TURN by name (00)

for IETF 92, 2015 March 23-27

Ben Schwartz
Justin Uberti

The setup

- Suppose you want to send a UDP packet to a DNS-named endpoint through TURN, e.g.
    - Your ICE peer sends you:
      `candidate:1 1 udp 99999 some.endpoint.example.com`
      (yes, this can happen!)
    - Network admin requires all UDP traffic to go through an auto-discovered TURN server (“escape hatch”)
    - Using a TURN server as a proxy for any reason

How it works today

- First, the client resolves the DNS name to an IP address.
  - Or possibly more than one, especially IPv4 + IPv6
- Then it uses TURN to connect to the IP address.

But what if...

- the network admin doesn’t allow some/most DNS lookups?
  - e.g. to prevent SSH-DNS, and use TURN as a pinhole for trusted users
- the DNS is supposed to be different for different clients?
  - e.g. GeoDNS, split-horizon DNS

Solution: name resolution via TURN

- Client only contacts the TURN server
- DNS packets flow between TURN server and DNS server.
- Unified access control for network administrators
  - Just give trusted users TURN access

How? Option 0: Hardcode DNS

- Tunnel DNS queries through the TURN server, just like any other UDP packet.
- Advantages
  - No change to TURN server!
- Disadvantages
  - Client has to **hardcode the IP address** of a D
    server.
  - Breaks recursive DNS caching, DHCP config, etc.

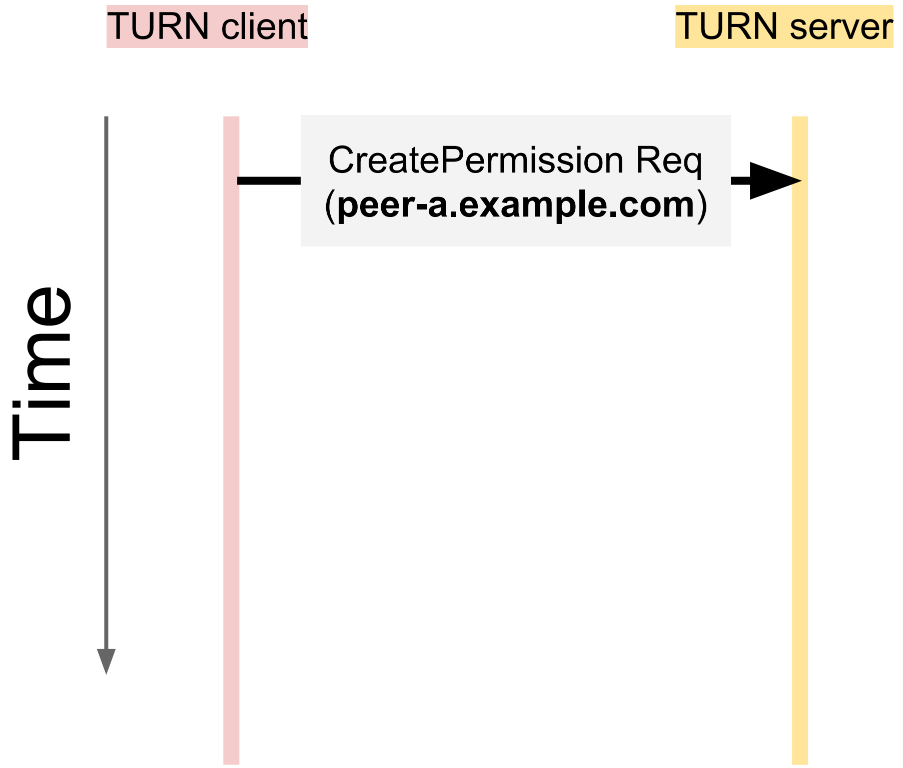
How? Option 1: ResolveRequest

- Add a new request type, **ResolveRequest**
  - The request includes a DNS name
  - The TURN server performs name resolution
  - The response includes the IP address
  - Client sends traffic to the IP address from then on
- Advantages:
  - Respects DHCP, caching, GeoDNS, etc.
- Disadvantage:
  - Totally unlike any current TURN functionality!

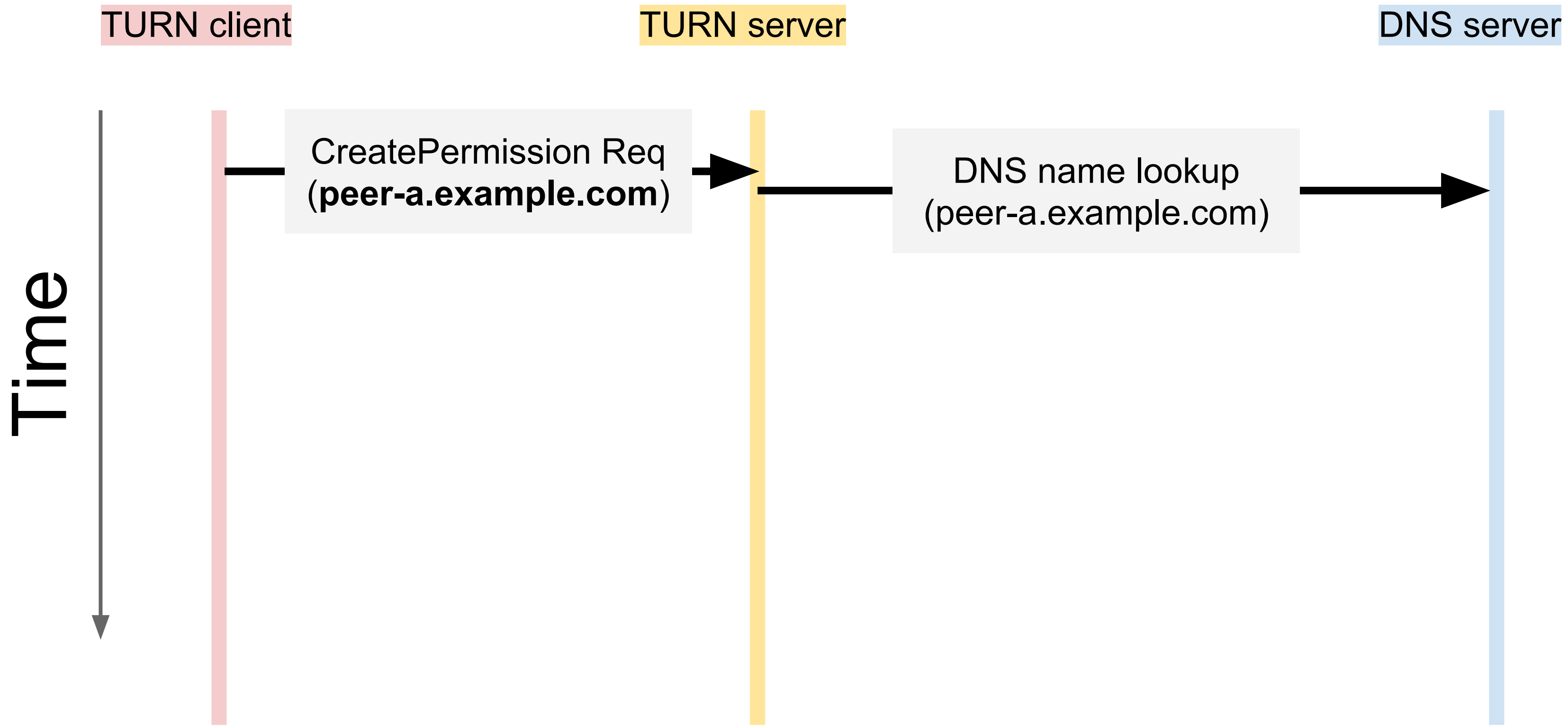
How? Option 2: *TURN by name*

- Allow endpoints to be identified **by name**
  - extends CreatePermission, Send, ChannelBind, etc.
- Advantages:
  - Respects DHCP, caching, GeoDNS, etc.
  - Completely abstracts DNS from client (like SO
    and HTTP(S) CONNECT). Never reveals the IP.
- Disadvantages:
  - More complex to implement
    - requires TURN to maintain IP->DNS mapping

