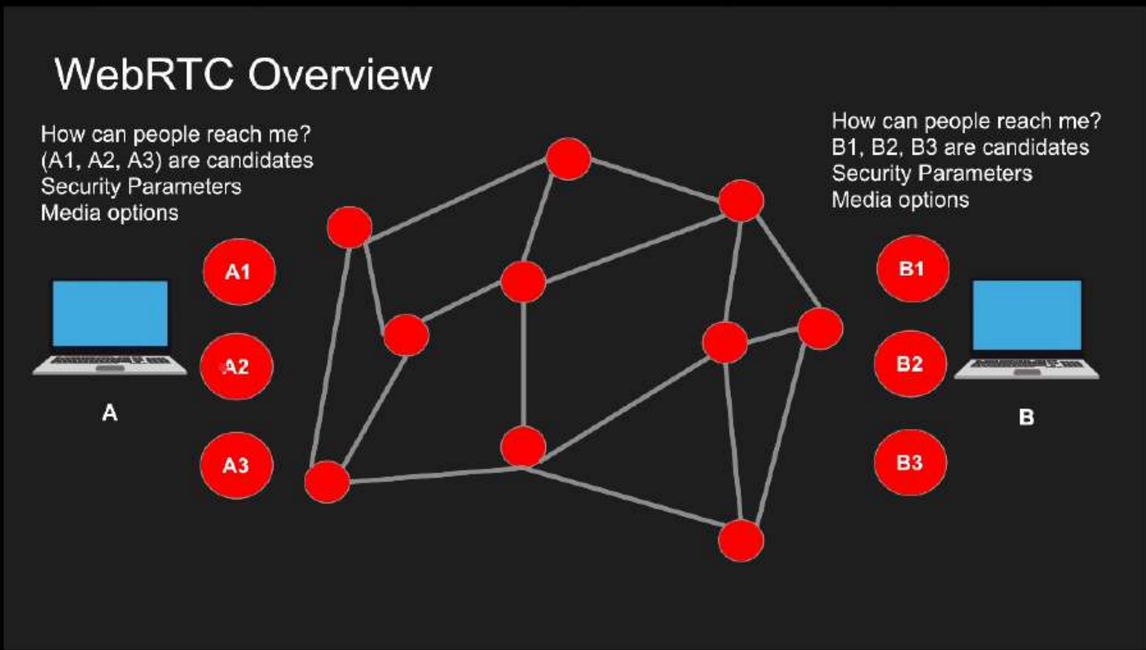
Web Real-Time Communication

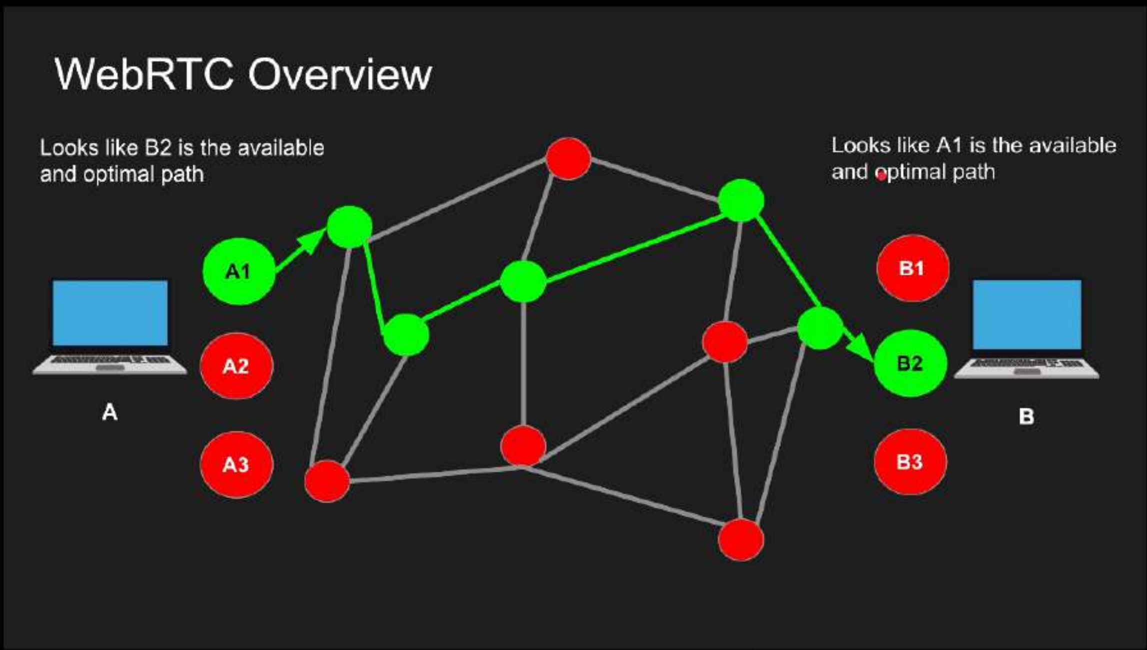
WebRTC Overview

- Stands for Web Real-Time Communication
- Find a peer to peer path to exchange video and audio in an efficient and low latency manner
- Standardized API
- Enables rich communications browsers, mobile, IOT devices

WebRTC Overview

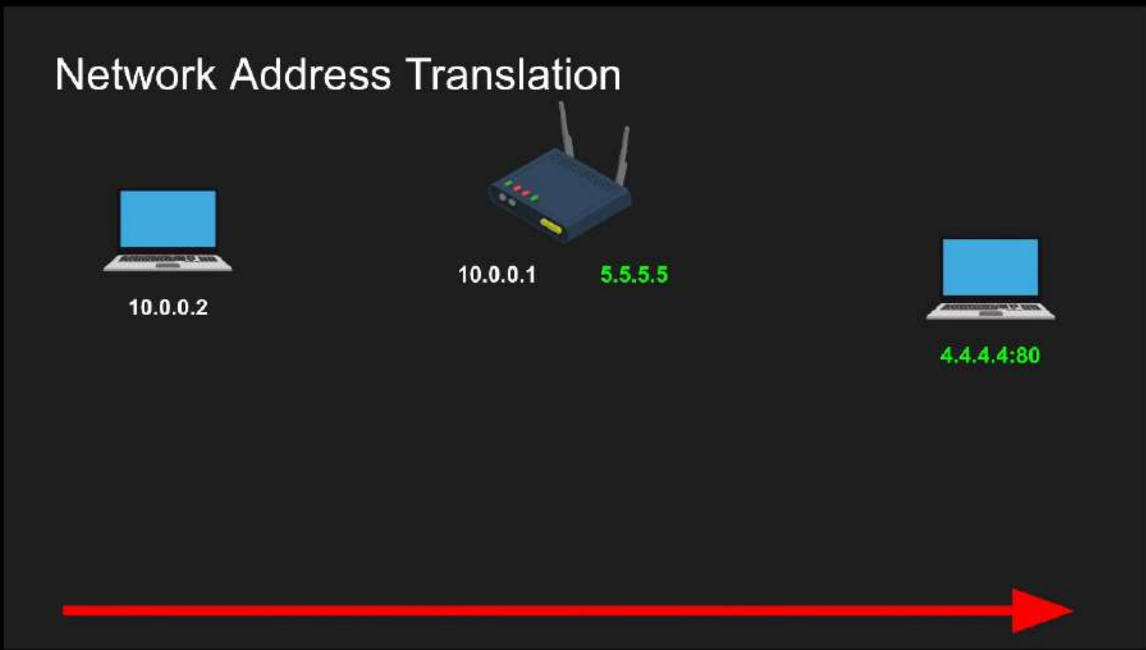
- A wants to connect to B
- A finds out all possible ways the public can connect to it
- B finds out all possible ways the public can connect to it
- A and B signal this session information via other means
    - WhatsApp, QR, Tweet, WebSockets, HTTP Fetch..
- A connects to B via the most optimal path
- A & B also exchanges their supported media and security

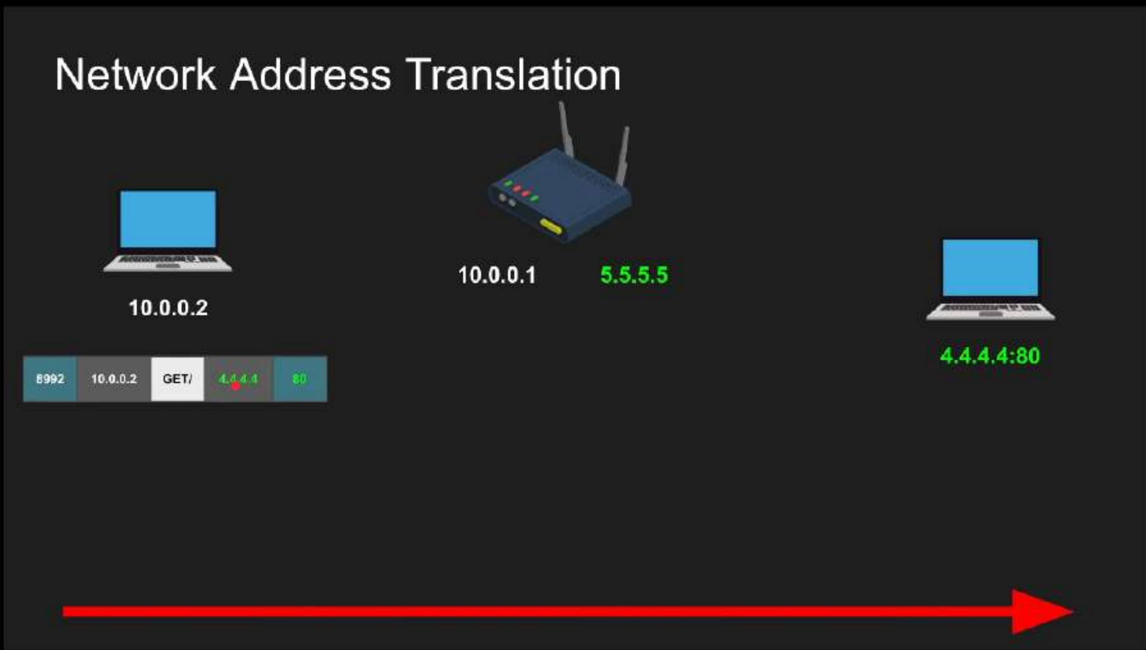

WebRTC Overview
How can people reach me?
(A1, A2, A3) are candidates
Security Parameters
Media options
A1
A2
A3
A
How can people reach me?
B1, B2, B3 are candidates
Security Parameters
Media options
B1
B2
B3
B

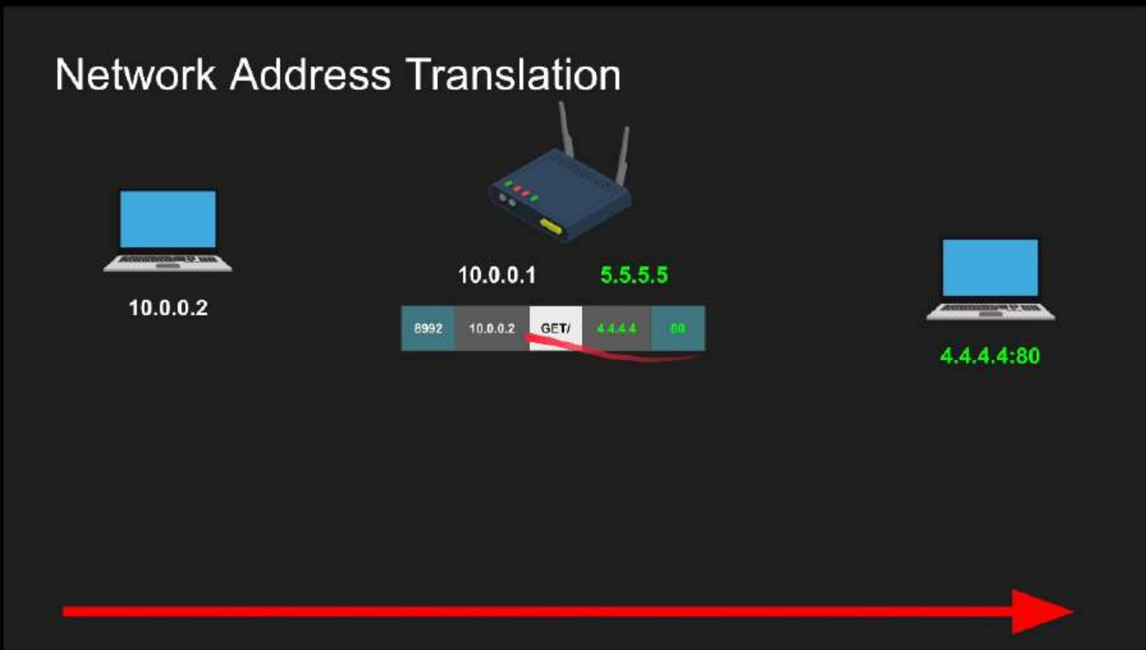

WebRTC Overview
Looks like B2 is the available
and optimal path
Looks like A1 is the available
and optimal path
A1
A2
A3
A
B1
B2
B3
B

WebRTC Demystified

- NAT
- STUN, TURN
- ICE
- SDP
- Signaling the SDP


Network Address Translation
10.0.0.2
10.0.0.1
5.5.5.5
4.4.4.4:80


Network Address Translation
10.0.0.1
5.5.5.5
10.0.0.2
4.4.4.4:80
8992
10.0.0.2
GET/
4.4.4.4
80


