ICE, TURN and STUN for NAT Traversal

Stephen Strowes
ENDS Seminar, 19/Nov/2008

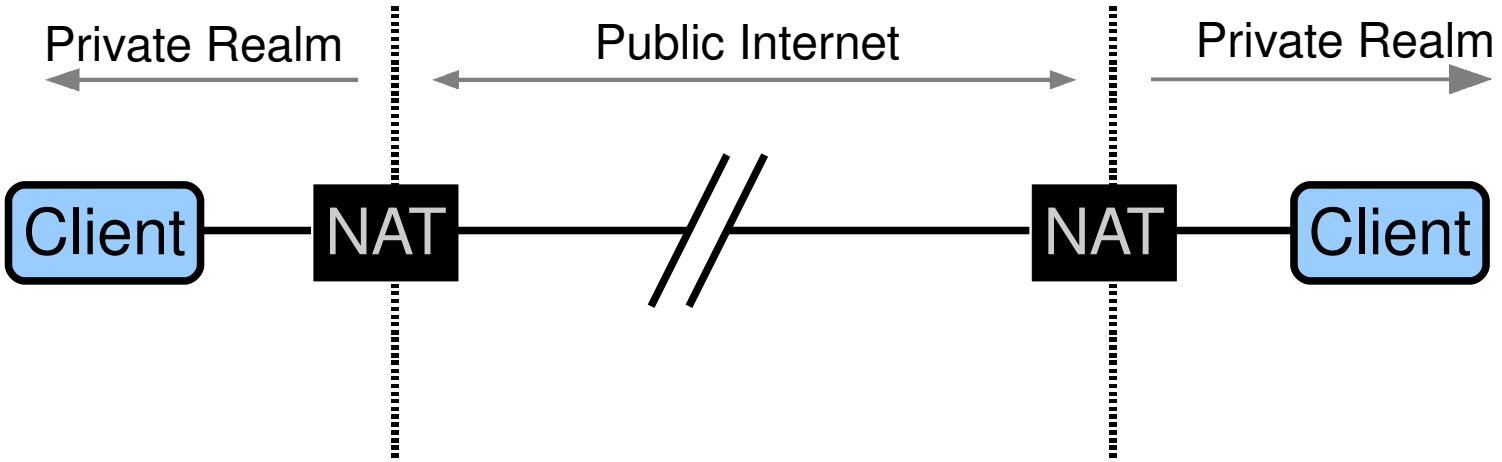
Nokia Research Center

Nokia

- 4 month internship, working with Nokia Research Centre
  - Based at a Nokia lablet at the Helsinki University of Technology (TKK)
- Part of the Future Internet Team
  - Learning about ICE *et al* for NAT traversal
  - Instrumenting an existing implementation for cross-platform deployment
  - Building a server-side platform for test management and data collection
- This talk is more of an overview of ICE for NAT traversal than it is the details of my work at Nokia

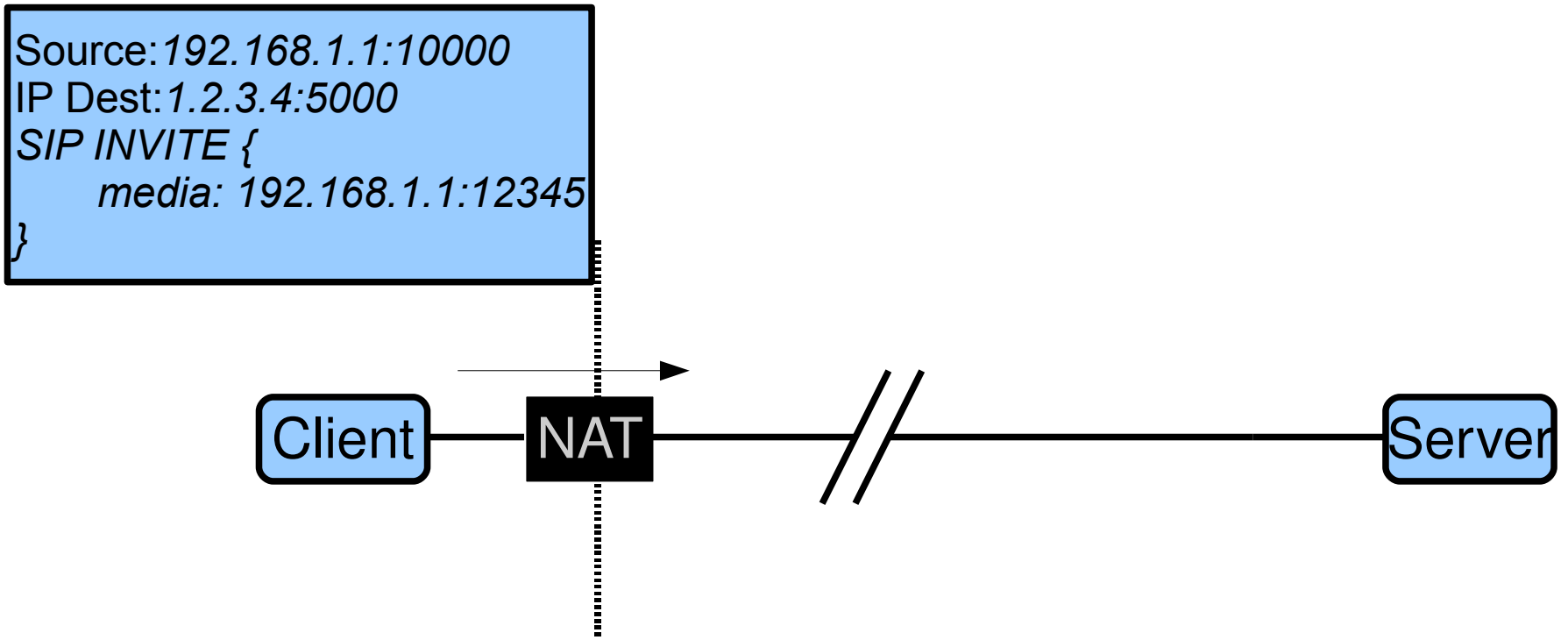
Network Address Translation

- Breaks the end-to-endianness of the network, which various protocols expect.



