Legacy Network Address Translator Traversal Using the Host Identity Protocol

(Perinteisten osoitteenmuuntajien läpäisy käyttäen koneen identiteetti protokollaa)

5.6.2007

Author: Lauri Silvennoinen
Supervisor: Professor Jörg Ott
Instructor: M.Sc. Miika Komu

InfraHIP project

- HIIT (University of Helsinki) and TML (HUT) joint effort
  - Running since November 18th 2004
  - Develops C-based HIP for Linux
  - Financial Support from
    - TEKES
    - Nokia
    - Ericsson
    - Elisa
    - Finnish Defence Forces

Agenda

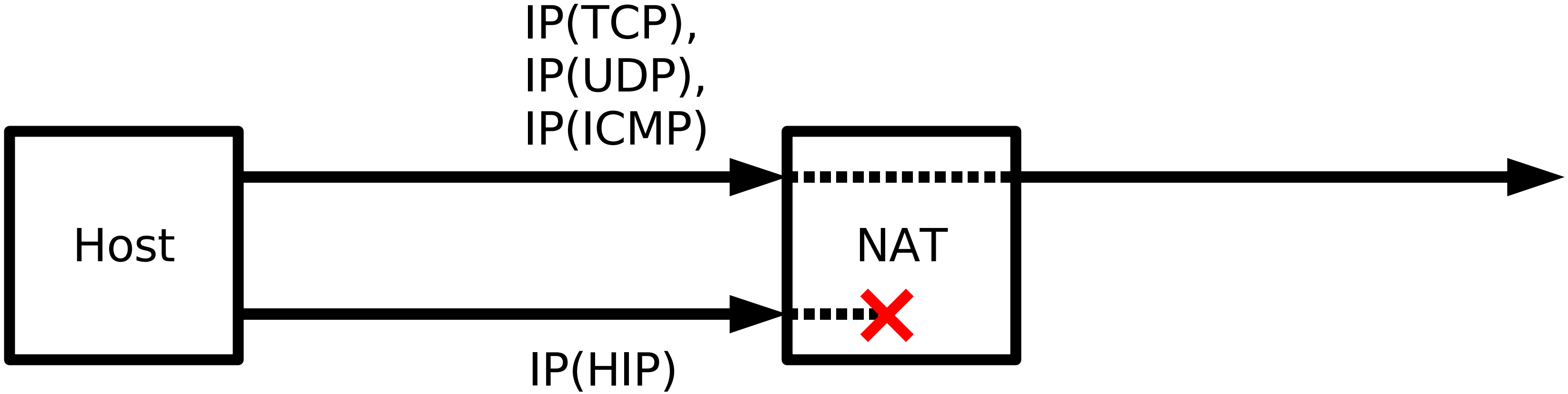
- **Host Identity Protocol (HIP)**
- Network Address Translation (NAT)
- Legacy NAT traversal using HIP
- Conclusions

HIP Background

- Host Identity Protocol (HIP) is being specified in IETF and IRTF work groups
  - Initial ideas, late 1999
  - Work groups since November 2003
- HIP is being implemented in various projects
  - **HIP for Linux (InfraHIP project)**
  - HIP for BSD (HIP for inter.net project)
  - OpenHIP Project

Research Problem

- Currently deployed NAT devices can translate TCP, UDP and ICMP packets
- NATs block unknown protocols, thus HIP won't work

