NSIS Working Group   M. Stiemerling
Internet-Draft   NEC
Expires: November 17, 2005   H. Tschofenig
                                                                 Siemens
                                                                 C. Aoun
                                                                  Nortel
                                                            May 16, 2005

NAT/Firewall NSIS Signaling Layer Protocol (NSLP)
draft-ietf-nsis-nslp-natfw-06

Status of this Memo

Copyright Notice

Abstract

This memo defines the NSIS Signaling Layer Protocol (NSLP) for Network Address Translators and Firewalls. This NSLP allows hosts to signal along a data path for Network Address Translators and Firewalls to be configured according to the data flow needs. The network scenarios, problems and solutions for path-coupled Network

Address Translator and Firewall signaling are described. The overall architecture is given by the framework and requirements defined by the Next Steps in Signaling (NSIS) working group.

Table of Contents

1. Introduction 5
1.1 Terminology and Abbreviations 7
1.2 Middleboxes 9
1.3 Non-Goals 10
1.4 General Scenario for NATFW Traversal 11

2. Network Deployment Scenarios using NATFW NSLP 13
2.1 Firewall Traversal 13
2.2 NAT with two private Networks 14
2.3 NAT with Private Network on Sender Side 15
2.4 NAT with Private Network on Receiver Side Scenario 15
2.5 Both End Hosts behind twice-NATs 16
2.6 Both End Hosts Behind Same NAT 17
2.7 IPv4/v6 NAT with two Private Networks 18
2.8 Multihomed Network with NAT 19
2.9 Multihomed Network with Firewall 20

3. Protocol Description 21
3.1 Policy Rules 21
3.2 Basic protocol overview 21
3.3 Protocol Operations 25
3.3.1 Creating Sessions 25
3.3.2 Reserving External Addresses 28
3.3.3 NATFW Session refresh 32
3.3.4 Deleting Sessions 34
3.3.5 Reporting Asynchronous Events 35
3.3.6 Query and diagnosis capabilities within the NATFW NSLP protocol 35
3.3.7 Proxy Mode for Data Receiver behind NAT 39
3.3.8 Proxy Mode for Data Sender behind Middleboxes 43
3.3.9 Proxy Mode for Data Receiver behind Firewall 44
3.4 Calculation of Session Lifetime 46
3.5 Message Sequencing 48
3.6 De-Multiplexing at NATs 49
3.7 Selecting Opportunistic Addresses for REA 49

4. NATFW NSLP Message Components 51
4.1 NSLP Header 51
4.2 NSLP message types 51
4.3 NSLP Objects 52
4.3.1 Session Lifetime Object 53
4.3.2 External Address Object 53

4.3.3 Extended Flow Information Object 54
4.3.4 Response Code Object 55
4.3.5 Proxy Support Type Object 56
4.3.6 Message Sequence Number Object 56
4.3.7 Bound Session ID Object 56
4.3.8 Data Sender Information Object 57
4.3.9 NATFW NF Hop Count Object 57
4.3.10 Maximum Hops Object 58
4.3.11 Session Status object 58
4.3.12 QDRQ type 59
4.3.13 QDRQ Response object 59
4.4 Message Formats 60
4.4.1 CREATE 60
4.4.2 RESERVE-EXTERNAL-ADDRESS (REA) 60
4.4.3 RESPONSE 61
4.4.4 QDRQ 61
4.4.5 NOTIFY 62
4.4.6 UCREATE 62

5. NATFW NSLP NTLP Requirements 63

6. NSIS NAT and Firewall Transition Issues 64

7. Security Considerations 65
7.1 Trust Relationship and Authorization 65
7.1.1 Peer-to-Peer Trust Relationship 66
7.1.2 Intra-Domain Trust Relationship 66
7.1.3 End-to-Middle Trust Relationship 67
7.2 Security Threats and Requirements 68
7.2.1 Attacks related to authentication and authorization . 68
7.2.2 Denial-of-Service Attacks 75
7.2.3 Man-in-the-Middle Attacks 76
7.2.4 Message Modification by non-NSIS on-path node 77
7.2.5 Message Modification by malicous NSIS node 77
7.2.6 Session Modification/Deletion 78
7.2.7 Misuse of unreleased sessions 81
7.2.8 Data traffic injection 82
7.2.9 Eavesdropping and traffic analysis 84
7.3 Security Framework for the NAT/Firewall NSLP 85
7.3.1 Security Protection between neighboring NATFW NSLP Nodes 85
7.3.2 Security Protection between non-neighboring NATFW NSLP Nodes 85
7.3.3 End-to-End Security 87

8. Open Issues 88

9. Contributors 89

10. References 90
10.1 Normative References 90
10.2 Informative References 90

Authors' Addresses 92

A. Firewall and NAT Resources 94
A.1 Mapping to Firewall Rules 94
A.2 Mapping to NAT Bindings 94
A.3 Mapping for combined NAT and Firewall 94

B. Problems and Challenges 95
B.1 Missing Network-to-Network Trust Relationship 95
B.2 Relationship with routing 96
B.3 Affected Parts of the Network 96
B.4 NSIS backward compatibility with NSIS unaware NAT and Firewalls 96
B.5 Authentication and Authorization 97
B.6 Directional Properties 97
B.7 Addressing 98
B.8 NTLP/NSLP NAT Support 98
B.9 Combining Middlebox and QoS signaling 98
B.10 Inability to know the scenario 99

C. Object ID allocation for testing 100

D. Acknowledgments 101

Intellectual Property and Copyright Statements 102

1. Introduction

Firewalls and Network Address Translators (NAT) have both been used throughout the Internet for many years, and they will remain present for the foreseeable future. Firewalls are used to protect networks against certain types of attacks from the outside, and in times of IPv4 address depletion, NATs virtually extend the IP address space. Both types of devices may be obstacles to some applications, since they only allow traffic created by a limited set of applications to traverse them (e.g., most HTTP traffic, and client/server applications), due to the relatively static properties of the protocols used. Other applications, such as IP telephony and most other peer-to-peer applications, which have more dynamic properties, create traffic which is unable to traverse NATs and Firewalls unassisted. In practice, the traffic from many applications cannot traverse autonomous Firewalls or NATs, even when they have added functionality which attempts to restore the transparency of the network.