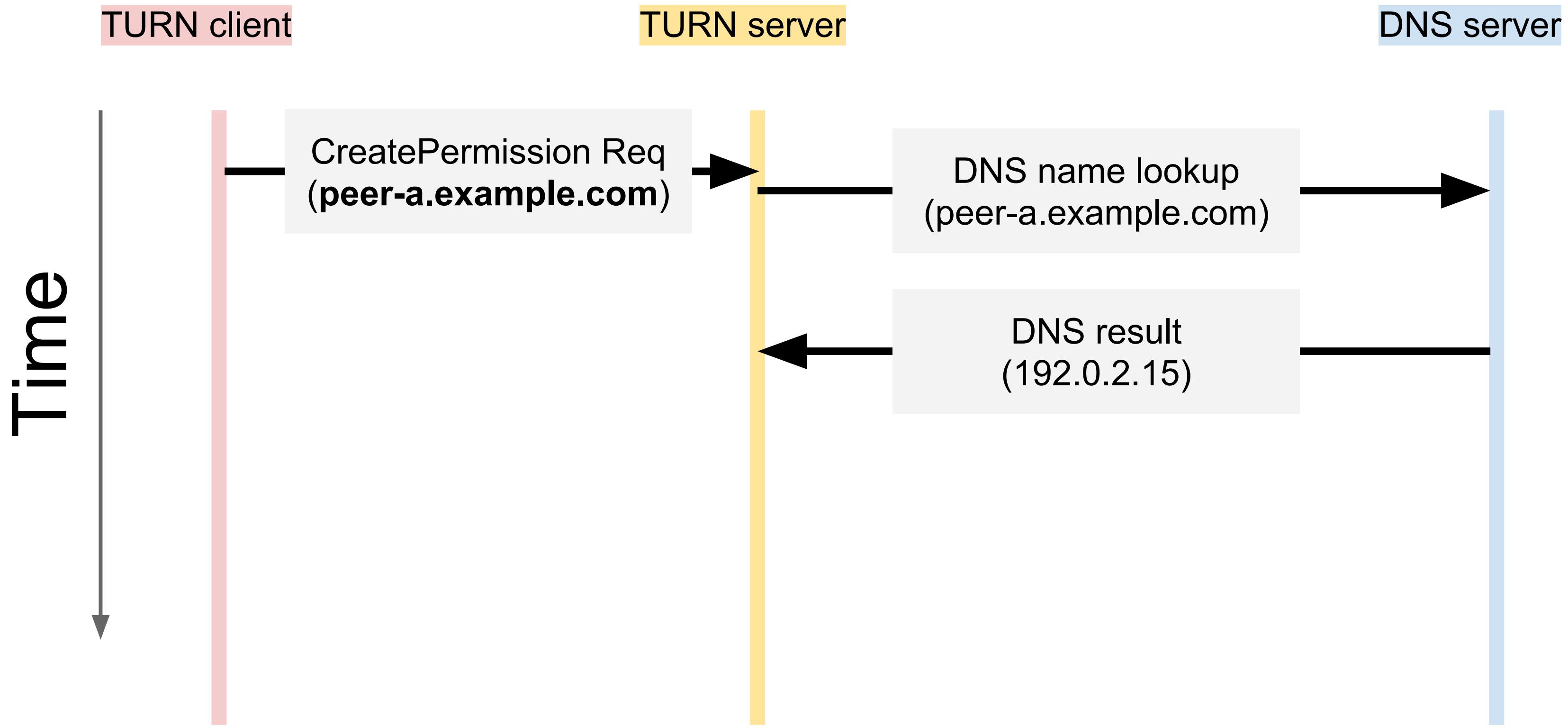
TURN by name message example



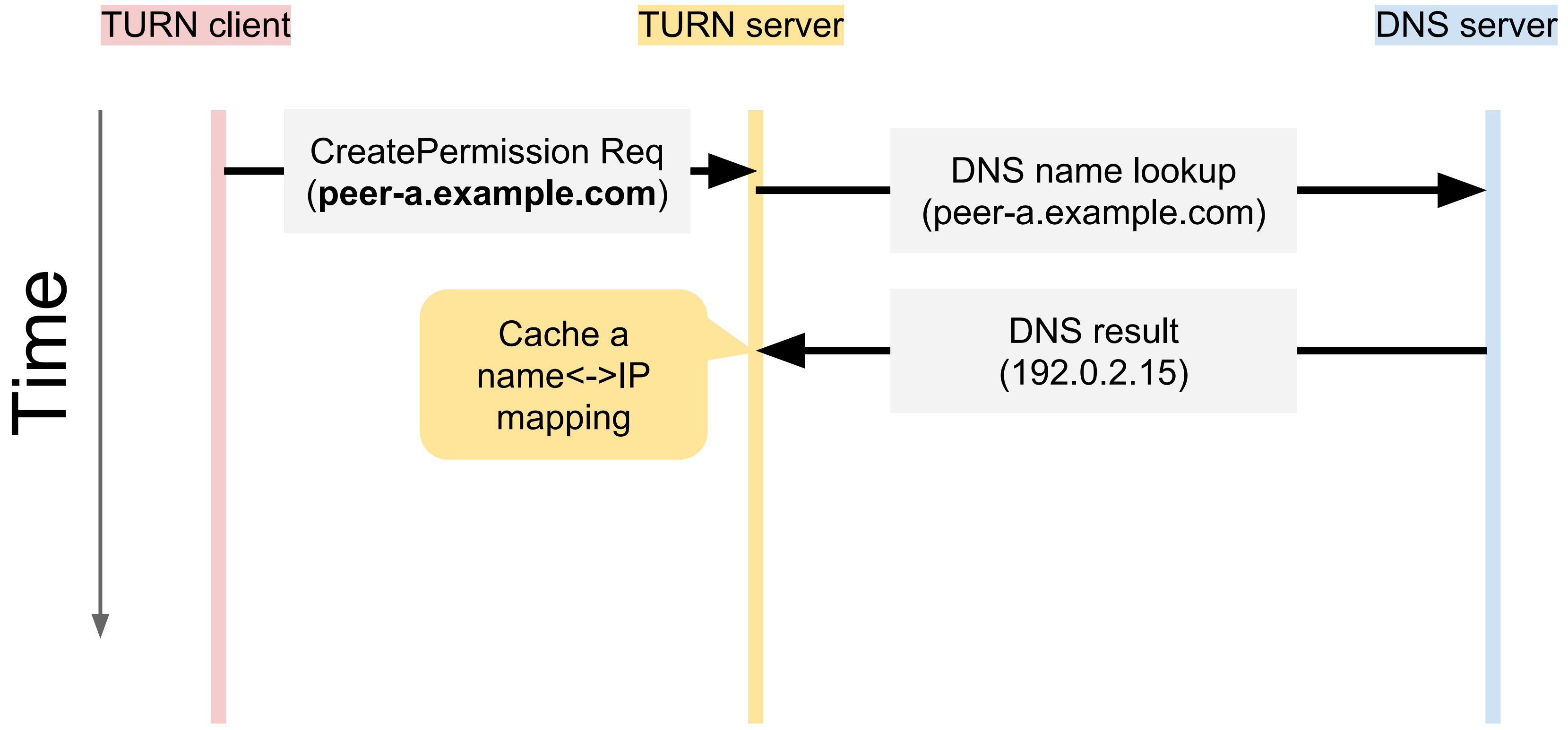
TURN by name message example



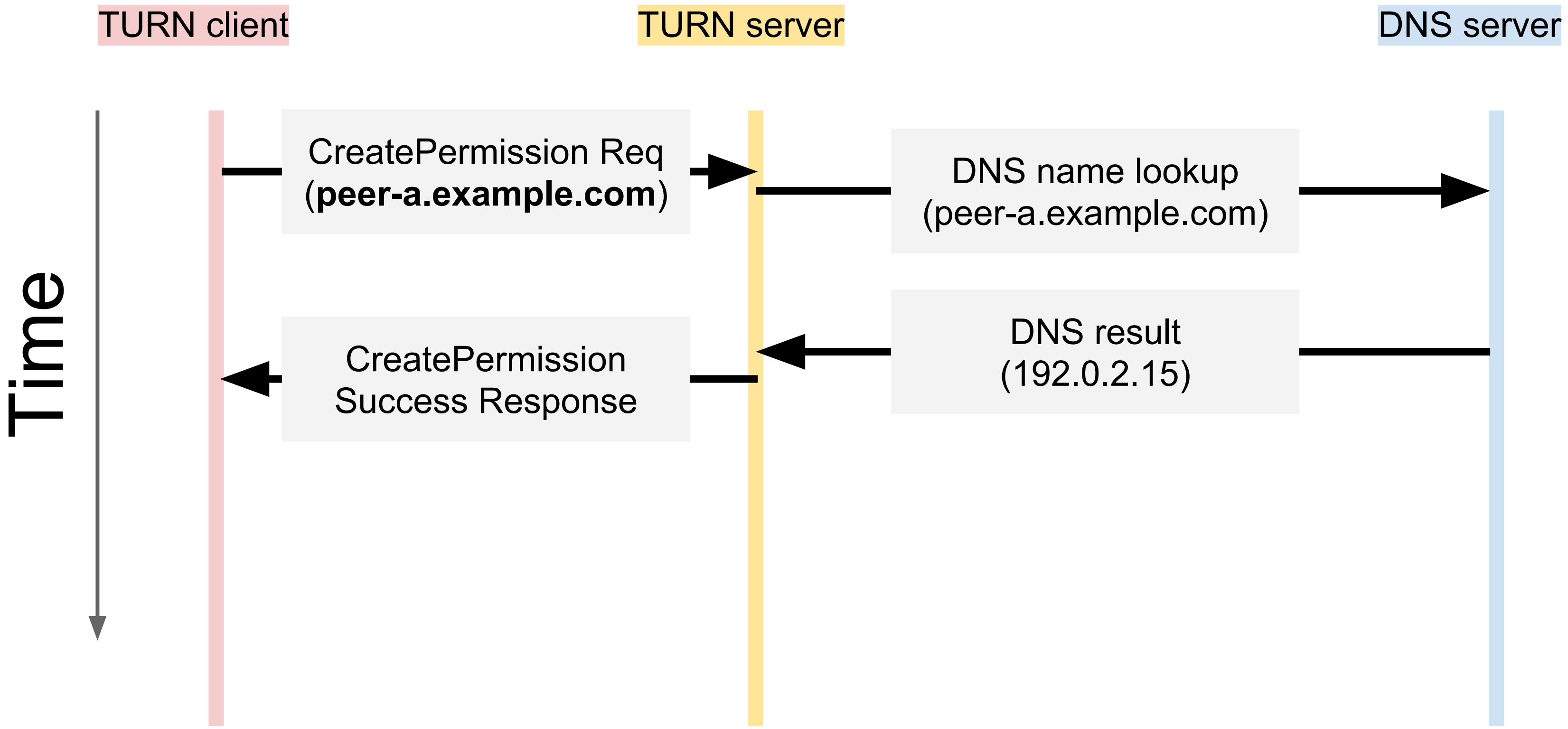
TURN by name message example