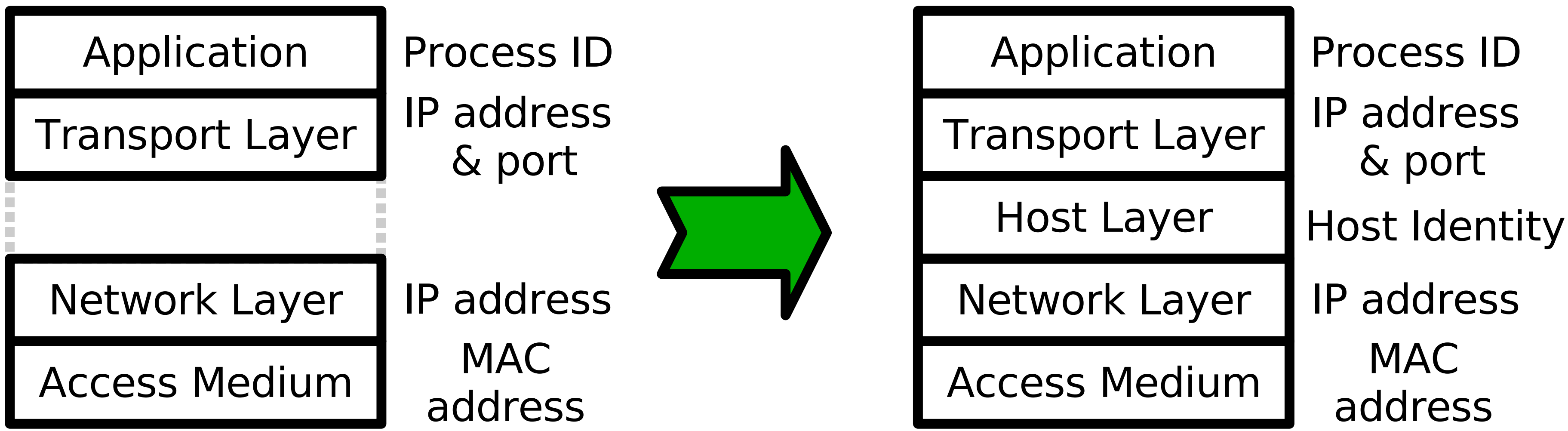


Motivation behind HIP

- IP addresses are used both to
  - identify a host
  - locate a host
- This duality makes many things hard
  - for example dynamic readdressing
- In HIP architecture
  - Host Identifier (HI) identifies a host
  - IP address locates a host

HIP Fundamentals (1/3)

- HIP is a concrete proposal for adding a new name space to the TCP/IP stack



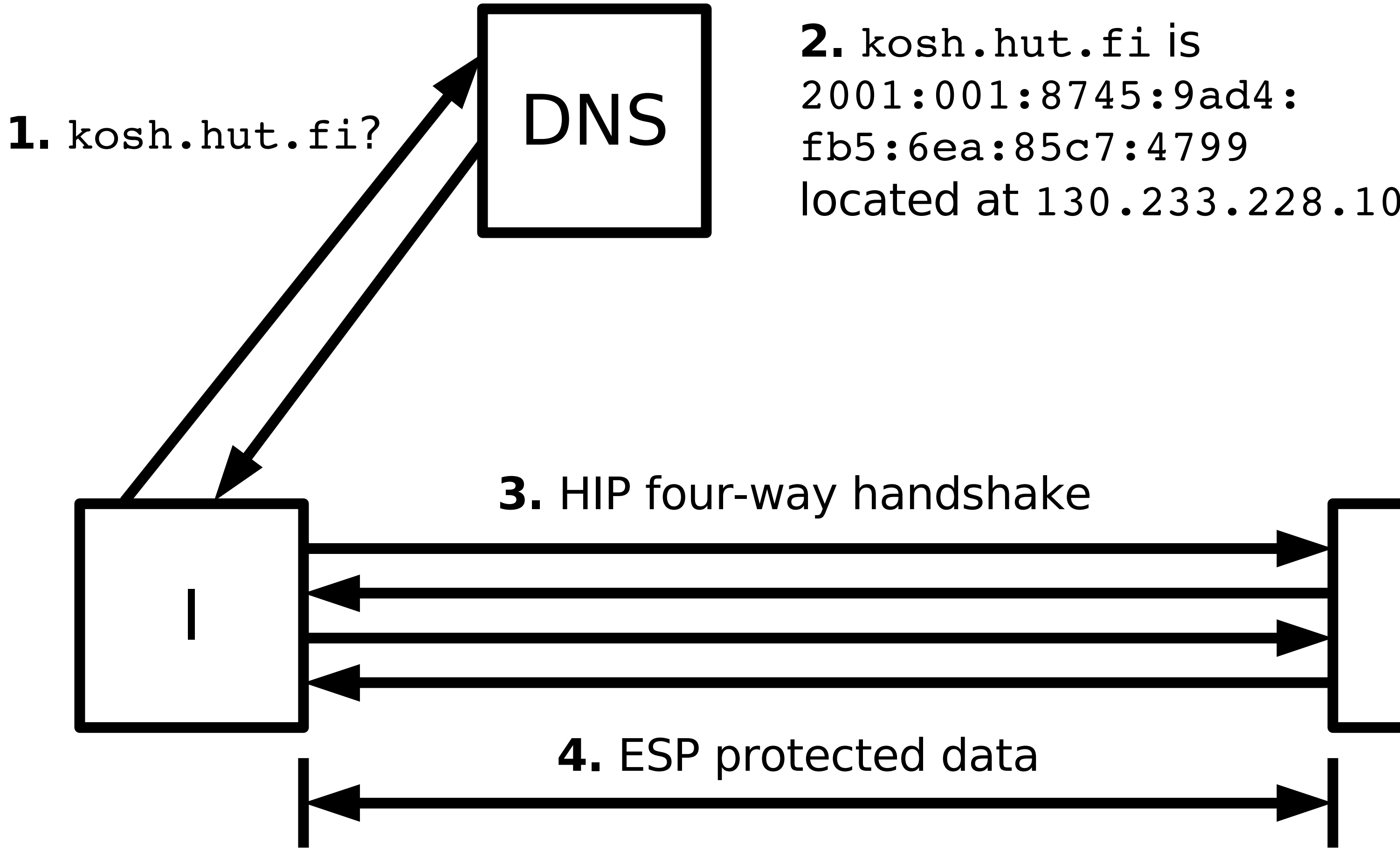
HIP Fundamentals (2/3)

