
QUIC Address Discovery

IETF 121, Dublin
Marten Seemann

[draft-seemann-quic-address-discovery](#)

Context: p2p QUIC

1. Nodes discover their reflexive addresses
2. Nodes connect to each other via a MASQUE relay
3. Nodes hole punch a direct connection using QUIC connection migration

<https://seemann.io/posts/2024-10-26---p2p-quic/>

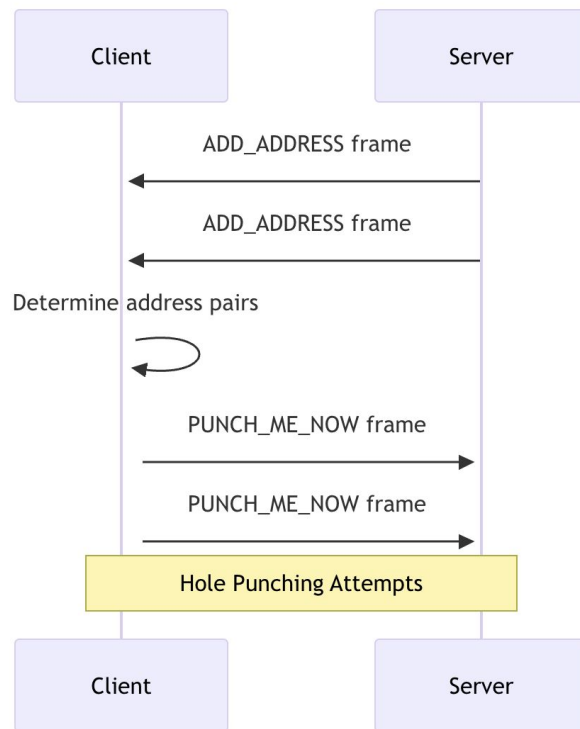
QUIC Relaying using MASQUE

- RFC 9298 defines how to send UDP packets in HTTP
- [CONNECT-UDP Listen](#) defines how to proxy a UDP Listener
- Relay server reserves an IP:port for the client

Use QUIC Connection Migration for NAT Traversal

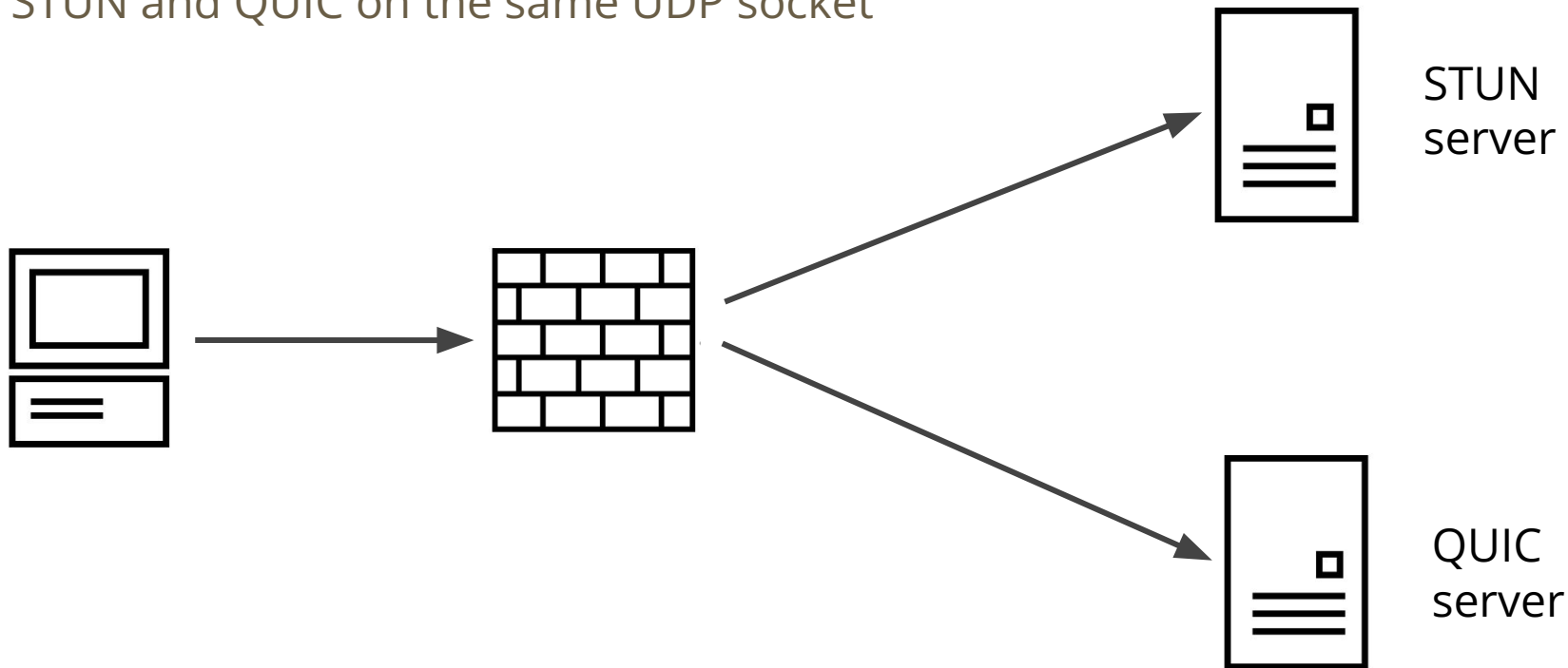
- migrate the proxied QUIC connection to a direct connection
- fully transparent to the application