Network Address Translation
10.0.0.2
10.0.0.1
5.5.5.5
8992
10.0.0.2
GET/
4.4.4.4
80
4.4.4.4:80

Network Address Translation

Internal IP	Internal Port	External IP	Ext. Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80

Network Address Translation

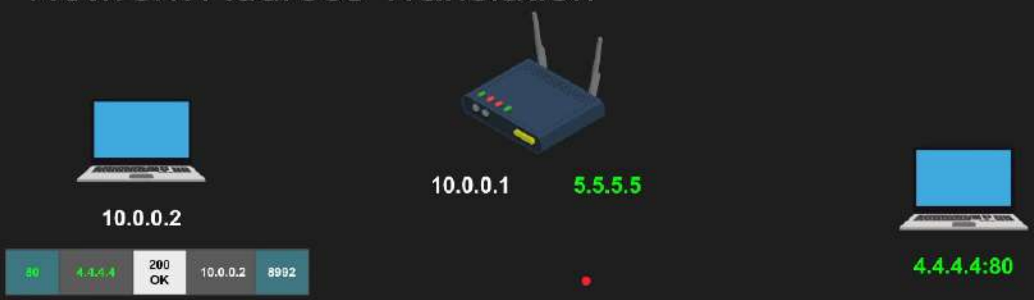
Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80

Network Address Translation

Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80

Slide 9 Q&A Notes Pointer Captions Tips EXIT

Network Address Translation



Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80

NAT Translations Method

- One to One NAT (Full-cone NAT)
- Address restricted NAT
- Port restricted NAT
- Symmetric NAT

One to One NAT (Full cone NAT)

- Packets to external IP:port on the router always maps to internal IP:port without exceptions

Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80

One to One NAT

Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80
10.0.0.2	9999	5.5.5.5	4444	3.3.3.3	80

Address Restricted NAT

- Packets to external IP:port on the router always maps to internal IP:port as long as source address from packet matches the table (regardless of port)
- Allow if we communicated with this host before

Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80

Address Restricted NAT

Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80
10.0.0.2	9999	5.5.5.5	4444	3.3.3.3	80

Port Restricted NAT

Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80
10.0.0.2	9999	5.5.5.5	4444	3.3.3.3	80
10.0.0.2	8888	5.5.5.5	2222	3.3.3.3	8080

Symmetric NAT

- Packets to external IP:port on the router always maps to internal IP:port as long as source address and port from packet matches the table
- Only Allow if the full pair match

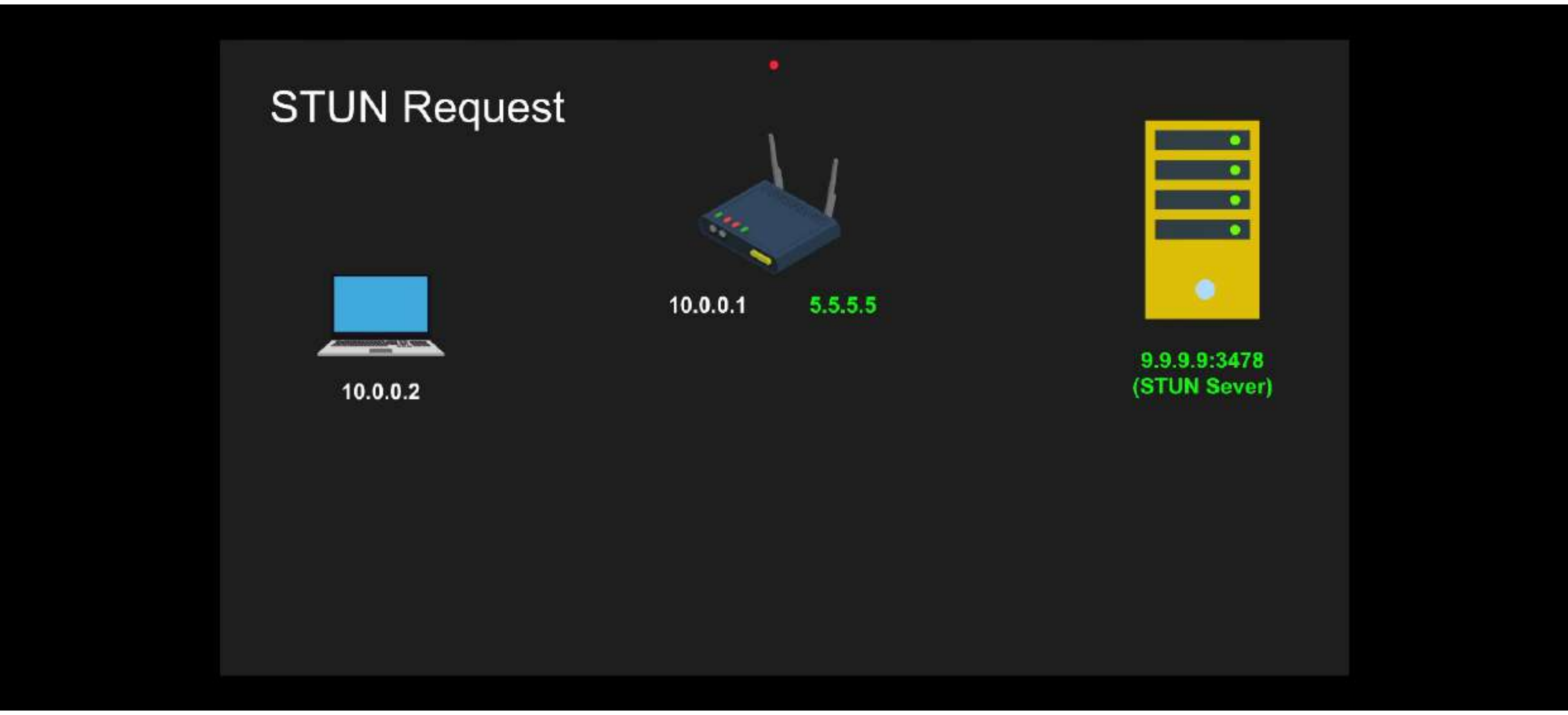
Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80

Symmetric NAT

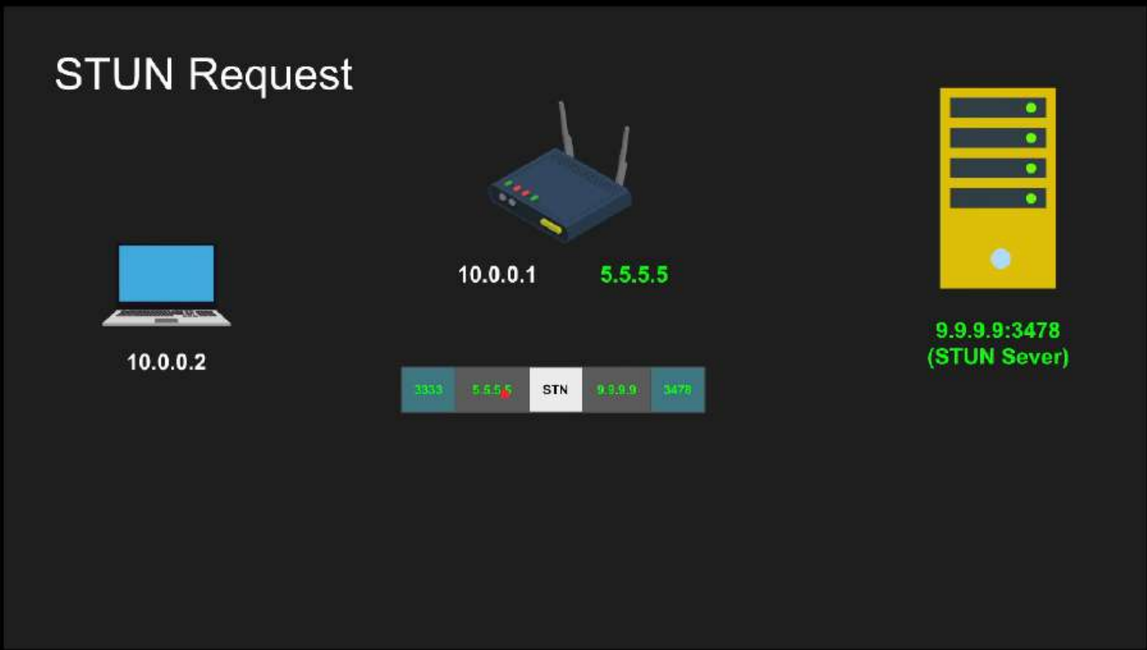
Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	4.4.4.4	80
10.0.0.2	9999	5.5.5.5	4444	3.3.3.3	80
10.0.0.2	8888	5.5.5.5	2222	3.3.3.3	8080

STUN

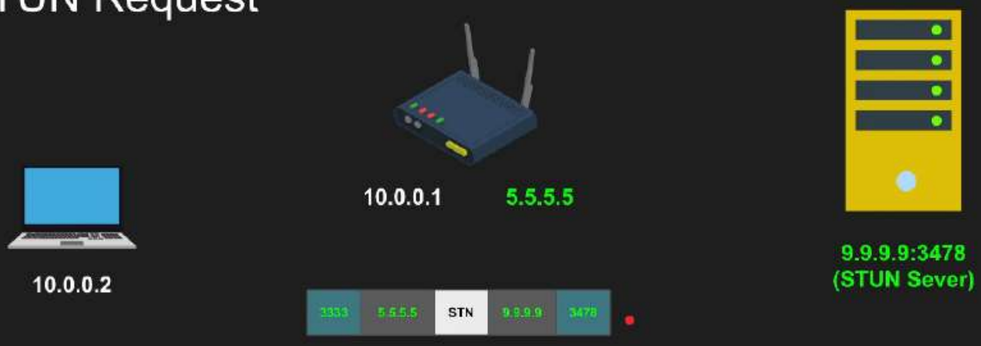
- Session Traversal Utilities for NAT
- Tell me my public ip address/port through NAT
- Works for Full-cone, Port/Address restricted NAT
- Doesn't work for symmetric NAT
- STUN server port 3478, 5349 for TLS
- Cheap to maintain


STUN Request
10.0.0.1
5.5.5.5
10.0.0.2
9.9.9.9:3478
(STUN Sever)

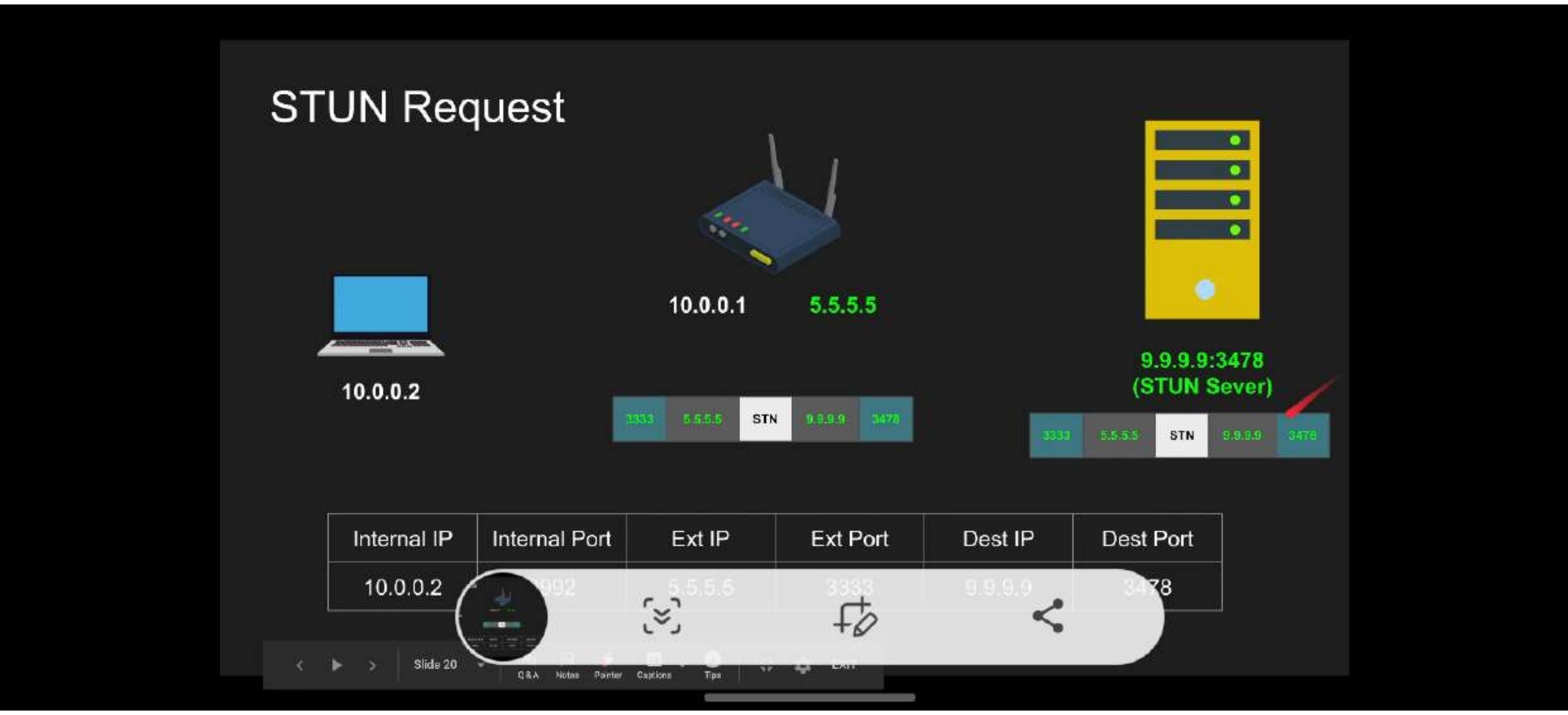

STUN Request
10.0.0.1
5.5.5.5
10.0.0.2
8992
10.0.0.2
STN
9.9.9.9
3478
9.9.9.9:3478
(STUN Sever)