- Current Internet has two global name spaces
  - DNS names, `kosh.hut.fi`
  - IP addresses, `130.233.228.10`
- New Host Identity name space
  - Host Indentifiers, Host Identity Tags `2001:001:8745:9ad4:fb5:6ea:85c7:4799`
    - Statistically globally unique

HIP Fundamentals (3/3)

- Each host has at least one Host Identifier (HI)
- HIP communication is based on two different protocols
  - HIP protocol itself for establishment and management of communications
  - IPSec ESP to exchange user and application data in a secured way

Example HIP session



Host Identity

- A public key of an asymmetric key pair
- Public keys tend to be rather long
  - A Host Identity Tag (HIT) is a 128-bit hash of the HI
    - HITs are exchanged during the four-way handshake
- The length of HIT equals the length of IPv6 address
  - HITs can be used in IPv6-sized fields in APIs

Internet Sockets in HIP

- Sockets bound to HIs, not to IP addresses

- Current
  - `protocol`
  - **`source IP address`**
  - `source port`
  - **`dest. IP address`**
  - `destination port`

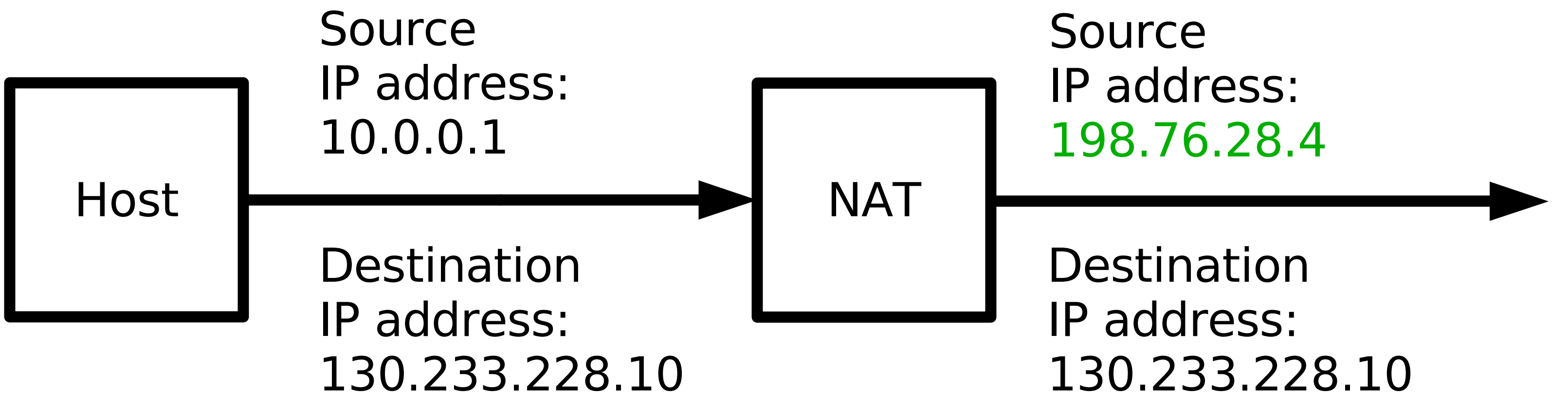
- HIP
  - `protocol`
  - **`source HI`**
  - `source port`
  - **`destination HI`**
  - `destination port`

