Self-Address Fixing Evolution BOF

https://www1.ietf.org/mailman/listinfo/safe

Chairs:

- Colin Perkins <csp@csperkins.org>
- Markus Isomaki <Markus.Isomaki@nokia.com>

Agenda

09:00	Introduction	(Chairs)
09:10	Problem statement and scope	(Wing)
09:25	Survey of existing work	(Barnes)
09:55	NAT/Firewall control with STUN	(Wing)
10:10	Discussion	
10:50	Future directions	(Chairs)

Intellectual Property

- When starting a presentation you MUST say if:
  – There is IPR associated with your draft
  – The restrictions listed in section 5 of RFC 3978 apply to your draft

- When asking questions or making comments:
  – You MUST disclose any IPR you know of relating to the technology under discussion

- Reference: RFC 3978/3979 and “Note Well” text

Aims of this BoF

- To discuss a newly-proposed technique for using STUN to discover, query and control firewalls and NATs, that can eliminate UDP keep-alive traffic.
- To review the problem space and existing work, and decide if there is a need for new work in the area, and if the IETF is an appropriate home for that work.
  - The intent is not to form a new working group at this time, but to gauge interest in work in this area, and consider an appropriate future home for that work.

Problem Statement and Scope

Dan Wing

I E T F®

Problem Statement

- UDP applications that do not control their NATs need frequent UDP keepalives
  - IPsec NAT traversal
  - STUN
  - SIP-Outbound
- Frequent UDP keepalives consume battery power on wireless devices (e.g., 802.11, W-CDMA, WiMax)

SAFE Scope

- Create a NAT control technique that:
  - Determines NAT and firewall keepalive interval
  - Adjusts NAT and firewall keepalive interval
  - Works with nested NATs and nested firewalls
  - Detects non-upgraded NATs, and reverts to pre-SAFE behavior
  - Uses source transport address for authorization

Survey of Protocols to Control NAT and Firewalls

Mary Barnes

Authors: Lars Eggert, Pasi Sarolahti, Remi Denis-Courmont, Hannes Tschofenig

draft-eggert-middlebox-control-survey-01.txt

IETF®

Summary of Protocols Analyzed

- SOCKS
- NSIS NATFW NSLP
- MIDCOM
- SIMCO
- UPnP
- Diameter Gq', Rx+, Gx+
- NAT-PMP
- STUN
- RSIP
- ALD
- NLS
- AFWC

General Categorization of Protocols

End-System-Initiated Protocols

- Two Party Approach
  - UPnP
  - SOCKS
  - NAT-PMP
- Multi-Party Approach
  - STUN
  - STUN controlled NAT
  - NSIS NATFW NSLP
  - NLS

General Categorization of Protocols

Third-Party-Initiated Approaches (with similar, general operational models):

- MIDCOM
- Diameter Gq', Rx+, Gx+
- SIMCO

Other more specialized approaches:

- RSIP
- AWFC
- ALD (v6 specific)

Protocol Summaries

UPnP (Universal Plug and Play):

- Protocol between clients and IPv4 gateways.
- Provides “Edge” interconnection device between a residential LAN and a WAN
- Limited to middleboxes in the local network, as middlebox discovery is based on broadcasting.
- References: UPnP Forum Internet Gateway Device (IGD)Standardized Device Control Protocol v 1.0.

SOCKS:

- Uses “sockets” to represent and keep track of individual connections
- Allows application layer protocols to securely and transparently traverse firewalls, by providing a “shim” layer between application and transport layers.
- Reference: RFC 1928

NAT-PMP (NAT Port Mapping Protocol):

- Lightweight protocol between clients and IPv4 gateways.
- If first hop GW supports NAT-PMP, client can learn external IPv4 address.
- Expects the NAT to be the default gateway, thus doesn’t work well in routed networks.
- Reference: draft-cheshire-nat-pmp

Protocol Summaries

STUN (Simple Traversal of UDP through NATs):

- Allows clients to discover the presence of NATs and determine public addresses, while requiring no special behavior from NATs, but NATs should abide by RFC 4787.
- Requires STUN server on public network
- With proposed enhancements, incremental deployment and nested NATs can be supported. Optimized behavior requires support in the middleboxes.
- References: RFC 3489, draft-ietf-behave-3489bis, draft-wing-behave-nat-control-stun-usage-04

NSIS NATFW NSLP

- NSIS uses a two layer architecture with a lower-layer transport protocol (NSIS Transport Layer Protocol (NTLP)).
- NAT/FW Network Signaling Layer protocol (an NSLP) is built on the NTLP.
- References: RFC 4080, draft-ietf-nsis-ntlp, draft-ietf-nsis-nslp-natfw