STUN Request
10.0.0.1
5.5.5.5
10.0.0.2
3333
5.5.5.5
STN
9.9.9.9
3478
9.9.9.9:3478
(STUN Sever)

STUN Request

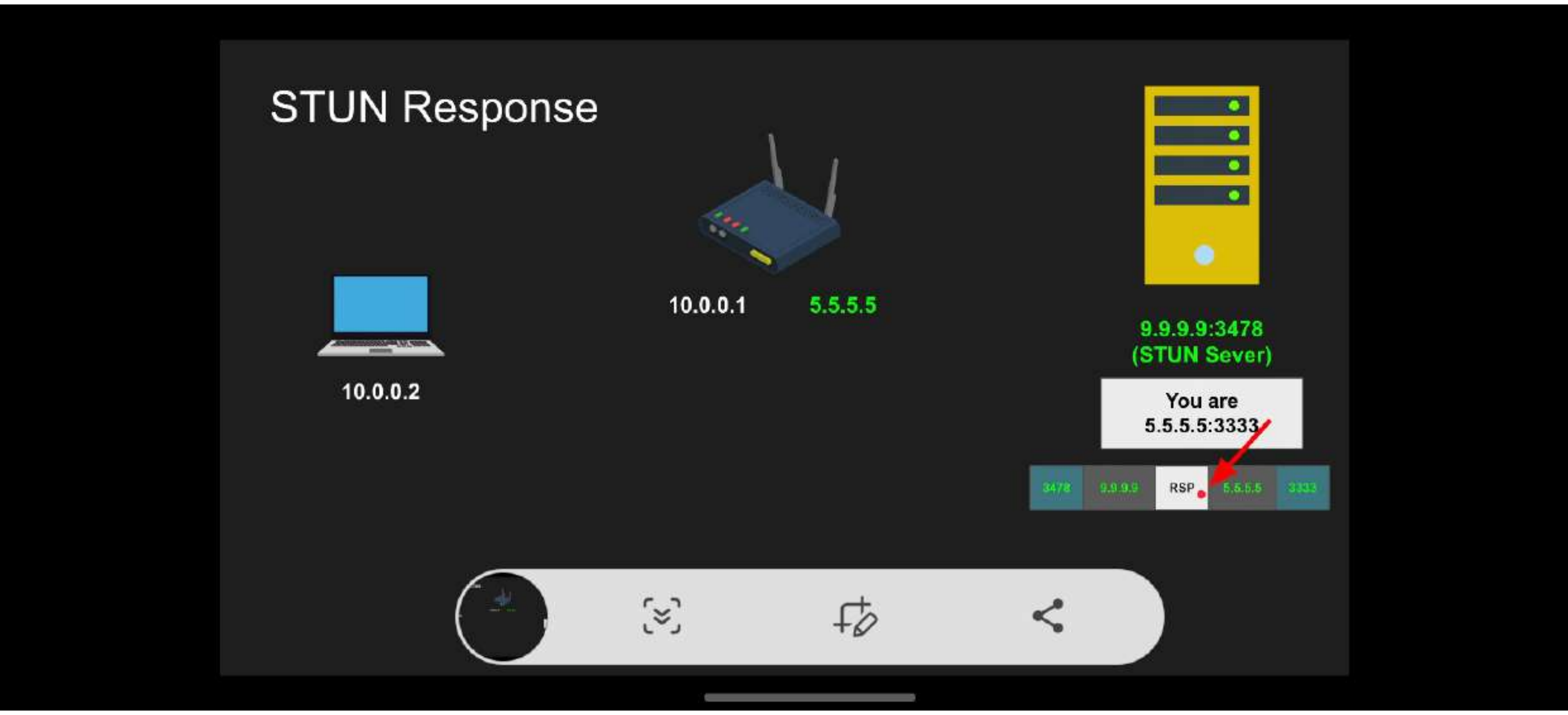


Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	9.9.9.9	3478

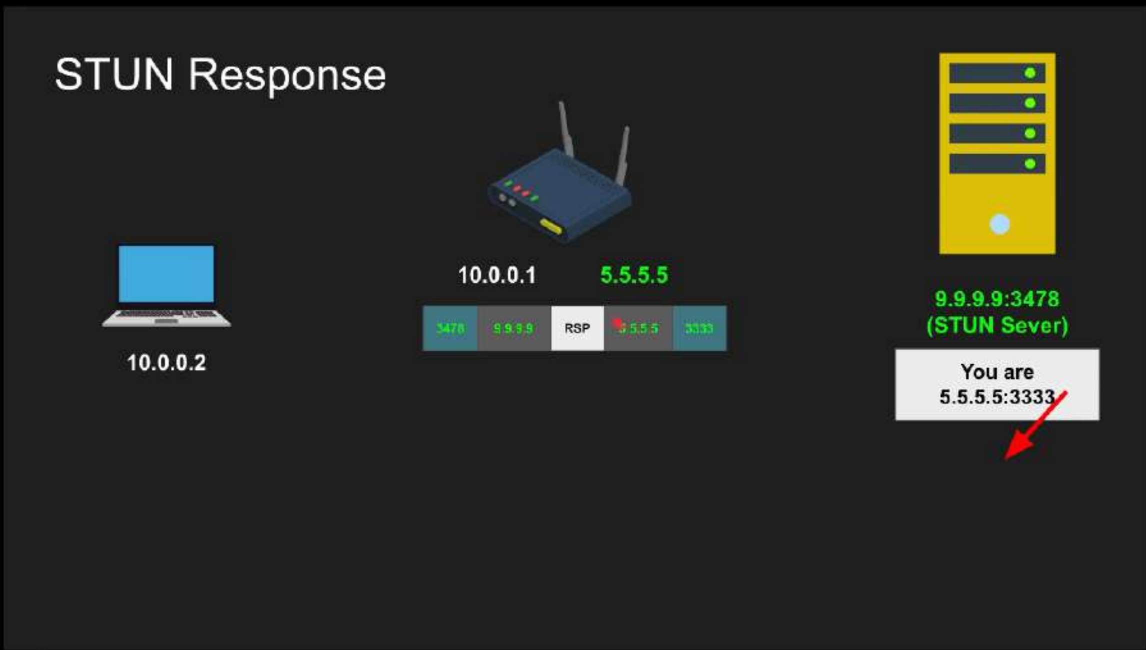


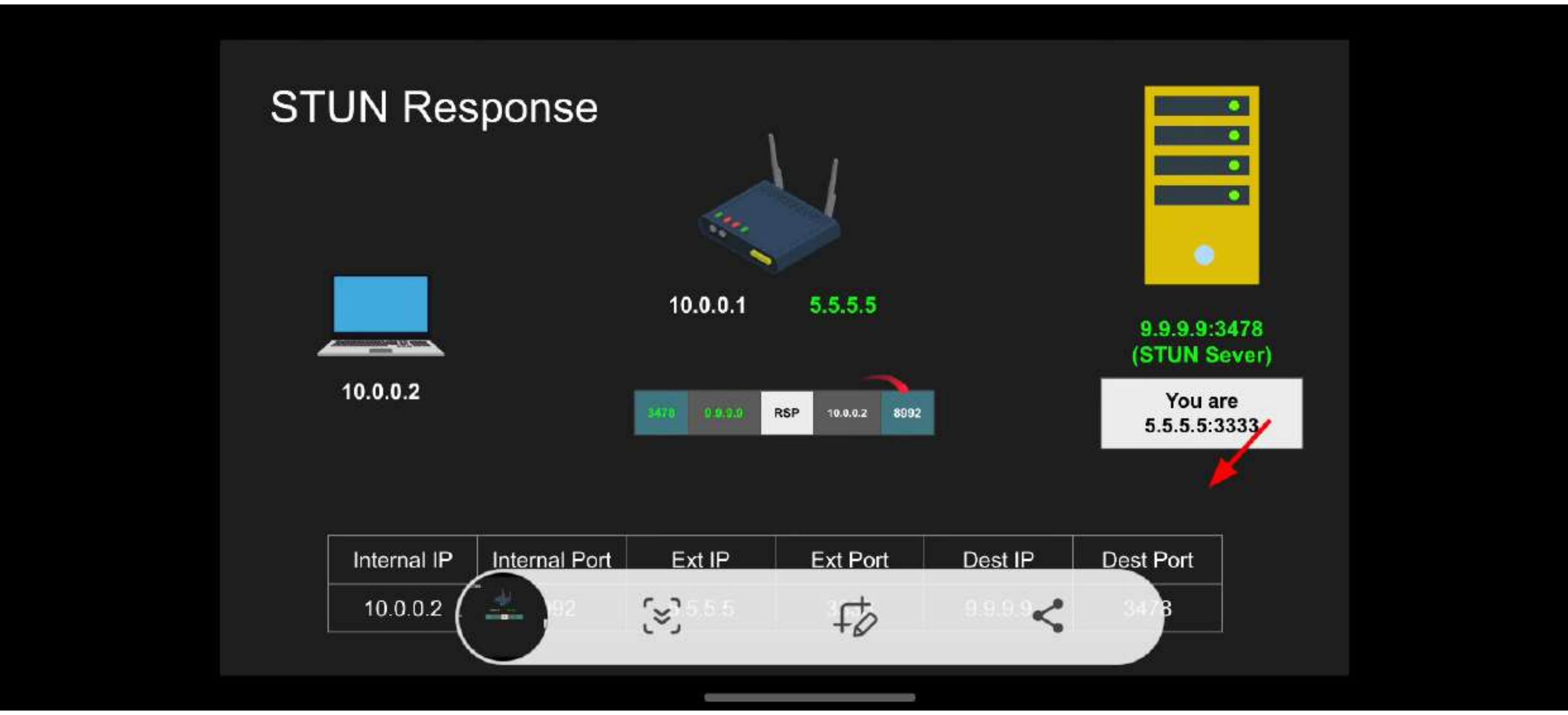
STUN Request
10.0.0.2
10.0.0.1
5.5.5.5
9.9.9.9:3478
(STUN Sever)
3333
5.5.5.5
STN
9.9.9.9
3478
Internal IP
Internal Port
Ext IP
Ext Port
Dest IP
Dest Port
10.0.0.2
5.5.5.5
3333
9.9.9.9
Slide 20
Q&A
Notes
Pointer
Captions
Tips


STUN Response
10.0.0.2
10.0.0.1
5.5.5.5
9.9.9.9:3478
(STUN Sever)
You are
5.5.5.5:3333

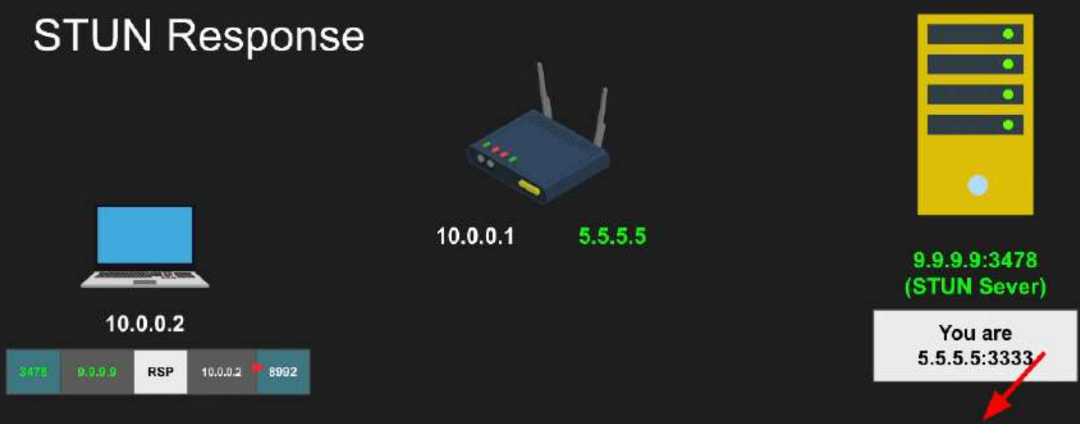


STUN Response
10.0.0.2
10.0.0.1
5.5.5.5
9.9.9.9:3478
(STUN Sever)
You are
5.5.5.5:3333
3478
9.9.9.9
RSP
5.5.5.5
3333


STUN Response
10.0.0.2
10.0.0.1
5.5.5.5
3478
9.9.9.9
RSP
5.5.5.5
3333
9.9.9.9:3478
(STUN Sever)
You are
5.5.5.5:3333