Agenda

- Host Identity Protocol (HIP)
- **Network Address Translation (NAT)**
- Legacy NAT traversal using HIP
- Conclusions

Network Address Translation

- A function, ..., that dynamically assigns a globally unique address to a host that doesn't have one, without that host's knowledge. [RFC 3234]



Motivation Behind NAT

- Public IPv4 address shortage
  - addresses are becoming more and more difficult to reserve
    - public IPv4 addresses are expensive
- Whenever external network topology changes, address assignment for local domain must reflect these changes
  - NATs can hide these kind of changes from local domain users

NAT Fundamentals (1/2)

- Only outbound traffic is permitted
  - static address maps are exceptions
- Outbound traffic creates a translation state in NAT
  - traffic sent in response from public realm uses this translation state and is permitted
- Private realm IP addresses are local to that domain

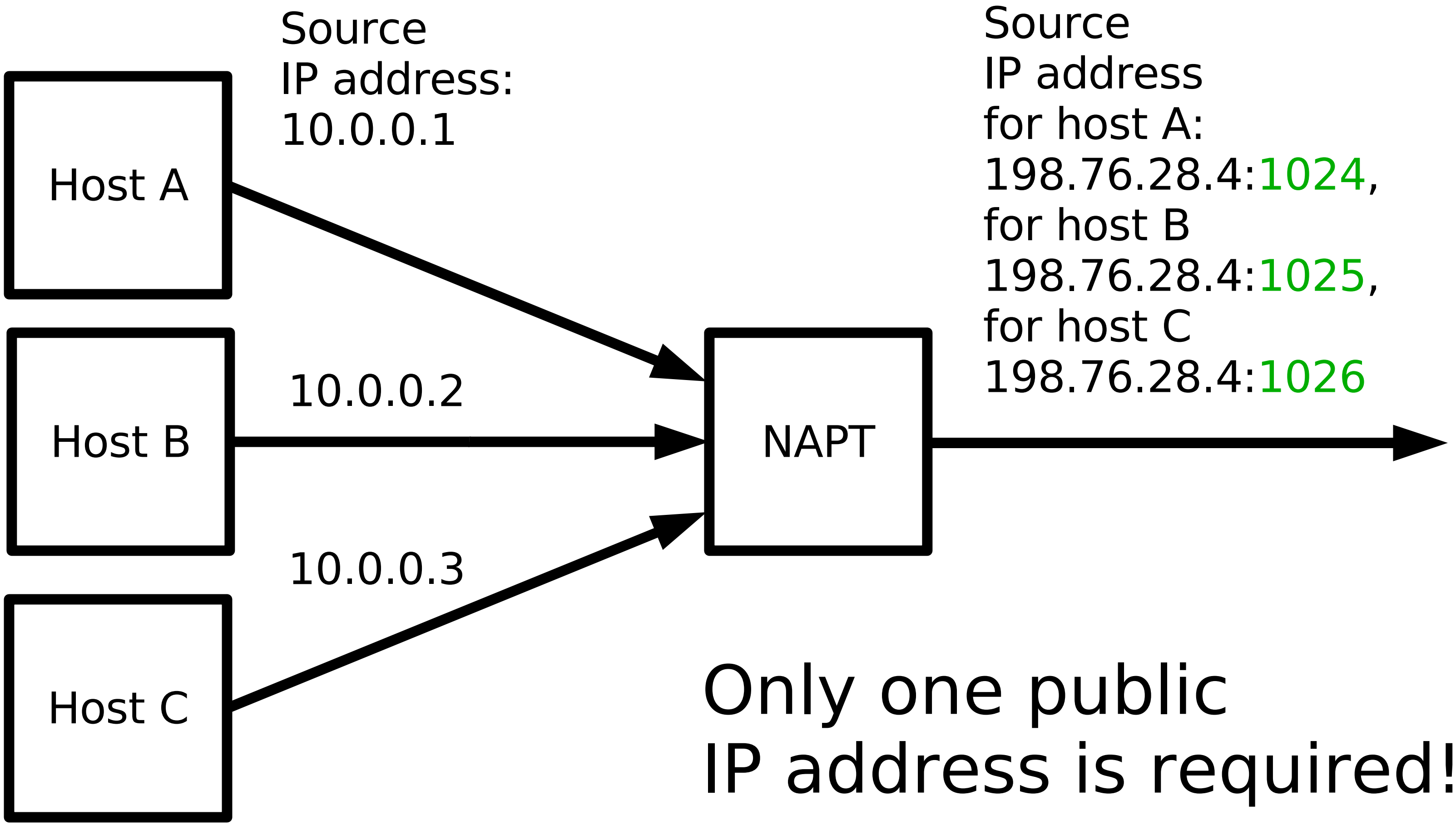
NAT Fundamentals (2/2)