Several solutions to enable applications to traverse such entities have been proposed and are currently in use. Typically, application level gateways (ALG) have been integrated with the Firewall or NAT to configure the Firewall or NAT dynamically. Another approach is middlebox communication (MIDCOM, currently under standardization at the IETF). In this approach, ALGs external to the Firewall or NAT configure the corresponding entity via the MIDCOM protocol [5]. Several other work-around solutions are available, including STUN [24] and TURN [27]. However, all of these approaches introduce other problems that are generally hard to solve, such as dependencies on the type of NAT implementation (full-cone, symmetric, ...), or dependencies on certain network topologies.

NAT and Firewall (NATFW) signaling shares a property with Quality of Service (QoS) signaling. The signaling of both must reach any device on the data path that is involved in QoS or NATFW treatment of data packets. This means, that for both, NATFW and QoS, it is convenient if signaling travels path-coupled, meaning that the signaling messages follow exactly the same path that the data packets take. RSVP [11] is an example of a current QoS signaling protocol that is path-coupled.

This memo defines a path-coupled signaling protocol for NAT and Firewall configuration within the framework of NSIS, called the NATFW NSIS Signaling Layer Protocol (NSLP). The general requirements for NSIS are defined in [3]. The general framework of NSIS is outlined in [2]. It introduces the split between an NSIS transport layer and an NSIS signaling layer. The transport of NSLP messages is handled by an NSIS Network Transport Layer Protocol (NTLP, with GIMPS [1]

being the implementation of the abstract NTLP). The signaling logic for QoS and NATFW signaling is implemented in the different NSLPs. The QoS NSLP is defined in [4], while the NATFW NSLP is defined in this memo.

The NATFW NSLP is designed to request the dynamic configuration of NATs and/or Firewalls along the data path. Dynamic configuration includes enabling data flows to traverse these devices without being obstructed as well as blocking of particular data flows at upstream firewalls. Enabling data flows requires the loading of firewall pin holes (loading of firewall rules with action allow) and creating NAT bindings. Blocking of data flows requires the loading of firewalls rules with action deny/drop. A simplified example for enabling data flows: A source host sends a NATFW NSLP signaling message towards its data destination. This message follows the data path. Every NATFW NSLP NAT/Firewall along the data path intercepts these messages, processes them, and configures itself accordingly. Afterwards, the actual data flow can traverse every configured Firewall/NAT.

It is necessary to distinguish between two different basic scenarios when operating the NATFW NSLP, independent of the type of middlebox to be configured.

1. Both data sender and data receiver of the network are NSIS NATFW NSLP aware. This includes the cases where the data sender is logically decomposed from the NSIS initiator or the data receiver logically decomposed from the NSIS receiver, but both sides support NSIS. This scenario assumes deployment of NSIS all over the Internet, or at least at all NATs and firewalls.

2. Only one end host is NSIS NATFW NSLP aware, either data receiver or data sender.

NATFW NSLP provides three modes to cope with various possible scenarios likely to be encountered before and after widespread deployment of NSIS. Once there is full deployment of NSIS (in the sense that both end hosts support NATFW NSLP signaling), the requisite NAT and firewall state can be created using either just CREATE mode if the data receiver resides in a public addressing realm, or a combination of RESERVE-EXTERNAL-ADDRESS and CREATE modes if the data receiver resides in a private addressing realm and needs to preconfigure the boundary NAT to provide a publically reachable address for use by the data sender. During the introduction of NSIS, it is likely that one or other of the data sender and receiver will not be NSIS capable. In these cases the NATFW NSLP can utilise NSIS aware middleboxes on the path between the sender and receiver to provide proxy NATFW NSLP services. Typically these boxes will be at

the boundaries of the realms in which the end hosts are located. If the data receiver is NSIS unaware, the normal modes can be employed but the NSIS signaling terminates at the NSIS aware node topologically closest to the receiver which then acts as a proxy for the receiver. If the data sender is unaware a variant of the RESERVE-EXTERNAL-ADDRESS mode can be used by a data receiver behind a NAT and the specialised UCREATE mode can be used by a data receiver behind a firewall.

All modes of operation create NATFW NSLP and NTLP state in NSIS entities. NTLP state allows signaling messages to travel in the forward (downstream) and the reverse (upstream) direction along the path between a NAT/Firewall NSLP sender and a corresponding receiver. NAT bindings and firewall rules are NAT/Firewall specific state. This state is managed using a soft-state mechanism, i.e., it expires unless it is refreshed from time to time.

Section 2 describes the network environment for NATFW NSLP signaling, highlighting the trust relationships and authorization required. Section 3 defines the NATFW signaling protocol. Section 4 defines the messages and and message components. In the remaining parts of the main body of the document, Section 6 covers transition issues and Section 7 addresses security considerations. Currently unsolved problems and challenges are listed and discussed in Appendix B. Please note that readers familiar with Firewalls and NATs and their possible location within networks can safely skip Section 2.