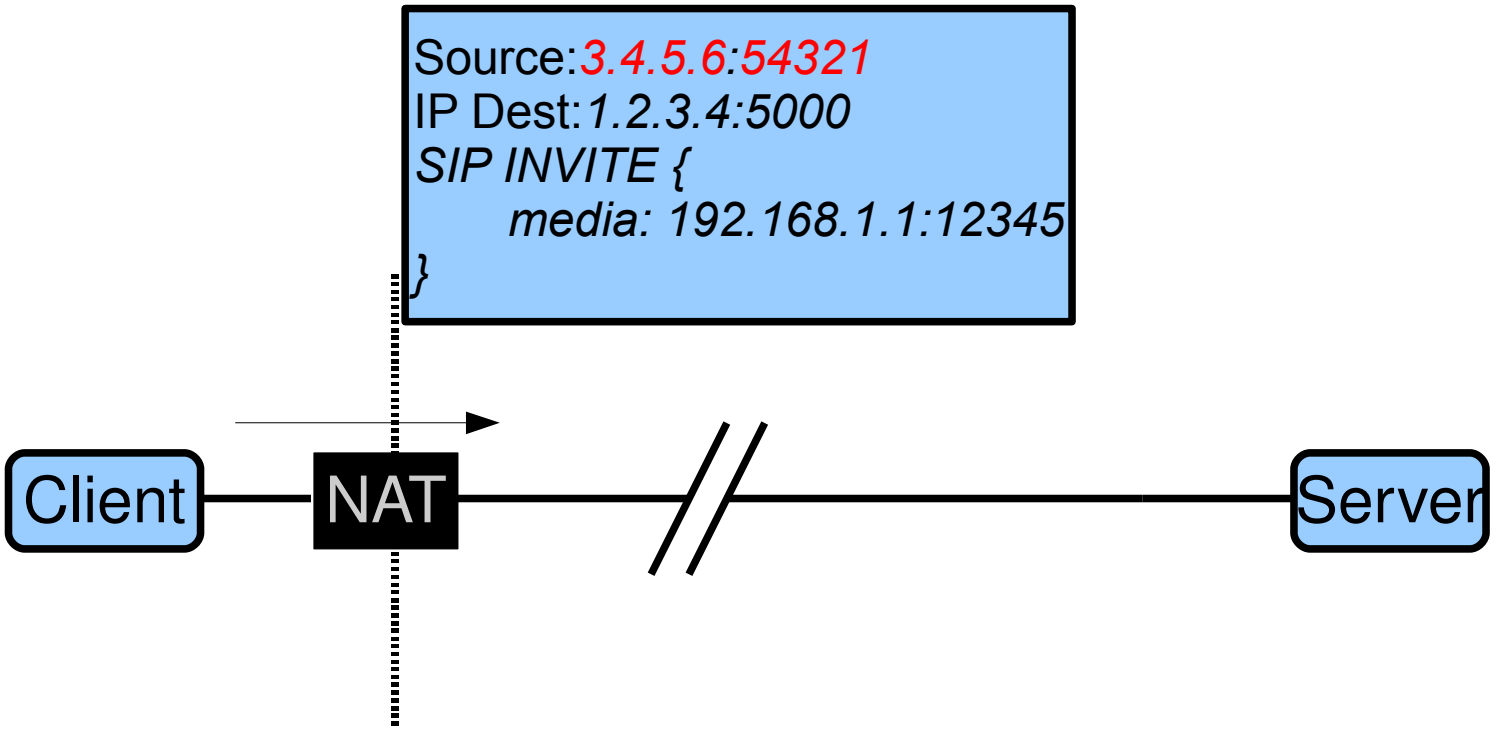
Network Address Translation

- Breaks the end-to-endianness of the network, which various protocols expect.



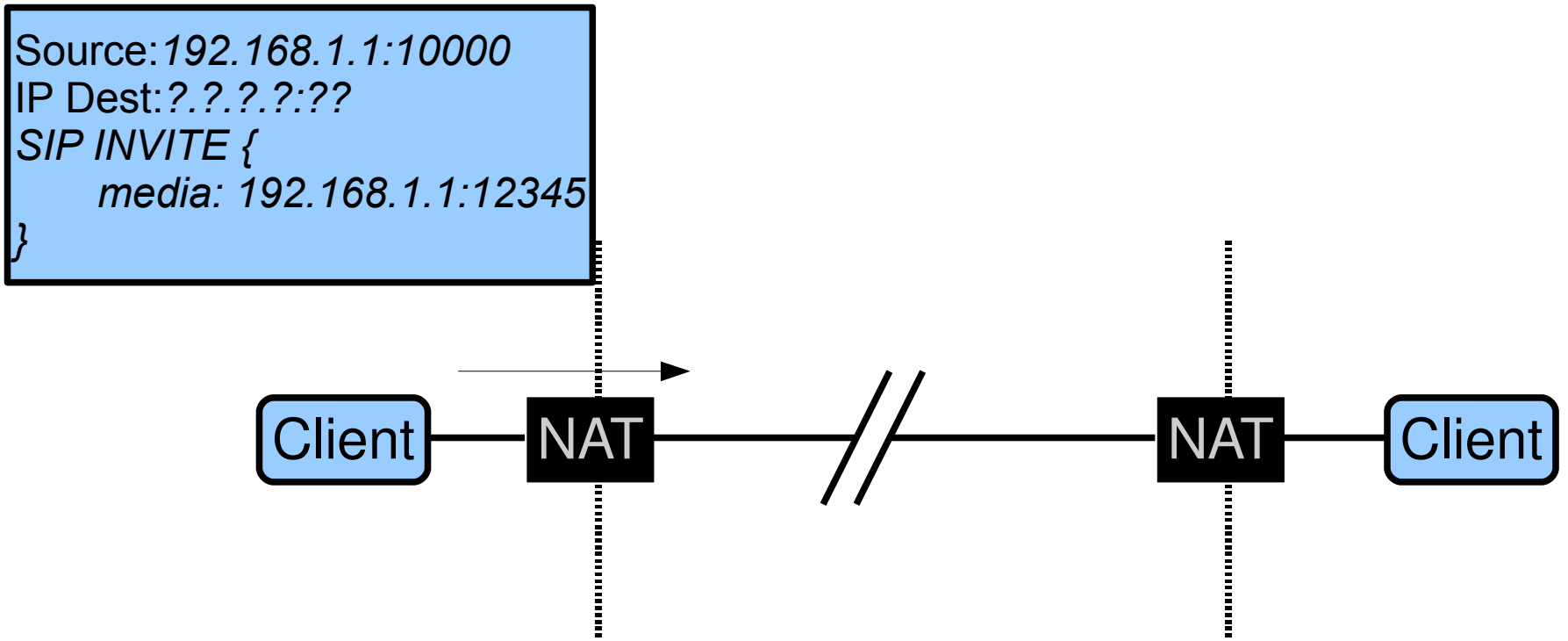
Network Address Translation

- Breaks the end-to-endianness of the network, which various protocols expect.



Network Address Translation

- Breaks the end-to-endianness of the network, which various protocols expect.



Network Address Translation, Terminology

  - Full cone
  - Restricted cone
  - Port-restricted cone
  - Symmetric
  - ... and then different behaviours within the *same* NAT
- Packet rewriting (ALGs)...
- NATs are black-boxes
  - *Make few assumptions*

ICE, Interactive Connectivity Establishment

- ICE is a mechanism to allow media streams to flow between two peers in a NATed environment
- An important extension to SIP, it can be used by other signalling mechanisms
  - Allows hosts in the same private realm to communicate directly
  - Allows two hosts, each located behind their own symmetric NAT, to communicate via relays
    - ... and variations in-between ...

ICE, Interactive Connectivity Establishment

- In essence:
  - Peer learns about its network environment
  - Peers exchange this information over a signalling channel (e.g., SIP)
  - Systematically probe possible combinations of transport addresses to find one which works