- Sessions other than UDP, TCP and ICMP query type (ping) are not permitted
- Address translation is application independent
  - Application Level Gateways (ALGs) are needed to perform payload monitoring
  - NATs cause trouble to applications that carry IP addresses in payload

NAT types

- Basic NAT maps private realm IP addresses to public realm IP addresses
  - Translates only IP addresses
- Network Address and Port Translator (NAPT)
  - Translation includes IP address and transport layer port number
  - The most commonly deployed NAT type
- NAPT and Basic NAT = Traditional NAT

NAPT



Only one public
IP address is required!

NAT Classification

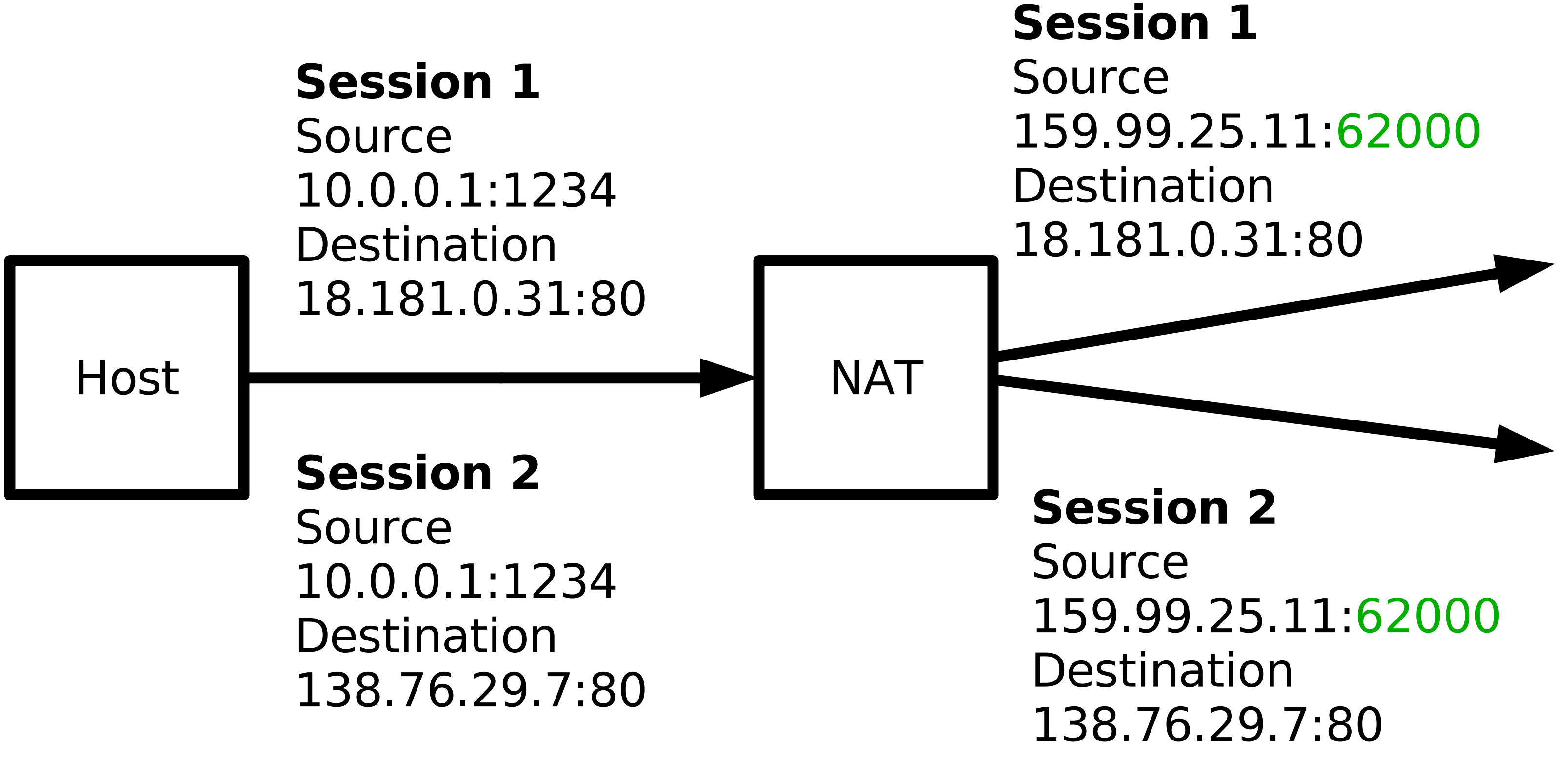
- Based on the concept of an Endpoint
  - Endpoint is the combination of an IP address and a port number
- Endpoint mapping refers to outgoing traffic
- Endpoint filtering refers to incoming traffic
  - 94.2% employ Endpoint Dependent Filtering