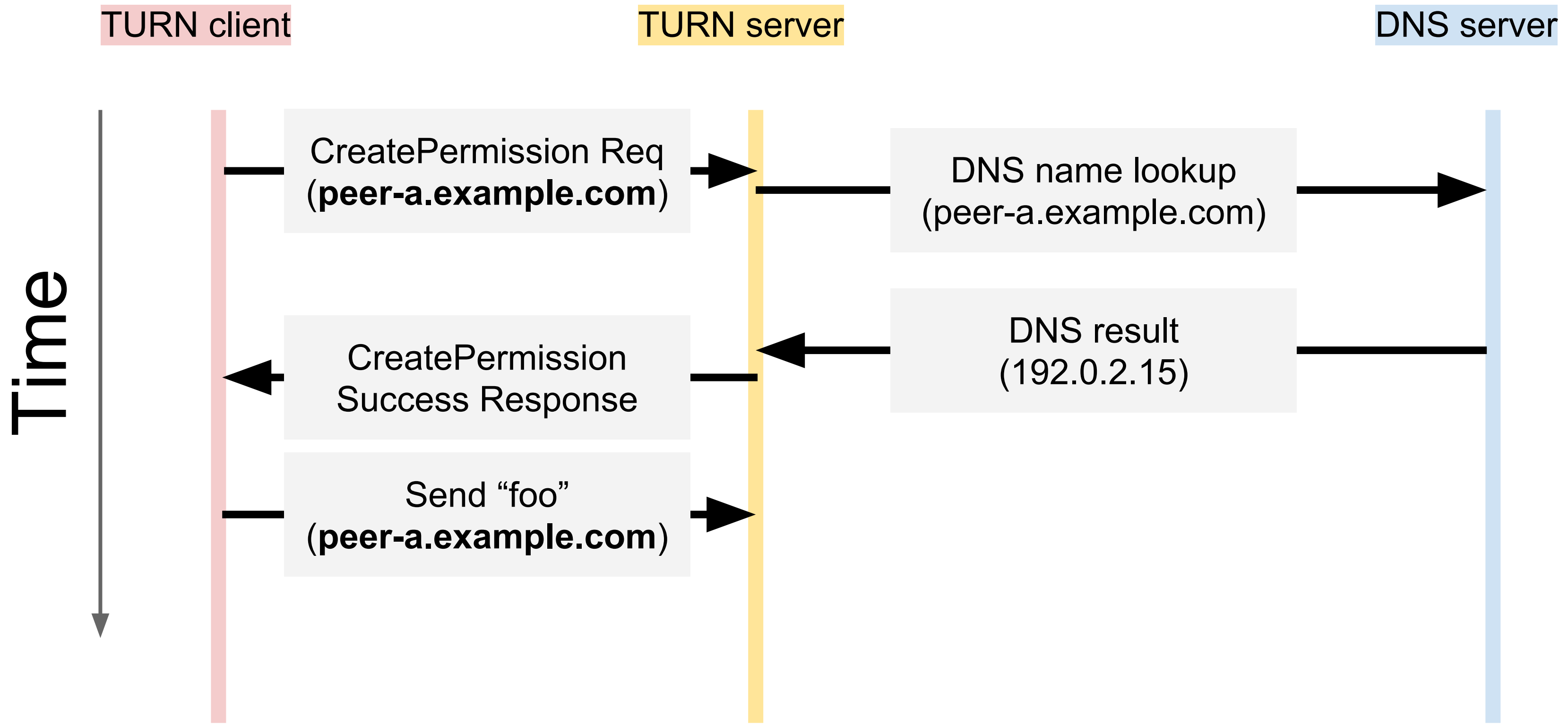
TURN by name message example


TURN client
TURN server
DNS server
Time
CreatePermission Req (**peer-a.example.com**)
DNS name lookup (peer-a.example.com)
DNS result (192.0.2.15)
Cache a name<->IP mapping