1.1 Terminology and Abbreviations

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

This document uses a number of terms defined in [3]. The following additional terms are used:

- Policy rule: A policy rule is "a basic building block of a policy-based system. It is the binding of a set of actions to a set of conditions - where the conditions are evaluated to determine whether the actions are performed" [26]. In the context of NSIS NATFW NSLP, the condition is a specification of a set of packets to which rules are applied. The set of actions always contains just a single element per rule, and is limited to either action "reserved", "deny" or action "allow".

- Firewall: A packet filtering device that matches packets against a set of policy rules and applies the actions. In the context of NSIS NATFW NSLP we refer to this device as a Firewall.

- Network Address Translator: Network Address Translation is a method by which IP addresses are mapped from one IP address realm to another, in an attempt to provide transparent routing between hosts (see [7]). Network Address Translators are devices that perform this work.

- Middlebox: "A middlebox is defined as any intermediate device performing functions other than the normal, standard functions of an IP router on the datagram path between a source host and a destination host" [9]. In the context of this document, the term middlebox refers to Firewalls and NATs only. Other types of middlebox are currently outside of the scope of this document.

- Security Gateway: IPsec based gateways.

- (Data) Receiver (DR or R): The node in the network that is receiving the data packets of a flow.

- (Data) Sender (DS or S): The node in the network that is sending the data packets of a flow.

- NATFW NSLP session: An application layer flow of information for which some network control state information is to be manipulated or monitored (as defined in [2]). The control state for NATFW NSLP consists of NSLP state and associated policy rules at a middlebox.

- NSIS peer or peer: An NSIS node with which an NSIS adjacency has been created as defined in [1].

- Edge-NAT: An edge-NAT is a NAT device that is reachable from the public Internet and that has a globally routable IP address.

- Edge-Firewall: An edge-Firewall is a Firewall device that is located on the demarcation line of an administrative domain.

- Public Network: "A Global or Public Network is an address realm with unique network addresses assigned by Internet Assigned Numbers Authority (IANA) or an equivalent address registry. This network is also referred as external network during NAT discussions" [7].

- Private/Local Network: "A private network is an address realm independent of external network addresses. Private network may also be referred alternately as Local Network. Transparent routing between hosts in private realm and external realm is facilitated by a NAT router" [7]. IP address space allocation for private networks is recommended in [25]

- Public/Global IP address: An IP address located in the public network according to Section 2.7 of [7].

- Private/Local IP address: An IP address located in the private network according to Section 2.8 of [7].

- Initial CREATE: A CREATE message creating a new session.

1.2 Middleboxes

The term middlebox covers a range of devices which intercept the flow of packets between end hosts and perform actions other than standard forwarding expected in an IP router. As such, middleboxes fall into a number of categories with a wide range of functionality, not all of which is pertinent to the NATFW NSLP. Middlebox categories in the scope of this memo are Firewalls that filter data packets against a set of filter rules, and NATs that translate packet addresses from one address realm to another address realm. Other categories of middleboxes, such as QoS traffic shapers and security gateways, are out of scope.

The term NAT used in this document is a placeholder for a range of different NAT flavors. We consider the following types of NATs:

- traditional NAT (basic NAT and NAPT)

- Bi-directional NAT

- Twice-NAT

- Multihomed NAT

For definitions and a detailed discussion about the characteristics of each NAT type please see [7].

Both types of middleboxes under consideration here use policy rules to make a decision on data packet treatment. Policy rules consist of a flow identifier which selects the packets to which the policy applies and an associated action; data packets matching the flow identifier are subjected to the policy rule action. A typical flow identifier is the 5-tuple selector which matches the following fields of a packet to configured values:

- Source and destination IP addresses

- Transport protocol number

- Transport source and destination port numbers

For further examples of flow identifiers see Section 5.2.2 of [1].

Actions for Firewalls are usually one or more of:

- Allow: forward data packet

- Deny: block data packet and discard it

- Other actions such as logging, diverting, duplicating, etc

Actions for NATs include (amongst many others):

- Change source IP address and transport port number to a globally routeable IP address and associated port number.

- Change destination IP address and transport port number to a private IP address and associated port number.

1.3 Non-Goals

Traversal of non-NSIS and non-NATFW NSLP aware NATs and Firewalls is outside the scope of this document.

Only Firewalls and NATs are considered in this document, any other types of devices, for instance IPSec security gateway, are out of scope.

The exact implementation of policy rules and their mapping to firewall rule sets and NAT bindings or sessions at the middlebox is an implementation issue and thus out of scope of this document.

Some devices categorized as firewalls only accept traffic after cryptographic verification (i.e., IPsec protected data traffic). Particularly for network access scenarios, either link layer or network layer data protection is common. We do not address these types of devices (referred to as security gateways) since per-flow signaling is typically not used in this environment.

Another application, for which NSIS signaling has been proposed but which is out of scope for this document, is discovering security gateways, for the purpose of executing IKE to create an IPsec SA.

In mobility scenarios, a common problem is the traversal of a security gateway at the edge of a corporate network. Network

administrators allow only authenticated data to enter the network. A problem statement for the traversal of these security gateways in the context of Mobile IP can be found in [22]). This topic is not within the scope of the present document.

1.4 General Scenario for NATFW Traversal

The purpose of NSIS NATFW signaling is to enable communication between endpoints across networks even in the presence of NAT and Firewall middleboxes. It is assumed that these middleboxes will be statically configured in such a way that NSIS NATFW signaling messages themselves are allowed to traverse them. NSIS NATFW NSLP signaling is used to dynamically install additional policy rules in all NATFW middleboxes along the data path. Firewalls are configured to forward data packets matching the policy rule provided by the NSLP signaling. NATs are configured to translate data packets matching the policy rule provided by the NSLP signaling. However, there is an exception to the primary goal of NSIS NATFW signaling, NSIS NATFW nodes can request blocking of particular data flows instead of enabling these flows at upstream firewalls.