Key Question in NAT Traversal

- Does the NAT assign the same Endpoint to two simultaneous transport layer sessions originating from the same Endpoint and destined to different targets?
  - From the same local computer
  - Using the same protocol
  - While the translation state is valid in NAT
  - Targeted to two different locations

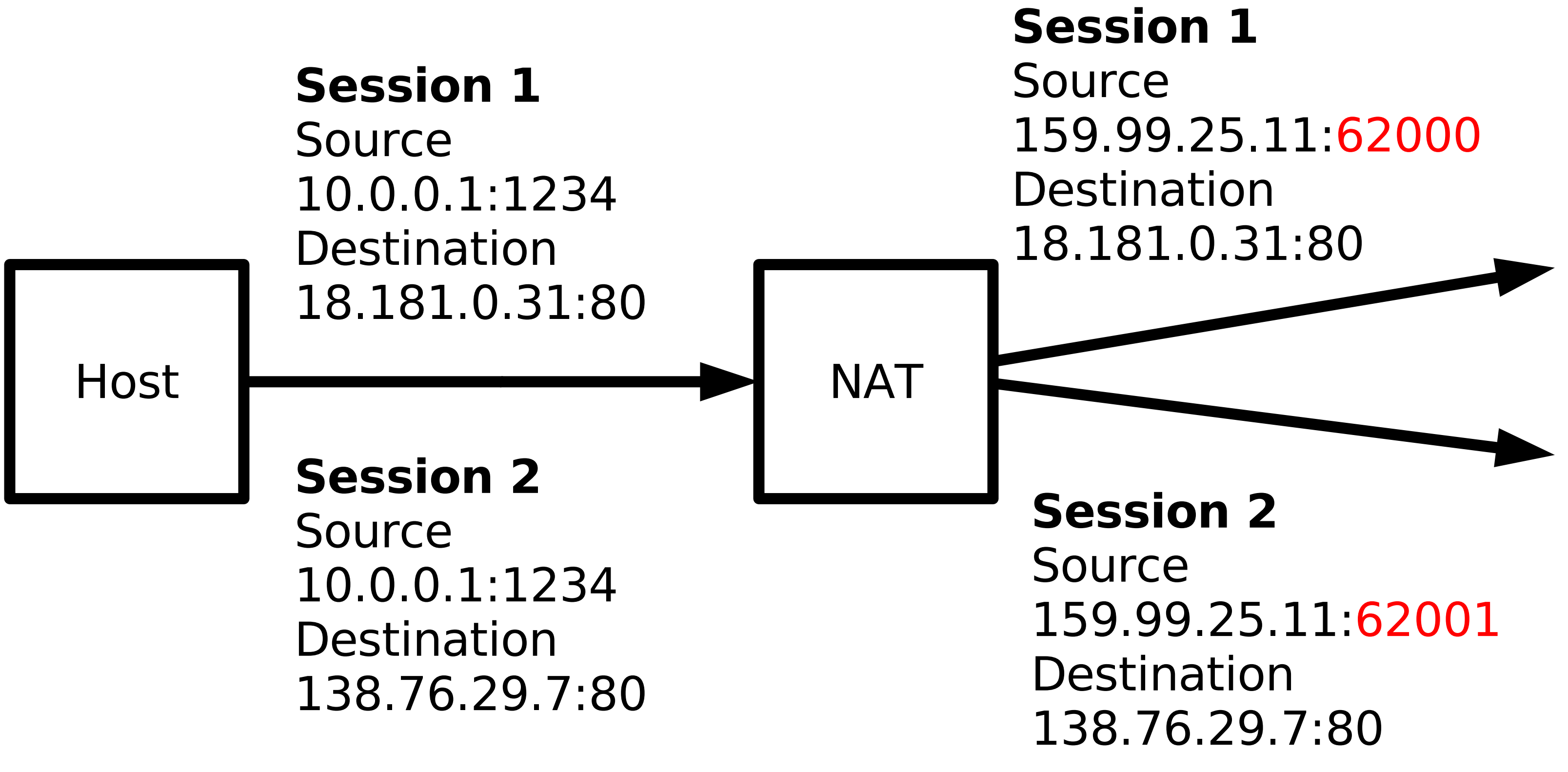
”Good” NAT

- Address Independent Mapping



”Bad” NAT

- Address Dependent Mapping



NATs deployed

- Of all NATs deployed approximately 70% are ”good” NATs
- The remaining 30% are ”bad” NATs