ICE, Interactive Connectivity Establishment

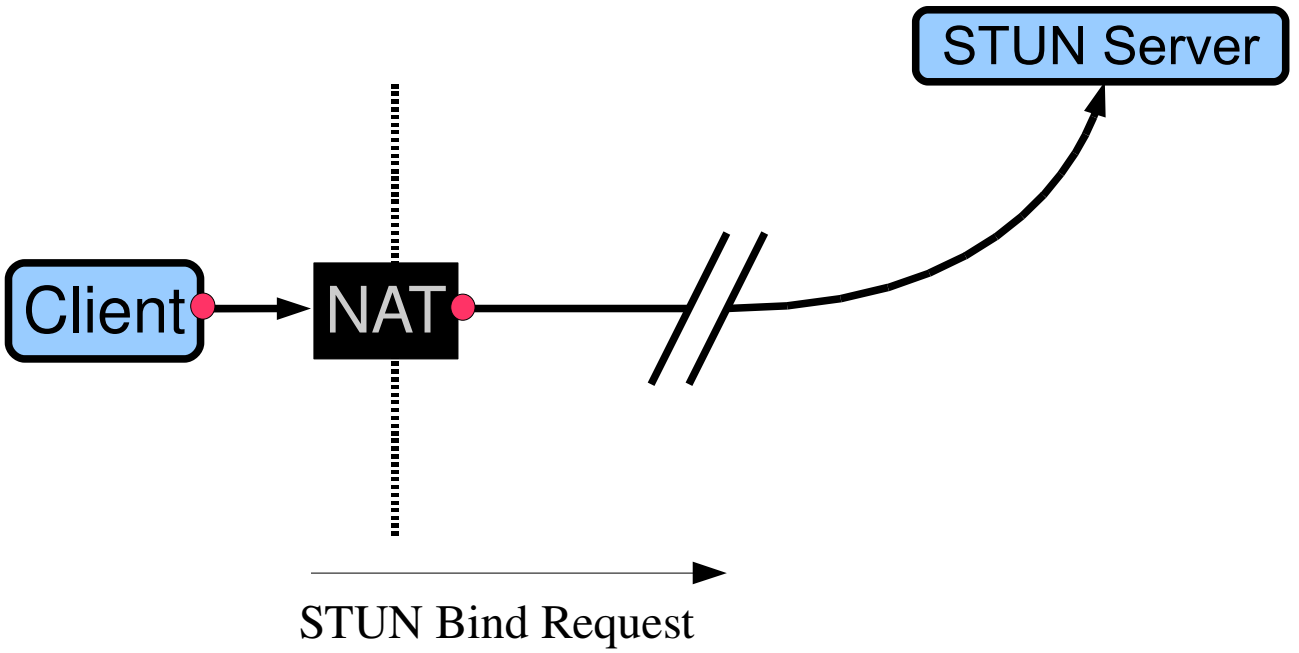
- In a little more detail:
  1. Candidate gathering
     - STUN
     - TURN
  2. Prioritisation
  3. Exchange
  4. Connectivity checks
  5. Coordination
  6. Communication

ICE, Candidate Gathering

- Uses STUN and TURN
  - Each host possibly has multiple candidates *per component*

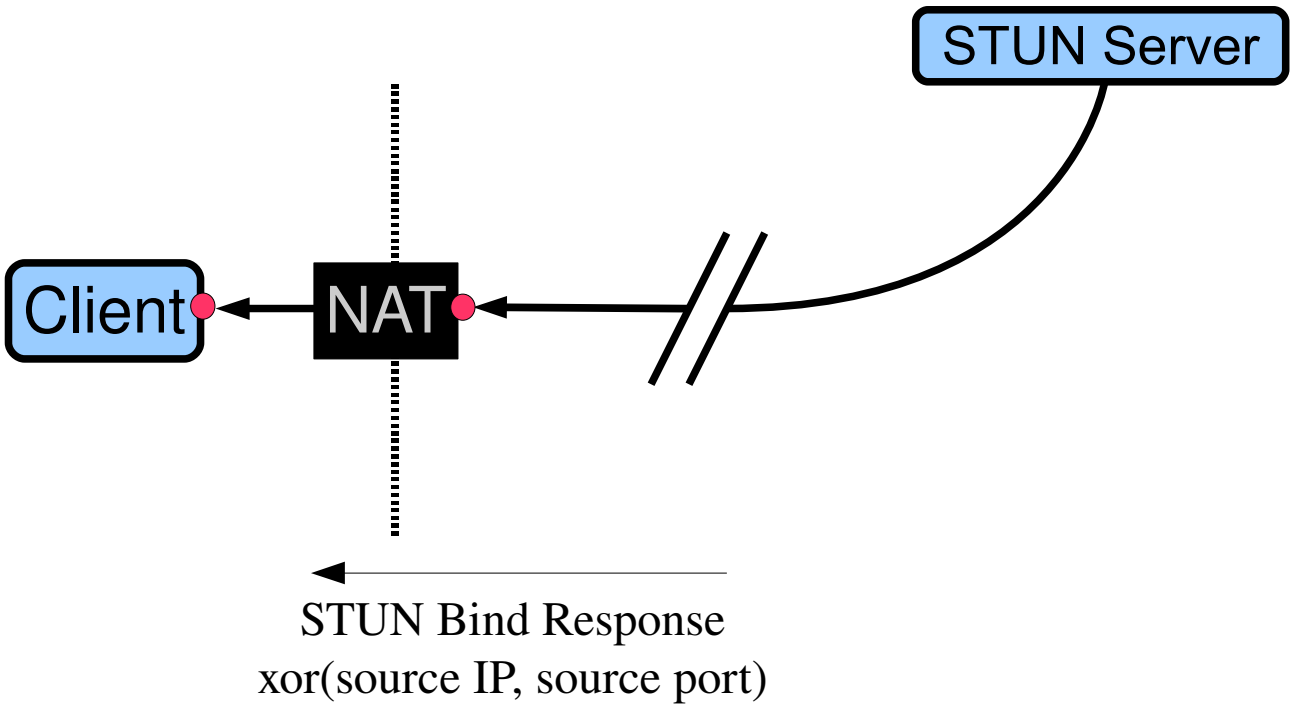
STUN, Session Traversal Utilities for NAT

- Returns the public-side of a NAT binding



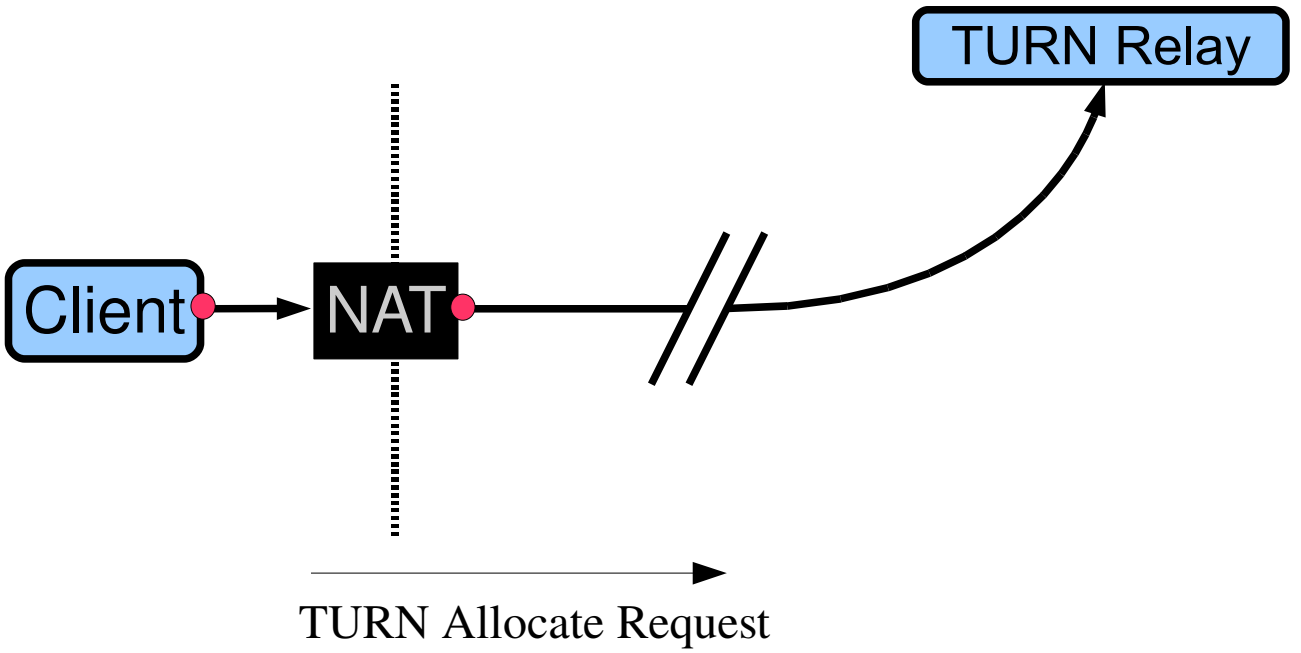
STUN, Session Traversal Utilities for NAT