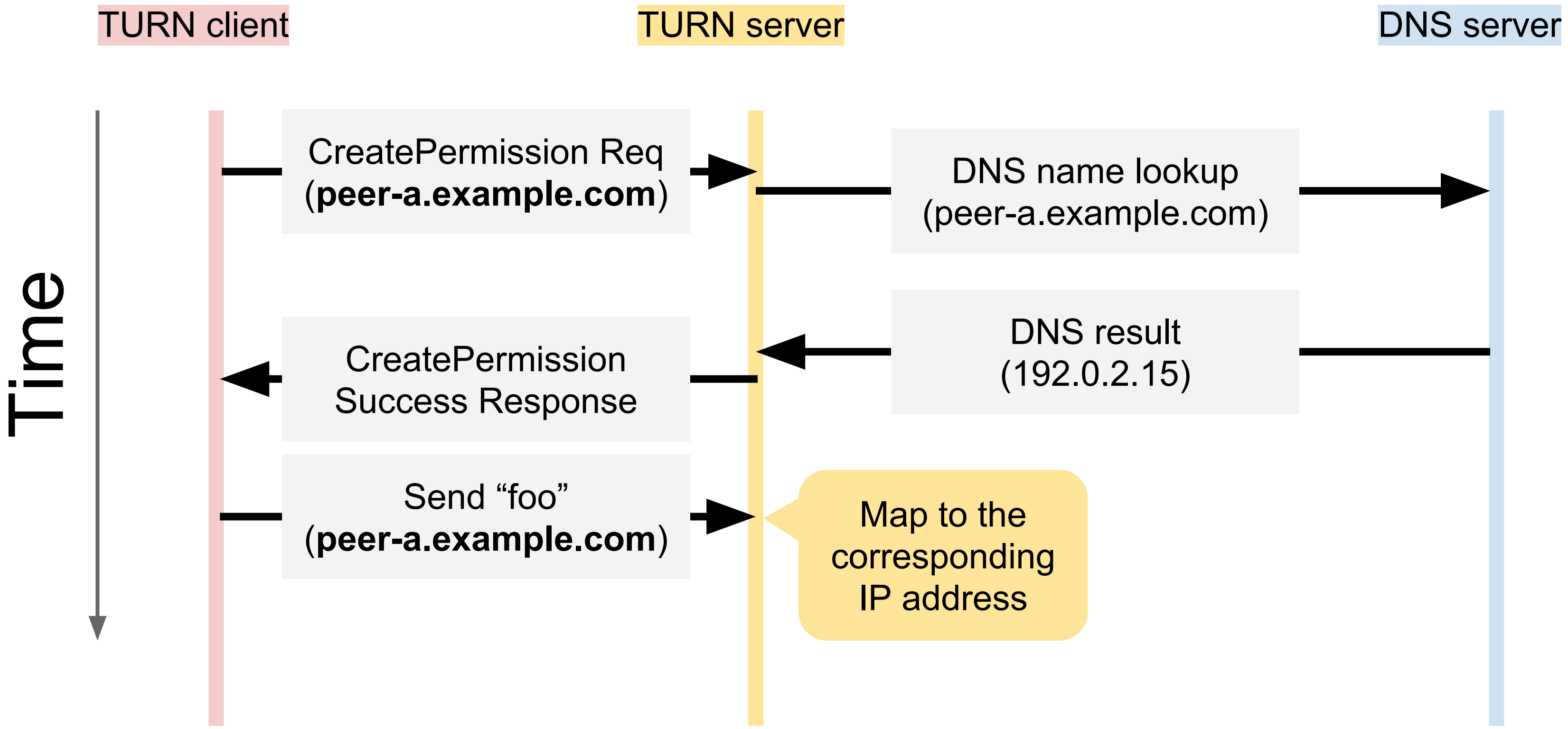
TURN by name message example



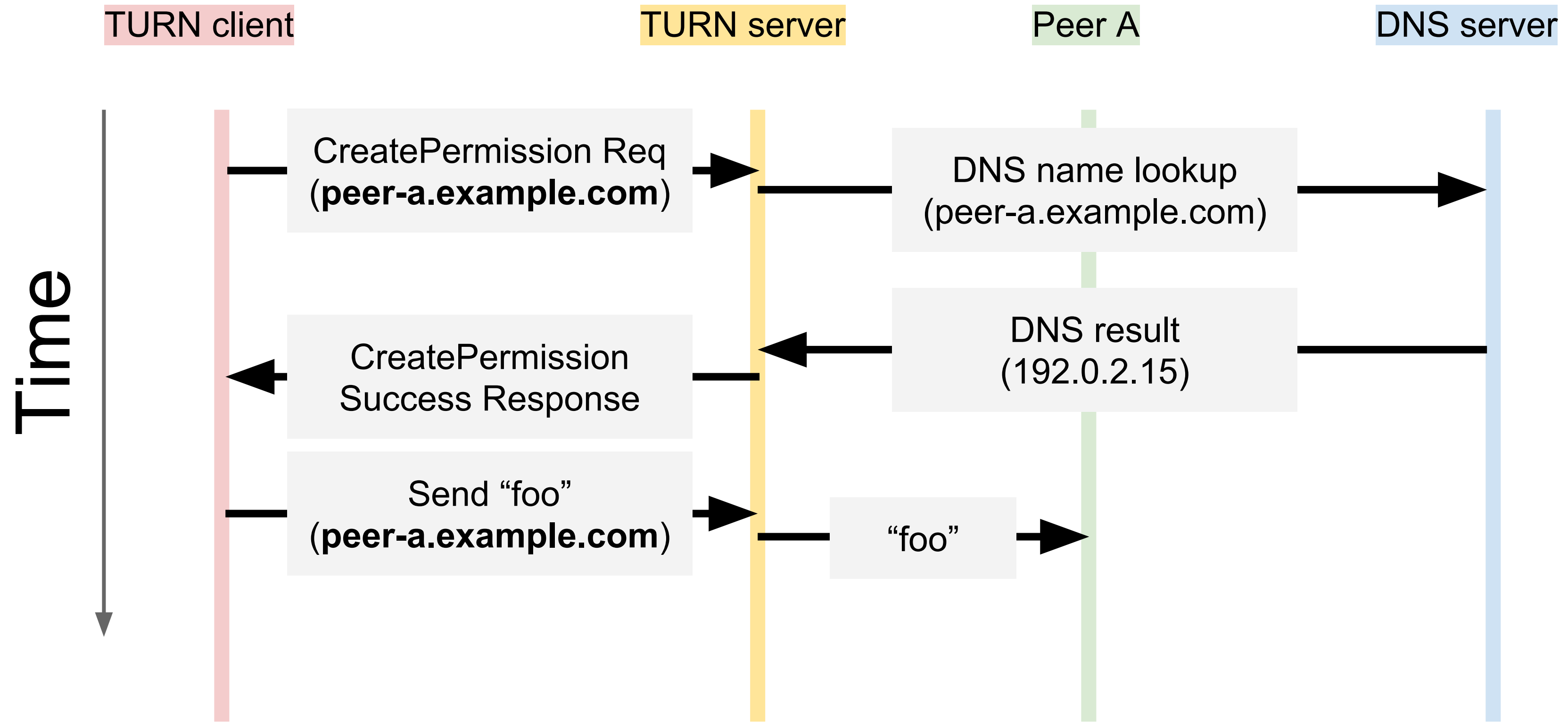
TURN by name message example



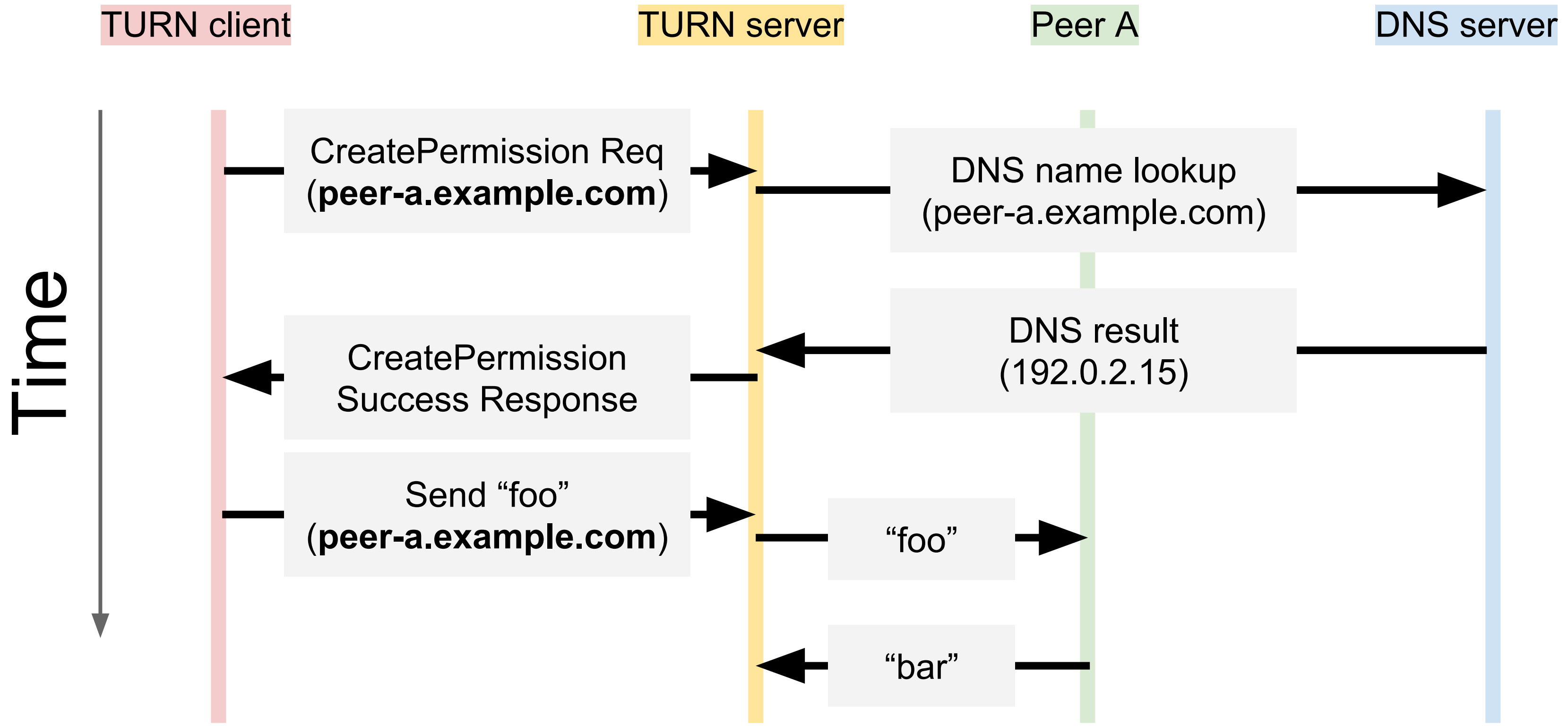
TURN by name message example


TURN client
TURN server
DNS server
Time
CreatePermission Req (**peer-a.example.com**)
DNS name lookup (peer-a.example.com)
DNS result (192.0.2.15)
CreatePermission Success Response
Send “foo” (**peer-a.example.com**)
Map to the corresponding IP address