NLS (Network Layer Signaling):

- Lightweight firewall pin-holing application, designed to carry requests for firewall resources to firewalls along a path between two endpoints.
- Based on generic Network Layer Signaling Transport Layer
- References: draft-shore-nls-fw-00

Protocol Summaries

MIDCOM

- Allows the endpoint to control a middlebox using a control protocol. Requires the middlebox vendors to implement and support the protocol.
- SNMP selected as the control protocol, thus a MIB has been defined.
- References: RFC 3303, RFC 4097, draft-ietf-midcom-mib

SIMCO:

- NEC's "SIMPLE" Middlebox Communication protocol
- Complies with the MIDCOM Semantics (RFC 3989, draft-ietf-midcom-rfc3989bis)
- Reference: RFC 4540

Diameter Gq', Rx+, Gx+

- Generally complies with MIDCOM requirements (RFC 3304) and was originally based on DIAMETER proposal in MIDCOM protocol evaluation (RFC 4097).
- The protocol is connection-oriented at both the transport and application levels.
- References: RFC 4097, ITU

Protocol Summaries

RSIP (Realm Specific IP)

- With RSIP with tunneling, the private realm host application knows the public realm IP addresses and port numbers. This requires an RSIP server and a tunneling protocol be implemented in the middlebox and an RSIP client and the tunneling protocol be implemented in the private realm host.
- One of 5 protocols proposed as the MIDCOM Protocol.
- References: RFC 3103, RFC 4097

ALD (Application Listener Discovery):

- Specifically for IPv6 stateful firewalls.
- Uses ICMPv6 for signaling
- Auto-configured through a specific router advertisement.
- Reference: draft-woodyatt-ald-01

AFWC (Authorized IP Firewall Control Application):

- Provides an interface that allows network entities to request firewall and NAT services and resources. An instance of a protocol that provides authorizations and other security services, and inter-works with other such instances
- AFWC uses its authorization facilities to provide network administrators more control over network border admission. Relies on crypto layer for authorization.
- References: draft-shore-afwc-00

Protocol Comparison: Deployment

Protocol	Implemented (Yes/No)	Widely Deployed (Yes/No)	Supports Incremental deployment (Yes/No)
UPnP	Yes	Yes	No
SOCKS	Yes	Yes	Yes
NAT-PMP	Yes	No	No
STUN	Yes	Yes	Yes
STUN (Control)	Yes	No	Yes
NSIS NATFW NSLP	Yes	No	No
NLS	Yes	No	No
MIDCOM	No	No	No
SIMCO	Yes	No	No
Diameter Gq', Rx+, Gx+	?	No	No
RSIP	Yes	No	No
ALD	Yes	No	No
AFWC	Yes	No	Yes

Protocol Comparison: Middle-box interactions

Protocol	Keepalive required (Yes/No)	Interacts directly with MB? (Yes/No)	Security between MB and endpoint?
UPnP	No	Yes	Yes (but unused)
SOCKS	No	No	No
NAT-PMP	No	No	No
STUN	Yes	No	No
STUN (Control)	No	Yes	No
NSIS NATFW NSLP	No	Yes	Yes
NLS	No	Yes	Yes
MIDCOM	No	Yes	Yes
SIMCO	No	Yes	Yes
Diameter Gq', Rx+, Gx+	No	Yes	Yes
RSIP	No	Yes	Yes
ALD	No	Yes	No
AFWC	No	Yes	Yes (through crypto layer)

Protocol Comparison: Topology/environments

Protocol	Topology Aware	Supports Nested NATs (Yes/No)	Supports diverse environments/endpoints
UPnP	No	No	No
SOCKS	No	Yes	No
NAT-PMP	No	No	No
STUN	Yes	Yes	Yes
STUN (Control)	Yes	Yes	Yes
NSIS NATFW NSLP	Yes	Yes	Yes
NLS	Yes	Yes	Yes
MIDCOM	Yes	Yes	Yes
SIMCO	Yes	Yes	Yes
Diameter Gq', Rx+, Gx+	Yes	Yes	Yes
RSIP	Yes	Yes	No
ALD	No	No	No
AFWC	Yes	Yes	Yes

Summary (1)

- Many NAT/FW traversal mechanisms and protocols have been implemented, however only a few are widely deployed: SOCKS, UPnP, STUN
- Only a few of the solutions effectively support incremental deployment: STUN (per draft-wing-behave-nat-control-stun-usage-04), SOCKS, and AFWC
- Several of the protocols require Keep-alive mechanisms, which can result in excessive chattiness that has performance impacts in certain environments: STUN (without NAT control)

Summary (2)

- Majority require direct interactions with middle-box
  - This can be a barrier to widespread deployment of these protocols due to lack of middle-box vendor support.
  - In addition, several of the protocols (MIDCOM, SIMCO, DIAMETER) don't provide a way to find on-path protocol-controlled NATs/FWs.
- About half the protocols require security between the endpoint and the middle-box. In one sense, this security relationship provides a more robust solution, but it can also be a barrier to deployment.
- Over half current protocols are aware of topology
- The majority of the protocols support Nested NATs.
- Over half the protocols can be used in diverse environments, in terms of supporting a variety of types of network deployments, endpoints and applications.
  - For the other half, enterprise deployment is often an issue: UPnP and NAT-PMP.