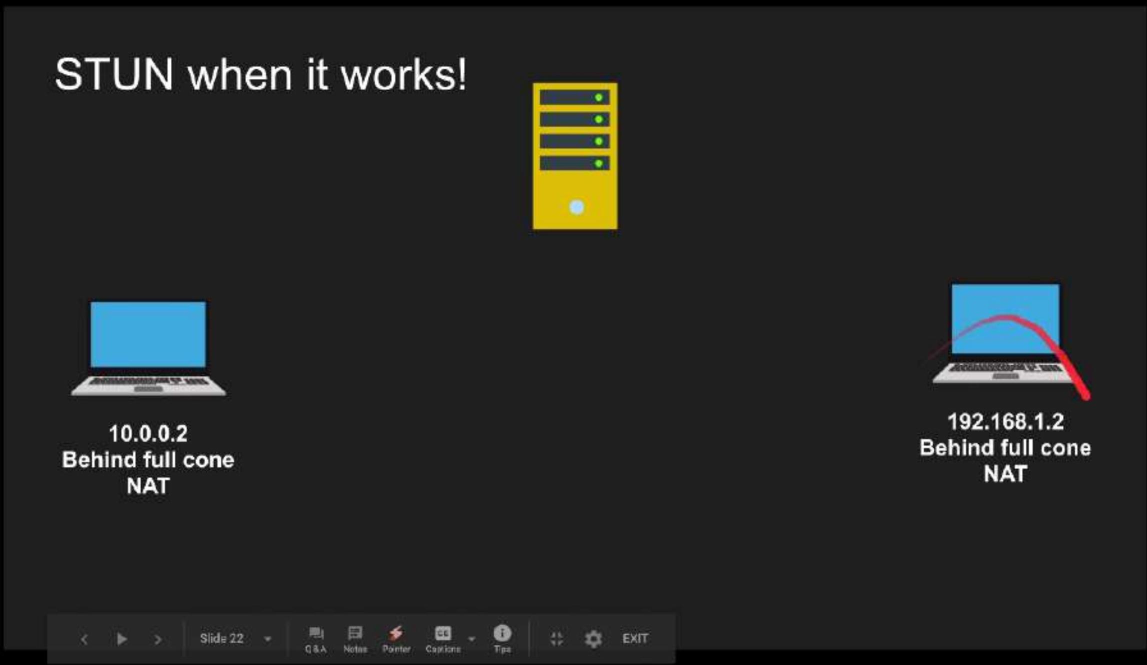

STUN Response
10.0.0.1
5.5.5.5
10.0.0.2
9.9.9.9:3478
(STUN Sever)
3478
9.9.9.9
RSP
10.0.0.2
8092
You are
5.5.5.5:3333
Internal IP
Internal Port
Ext IP
Ext Port
Dest IP
Dest Port
10.0.0.2

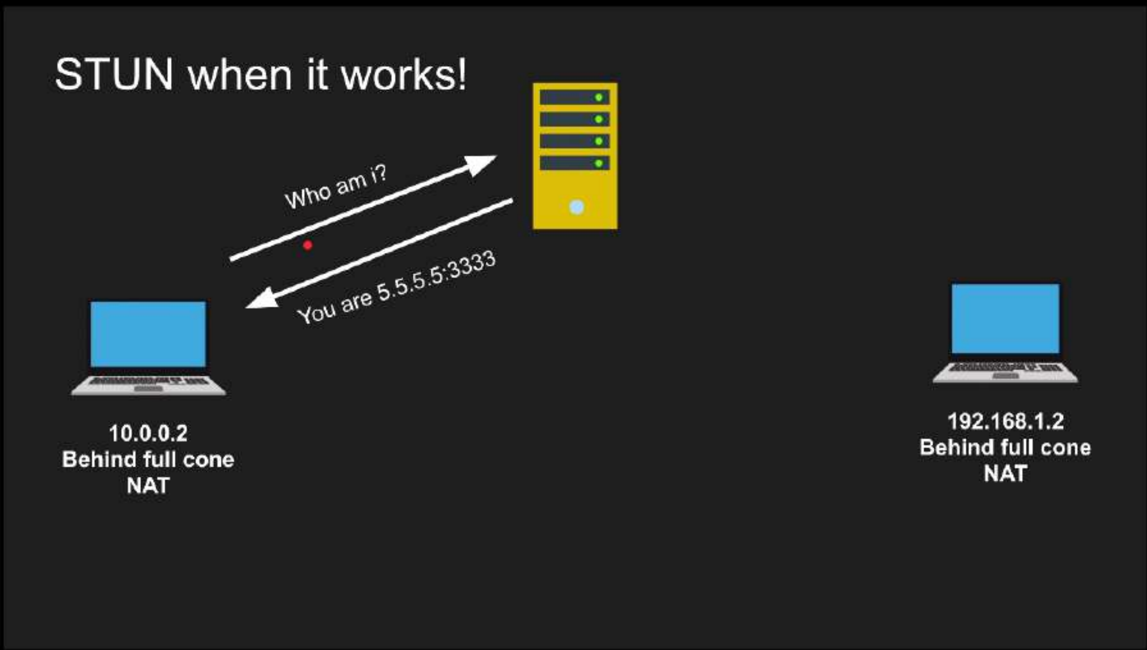
STUN Response

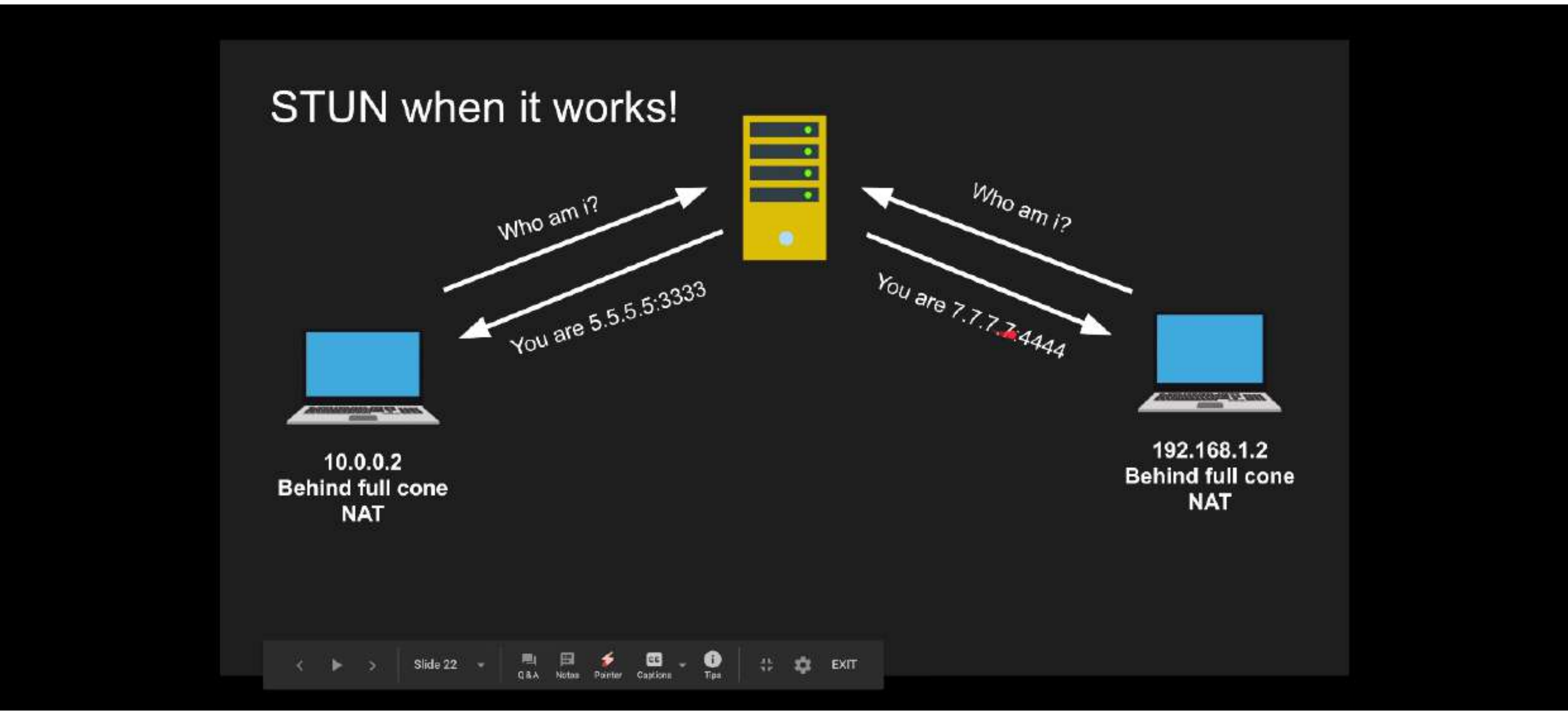


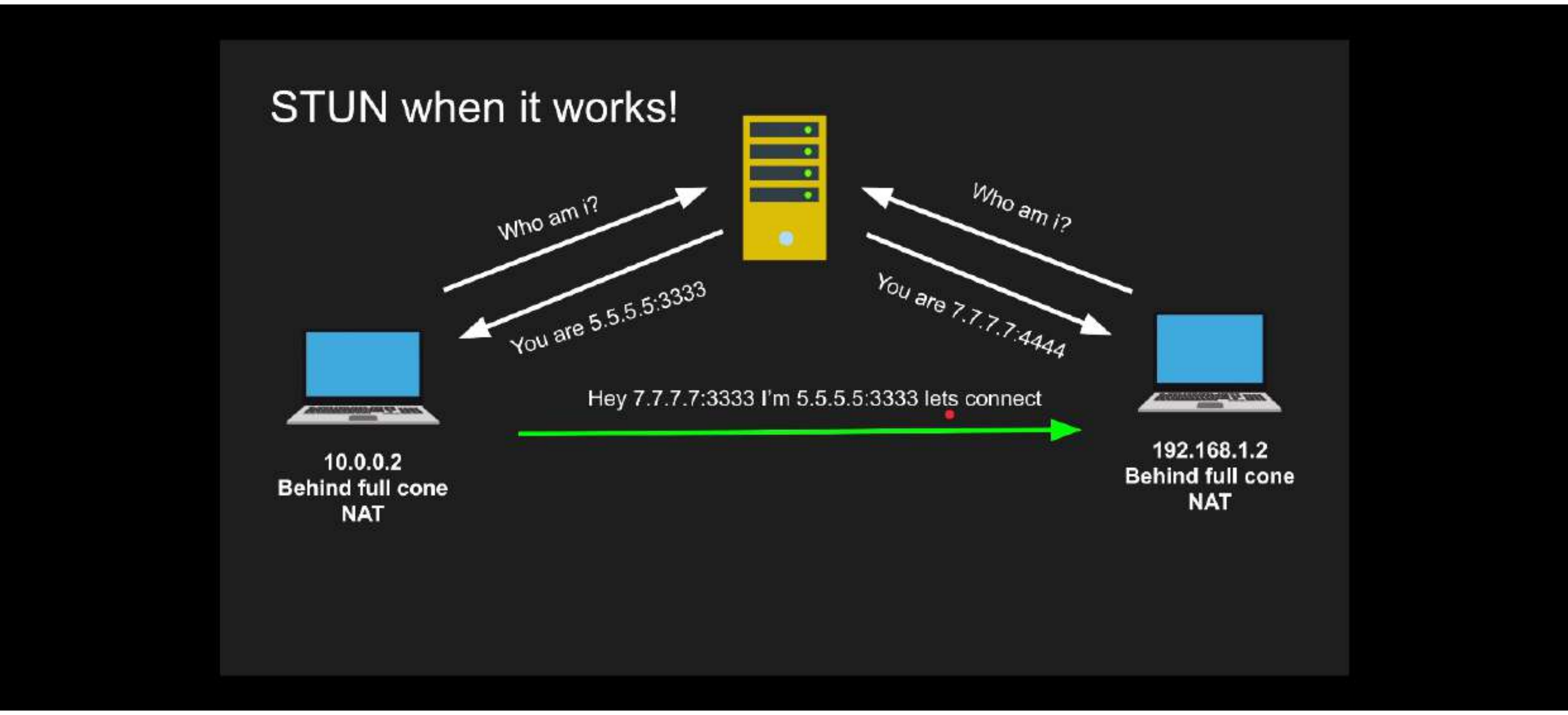
Internal IP	Internal Port	Ext IP	Ext Port	Dest IP	Dest Port
10.0.0.2	8992	5.5.5.5	3333	9.9.9.9	3478