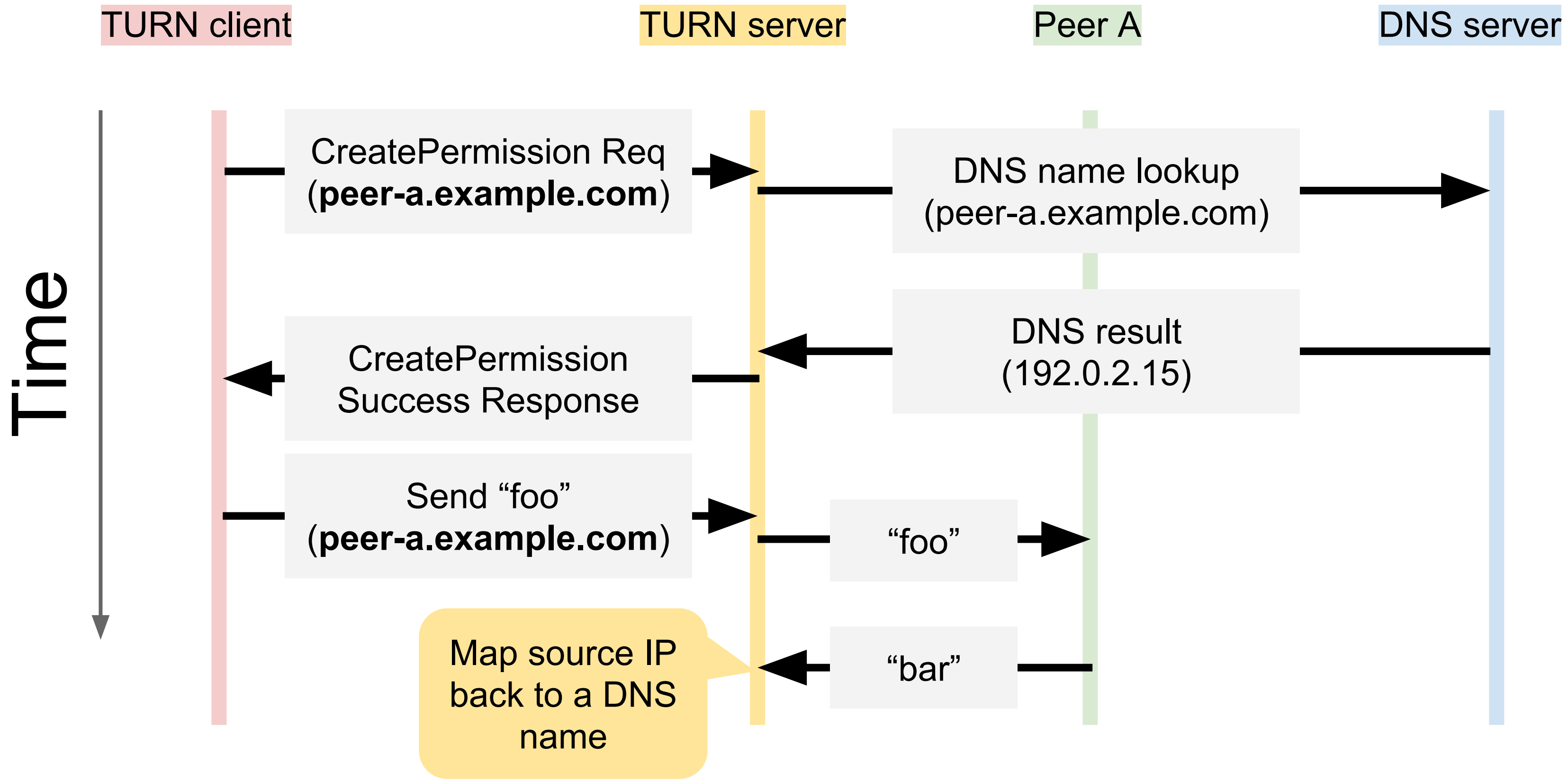
TURN by name message example



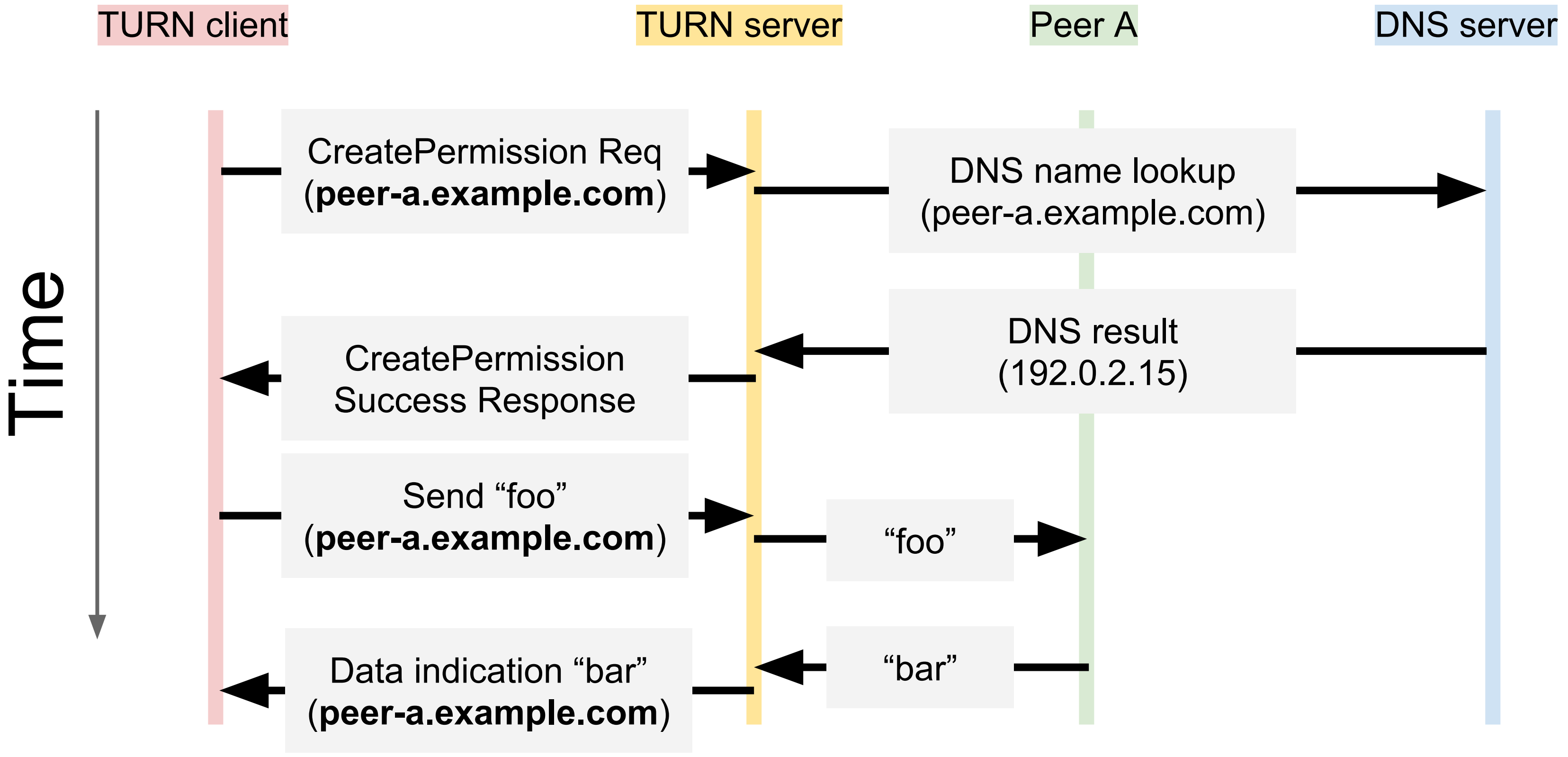
TURN by name message example



TURN by name message example


TURN client
TURN server
Peer A
DNS server
Time
CreatePermission Req (**peer-a.example.com**)
DNS name lookup (peer-a.example.com)
DNS result (192.0.2.15)
CreatePermission Success Response
Send “foo” (**peer-a.example.com**)