- Returns the public-side of a NAT binding
  - XOR-mapped address



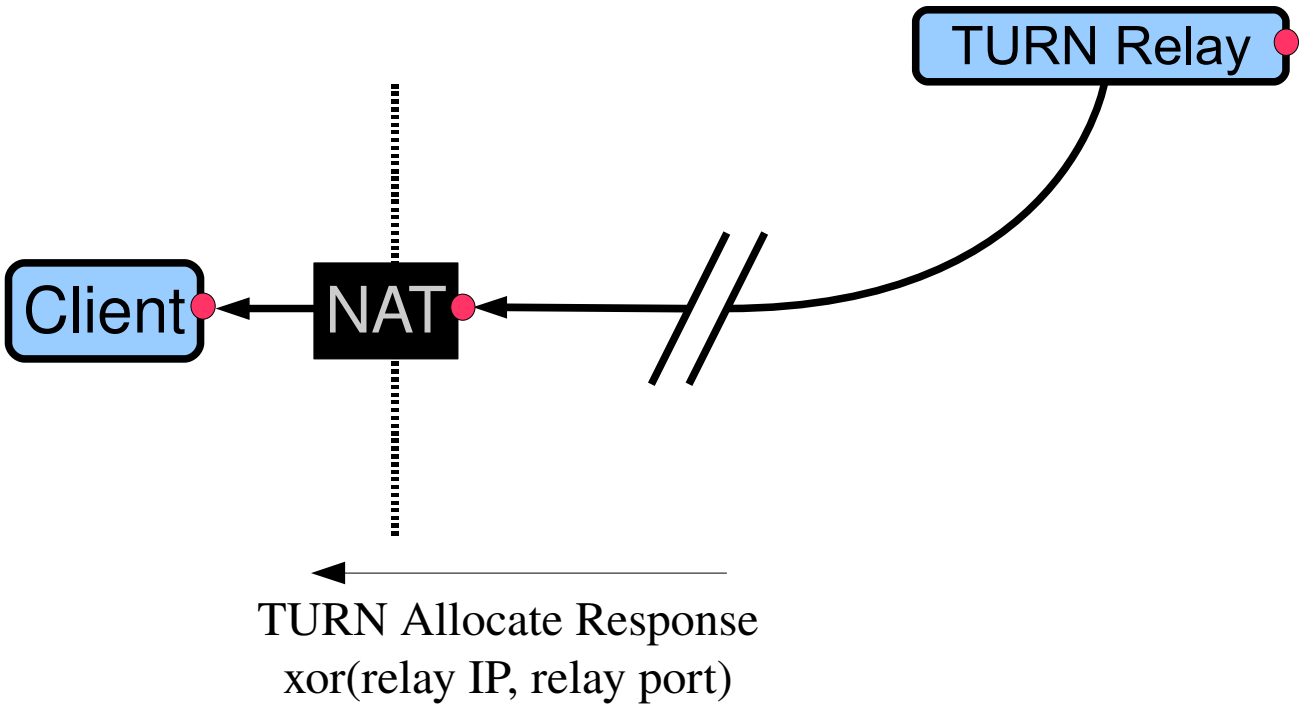
TURN, Traversal Using Relays around NAT

- Allocate a socket on a relay



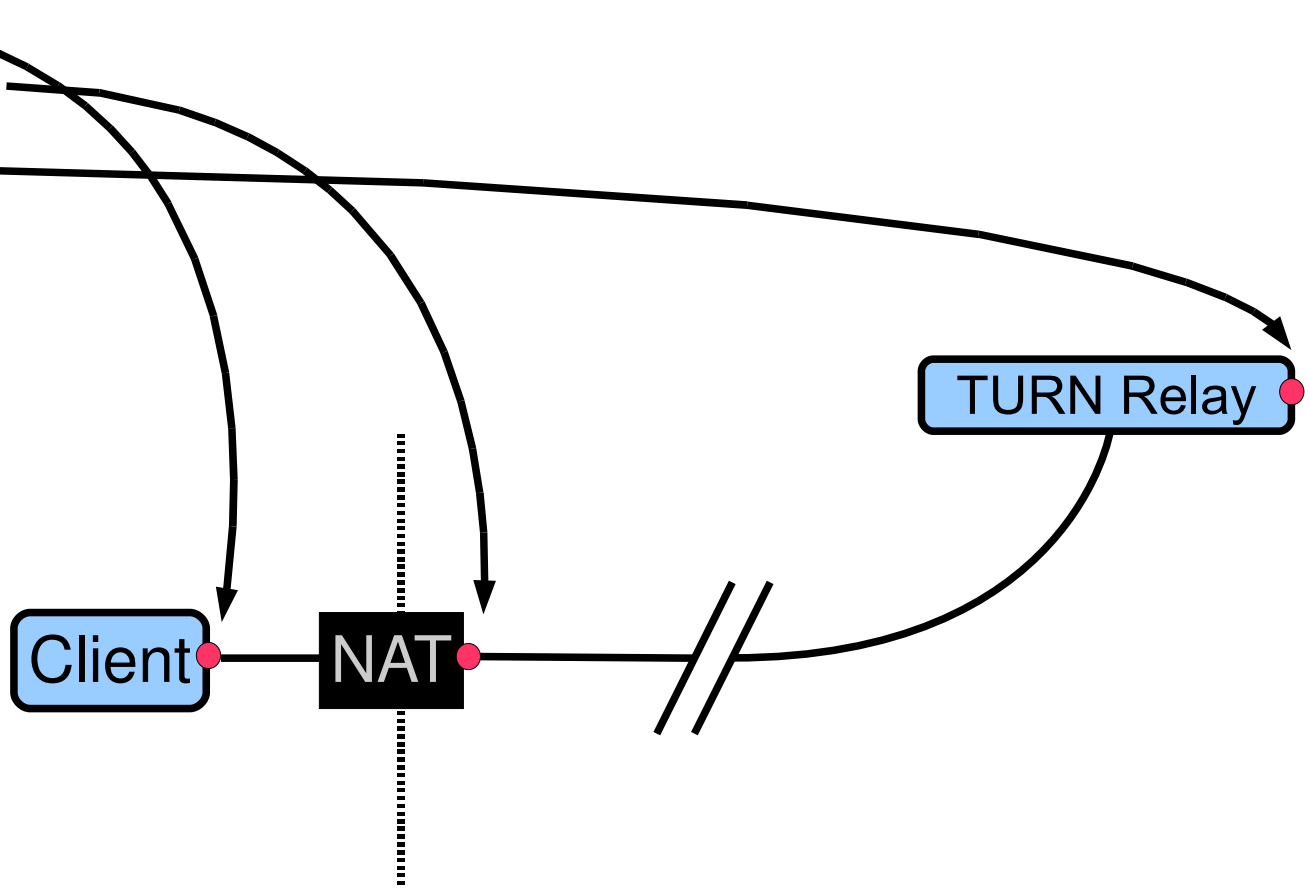
TURN, Traversal Using Relays around NAT

- Allocate a socket on a relay
- *Permissions* inform the relay which locations it should accept packets from for relaying back to the client



ICE, Candidate Gathering

- Uses STUN and TURN
- Each host possibly has multiple candidates *per component*
  - Host
  - Server reflexive
  - Relay candidate
  - Peer reflexive