NAT Control STUN Usage
"STUN Control"

Dan Wing

Jonathan Rosenberg

Hannes Tschofenig

draft-wing-behave-nat-control-stun-usage-05.txt

I E T F®

Outline

- Motivation and goals
- Procedures:
  - with firewalls
  - with one NAT
  - with nested NATs
  - with nested NATs with overlapping IP addresses
- Summary of benefits
  - Why STUN Control will succeed

Motivation

- Reduce network traffic
  - Keepalive chatter to STUN server
    - Battery-operated wireless devices
  - Binding discovery chatter to STUN server
- Retain STUN/ICE's ability to work on any network
  - Enterprise networks
  - ISPs that NAT their subscribers
  - Home networks

STUN Control: Initial Goals

- UDP only
- Extend the NAT’s binding lifetime
  - Reduces keepalive chatter

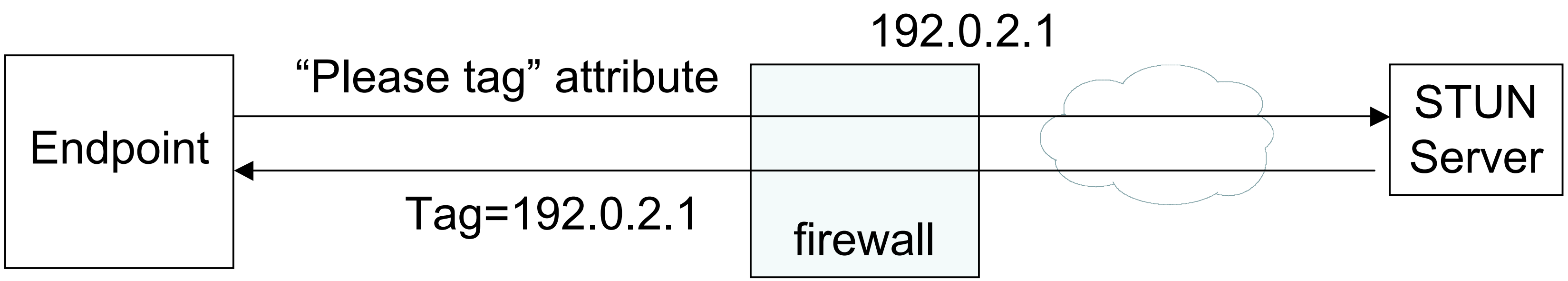
Implementation Available

- http://www.christian-dickmann.de/stun.php

Procedure with Firewall

Tagging Procedure with Firewalls

- Endpoint sends STUN request and includes ‘please tag’ attribute
- Firewall sees STUN request with that attribute, remembers it
- Firewall tags the response (with same STUN transaction-id and inverted 5-tuple) with firewall’s IP address