The basic high-level picture of NSIS usage is that end hosts are located behind middleboxes, meaning that there is a middlebox on the data path from the end host in a private network and the external network (NAT/FW in Figure 1). Applications located at these end hosts try to establish communication with corresponding applications on other such end hosts. They trigger the NSIS entity at the local host to provide for middlebox traversal along the prospective data path (e.g., via an API call). The NSIS entity in turn uses NSIS NATFW NSLP signaling to establish policy rules along the data path, allowing the data to travel from the sender to the receiver unobstructed.

```
   Application          Application Server (0, 1, or more)   Application

   +----+                        +----+                        +----+
   |    +------------------------+    +------------------------+    |
   +-+--+                        +----+                        +-+--+
     |                                                           |
     |         NSIS Entities                    NSIS Entities  |
   +-+--+        +----+                        +-----+     +-+--+
   |    +--------+    +----------------------------+     +-----+    |
   +-+--+        +-+--+                        +--+--+     +-+--+
     |             |               ------             |          |
     |             |           ////      \\\\\        |          |
   +-+--+        +-+--+      |/               |     +-+--+     +-+--+
   |    |        |    |     |   Internet      |    |    |     |    |
   |    +--------+    +-----+                  +----+    +-----+    |
   +----+        +----+      |\               |     +----+     +----+
                               \\\\      /////
   sender    NAT/FW (1+)           ------          NATFW (1+) receiver
```

Figure 1: Generic View on NSIS in a NAT / Firewall case

For end-to-end NATFW signaling, it is necessary that each firewall and each NAT along the path between the data sender and the data receiver implements the NSIS NATFW NSLP. There might be several NATs and FWs in various possible combinations on a path between two hosts. Section 2 presents a number of likely scenarios with different combinations of NATs and firewalls.

2. Network Deployment Scenarios using NATFW NSLP

This section introduces several scenarios for middlebox placement within IP networks. Middleboxes are typically found at various different locations, including at Enterprise network borders, within enterprise networks, as mobile phone network gateways, etc. Usually, middleboxes are placed more towards the edge of networks than in network cores. Firewalls and NATs may be found at these locations either alone, or they may be combined; other categories of middleboxes may also be found at such locations, possibly combined with the NATs and/or Firewalls. To reduce the number of network elements needed, combined Firewall and NATs have been made available.

NSIS initiators (NI) send NSIS NATFW NSLP signaling messages via the regular data path to the NSIS responder (NR). On the data path, NATFW NSLP signaling messages reach different NSIS nodes that implement the NATFW NSLP. Each NATFW NSLP node processes the signaling messages according to Section 3 and, if necessary, installs policy rules for subsequent data packets.

Each of the following sub-sections introduces a different scenario for a different set of middleboxes and their ordering within the topology. It is assumed that each middlebox implements the NSIS NATFW NSLP signaling protocol.

2.1 Firewall Traversal

This section describes a scenario with Firewalls only; NATs are not involved. Each end host is behind a Firewall. The Firewalls are connected via the public Internet. Figure 2 shows the topology. The part labeled "public" is the Internet connecting both Firewalls.

```
                  +-----+    //----\\       +-----+
          NI -----| FW |---|        |------| FW |--- NR
                  +-----+    \\----//       +-----+

                 private     public        private


             FW: Firewall
             NI: NSIS Initiator
             NR: NSIS Responder
```

Figure 2: Firewall Traversal Scenario

Each Firewall on the data path must provide traversal service for NATFW NSLP in order to permit the NSIS message to reach the other end host. All Firewalls process NSIS signaling and establish appropriate

policy rules, so that the required data packet flow can traverse them.

Placing firewalls in a network topology can be done in several very different ways. To distinguish firewalls located at network borders, such as administrative domains, from others located internally, the term edge-Firewall is used. A similar distinction can be made for NATs, with an edge-NAT fulfilling the equivalent role.

2.2 NAT with two private Networks

Figure 3 shows a scenario with NATs at both ends of the network. Therefore, each application instance, NSIS initiator and NSIS responder, are behind NATs. The outermost NAT, called edge-NAT, at each side is connected to the public Internet. The NATs are generically labeled as MB (for middlebox), since those devices certainly implement NAT functionality, but can implement firewall functionality as well.

Only two middleboxes MB are shown in Figure 3 at each side, but in general, any number of MBs on each side must be considered.

```
        +----+     +----+    //----\\    +----+     +----+
   NI --| MB |-----| MB |---|        |---| MB |-----| MB |--- NR
        +----+     +----+    \\----//    +----+     +----+

             private          public          private

          MB: Middlebox
          NI: NSIS Initiator
          NR: NSIS Responder
```

Figure 3: NAT with two Private Networks Scenario

Signaling traffic from NI to NR has to traverse all the middleboxes on the path, and all the middleboxes must be configured properly to allow NSIS signaling to traverse them. The NATFW signaling must configure all middleboxes and consider any address translation that will result from this configuration in further signaling. The sender (NI) has to know the IP address of the receiver (NR) in advance, otherwise it will not be possible to send any NSIS signaling messages towards the responder. Note that this IP address is not the private IP address of the responder. Instead a NAT binding (including a public IP address) has to be previously installed on the NAT that subsequently allows packets reaching the NAT to be forwarded to the receiver within the private address realm. This generally requires further support from an application layer protocol for the purpose of discovering and exchanging information. The receiver might have a

number of ways to learn its public IP address and port number and might need to signal this information to the sender using the application level signaling protocol.