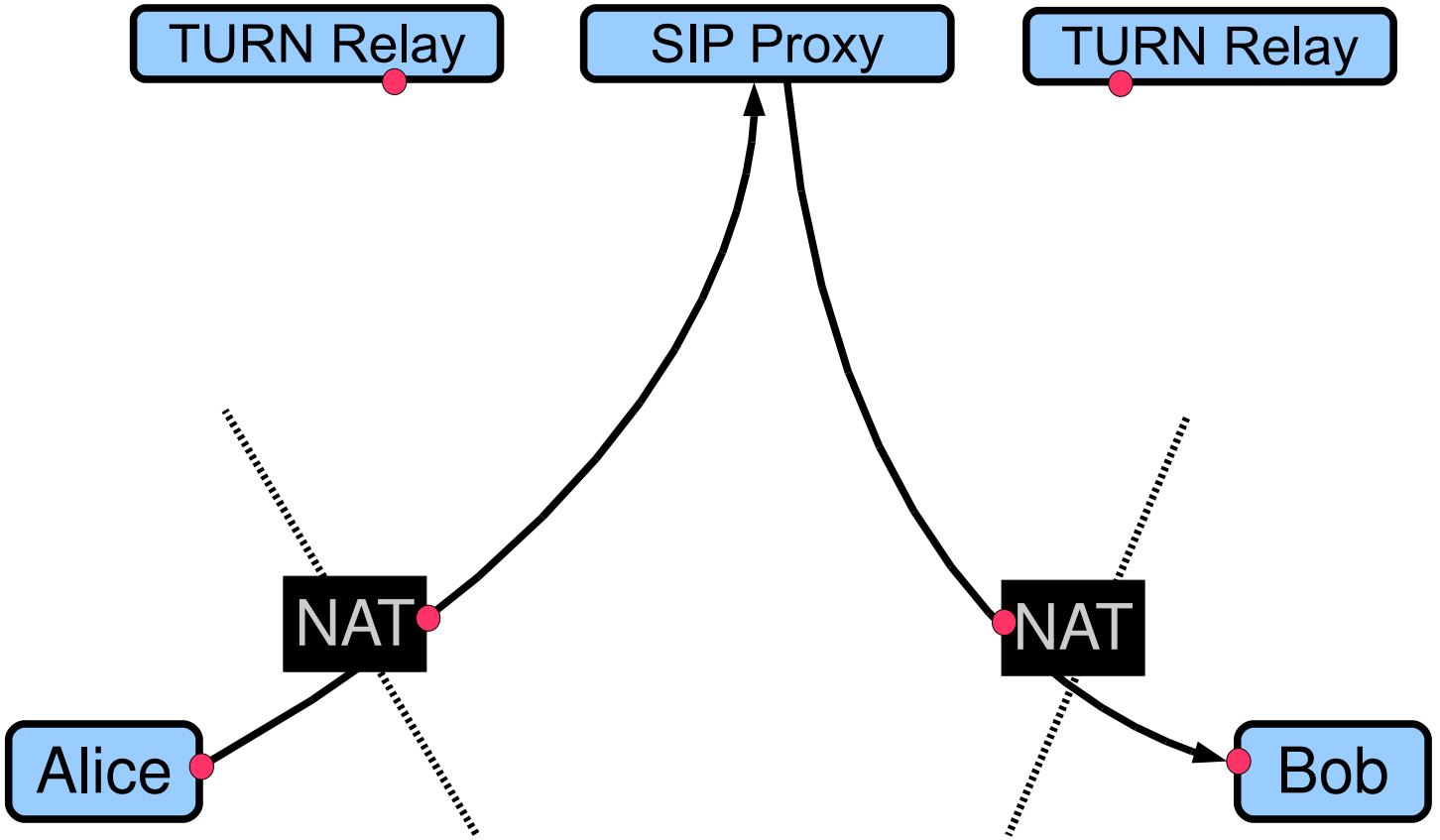


ICE, Prioritisation

- Candidates prioritised
  - In essence: host candidates good, relay candidates bad
- Hosts can exert preference over:
  - *Type* (host, prflx, srflx, relay)
  - *Local considerations* (e.g., specific interfaces)
  - *Component ID* (so that, e.g., data streams are probed prior to control streams, to move data faster)

ICE, Candidate Exchange

- Signalling carries the gathered candidates
  - In SIP, *INVITE* & response
- Candidates carried in SDP description for ICE usage



ICE, Connectivity Checks

- Pair-up candidates
- Prioritise according to magic formula
- Prune duplicates (and retain highest priority of the two)

Alice's host candidate – Bob's host candidate

Alice's server reflexive candidate – Bob's host candidate

ICE, Connectivity Checks

- Pair-up candidates
- Prioritise according to magic formula
- Prune duplicates (and retain highest priority of the two)

Alice's host candidate – Bob's host candidate

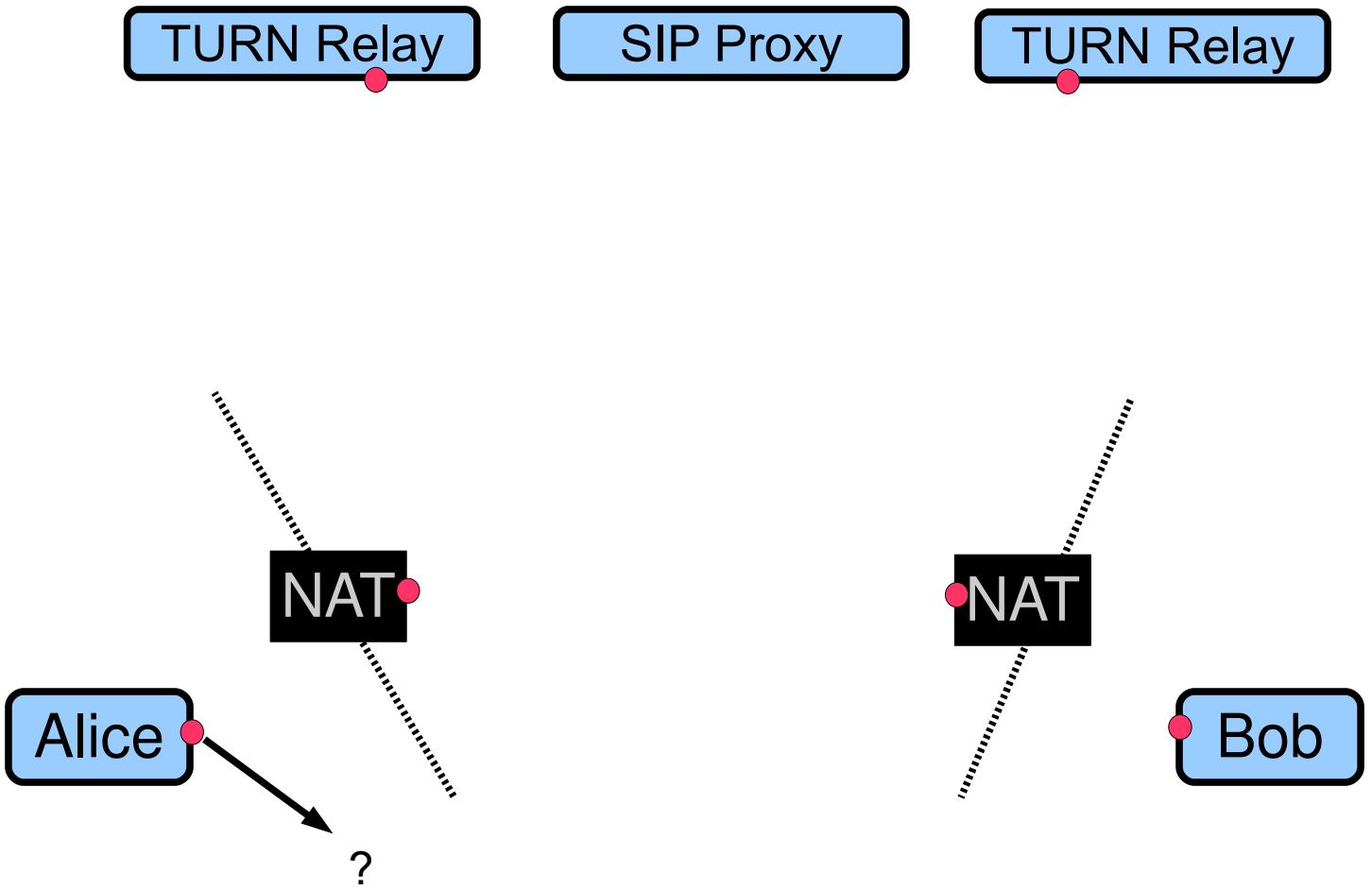
Alice's host candidate – Bob's host candidate