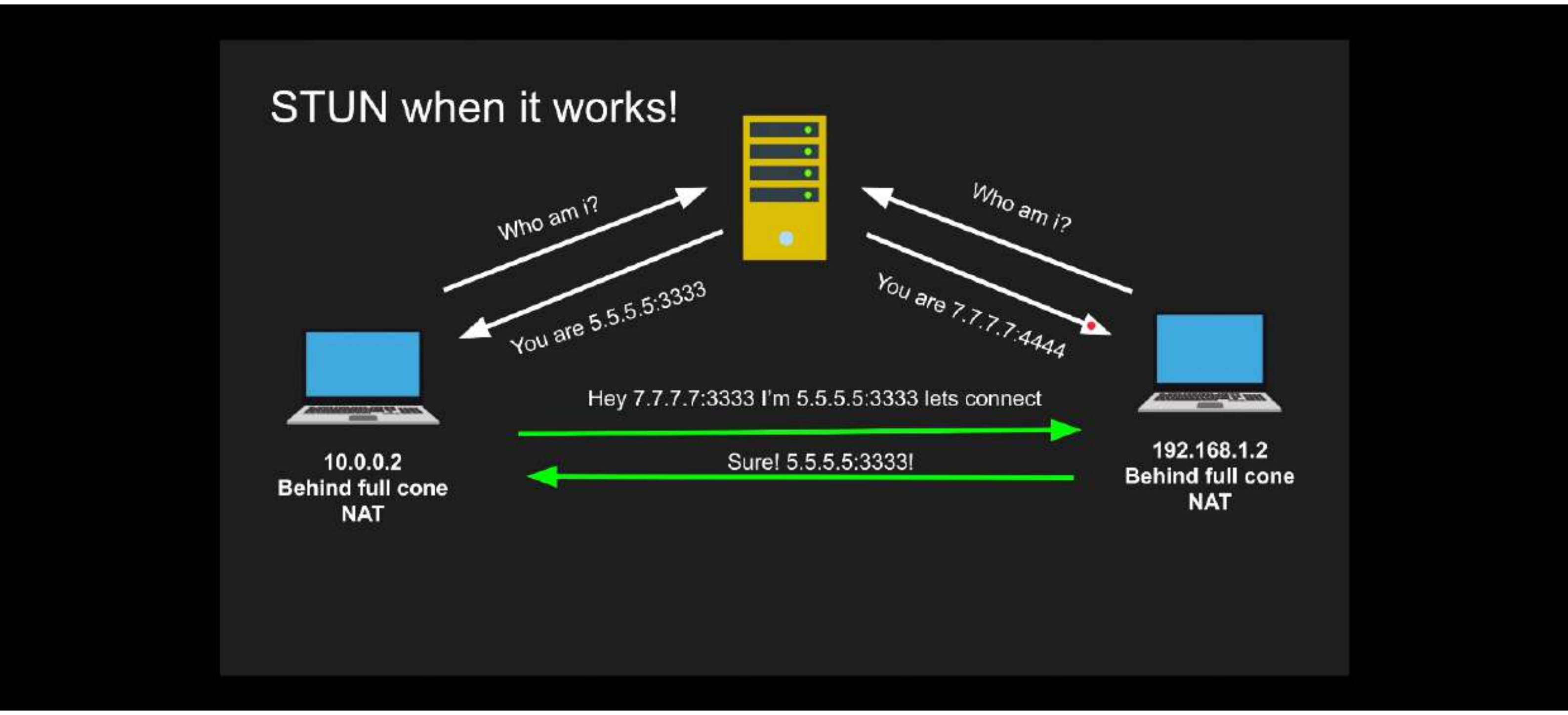

STUN Response
10.0.0.1
5.5.5.5
9.9.9.9:3478
(STUN Sever)
You are
5.5.5.5:3333
10.0.0.2
3478
9.9.9.9
RSP
10.0.0.2
8992
You are
5.5.5.5:3333
Internal IP
Internal Port
Ext IP
Ext Port
Dest IP
Dest Port
10.0.0.2

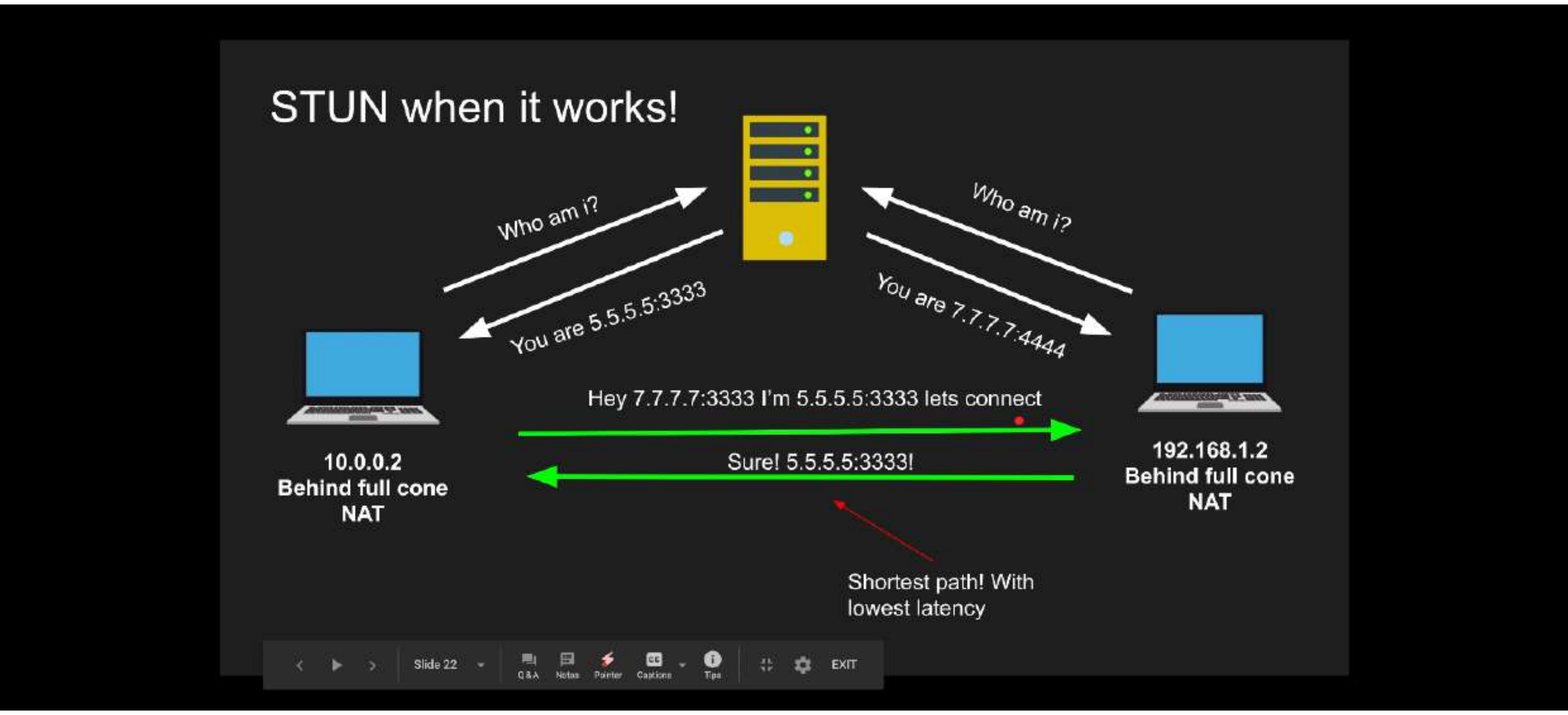

STUN when it works!
10.0.0.2
Behind full cone
NAT
192.168.1.2
Behind full cone
NAT
Slide 22
Q&A
Notes
Pointer
Captions
Tips
EXIT


STUN when it works!
Who am i?
You are 5.5.5.5:3333
10.0.0.2
Behind full cone
NAT
192.168.1.2
Behind full cone
NAT

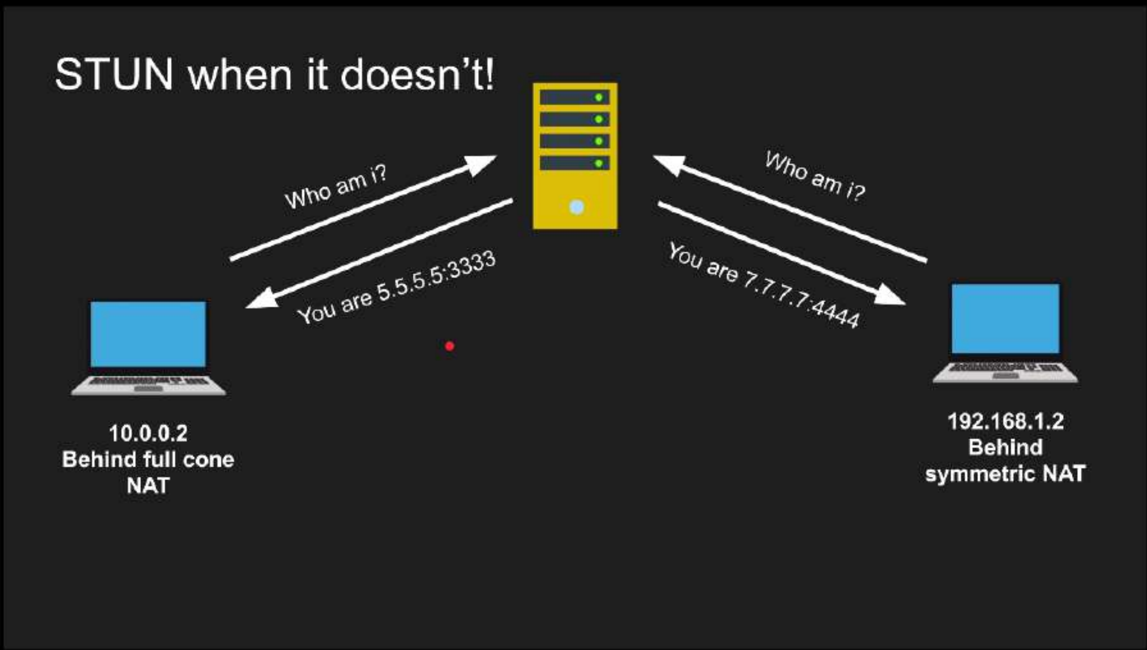

STUN when it works!
Who am i?
You are 5.5.5.5:3333
Who am i?
You are 7.7.7.7:4444
10.0.0.2
Behind full cone
NAT
192.168.1.2
Behind full cone
NAT
Slide 22
Q&A
Notes
Pointer
Captions
Tips
EXIT

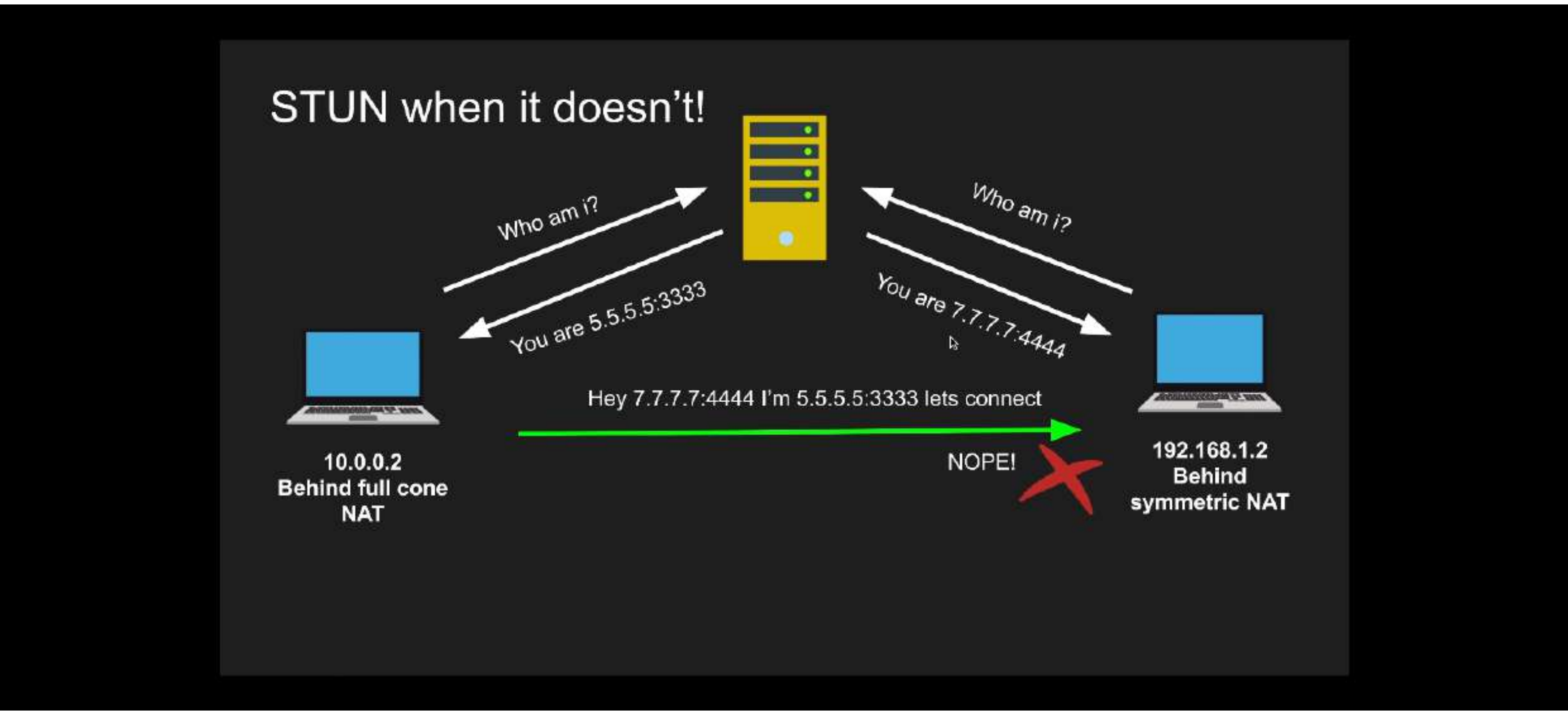

STUN when it works!
Who am i?
You are 5.5.5.5:3333
Who am i?
You are 7.7.7.7:4444
Hey 7.7.7.7:3333 I'm 5.5.5.5:3333 lets connect
10.0.0.2
Behind full cone
NAT
192.168.1.2
Behind full cone
NAT


STUN when it works!
Who am i?
You are 5.5.5.5:3333
Who am i?
You are 7.7.7.7:4444
Hey 7.7.7.7:3333 I'm 5.5.5.5:3333 lets connect
Sure! 5.5.5.5:3333!
10.0.0.2
Behind full cone
NAT
192.168.1.2
Behind full cone
NAT



STUN when it works!
Who am i?
You are 5.5.5.5:3333
Who am i?
You are 7.7.7.7:4444
Hey 7.7.7.7:3333 I'm 5.5.5.5:3333 lets connect
Sure! 5.5.5.5:3333!
10.0.0.2
Behind full cone
NAT
192.168.1.2
Behind full cone
NAT
Shortest path! With
lowest latency
Slide 22
Q&A
Notes
Pointer
Captions
Tips
EXIT