“foo”
“bar”
Map source IP back to a DNS name

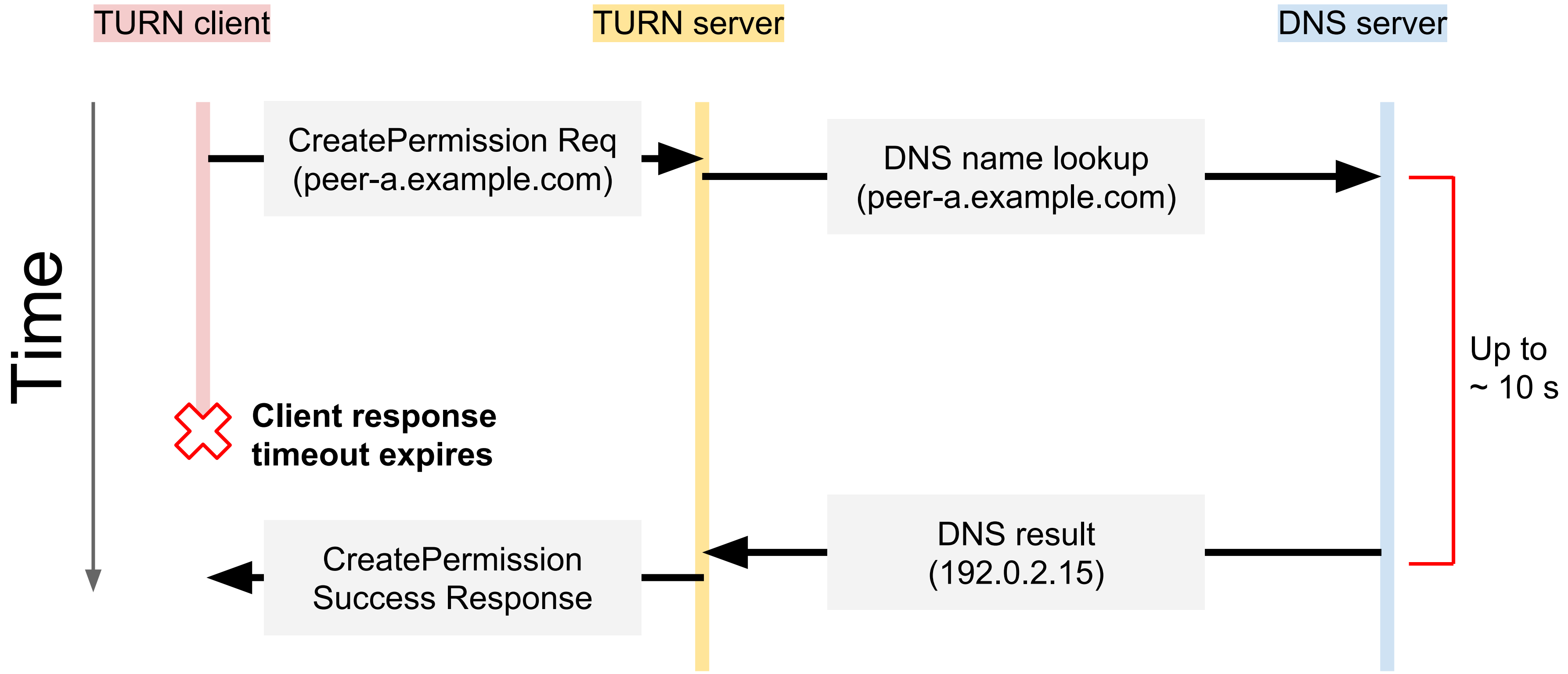
TURN by name message example


TURN client
TURN server
Peer A
DNS server
Time
CreatePermission Req (**peer-a.example.com**)
DNS name lookup (peer-a.example.com)
DNS result (192.0.2.15)
CreatePermission Success Response
Send “foo” (**peer-a.example.com**)
“foo”
“bar”
Data indication “bar” (**peer-a.example.com**)