ICE, Connectivity Checks

- Series of STUN requests and responses between peers
- Checks are paced
  - 1 every ~20ms
- Frozen algorithm
  - Normal checks (following prioritisation)
  - Triggered checks (optimisation)

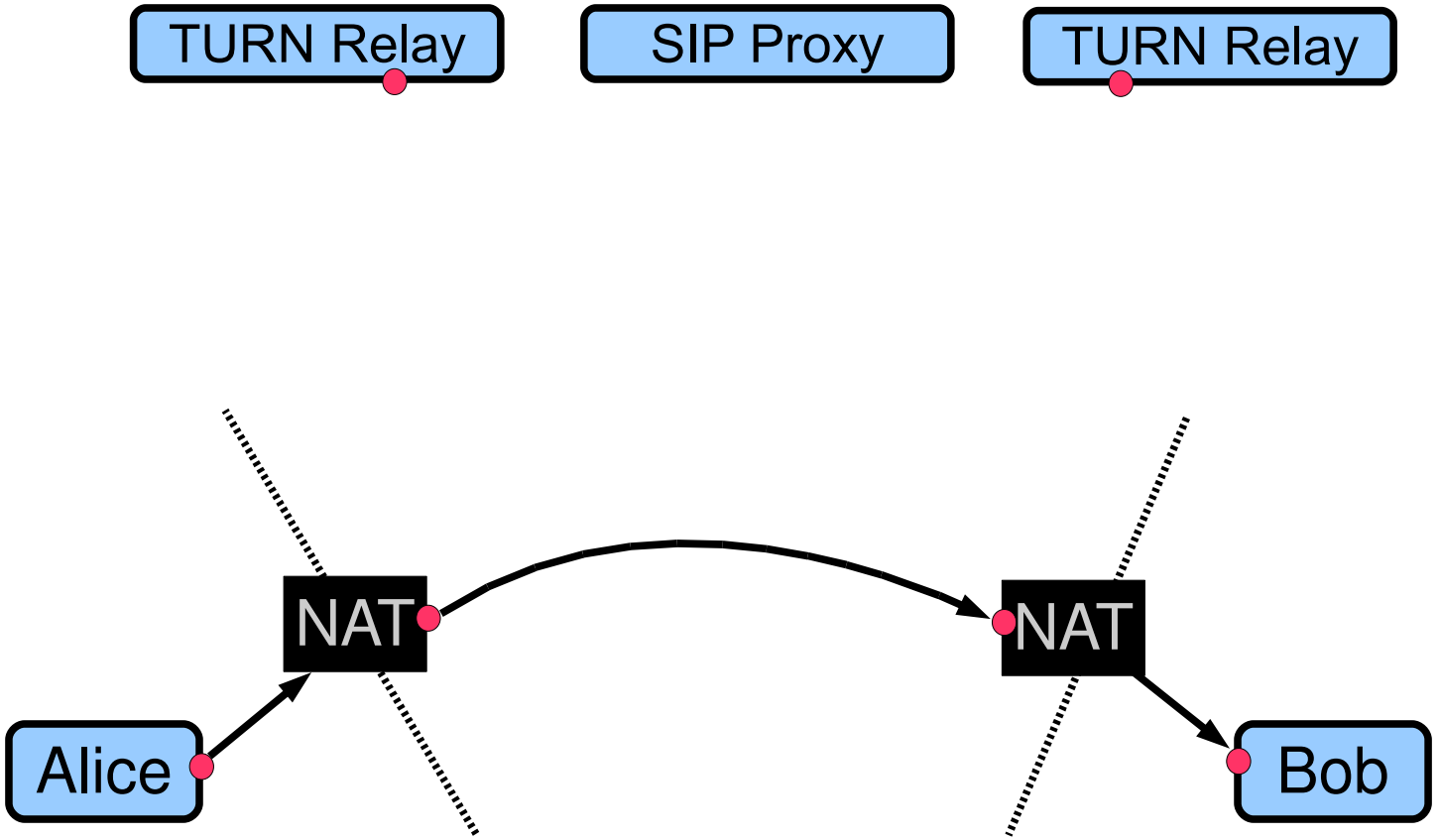
ICE, Connectivity Checks

- Alice's host cand – Bob's host cand



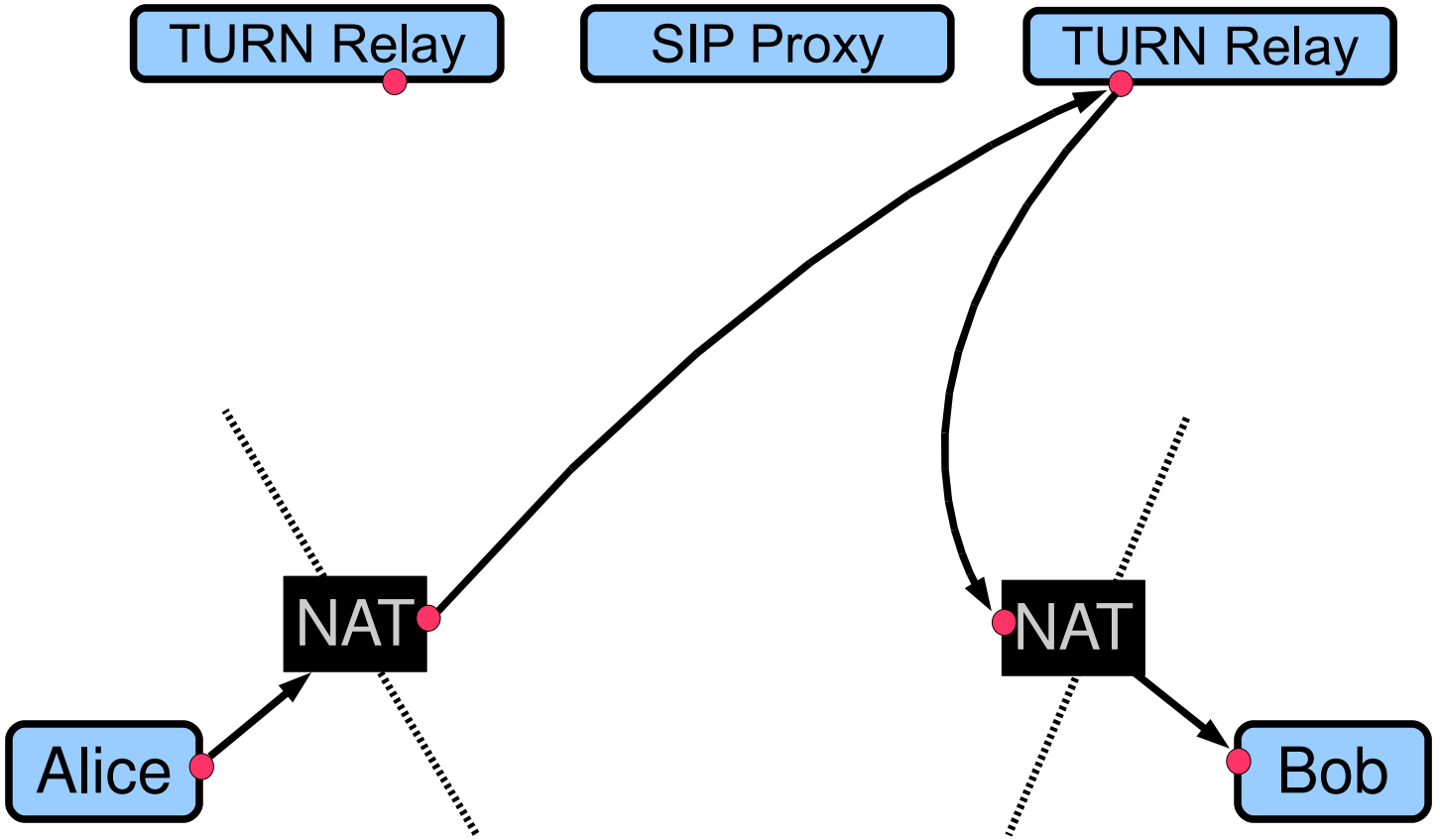
ICE, Connectivity Checks

- Alice's host cand – Bob's server reflexive cand



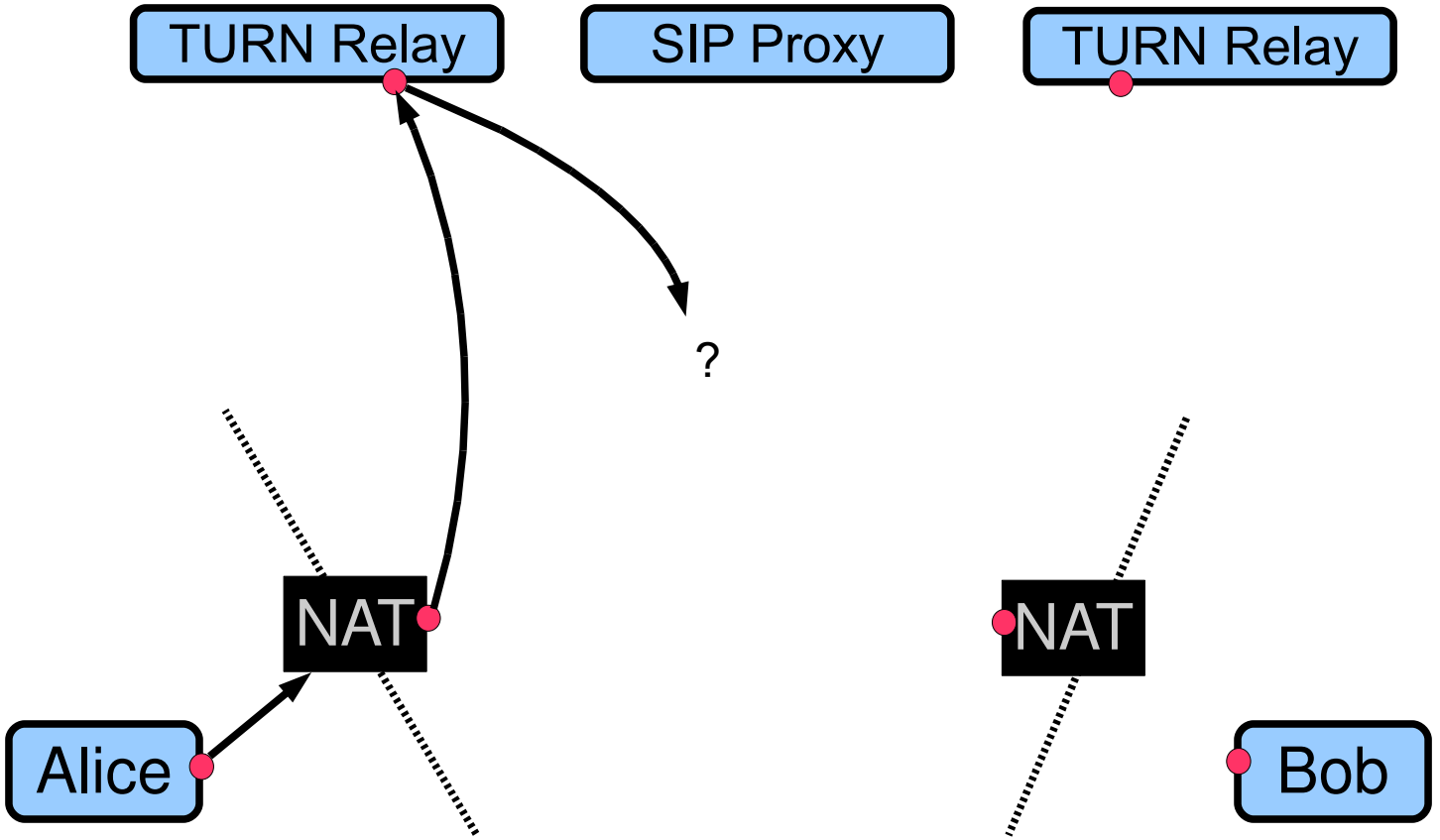
ICE, Connectivity Checks

- Alice's host cand – Bob's relay cand