2.3 NAT with Private Network on Sender Side

This scenario shows an application instance at the sending node that is behind one or more NATs (shown as generic MB, see discussion in Section 2.2). The receiver is located in the public Internet.

```
             +-----+    +-----+    //----\\
        NI --| MB |-----| MB |---|        |--- NR
             +-----+    +-----+    \\----//

                  private          public

             MB: Middlebox
             NI: NSIS Initiator
             NR: NSIS Responder
```

Figure 4: NAT with Private Network on Sender Side Scenario

The traffic from NI to NR has to traverse middleboxes only on the sender's side. The receiver has a public IP address. The NI sends its signaling message directly to the address of the NSIS responder. Middleboxes along the path intercept the signaling messages and configure the policy rules accordingly.

Note that the data sender does not necessarily know whether the receiver is behind a NAT or not, hence, it is the receiving side that has to detect whether itself is behind a NAT or not. As described in Section 3.3.2 NSIS can also provide help for this procedure.

2.4 NAT with Private Network on Receiver Side Scenario

The application instance receiving data is behind one or more NATs shown as MB (see discussion in Section 2.2).

```
               //----\\    +----+     +----+
        NI ---|        |---| MB |-----| MB |--- NR
               \\----//    +----+     +----+

                public          private


             MB: Middlebox
             NI: NSIS Initiator
             NR: NSIS Responder
```

Figure 5: NAT with Private Network on Receiver Scenario

Initially, the NSIS responder must determine its publicly reachable IP address at the external middlebox and notify the NSIS initiator about this address. One possibility is that an application level protocol is used, meaning that the public IP address is signaled via this protocol to the NI. Afterwards the NI can start its signaling towards the NR and so establish the path via the middleboxes in the receiver side private network.

This scenario describes the use case for the RESERVE-EXTERNAL-ADDRESS mode of the NATFW NSLP.

2.5 Both End Hosts behind twice-NATs

This is a special case, where the main problem arises from the need to detect that both end hosts are logically within the same address space, but are also in two partitions of the address realm on either side of a twice-NAT (see [7] for a discussion of twice-NAT functionality).

Sender and receiver are both within a single private address realm but the two partitions potentially have overlapping IP address ranges. Figure 6 shows the arrangement of NATs. This is a common configuration in networks, particularly after the merging of companies that have used the same private address space, resulting in overlapping address ranges.

```
                                   public
             +----+     +----+    //----\\
        NI --| MB |--+--| MB |---|        |
             +----+  |  +----+    \\----//
                     |
                     |  +----+
                     +--| MB |------------ NR
                        +----+

                   private

             MB: Middlebox
             NI: NSIS Initiator
             NR: NSIS Responder
```

Figure 6: NAT to Public, Sender and Receiver on either side of a twice-NAT Scenario

The middleboxes shown in Figure 6 are twice-NATs, i.e., they map IP addresses and port numbers on both sides, meaning the mapping of source and destination address at the private and public interfaces.

This scenario requires the assistance of application level entities, such as a DNS server. The application level gateways must handle requests that are based on symbolic names, and configure the middleboxes so that data packets are correctly forwarded from NI to NR. The configuration of those middleboxes may require other middlebox communication protocols, such as MIDCOM [5]. NSIS signaling is not required in the twice-NAT only case, since middleboxes of the twice-NAT type are normally configured by other means. Nevertheless, NSIS signaling might by useful when there are also Firewalls on path. In this case NSIS will not configure any policy rule at twice-NATs, but will configure policy rules at the Firewalls on the path. The NSIS signaling protocol must be at least robust enough to survive this scenario. This requires that twice-NATs must implement the NATFW NSLP also and participate in NATFW sessions but they do not change the configuration of the NAT, i.e., they only read the address mapping information out of the NAT and translate the Message Routing Information (MRI, [1])within the NSLP and NTLP accordingly.

2.6 Both End Hosts Behind Same NAT

When NSIS initiator and NSIS responder are behind the same NAT (thus being in the same address realm, see Figure 7), they are most likely not aware of this fact. As in Section 2.4 the NSIS responder must determine its public IP address in advance and transfer it to the NSIS initiator. Afterwards, the NSIS initiator can start sending the

signaling messages to the responder's public IP address. During this process, a public IP address will be allocated for the NSIS initiator at the same middlebox as for the responder. Now, the NSIS signaling and the subsequent data packets will traverse the NAT twice: from initiator to public IP address of responder (first time) and from public IP address of responder to responder (second time). This is the worst case in which both sender and receiver obtain a public IP address at the NAT, and the communication path is certainly not optimal in this case.

```
            NI              public
             \  +----+     //----\\
              +-| MB |----|        |
             /  +----+     \\----//
            NR
                private

          MB: Middlebox
          NI: NSIS Initiator
          NR: NSIS Responder
```

Figure 7: NAT to Public, Both Hosts Behind Same NAT

The NSIS NATFW signaling protocol should support mechanisms to detect such a scenario.

2.7 IPv4/v6 NAT with two Private Networks

This scenario combines the use case described in Section 2.2 with the IPv4 to IPv6 transition scenario involving address and protocol translation, i.e., using Network Address and Protocol Translators (NAT-PT, [8]).

The difference from the other scenarios is the use of IPv6 to IPv4 (and vice versa) address and protocol translation. Additionally, the base NTLP must support transport of messages in mixed IPv4 and IPv6 networks where some NSIS peers provide translation.

```
     +----+  +----+   //---\\   +----+  //---\\   +----+  +----+
NI --| MB |--| MB |--|       |--| MB |-|       |--| MB |--| MB |-- NR
     +----+  +----+   \\---//   +----+  \\---//   +----+  +----+

          private      public             public       private
                        IPv4               IPv6

          MB: Middlebox
          NI: NSIS Initiator
          NR: NSIS Responder
```

Figure 8: IPv4/v6 NAT with two Private Networks

This scenario needs the same type of application level support as described in Section 2.5, and so the issues relating to twice-NATs apply here as well.