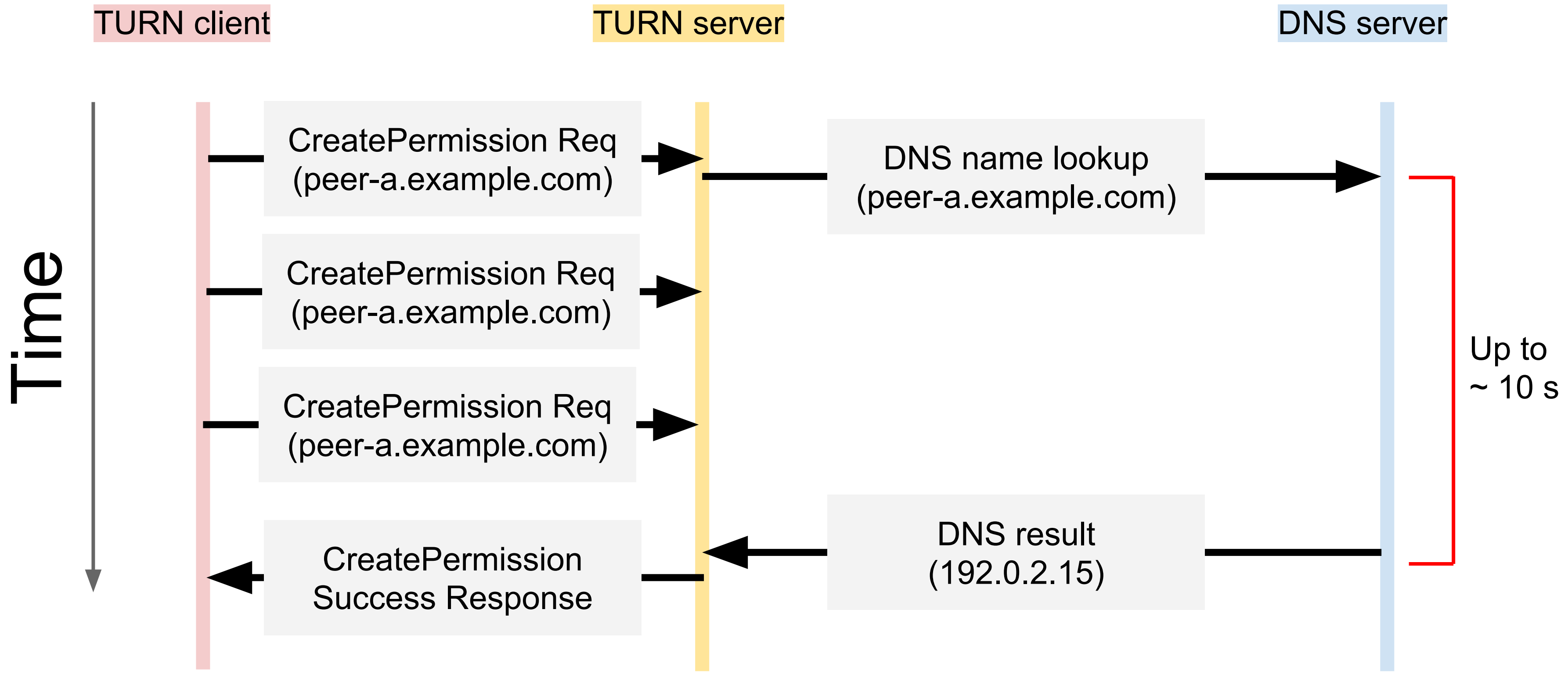
Open problem: timeout conflict

- DNS resolution can be slow (up to ~10 seconds)
- STUN (and hence TURN) request responses have to be fast, ≲79×RTT
    - Designed for actions that can be performed pu
      locally and in-memory
    - Thanks to Jonathan Lennox for spotting this!

Open problem: timeout conflict


TURN client
TURN server
DNS server
Time
CreatePermission Req (peer-a.example.com)
DNS name lookup (peer-a.example.com)
Up to ~ 10 s
Client response timeout expires
DNS result (192.0.2.15)
CreatePermission Success Response

Proposal 0: *Just extend the timeout*

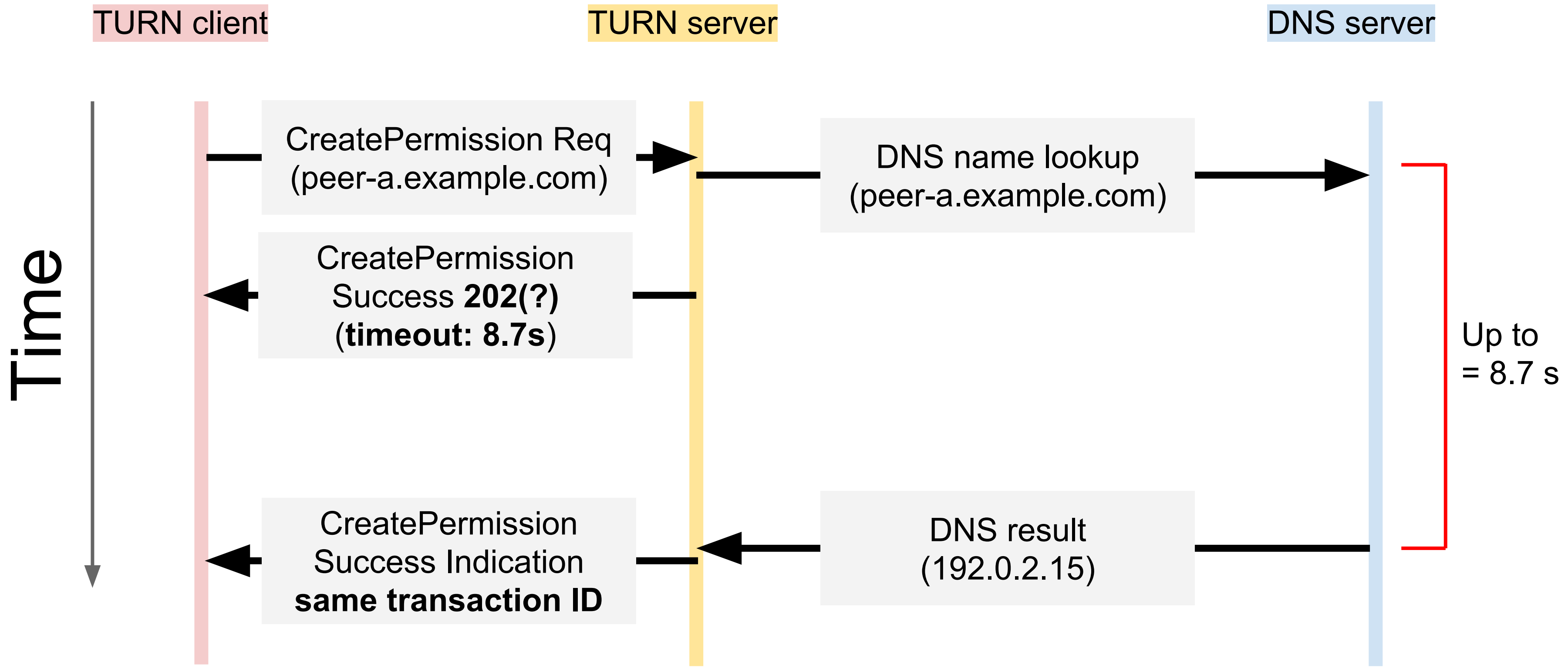


TURN client
TURN server
DNS server
Time
CreatePermission Req (peer-a.example.com)
DNS name lookup (peer-a.example.com)
CreatePermission Req (peer-a.example.com)
CreatePermission Req (peer-a.example.com)
Up to ~ 10 s
DNS result (192.0.2.15)
CreatePermission Success Response

Proposal 0: *Just extend the timeout*

- Problems:
  - Client sends a bunch of extra/redundant reque
    no reason.
  - If the response packet is lost, the retry waits th
    timeout (~10s).