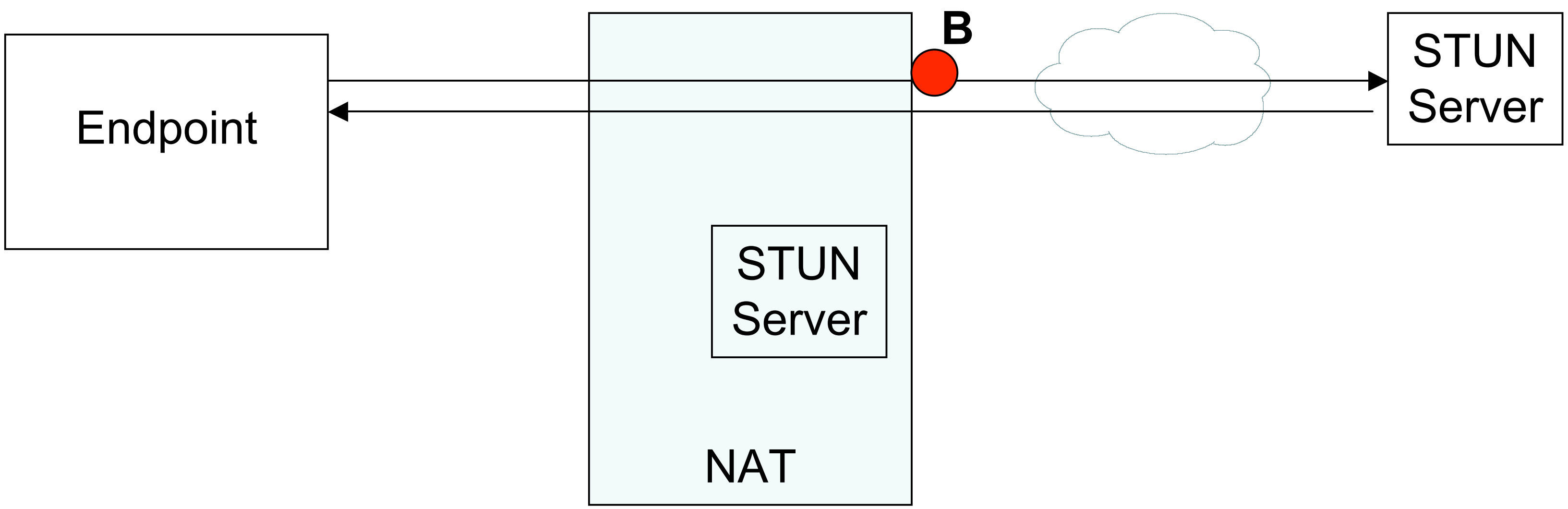
Procedure with one NAT

One NAT Procedure Overview

1. Learn IP address of outer-most NAT
2. Using that NAT’s embedded STUN server, query and extend UDP binding lifetime

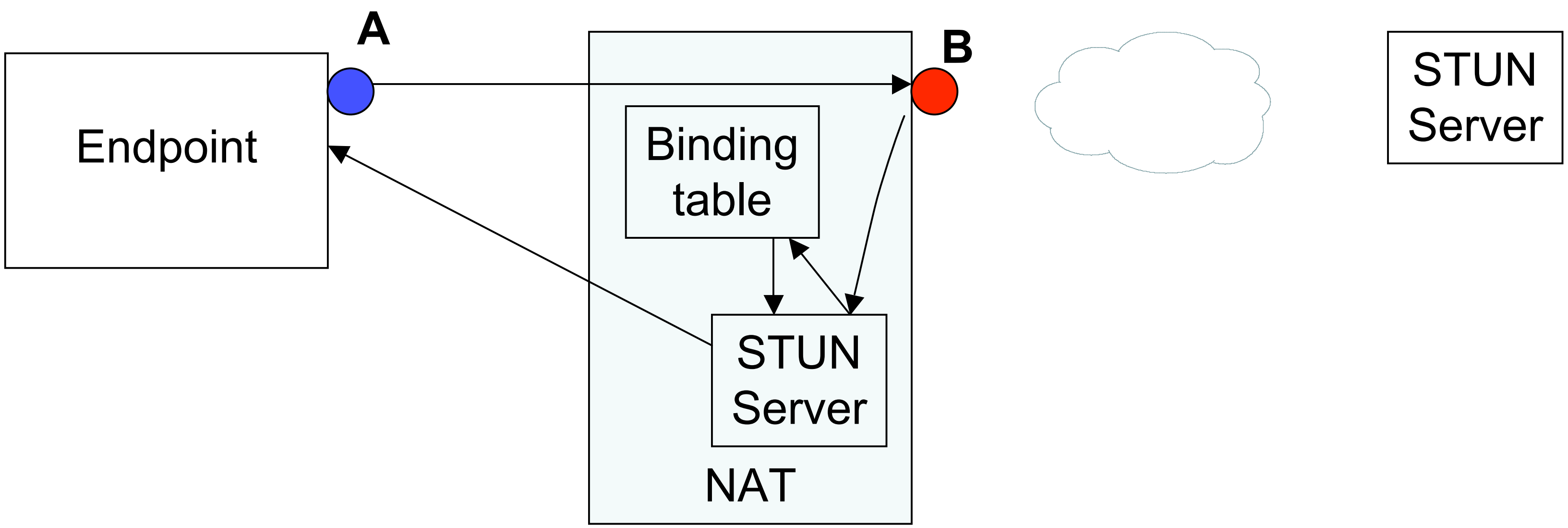
1. Learn IP address of outer-most NAT

- This is classic STUN (RFC3489)



2. Communicate to NAT's embedded STUN Server

- Adjust binding lifetime
- Learn UDP port "B"
- Learn IP address and UDP port "A" (ourself)