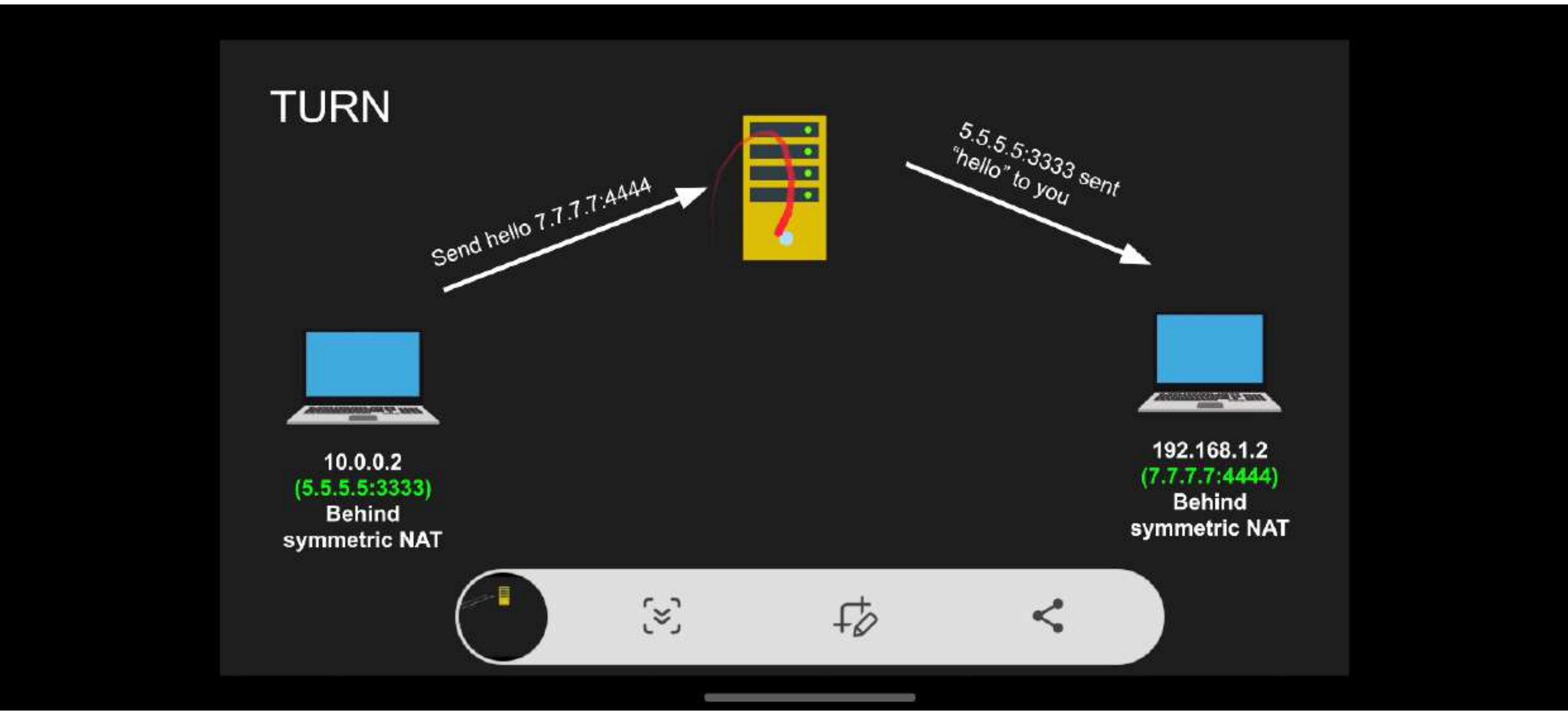

STUN when it doesn't!
Who am i?
You are 5.5.5.5:3333
Who am i?
You are 7.7.7.7:4444
10.0.0.2
Behind full cone
NAT
192.168.1.2
Behind
symmetric NAT

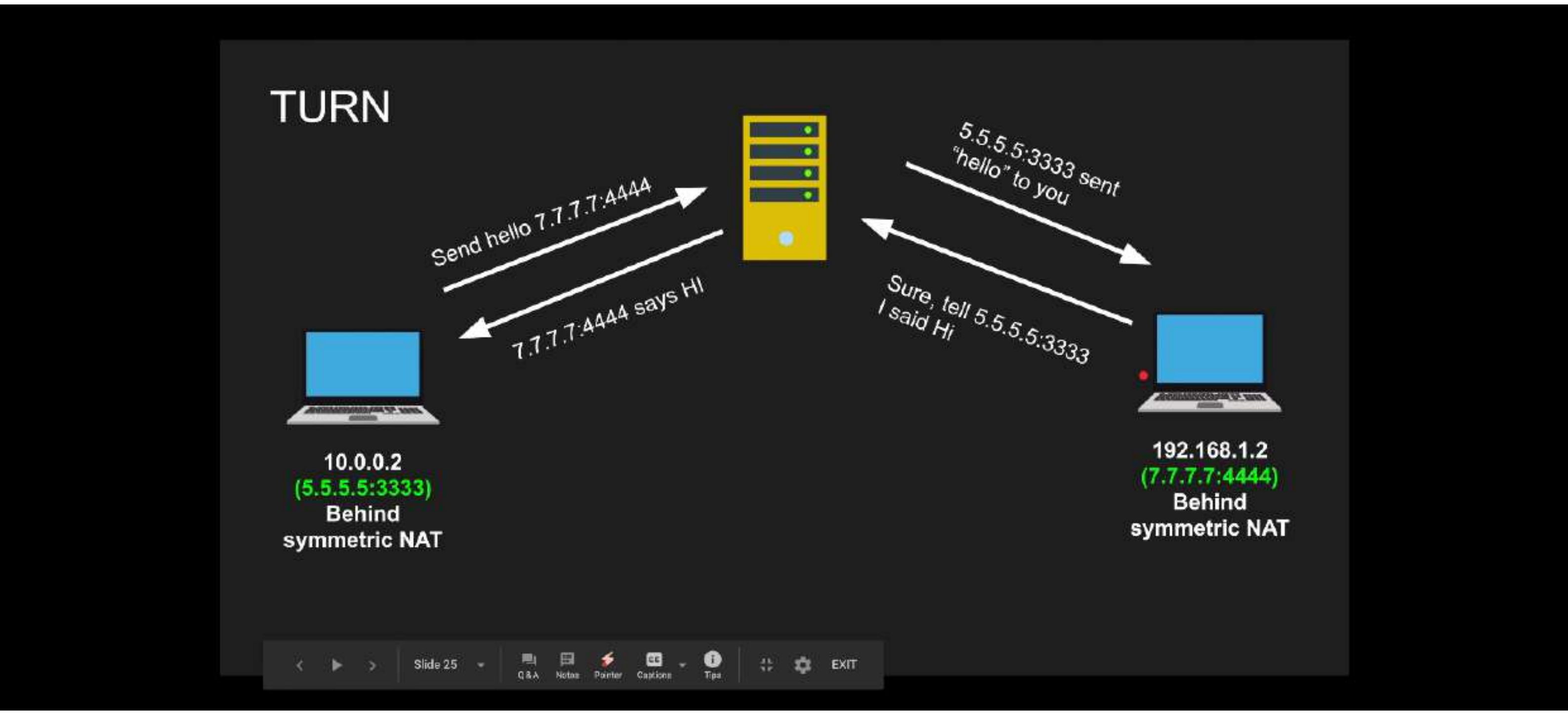

STUN when it doesn't!
Who am i?
You are 5.5.5.5:3333
Who am i?
You are 7.7.7.7:4444
Hey 7.7.7.7:4444 I'm 5.5.5.5:3333 lets connect
NOPE!
10.0.0.2
Behind full cone
NAT
192.168.1.2
Behind
symmetric NAT

TURN

- Traversal Using Relays around NAT
- In case of Symmetric NAT we use TURN
- It's just a server that relays packets
- TURN default server port 3478, 5349 for TLS
- Expensive to maintain and run

TURN
Send hello 7.7.7.7:4444
10.0.0.2
(5.5.5.5:3333)
Behind
symmetric NAT
192.168.1.2
(7.7.7.7:4444)
Behind
symmetric NAT


TURN
Send hello 7.7.7.7:4444
5.5.5.5:3333 sent
"hello" to you
10.0.0.2
(5.5.5.5:3333)
Behind
symmetric NAT
192.168.1.2
(7.7.7.7:4444)
Behind
symmetric NAT


TURN
Send hello 7.7.7.7:4444
7.7.7.7:4444 says Hi
5.5.5.5:3333 sent "hello" to you
Sure, tell 5.5.5.5:3333 I said Hi
10.0.0.2
(5.5.5.5:3333)
Behind
symmetric NAT
192.168.1.2
(7.7.7.7:4444)
Behind
symmetric NAT
Slide 25
EXIT

ICE

- Interactive Connectivity Establishment
- ICE collects all available candidates (local IP addresses, reflexive addresses – STUN ones and relayed addresses – TURN ones)
- Called ice candidates
- All the collected addresses are then sent to the remote peer via SDP

SDP

- Session Description Protocol
- A format that describes ice candidates, networking options, media options, security options and other stuff
- Not really a protocol its a format
- Most important concept in WebRTC
- The goal is to take the SDP generated by a user and send it “somehow” to the other party

SDP Example

```
v=0
o=- 9148204791819634656 3 IN IP4 127.0.0.1
s=-
t=0 0
a=group:BUNDLE audio video data
a=msid-semantic: WMS kyaiqbOs7S2h3EoSHabQ3JlBqZ67cFqZmWFN
m=audio 50853 RTP/SAVPF 111 103 104 0 8 107 106 105 13 126
c=IN IP4 192.168.1.64
a=rtcp:50853 IN IP4 192.168.1.64
a=candidate:3460887983 1 udp 2113937151 192.168.1.64 50853 typ host generation 0
a=candidate:3460887983 2 udp 2113937151 192.168.1.64 50853 typ host generation 0
...
```

Signaling

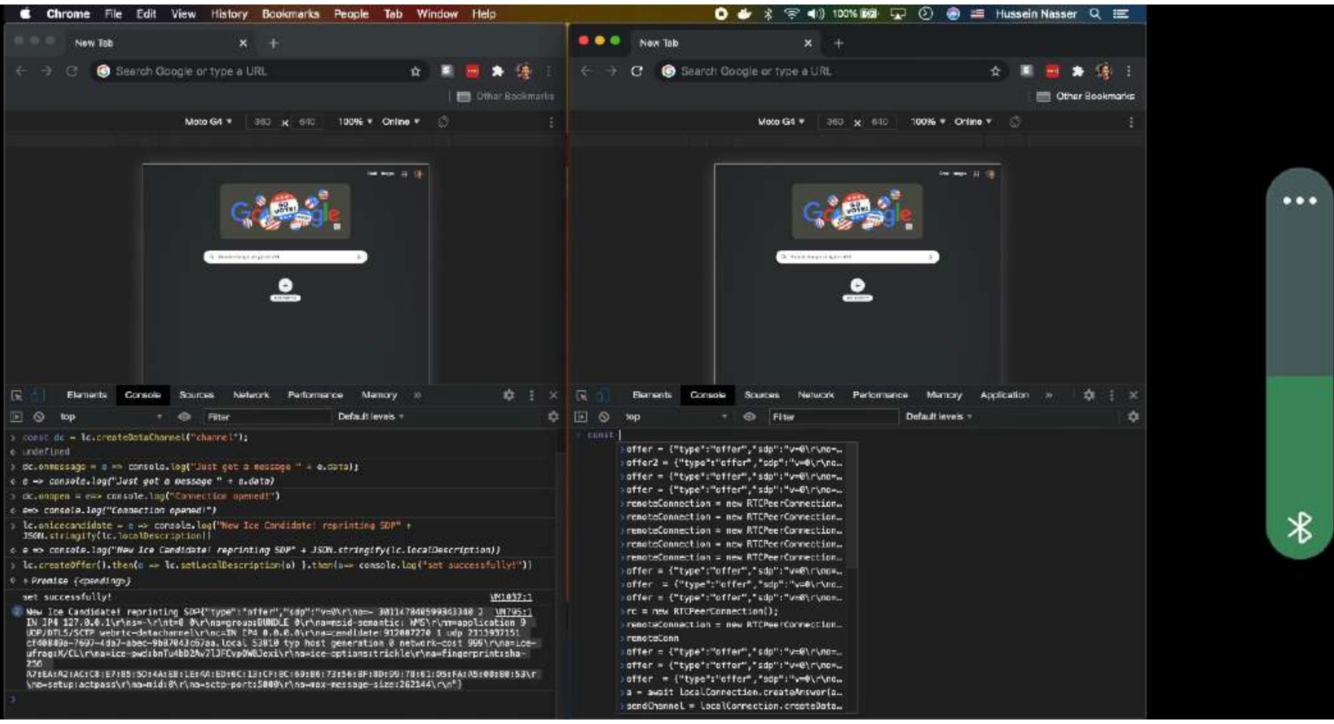
- SDP Signaling
- Send the SDP that we just generated somehow to the other party we wish to communicate with
- Signaling can be done via a tweet, QR code, Whatsapp, WebSockets, HTTP request DOESN'T MATTER! Just get that large string to the other party