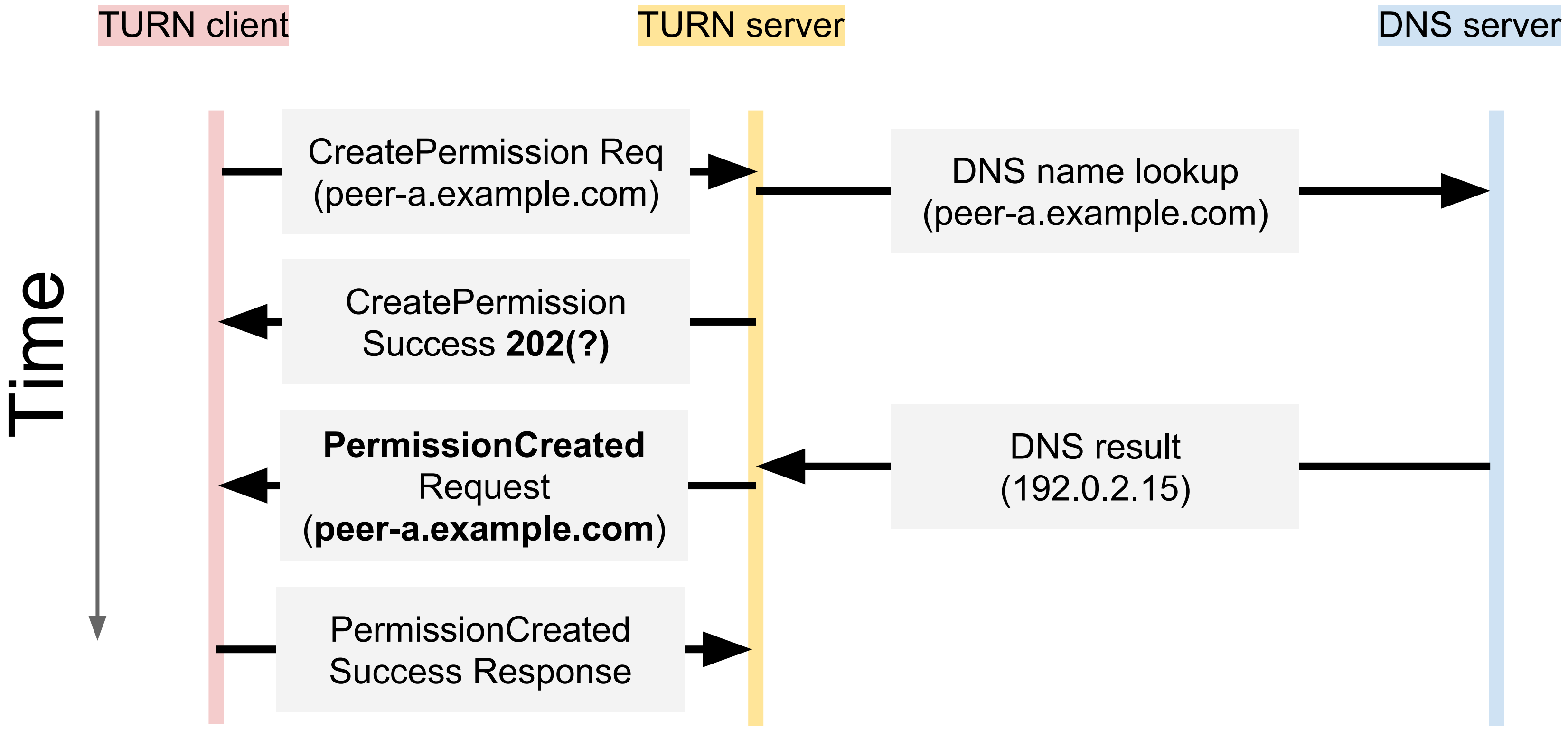
Proposal 1: *Late indication*


TURN client
TURN server
DNS server
Time
CreatePermission Req (peer-a.example.com)
DNS name lookup (peer-a.example.com)
CreatePermission Success **202(?)** (**timeout: 8.7s**)
Up to = 8.7 s
CreatePermission Success Indication **same transaction ID**
DNS result (192.0.2.15)

Proposal 1: *Late indication*

- Problems:
  - If the response packet is lost, the retry waits th
    timeout (~10s).

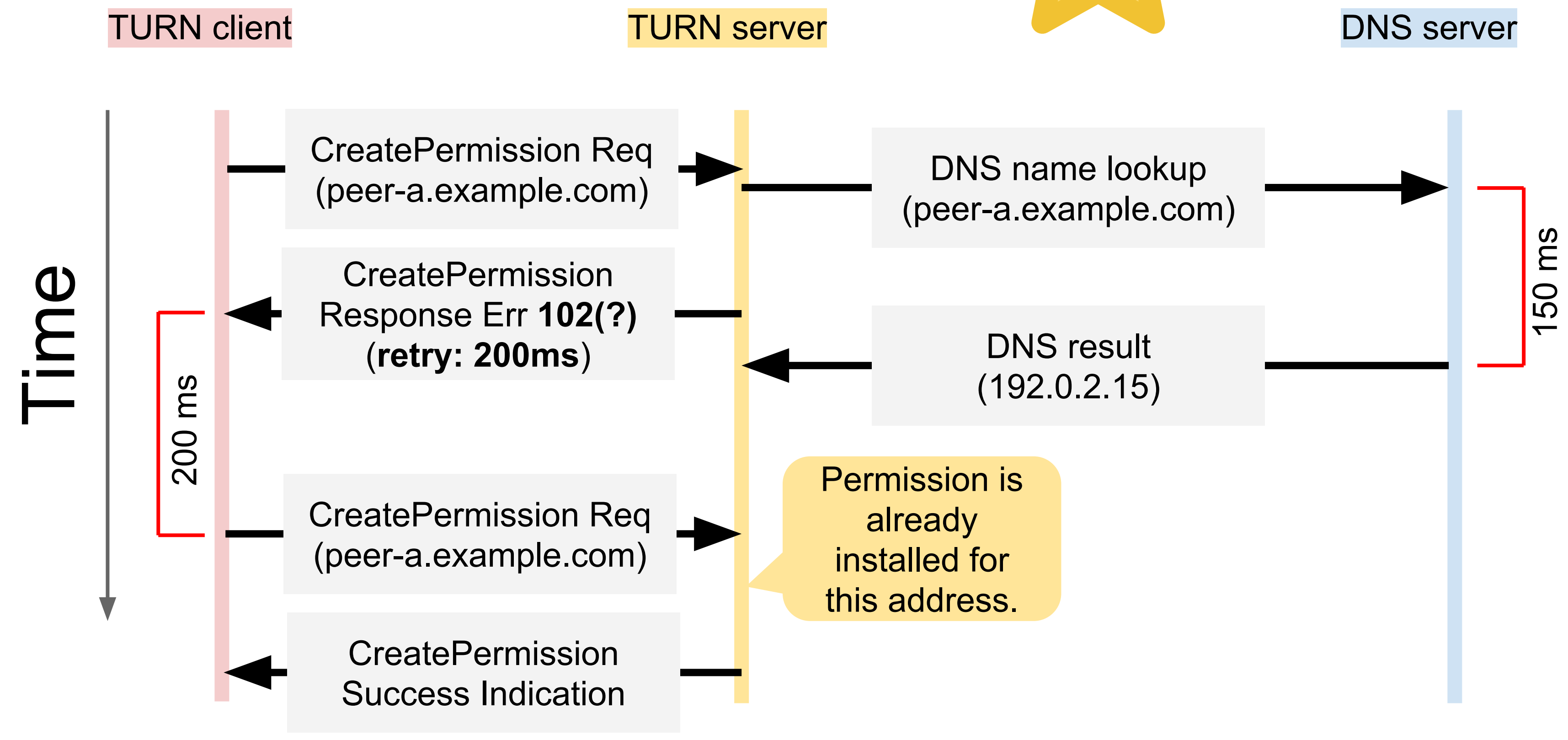
Proposal 2: *Reverse Request*


TURN client
TURN server
DNS server
Time
CreatePermission Req (peer-a.example.com)
DNS name lookup (peer-a.example.com)
CreatePermission Success **202(?)**
PermissionCreated Request (**peer-a.example.com**)
DNS result (192.0.2.15)
PermissionCreated Success Response

Proposal 2: *Reverse Request*

- Problems:
  - This would be the first Server-to-Client Reque
    STUN history.

Proposal 3: *EAGAIN*



TURN client
TURN server
DNS server
Time
CreatePermission Req (peer-a.example.com)
DNS name lookup (peer-a.example.com)
150 ms
CreatePermission Response Err **102(?)** (**retry: 200ms**)
DNS result (192.0.2.15)
200 ms
CreatePermission Req (peer-a.example.com)
Permission is already installed for this address.
CreatePermission Success Indication

In conclusion, TURN by name

- solves
  - TURN to DNS-named endpoints on restricted networks
  - making TURN parallel to SOCKS 5 or HTTP CONNECT
- but still needs discussion about
  - the choice to hide peer IPs from the client
  - the interaction with STUN timeouts