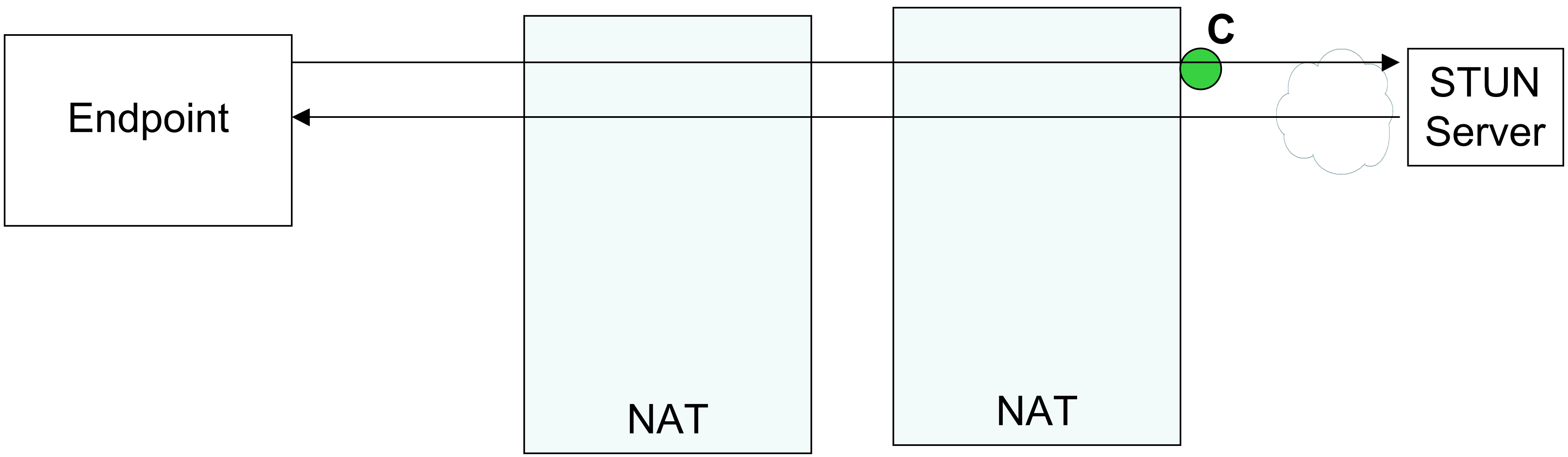
Procedure with nested NATs

Nested NATs Procedure Overview

1. Learn IP address of outer-most NAT
2. Using that NAT’s embedded STUN server, query and extend UDP binding lifetime, and learn next-inner NAT
3. Using next-inner NAT’s embedded STUN server, query and extend its UDP binding lifetime, and learn next-inner NAT
4. repeat

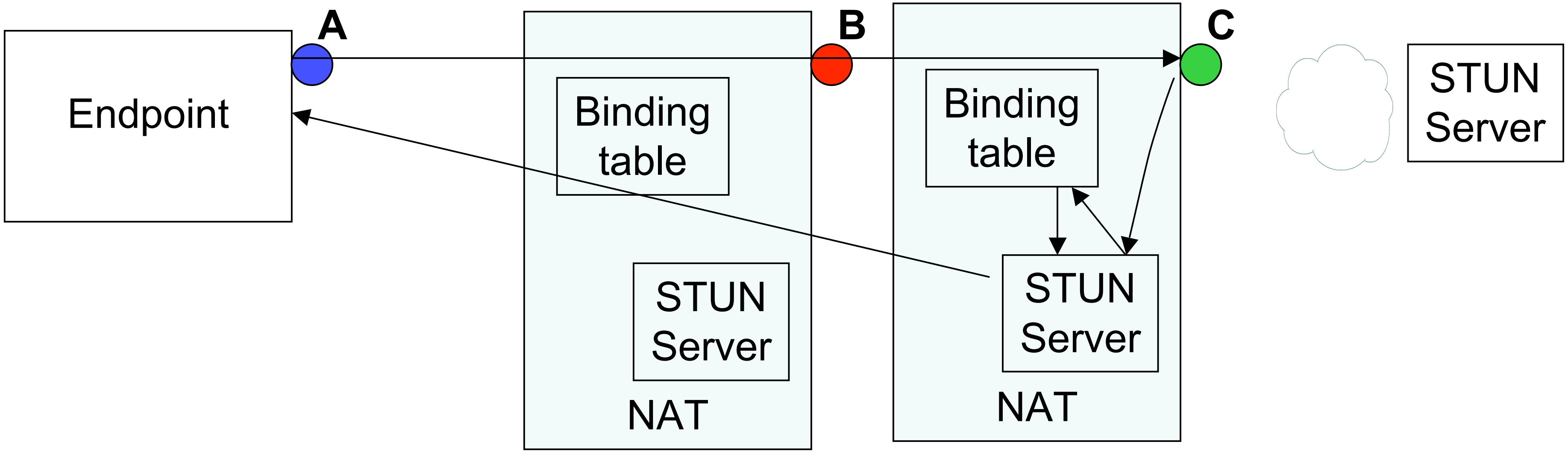
1. Learn IP address of outer-most NAT

- This is classic STUN (RFC3489)