2.8 Multihomed Network with NAT

The previous sub-sections sketched network topologies where several NATs and/or Firewalls are ordered sequentially on the path. This section describes a multihomed scenario with two NATs placed on alternative paths to the public network.

```
          +----+
NI -------| MB |\
    \     +----+ \  //---\\
     \            -|       |-- NR
      \             \\---//
       \  +----+       |
        --| MB |-------+
          +----+
          private
     private          public

          MB: Middlebox
          NI: NSIS Initiator
          NR: NSIS Responder
```

Figure 9: Multihomed Network with Two NATs

Depending on the destination or load balancing requirements, either one or the other middlebox is used for the data flow. Which middlebox is used depends on local policy or routing decisions. NATFW NSLP must be able to handle this situation properly, see Section 3.3.2 for an expanded discussion of this topic with respect to NATs.

2.9 Multihomed Network with Firewall

This section describes a multihomed scenario with two firewalls placed on alternative paths to the public network (Figure 10). The routing in the private and public network decided which firewall is being taken for data flows. Depending on the data flow's direction, either outbound or inbound, a different firewall could be traversed. This is a challenge for a certain mode of the NATFW NSLP where the NSIS responder is located behind these firewalls within the private network: the UCREATE mode. The UCREATE mode is used to block a particular data flow on an upstream firewall. NSIS must route the UCREATE mode message upstream from NR to NI without probably knowing the data traffic's subsequent path will take from NI to NR.

```
             +----+
   NR -------| MB |\
       \     +----+ \  //---\\
        \            -|       |-- NI
         \             \\---//
          \  +----+       |
           --| MB |-------+
             +----+
             private
        private          public

             MB: Middlebox
             NI: NSIS Initiator
             NR: NSIS Responder
```

Figure 10: Multihomed Network with Two Firewalls

3. Protocol Description

This section defines messages, objects, and protocol semantics for the NATFW NSLP. Section 3.1 introduces the base element of a NSLP session , the policy rule. Section 3.2 introduces the protocol and the protocol behavior is defined in Section 3.3. Section 4 defines the syntax of the messages and objects.

3.1 Policy Rules

Policy rules, bound to a session, are the building block of middlebox devices considered in the NATFW NSLP. For Firewalls the policy rule usually consists of a 5-tuple, source/destination addresses, transport protocol, and source/destination port numbers, plus an action, such as allow or deny. For NATs the policy rule consists of action 'translate this address' and further mapping information, that might be, in the simplest case, internal IP address and external IP address.

Policy rules are usually carried in one piece in signaling applications. In NSIS the policy rule is divided into the flow identifier, an allow or deny action, and additional information. The filter specification is carried within NTLP's message routing information (MRI) and additional information, including the specification of the action, is carried in NSLP's objects. Additional information is, for example, the lifetime of a policy rule or session.

3.2 Basic protocol overview

The NSIS NATFW NSLP is carried over the NSIS Transport Layer Protocol (NTLP) defined in [1]. The interworking with the NTLP and other components is shown in Figure 56. NATFW NSLP messages are initiated by the NSIS initiator (NI), handled by NSIS forwarders (NF) and finally processed by the NSIS responder (NR). It is required that at least NI and NR implement this NSLP, intermediate NFs only implement this NSLP when they provide relevant middlebox functions. NSIS forwarders that do not have any NATFW NSLP functions just forward these packets when they have no interest.

A Data Sender (DS), intending to send data to a Data Receiver (DR) must first initiate NATFW NSLP signaling. This causes the NI associated with the data sender (DS) to launch NSLP signaling towards the address of data receiver DR (see Figure 11). Although it is expected that the DS and the NATFW NSLP NI will usually reside on the same host, this specification does not rule out scenarios where the DS and NI reside on different hosts, the so-called proxy mode (see Section 1.)

```
             +-------+    +-------+    +-------+    +-------+
             | DS/NI |<~~~| MB1/  |<~~~| MB2/  |<~~~| DR/NR |
             |       |--->| NF1   |--->| NF2   |--->|       |
             +-------+    +-------+    +-------+    +-------+


                 ========================================>
                         Data Traffic Direction

                  --->  : NATFW NSLP request signaling
                  ~~~>  : NATFW NSLP response signaling
                  DS/NI : Data sender and NSIS initiator
                  DR/NR : Data receiver and NSIS responder
                  MB1   : Middlebox 1 and NSIS forwarder 1
                  MB2   : Middlebox 2 and NSIS forwarder 2
```

Figure 11: General NSIS signaling

```
             +-------+    +-------+    +-------+    +-------+
             | DS/NI |<~~~| MB1/  |<~~~| NR    |    |   DR  |
             |       |--->| NF1   |--->|       |    |       |
             +-------+    +-------+    +-------+    +-------+


                 ========================================>
                         Data Traffic Direction

                  --->  : NATFW NSLP request signaling
                  ~~~>  : NATFW NSLP response signaling
                  DS/NI : Data sender and NSIS initiator
                  DR/NR : Data receiver and NSIS responder
                  MB1   : Middlebox 1 and NSIS forwarder 1
                  MB2   : Middlebox 2 and NSIS forwarder 2
```

Figure 12: A NSIS proxy mode signaling

The sequence of NSLP events is as follows:

- NSIS initiators generate NATFW NSLP request messages and send those towards the NSIS responder. Note, that the NSIS initiator may not necessarily be the data sender but may be the data receiver, for instance, when using the RESERVE-EXTERNAL-ADDRESS

message.

- NSLP request messages are processed each time a NF with NATFW NSLP support is traversed. These nodes process the message, check local policies for authorization and authentication, possibly create policy rules, and forward the signaling message to the next NSIS node. The request message is forwarded until it reaches the NSIS responder.