WebRTC Demystified

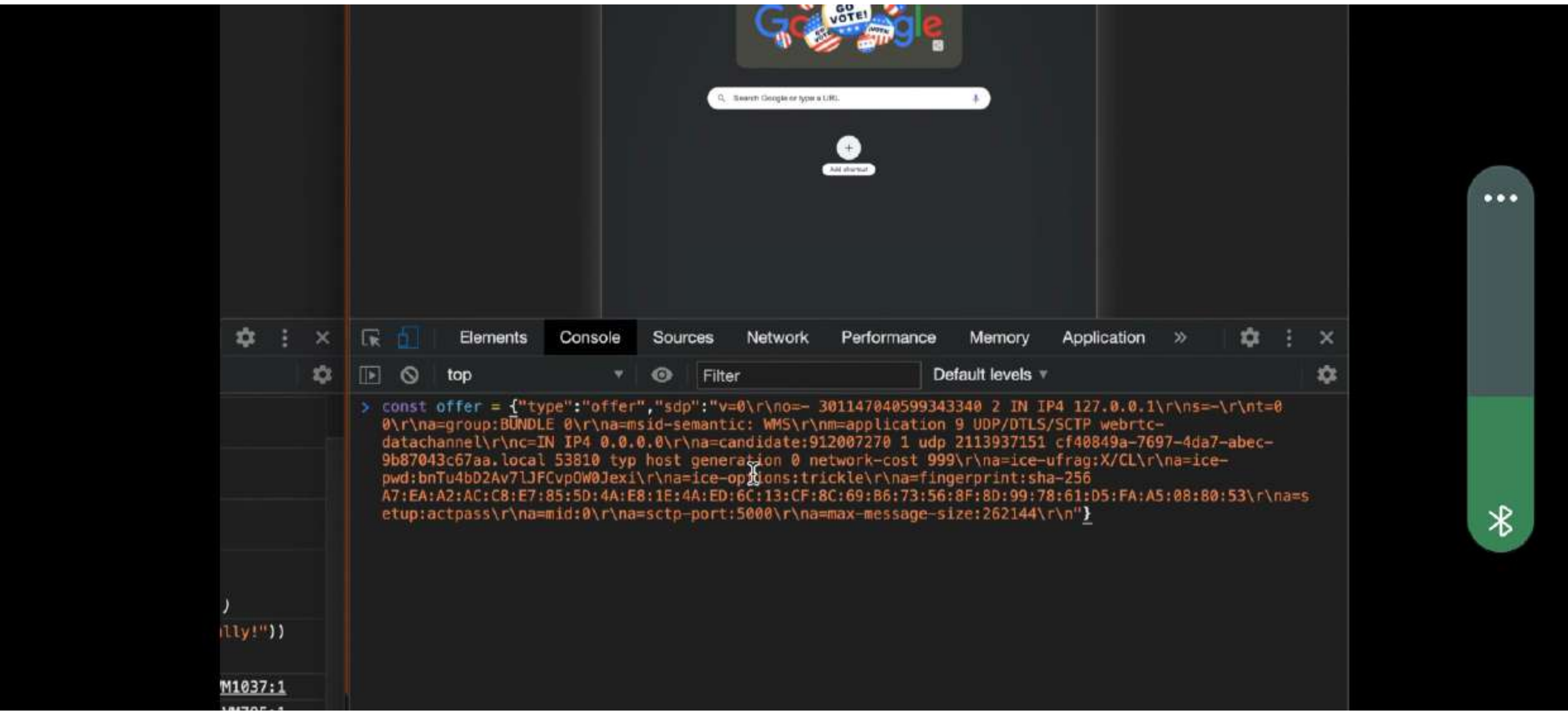
1. A wants to connect to B
2. A creates an “offer”, it finds all ICE candidates, security options, audio/video options and generates SDP, the offer is basically the SDP
3. A signals the offer somehow to B (whatsapp)
4. B creates the “answer” after setting A’s offer
5. B signals the “answer” to A
6. Connection is created

WebRTC Demo

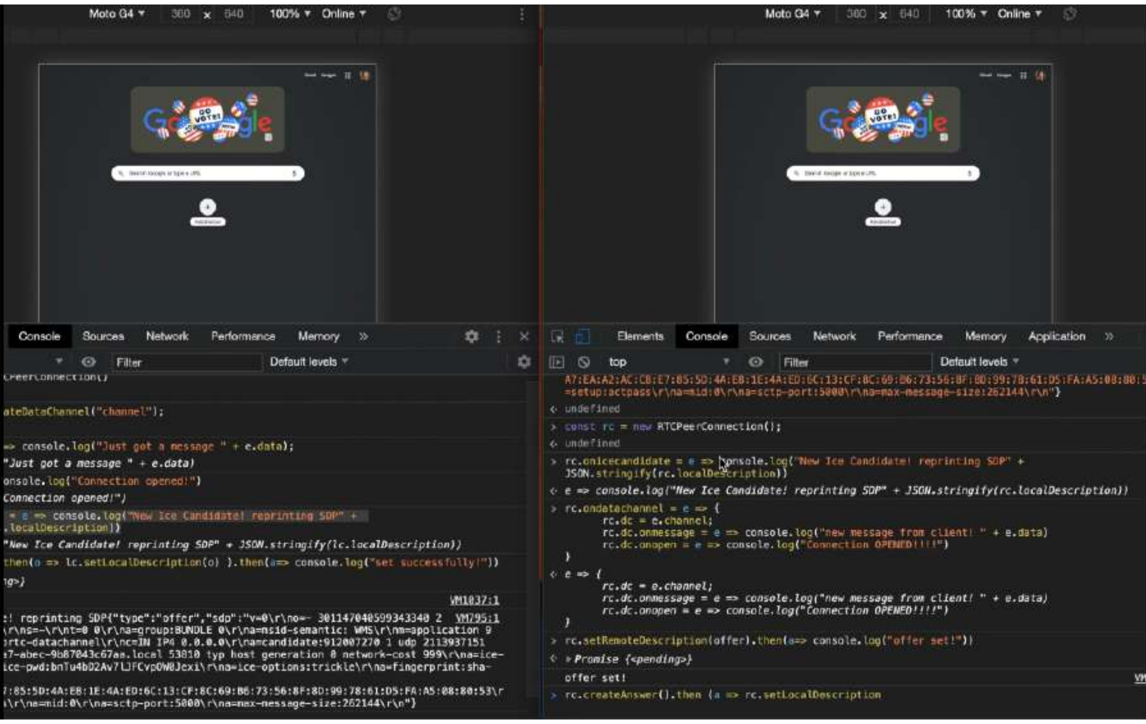
- We will connect two browsers (Browser A & Browser B)
- A will create an offer (sdp) and set it as local description
- B will get the offer and set it as remote description
- B creates an answer sets it as its local description and signal the answer (sdp) to A
- A sets the answer as its remote description
- Connection established, exchange data channel



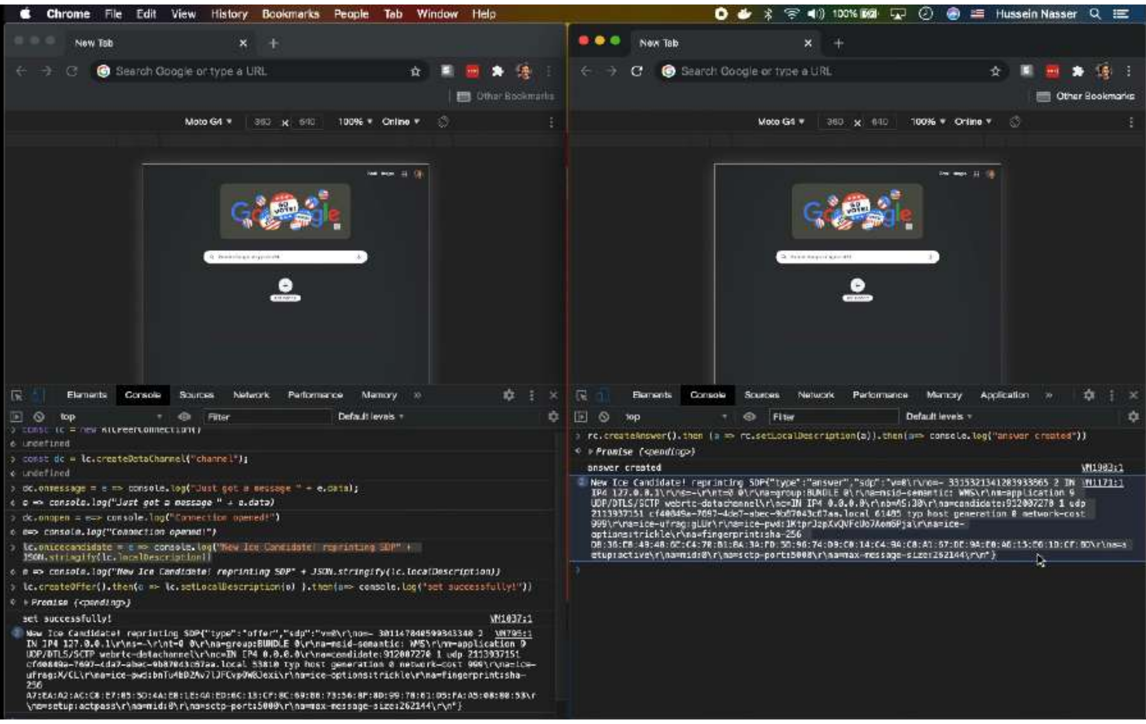
Chrome File Edit View History Bookmarks People Tab Window Help
Hussein Nasser
New Tab
Search Google or type a URL
Other Bookmarks
Moto G4
Elements Console Sources Network Performance Memory
Filter
Default levels
const dc = lc.createDataChannel("channel");
undefined
dc.onmessage = e => console.log("Just got a message " + e.data);
dc.onopen = e=> console.log("Connection opened!")
lc.onicecandidate = e => console.log("New Ice Candidate! reprinting SDP" + JSON.stringify(lc.localDescription))
lc.createOffer().then(o => lc.setLocalDescription(o)).then(a=> console.log("set successfully!"))
Promise {<pending>}
set successfully!
remoteConnection = new RTCPeerConnection…
rc = new RTCPeerConnection();
a = await localConnection.createAnswer(a…
sendChannel = localConnection.createData…



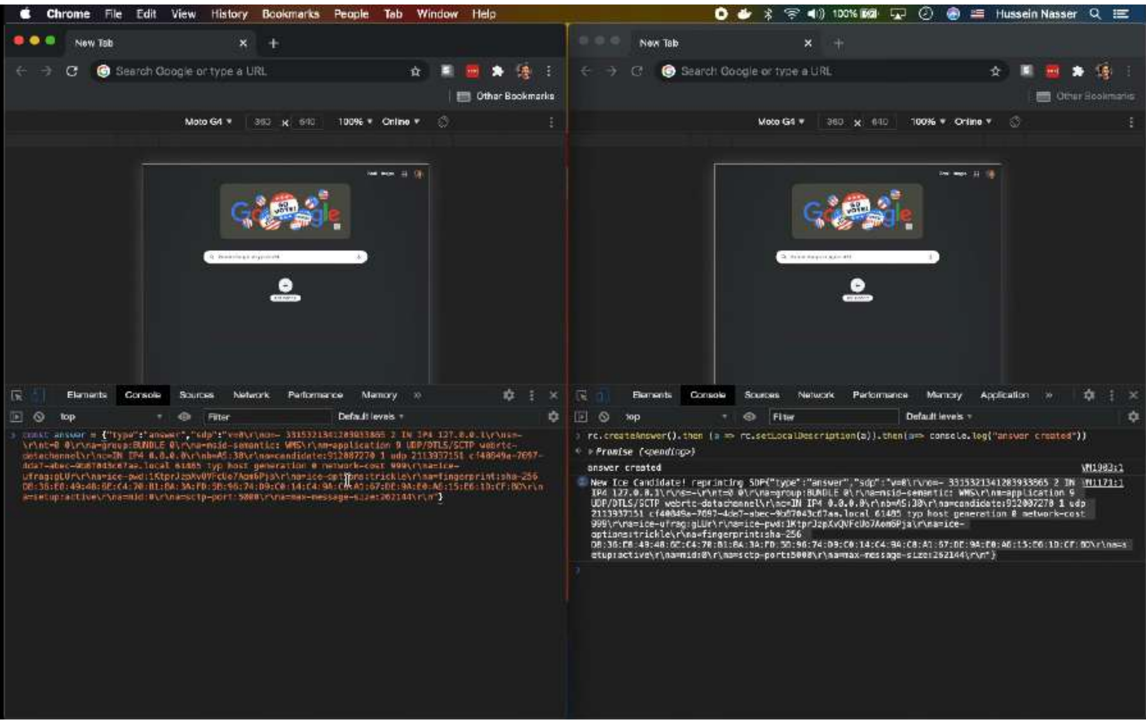
Elements
Console
Sources
Network
Performance
Memory
Application
top
Filter
Default levels
const offer = {"type":"offer","sdp":"v=0\r\no=- 3011470405993433402 IN IP4 127.0.0.1\r\ns=-\r\nt=0
0\r\na=group:BUNDLE 0\r\na=msid-semantic: WMS\r\nm=application 9 UDP/DTLS/SCTP webrtc-
datachannel\r\nc=IN IP4 0.0.0.0\r\na=candidate:912007270 1 udp 2113937151 cf40849a-7697-4da7-abec-
9b87043c67aa.local 53810 typ host generation 0 network-cost 999\r\na=ice-ufrag:X/CL\r\na=ice-
pwd:bnTu4bD2Av7lJFCvpOW0Jexi\r\na=ice-options:trickle\r\na=fingerprint:sha-256
A7:EA:A2:AC:C8:E7:85:5D:4A:E8:1E:4A:ED:6C:13:CF:8C:69:B6:73:56:8F:8D:99:78:61:D5:FA:A5:08:80:53\r\na=s
etup:actpass\r\na=mid:0\r\na=sctp-port:5000\r\na=max-message-size:262144\r\n"}