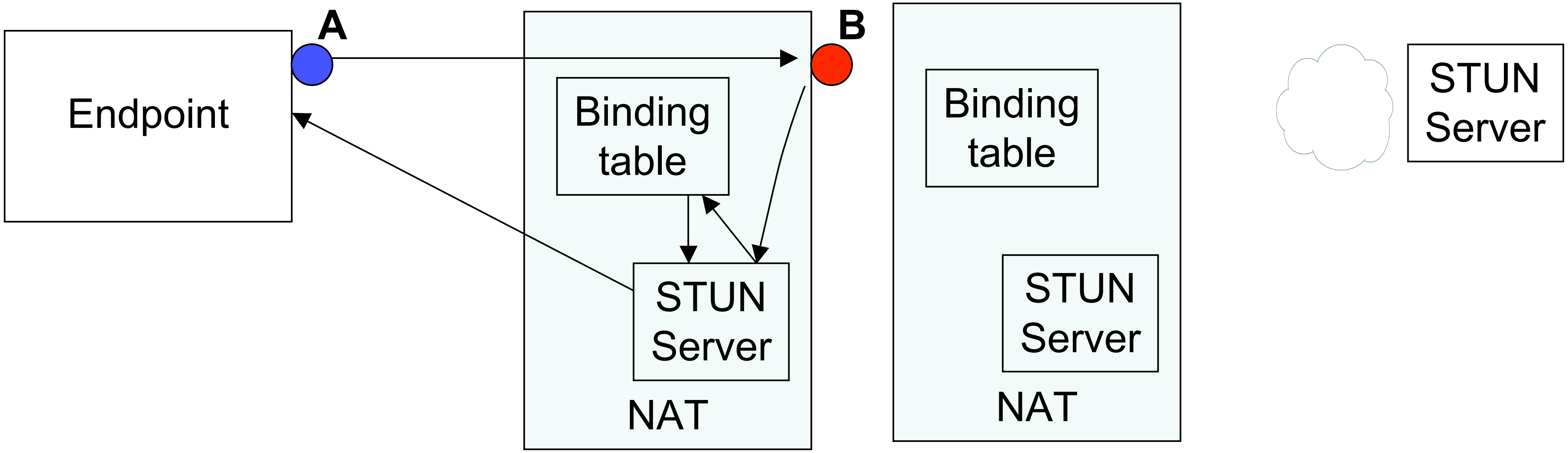
2. Communicate to outer-most NAT's embedded STUN Server

- Adjust binding lifetime of NAT "C"
- Learn UDP port "C"
- Learn IP address and UDP port "B"