- NSIS responders will check received messages and process them if applicable. NSIS responders generate response messages and send them hop-by-hop back to the NI via the same chain of NFs (traversal of the same NF chain is guaranteed through the established reverse message routing state in the NTLP). Note, that NSIS responder may not necessarily be the data receiver but may be any intermediate NSIS node that terminates the forwarding, for example, in a proxy mode case where an edge-NAT is replying to requests

- The response message is processed at each NF implementing the NATFW NSLP.

- Once the NI has received a successful response, the data sender can start sending its data flow to the data receiver.

Because NATFW NSLP signaling follows the data path from DS to DR, this immediately enables communication between both hosts for scenarios with only Firewalls on the data path or NATs on sender side. For scenarios with NATs on the receiver side certain problems arise, as described in Section 2.

When the NR and the NI are located in different address realms and the NR is located behind a NAT, the NI cannot signal to the NR directly. The DR and NR are not reachable from the NIs using the private address of the NR and thus NATFW signaling messages cannot be sent to the NR/DR's address. Therefore, the NR must first obtain a NAT binding that provides an address that is reachable for the NI. Once the NR has acquired a public IP address, it forwards this information to the DS via a separate protocol (such as SDP within SIP). This application layer signaling, which is out of scope of the NATFW NSLP, may involve third parties that assist in exchanging these messages.

NATFW NSLP signaling supports this scenario by using the RESERVE-EXTERNAL-ADDRESS mode of operation

1. The NR acquires a public address by signaling on the reverse path (NR towards NI) and thus making itself available to other hosts.

This process of acquiring a public addresses is called reservation. During this process the DR reserves publicly reachable addresses and ports suitable for NATFW NSLP signaling, but data traffic will not be allowed to use this address/port initially.

2. The NI signals directly to the NR as the NI would do if there is no NAT in between, and creates policy rules at middleboxes. Note, that the reservation mode will only allow the forwarding of signaling messages but not data flow packets. Data flow packets will be 'activated' by the signaling from NI towards NR. The RESERVE-EXTERNAL-ADDRESS mode of operation is detailed in Section 3.3.2

The above usage assumes that both ends of a communication support NSIS but fail when NSIS is only deployed at one end of the network. In this case only the receiving or sending side are NSIS aware and not both at the same time (see also Section 1). NATFW NSLP supports this scenario by using a proxy mode, as described in Section 3.3.7 and Section 3.3.8.

The basic functionality of the NATFW NSLP provides for opening firewall pin holes and creating NAT bindings to enable data flows to traverse these devices. Firewalls are expected to work on a deny-all policy, meaning that traffic that does not explicitly match any firewall filter rule will be blocked. In contrast, the normal behavior of NATs is to block all traffic that does not match any already configured/installed binding or session. However, in some scenarios it is required to support firewalls having allow-all policies, allowing data traffic to traverse unless it is blocked explicitly. Data receivers can utilize NATFW NSLP's UCREATE message to install policy rules at upstream firewalls to block unwanted traffic.

The protocol works on a soft-state basis, meaning that whatever state is installed or reserved on a middlebox will expire, and thus be de-installed/ forgotten after a certain period of time. To prevent this, the NATFW nodes involved will have to specifically request a session extension. An explicit NATFW NSLP state deletion capability is also provided by the protocol.

Middleboxes should return an error in case of a failure, such that appropriate actions can be taken; this ability would allow debugging and error recovery. Error messages could be sent upstream (for errors related to received messages as well as asynchronous error notification messages) towards the NI as well as downstream towards the NR (in the case of asynchronous error notification messages).

The next sections define the NATFW NSLP message types and formats, protocol operations, and policy rule operations.

3.3 Protocol Operations

This section defines the protocol operations including, how to create sessions, maintain them, and how to reserve addresses. All the NATFW NSLP protocol messages require C-mode handling by the NTLP and cannot be piggybacked into D-mode NTLP messages used during the NTLP path discovery/refresh phase. The usage of the NTLP by protocol messages is described in detail in Section 4.

The protocol uses six messages:

- CREATE: a request message used for creating, changing, refreshing and deleting NATFW NSLP sessions.

- RESERVE-EXTERNAL-ADDRESS (REA): a request message used for reserving an external address and probably port number, depending on the type of NAT.

- Query and Diagnosis ReQuest (QDRQ): a request message used by authorized NATFW NEs for querying and diagnosing installed NATFW states

- NOTIFY: an asynchronous message used by NATFW NEs to alert upstream and/or downstream NATFW NEs about specific events (especially failures).

- UCREATE: a request message used by data receivers to instruct upstream firewalls to block data traffic.

- RESPONSE: used as a response to CREATE, REA, UCREATE and QUERY messages with Success or Error information

3.3.1 Creating Sessions

Allowing two hosts to exchange data even in the presence of middleboxes is realized in the NATFW NSLP by the CREATE request message. The data sender generates a CREATE message as defined in Section 4.4.1 and hands it to the NTLP. The NTLP forwards the whole message on the basis of the message routing information towards the NR. Each NSIS forwarder along the path that implements NATFW NSLP, processes the NSLP message. Forwarding is thus managed NSLP hop-by-hop but may pass transparently through NSIS forwarders which do not contain NATFW NSLP functionality and non-NSIS aware routers between NSLP hop waypoints. When the message reaches the NR, the NR can

accept the request or reject it. The NR generates a response to the request and this response is transported hop-by-hop towards the NI. NATFW NSLP forwarders may reject requests at any time. Figure 13 sketches the message flow between NI (DS), a NF (NAT), and NR (DR).

```
       NI      Private Network        NF    Public Internet        NR
       |                              |                            |
       | CREATE                       |                            |
       |----------------------------->|                            |
       |                              |                            |
       | RESPONSE[Error](if necessary)|                            |
       |<-----------------------------| CREATE                     |
       |                              |--------------------------->|
       |                              |                            |
       |                              | RESPONSE[Success/Error]    |
       |    RESPONSE[Success/Error]   |<---------------------------|
       |<-----------------------------|                            |
       |                              |                            |
       |                              |                            |
```