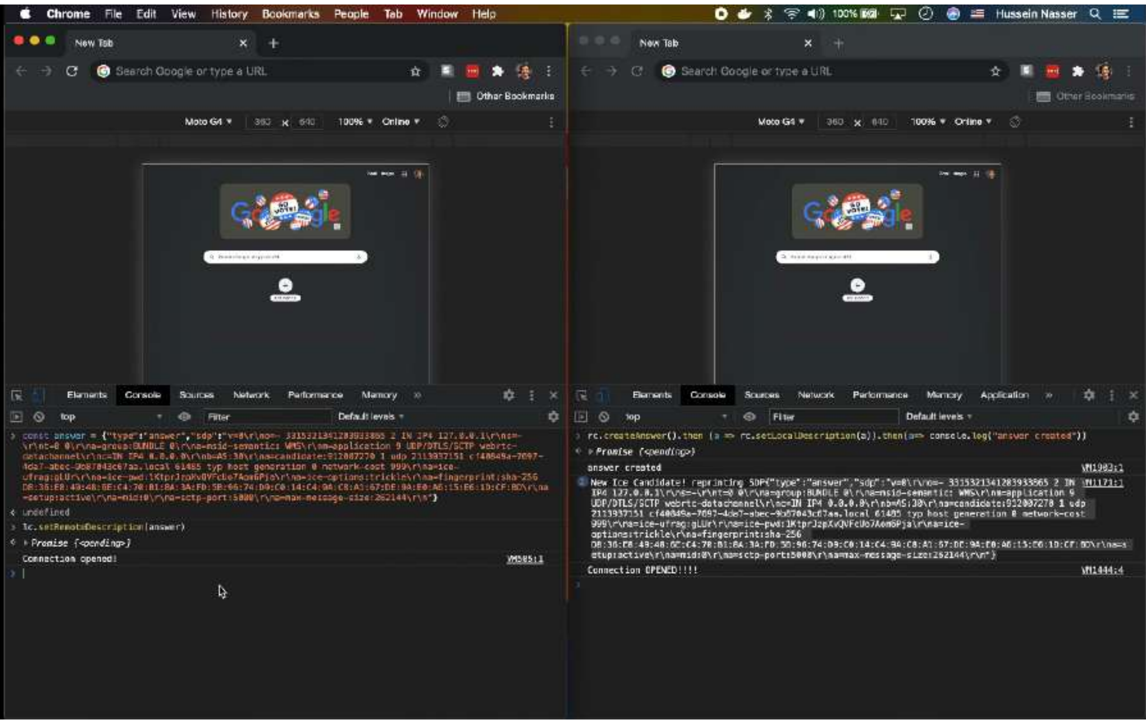


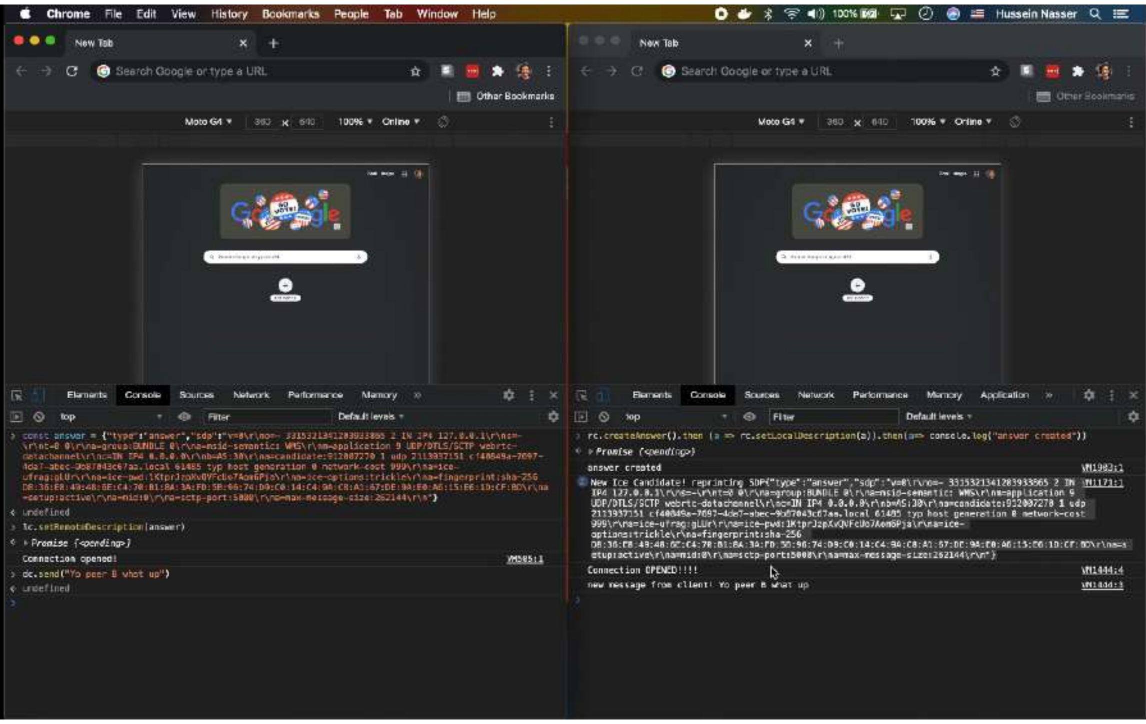
Moto G4 ▾ 360 x 640 100% ▾ Online ▾
Console Sources Network Performance Memory
Filter
Default levels
ateDataChannel("channel");
=> console.log("Just got a message " + e.data);
"Just got a message " + e.data)
onsole.log("Connection opened!")
Connection opened!")
= e => console.log("New Ice Candidate! reprinting SDP" +
.localDescription))
"New Ice Candidate! reprinting SDP" + JSON.stringify(lc.localDescription))
then(o => lc.setLocalDescription(o)).then(a=> console.log("set successfully!"))
VM1037:1
VM795:1
Elements Console Sources Network Performance Memory Application
top
Filter
Default levels
=setup:actpass\r\na=mid:0\r\na=sctp-port:5000\r\na=max-message-size:262144\r\n"}
undefined
const rc = new RTCPeerConnection();
undefined
rc.onicecandidate = e => console.log("New Ice Candidate! reprinting SDP" + JSON.stringify(rc.localDescription))
e => console.log("New Ice Candidate! reprinting SDP" + JSON.stringify(rc.localDescription))
rc.ondatachannel = e => {
rc.dc = e.channel;
rc.dc.onmessage = e => console.log("new message from client! " + e.data)
rc.dc.onopen = e => console.log("Connection OPENED!!!!")
}
e => {
rc.dc = e.channel;
rc.dc.onmessage = e => console.log("new message from client! " + e.data)
rc.dc.onopen = e => console.log("Connection OPENED!!!!")
}
rc.setRemoteDescription(offer).then(a=> console.log("offer set!"))
Promise {<pending>}
offer set!
rc.createAnswer().then (a => rc.setLocalDescription

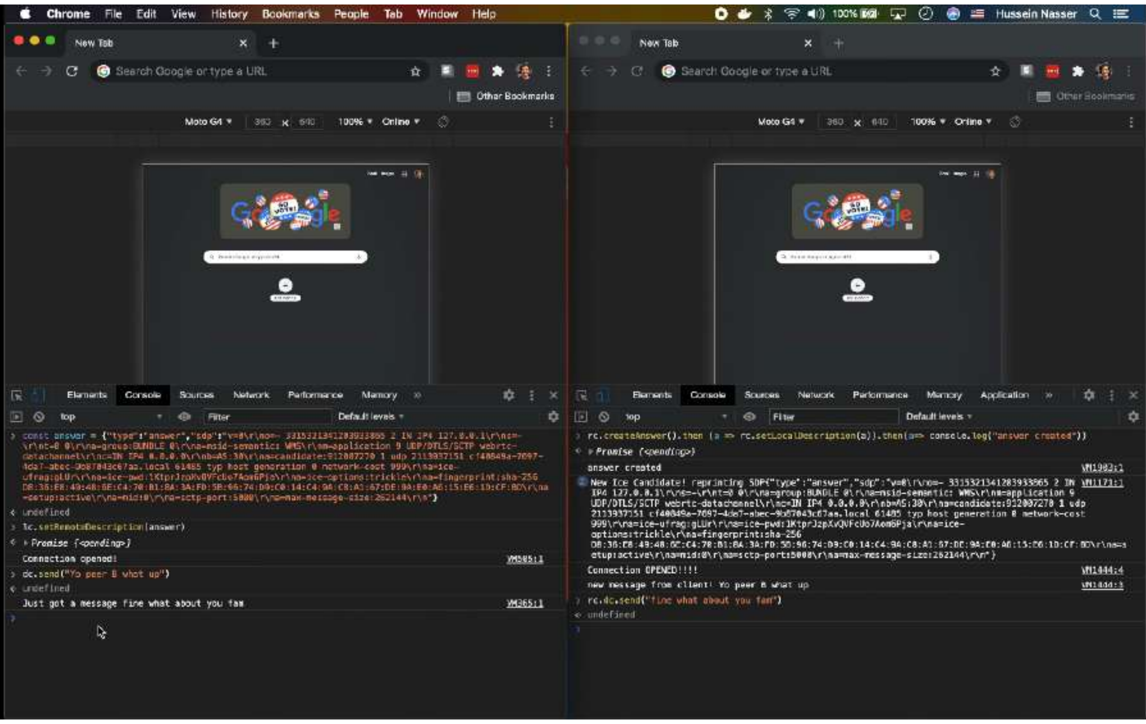


Chrome File Edit View History Bookmarks People Tab Window Help
Hussein Nasser
New Tab
Search Google or type a URL
Other Bookmarks
Moto G4 360 x 640 100% Online
Elements Console Sources Network Performance Memory
top Filter Default levels
undefined
const dc = lc.createDataChannel("channel");
undefined
dc.onmessage = e => console.log("Just got a message " + e.data);
e => console.log("Just got a message " + e.data)
dc.onopen = e=> console.log("Connection opened!")
e=> console.log("Connection opened!")
lc.onicecandidate = e => console.log("New Ice Candidate! reprinting SDP" + JSON.stringify(lc.localDescription))
e => console.log("New Ice Candidate! reprinting SDP" + JSON.stringify(lc.localDescription))
lc.createOffer().then(o => lc.setLocalDescription(o)).then(a=> console.log("set successfully!"))
▸ Promise {<pending>}
set successfully!
VM1037:1
New Ice Candidate! reprinting SDP{"type":"offer","sdp":"v=0\r\no=-
VM795:1
Elements Console Sources Network Performance Memory Application
top Filter Default levels
rc.createAnswer().then (a => rc.setLocalDescription(a)).then(a=> console.log("answer created"))
▸ Promise {<pending>}
answer created
VM1983:1
New Ice Candidate! reprinting SDP{"type":"answer","sdp":"v=0\r\no=-
VM1171:1


Chrome File Edit View History Bookmarks People Tab Window Help
Hussein Nasser
New Tab
Search Google or type a URL
Other Bookmarks
Elements Console Sources Network Performance Memory
Elements Console Sources Network Performance Memory Application
Default levels
answer created


Chrome File Edit View History Bookmarks People Tab Window Help
Hussein Nasser
New Tab
Search Google or type a URL
Other Bookmarks
Elements Console Sources Network Performance Memory
lc.setRemoteDescription(answer)
Connection opened!
answer created
Connection OPENED!!!!




Chrome File Edit View History Bookmarks People Tab Window Help
Hussein Nasser
New Tab
Search Google or type a URL
Other Bookmarks
Elements Console Sources Network Performance Memory
Elements Console Sources Network Performance Memory Application

WebRTC Pros & Cons

- Pros
  - P2p is great ! low latency for high bandwidth content
  - Standardized API I don't have to build my own
- Cons
  - Maintaining STUN & TURN servers
  - Peer 2 Peer falls apart in case of multiple participants (discord case)

More WebRTC stuff!

So more to discuss beyond this content

Media API

- getUserMedia to access microphone, video camera
- RTCPConnection.addTrack(stream)
- https://www.html5rocks.com/en/tutorials/webrtc/basics/

onIceCandidate and addIceCandidate

- To maintain the connection as new candidates come and go
- onIceCandidate tells user there is a new candidate after the SDP has already been created
- The candidate is signaled and sent to the other party
- The other party uses addiceCandidate to add it to its SDP

Set custom TURN and STUN Servers

```
const iceConfiguration = {
   iceServers: [{
           urls: 'turn:turnserver.company.com:3478',
           username: 'optional-username',
           credentials: 'auth-token'},
          { urls: "stun:stun.services.mozilla.com",
           username: "test@mozilla.com",
           credential: "webrtcdemo"}
        ]
}
const pc = new RTCPeerConnection(configuration);
```

Create your own STUN & TURN server

- COTURN open source project
- https://github.com/coturn/coturn

Public STUN servers

- `stun1.l.google.com:19302`
- `stun2.l.google.com:19302`
- `stun3.l.google.com:19302`
- `stun4.l.google.com:19302`
- `stun.stunprotocol.org:3478`

Thank You

Subscribe for deep dive engineering discussions