3. Communicate to next-closer NAT's embedded STUN Server

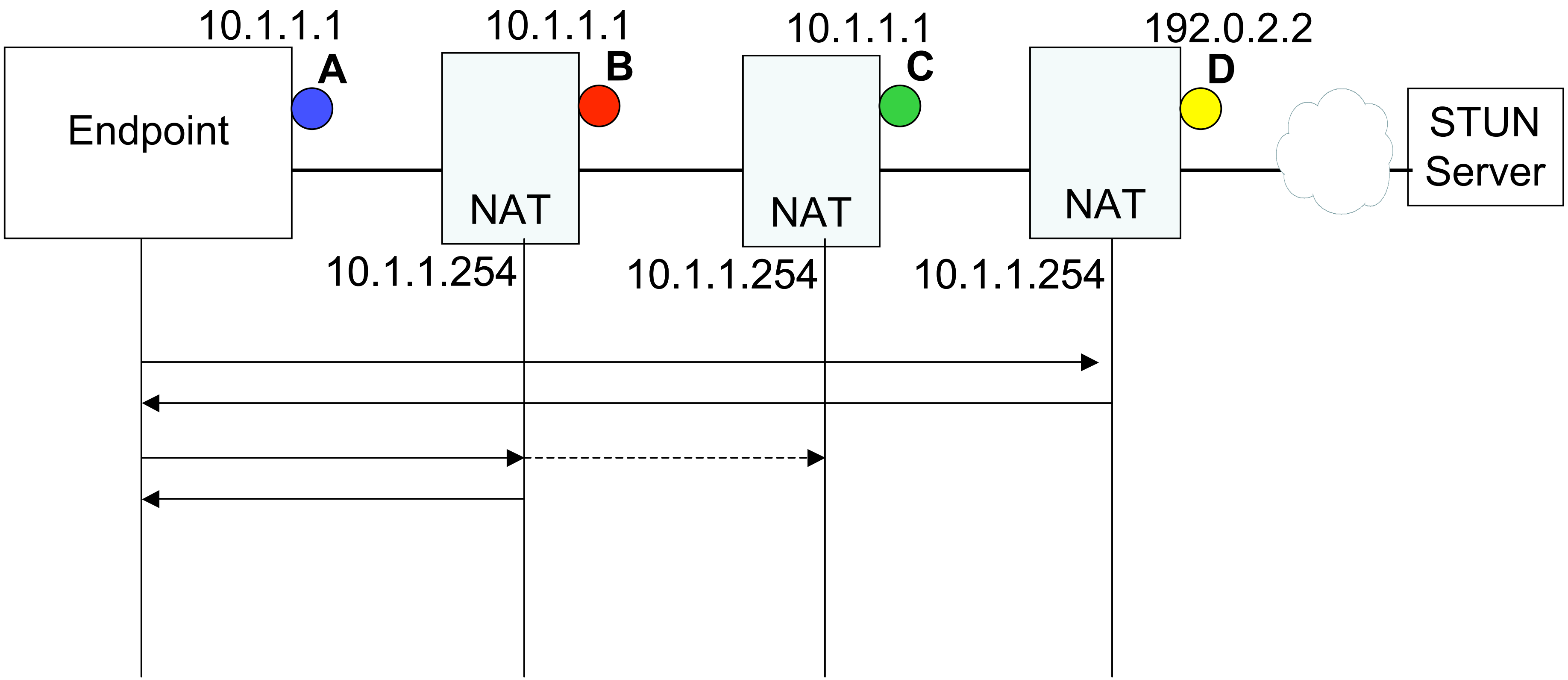
- Adjust binding lifetime of NAT "B"
- Learn IP address and UDP port "A" (ourself)