- Some measurents show, that 74% of all computers are behind a NAT
  - This is only a rough estimation

Agenda

- Host Identity Protocol (HIP)
- Network Address Translation (NAT)
- **Legacy NAT traversal using HIP**
- Conclusions

UDP Encapsulation

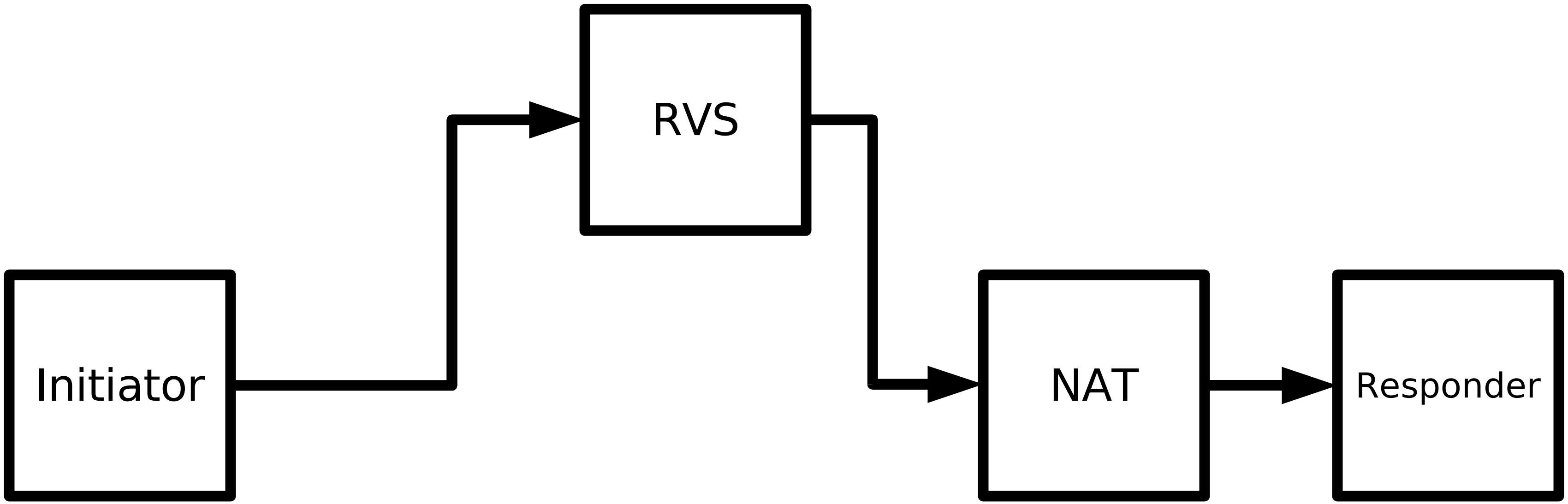
- Problem: only TCP, UDP and ICMP traffic permitted
- Solution: encapsulate all traffic in UDP
  - Instead of IP(HIP) use IP(UDP(HIP))
  - Instead of IP(ESP) use IP(UDP(ESP))

Responder behind NAT

- Problem: only outbound traffic can traverse NATs. How can a host behind a NAT be reached?