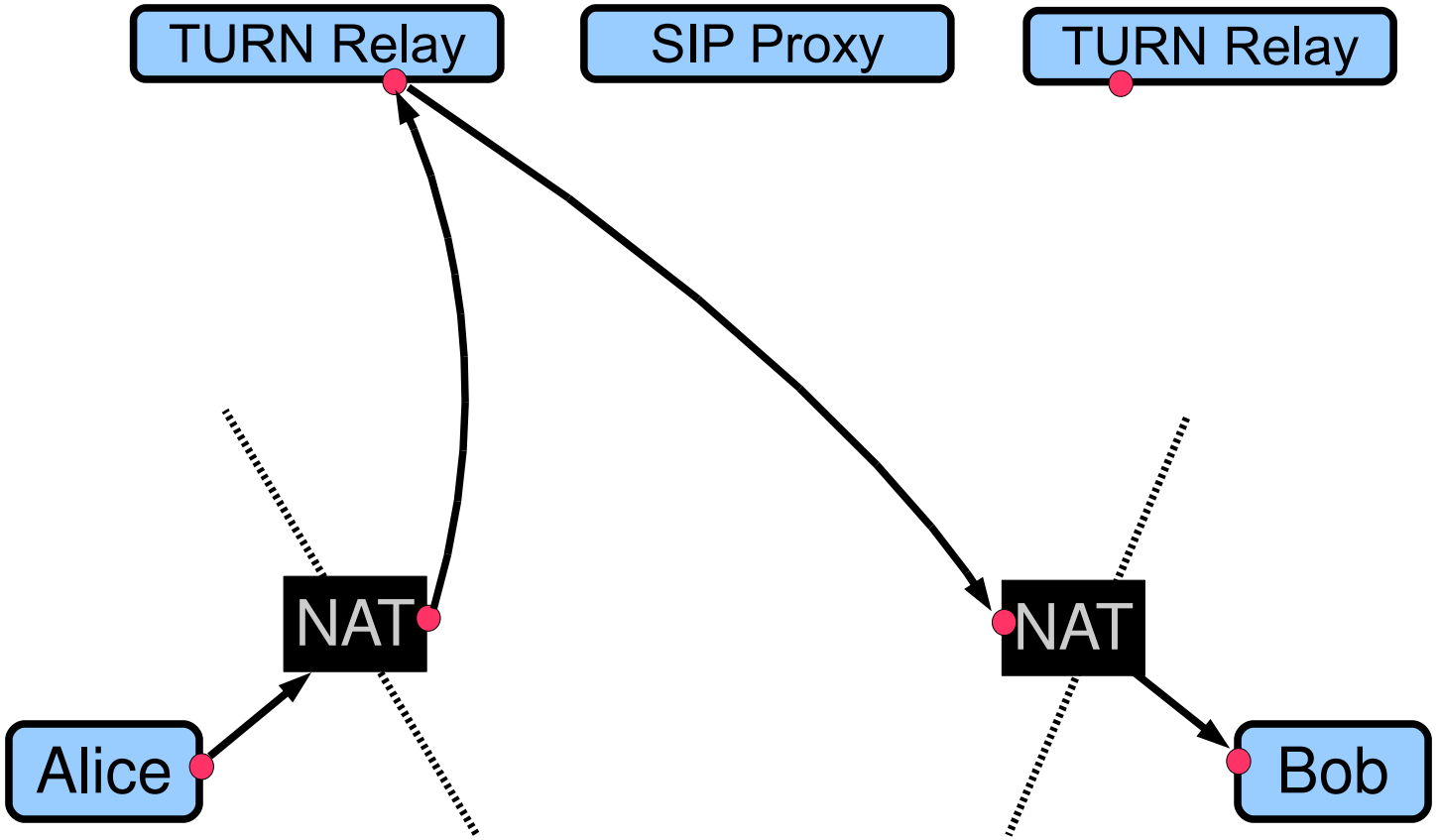
ICE, Connectivity Checks

- Alice's relay cand – Bob's host cand



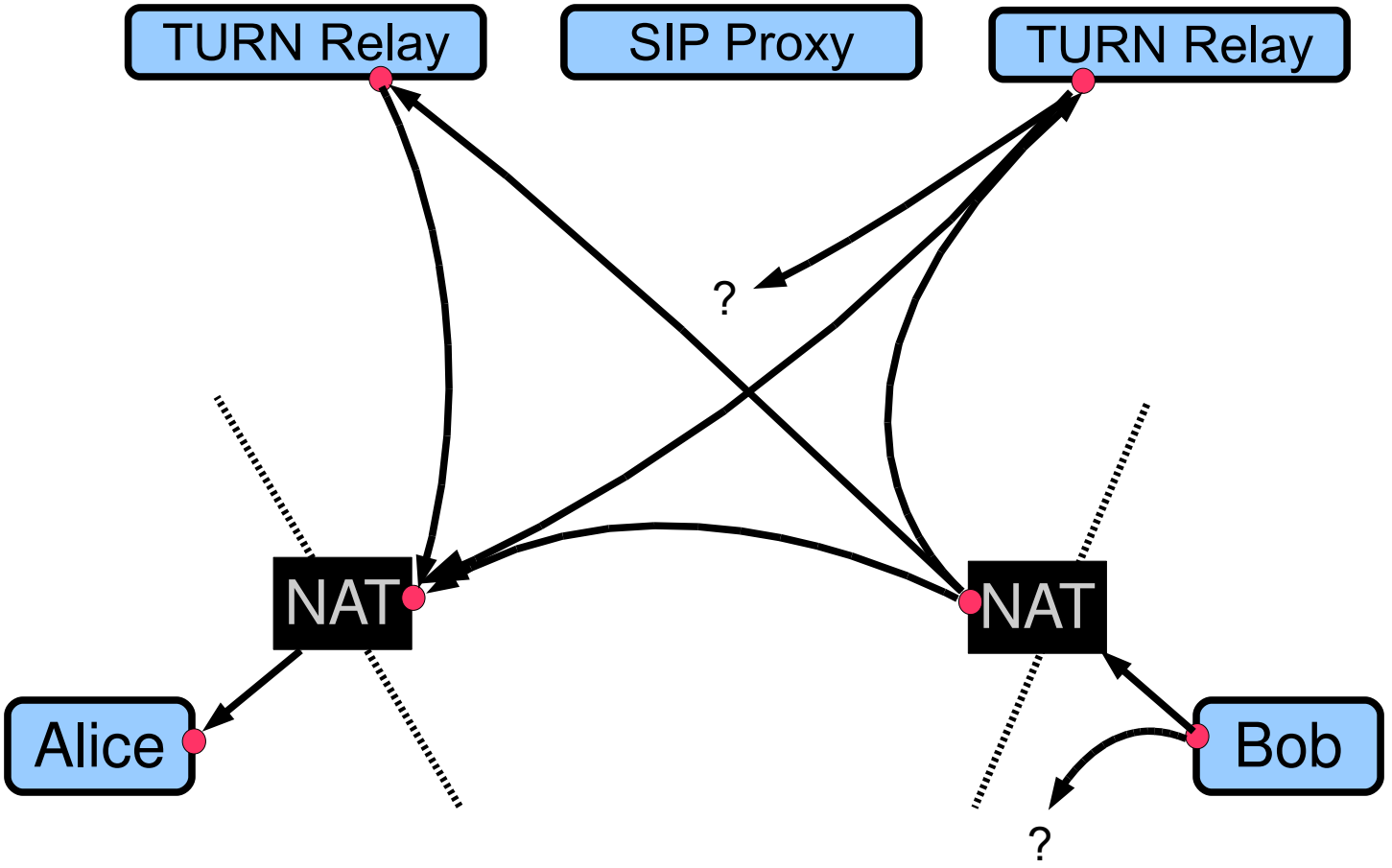
ICE, Connectivity Checks

- Alice's relay cand – Bob's server reflexive cand



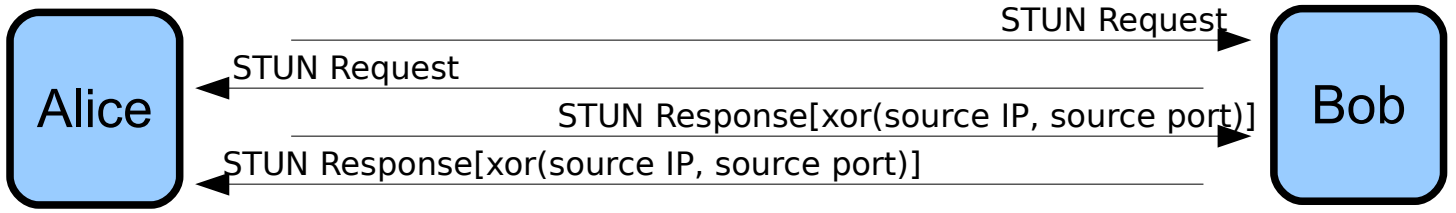
ICE, Connectivity Checks

- And Bob is doing the same...



ICE, Connectivity Checks

- Essentially forms a 4-way handshake



ICE, Coordination

- Signal completion (achieved directly between peers, not via signalling channel)
- Regular Nomination by controlling peer
  - Re-send a STUN check, with a flag set
- Aggressive nomination by controlling peer
  - Set flag in all STUN checks, such that the first working candidate is chosen

ICE, Communication

joy

Nokia

- This is essentially an on-going work
  - It's not *live* yet, and there are various technical and bureaucratic hurdles to cross before it *will* go live

Resources

- ICE: http://tools.ietf.org/html/draft-ietf-mmusic-ice
- STUN: http://tools.ietf.org/html/draft-ietf-behave-rfc3489bis
- TURN: http://tools.ietf.org/html/draft-ietf-behave-turn

Questions?
sds@dcs.gla.ac.uk