Procedure with nested NATs with overlapping IP addresses

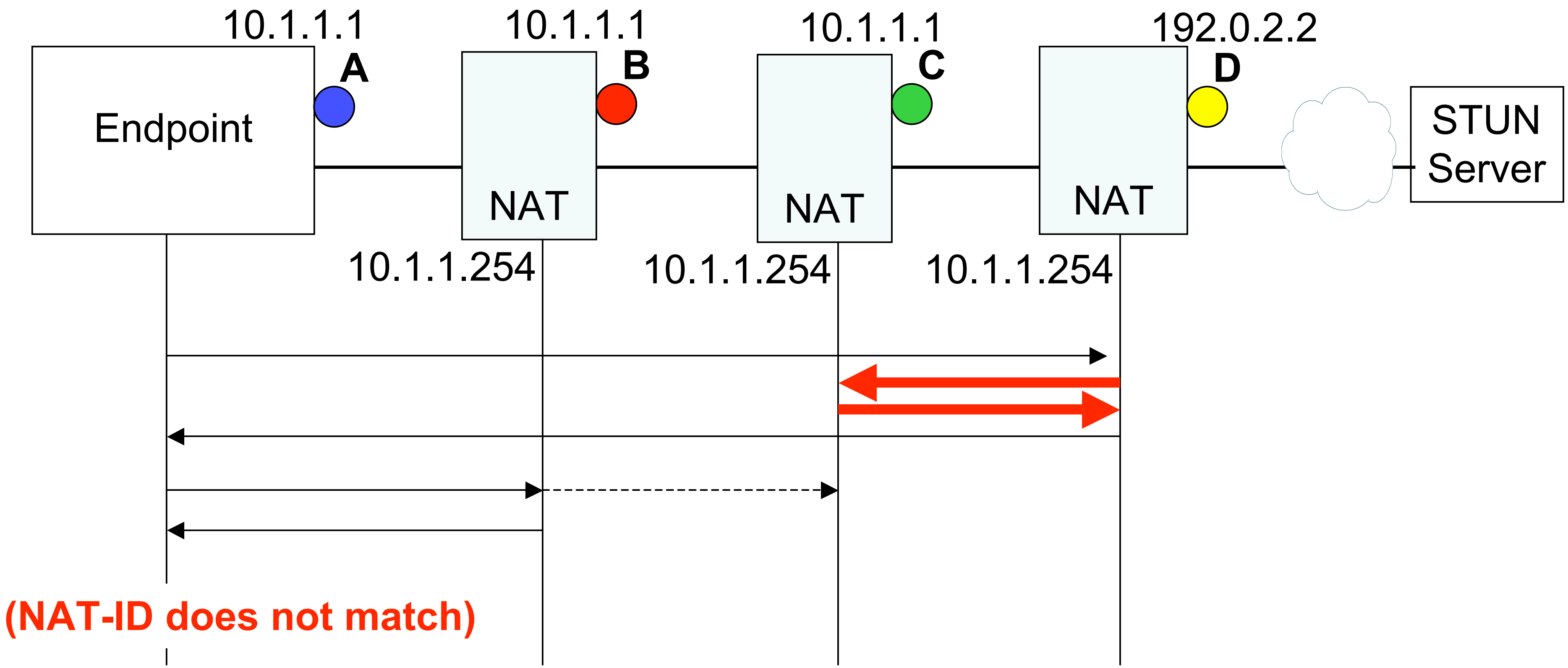
NATs with Overlapping IP addresses

- As described currently, this is not well detected



Proposed solution: NAT-ID

- Outer NAT query next-innermost NAT for its NAT-ID (shown in red)



Summary of Benefits

STUN Control: Summary of Benefits

- Preserves STUN’s ability to work with nested NATs
- Extend NAT binding duration of all NATs along path
  - Reduces keep-alive chatter
- Automatically learns NAT path topology
  - Allows ICE to better optimize media path

STUN Control: Middle box interactions

Protocol	Keepalive required (Yes/No)	Interacts directly with MB? (Yes/No)	Security between MB and endpoint?
UPnP	Yes	Yes	No
SOCKS	No	No	No
NAT-PMP	No	No	No
STUN	Yes	No	No
STUN Control	**No**	**Yes**	**No**
NSIS NATFW NSLP	No	Yes	Yes
NLS	No	Yes	Yes
MIDCOM	No	Yes	Yes
SIMCO	No	Yes	Yes
Diameter Gq', Rx+, Gx+	No	Yes	Yes
RSIP	No	Yes	Yes
ALD	No	Yes	No
AFWC	No	Yes	Yes (through crypto layer)

STUN Control: Topology/environments

Protocol	Topology Aware (Yes/No)	Supports Nested NATs (Yes/No)	Supports Diverse environments/endpoints
UPnP	No	No	No
SOCKS	No	Yes	No
NAT-PMP	No	No	No
STUN	Yes	Yes	Yes
STUN Control	**Yes**	**Yes**	**Yes**
NSIS NATFW NSLP	Yes	Yes	Yes
NLS	Yes	Yes	Yes
MIDCOM	Yes	Yes	Yes
SIMCO	Yes	Yes	Yes
Diameter Gq', Rx+, Gx+	Yes	Yes	Yes
RSIP	Yes	Yes	No
ALD	No	No	No
AFWC	Yes	Yes	Yes

Why STUN Control Will Succeed

- Works with nested NATs
- Works on routed networks
- Incrementally deployable
  - If STUN Control is unavailable, the host falls back to normal keepalive behavior
- No additional security policy/configuration in the NAT

Questions and Discussion

...on the technology

Future Directions

Colin Perkins
Markus Isomaki

Future Directions

- Aim of this BoF is not to form a new working group
- Rather, decide if there is a need for new work in the area, and if the IETF is an appropriate home for that work
  - If "yes" to both, will work with IESG to decide if the work fits an existing group, or if a working group forming BOF is needed at IETF 71

Future Directions

- Will ask the following three questions:
  - Are some functional requirements or deployment considerations left unsatisfied by existing protocols?
  - Is there agreement that the IETF should consider developing a new NAT control mechanism to address these requirements?
  - Is the NAT Control STUN usage a reasonable approach to NAT control, addressing the requirements?

Requirements

- Will ask the following three questions:
  - Are some functional requirements or deployment considerations left unsatisfied by existing protocols?
  - Is there agreement that the IETF should consider developing a new NAT control mechanism to address these requirements?
  - Is the NAT Control STUN usage a reasonable approach to NAT control, addressing the requirements?

NAT Control

- Will ask the following three questions:
  - Are some functional requirements or deployment considerations left unsatisfied by existing protocols?
  - Is there agreement that the IETF should consider developing a new NAT control mechanism to address these requirements?
  - Is the NAT Control STUN usage a reasonable approach to NAT control, addressing the requirements?

NAT Control STUN Usage

- Will ask the following three questions:
  - Are some functional requirements or deployment considerations left unsatisfied by existing protocols?
  - Is there agreement that the IETF should consider developing a new NAT control mechanism to address these requirements?
  - Is the NAT Control STUN usage a reasonable approach to NAT control, addressing the requirements?