<https://datatracker.ietf.org/doc/draft-seemann-quic-nat-traversal/>

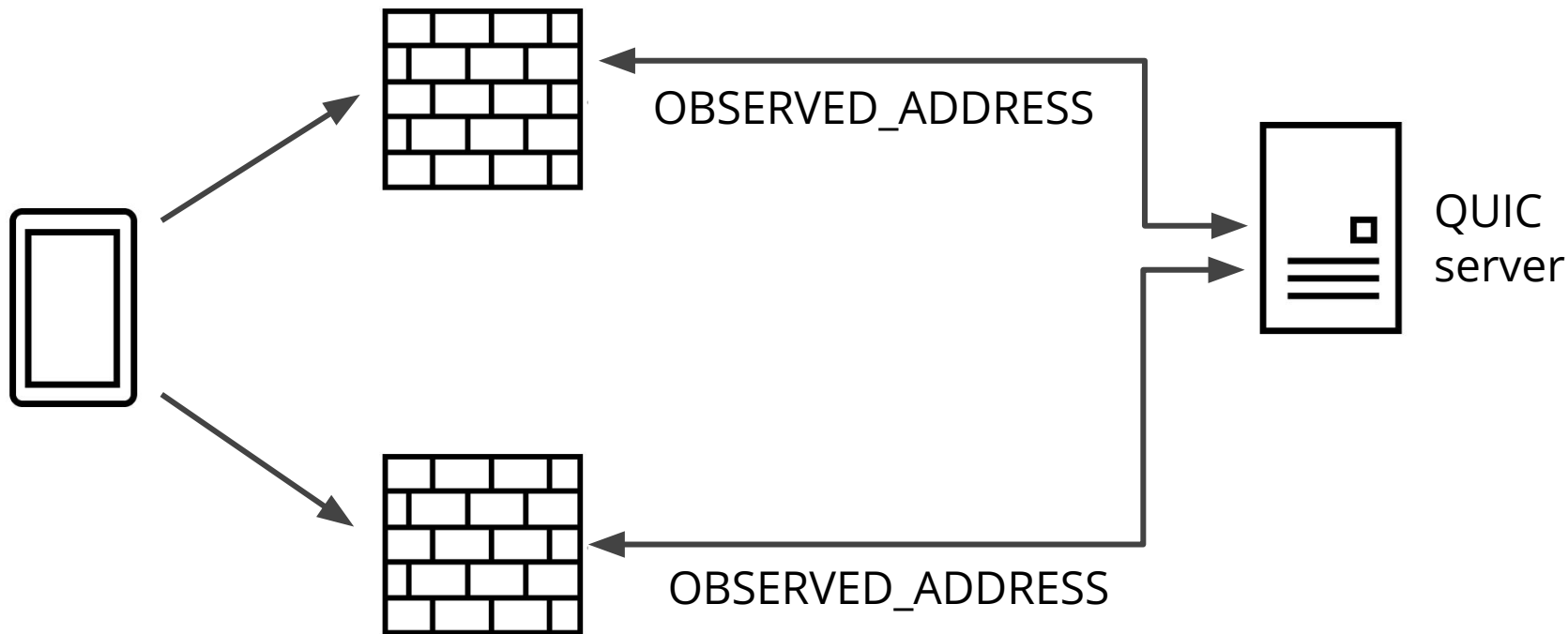


How to find out your (public) address: STUN

Use STUN and QUIC on the same UDP socket



How to find out your (public) address: QUIC



QUIC Address Discovery: Extension Negotiation

Negotiated using the *address_discovery* transport parameter:

- **0**: I can provide observations, but don't send me any
- **1**: I want to receive observations, but can't send any
- **2**: I want to receive observations, and I can provide observations

QUIC Address Discovery: Framing

- sent every time a new path is established
- sent every time the remote address changes (i.e. NAT rebinding)

```
OBSERVED_ADDRESS Frame {  
    Type (i) = 0x9f81a6..0x9f81a7,  
    Sequence Number (i),  
    [ IPv4 (32) ],  
    [ IPv6 (128) ],  
    Port (16),  
}
```


Progress since IETF 118 (Prague)

- Christian Huitema joined as co-author
- Removed request-response mechanism