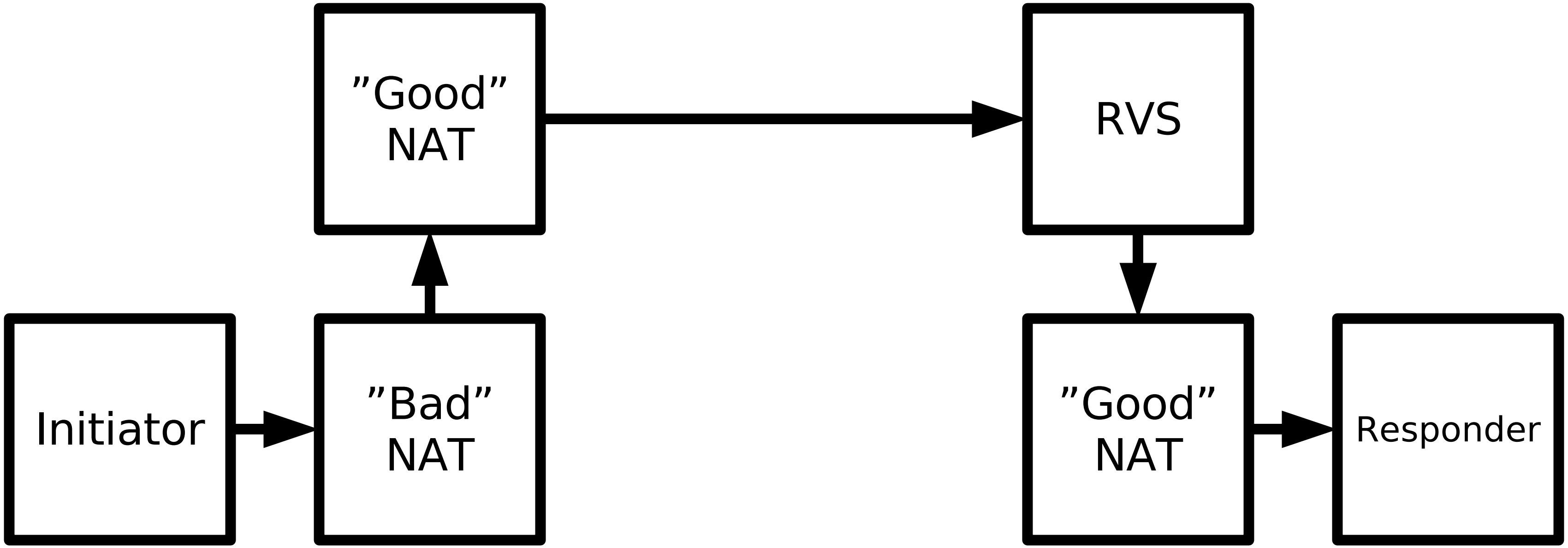
Rendezvous Server (RVS)

- Solution: use a Rendezvous Server where the Responder can register its whereabouts



Example Scenario

- Cascaded NATs
- A bad NAT enroute



Evaluation of the Solution

- UDP encapsulation and RVS together solve the main problems NATs cause
- However, further work still remains
  - A single ”bad” NAT enroute ruins our design
  - NAT detection
  - Cascaded NATs

Agenda

- Host Identity Protocol (HIP)
- Network Address Translation (NAT)
- Legacy NAT traversal using HIP
- **Conclusions**

Conclusions (1/2)

- Host Identity Protocol
  - Solves the dual role of IP addresses
  - Looks nice on paper but is still a work in progess
  - Widespread deployment requires
    - Endhosts that support HIP
    - DNSSEC for Host Identifiers
    - Deployment of RVSes
    - Deployment of NAT detection servers (STUN)

Conclusions (2/2)

- Network Address Translators
  - Cause a lot of headache to protocol designers
    - Protocols that carry IP addresses in payload are especially vulnerable
  - Are not standardized in anyway, thus many variations exist
    - RFC 4787 (January 2007) gives general guidelines on how a NAT should be designed
    - Buy RFC 4787 compliant NATs

Questions?

- Thank You!