Figure 13: Creation message flow

Since the CREATE message is used for several purposes within the lifetime of a session, there are several processing rules for NATFW NEs when generating and receiving CREATE messages. The different processing methods depend not only on the function which the CREATE is performing (to create, modify, refresh or delete a session) but also on the node at which the processing happens. For an initial CREATE message, the CREATE message creating a new NSIS session, the processing of CREATE messages is different for every NSIS node type:

- NSLP initiator: NI only generates initial CREATE messages and hands them over to the NTLP. After receiving a successful response, the data path is configured and the DS can start sending its data to the DR. After receiving an 'error' response message the NI MAY try to generate the CREATE message again or give up and report the failure to the application, depending on the error condition.

- NATFW NSLP forwarder: NFs receiving an initial CREATE message MUST first check authentication and authorization before any further processing is executed. The NF SHOULD check with its local policies if it can accept the desired policy rule given the combination of the NTLP's 'Message-Routing-Information' (MRI) [1] (the flow description information) and the CREATE payload

(behavior to be enforced on the packet stream). An initial CREATE is distinguished from subsequent CREATE messages by the absence of existing NSLP session state related to the same session ID or the same MRI. The NSLP message processing depends on the middlebox type:

- NAT: When the initial CREATE message is received at the public side of the NAT, it looks for a reservation made in advance, by using a REA message Section 3.3.2, that matches the destination address/port of the MRI provided by the NTLP. If no reservation had been made in advance the NSLP MAY return an error response message of type 'no reservation found' and discard the request. If there is a reservation, NSLP stores the data sender's address as part of the policy rule to be loaded and forwards the message with the address set to the internal (private in most cases) address of the next NSIS node. When the initial CREATE message, for a new session, is received at the private side the NAT binding is reserved, but not activated. The NSLP message is forwarded to the next NSIS hop with source address set to the NAT's external address from the newly reserved binding.

- Firewall: When the initial CREATE message is received the NSLP just remembers the requested policy rule, but does not install any policy rule. Afterwards, the message is forwarded to the next NSLP hop. There is a difference between requests from trusted (authorized NIs) and un-trusted (un-authorized NIs); requests from trusted NIs will be pre-authorized, whereas requests from un-trusted NIs will not be pre-authorized. This difference is required to speed-up the protocol operations as well as for proxy mode usage (please refer to Section 3.3.7 and [13]).

- Combined NAT and Firewall: Processing at combined Firewall and NAT middleboxes is the same as in the NAT case. No policy rules are installed. Implementations MUST take into account the order of packet processing in the Firewall and NAT functions within the device. This will be referred to as 'order of functions' and is generally different depending on whether the packet arrives at the external or internal side of the middlebox.

o NSLP receiver: NRs receiving initial CREATE messages MUST reply with a 'success' (response object has success information) RESPONSE message if they accept the CREATE request message and the authorization and authentication checks have been successful. Otherwise they SHOULD generate a RESPONSE message with an error code. RESPONSE messages are sent back NSLP hop-by-hop towards the

NI, independently of the response codes, either success or error.

Policy rules at middleboxes MUST be only installed upon receiving a successful response. This is a countermeasure to several problems, for example wastage of resources due to loading policy rules at intermediate NF when the CREATE message does not reach the final NR for some reason.

3.3.2 Reserving External Addresses

NSIS signaling is intended to travel end-to-end, even in the presence of NATs and Firewalls on-path. This works well in cases where the data sender is itself behind a NAT as described in Section 3.3.1. For scenarios where the data receiver is located behind a NAT and needs to receive data flows from outside its own network (see Figure 5) the problem is more troublesome. NSIS signaling, as well as subsequent data flows, are directed to a particular destination IP address that must be known in advance and reachable.

```
                      +-------------+   AS-Data Receiver Communication
            +-------->| Application |<-----------------------------+
            |         | Server      |                              |
            |         +-------------+                              |
            |                                          IP(R-NAT_B) |
            |         NSIS Signaling Message               +-------+--+
            |  +------------------------------------------>| NAT/NAPT |
            |  |                                           | B        |
            |  |                                           +-------+--+
            |  |                                                   |
     AS-Data|  |                                                   |
    Receiver|  |                       +----------+                |
       Comm.|  |                       | NAT/NAPT |                |
            |  |                       | A        |                |
            |  |                       +----------+                |
            |  |                                                   |
            |  |                                                   |
            |  |                                                   |
            |  |                                                   |
            v  |                                             IP(R) v
        +--------+                                          +---------+
        | Data   |                                          | Data    |
        | Sender |                                          | Receiver|
        +--------+                                          +---------+
```

Figure 14: The Data Receiver behind NAT problem

Figure 14 describes a typical message communication in a peer-to-peer networking environment whereby the two end points learn of each others existence with the help of a third party (referred to as an Application Server). Communication between the application server and each of the two end points (data sender and data receiver) enables the two end hosts to learn each other's IP addresses. The approach described in this memo supports this peer-to-peer approach, but is not limited to it.

Some sort of communication between the data sender/data receiver and a third party is typically necessary (independently of whether NSIS is used). NSIS signaling messages cannot be used to communicate the relevant application level end point identifiers (in the generic case at least) as a replacement for communication with the application server.

If the data receiver is behind a NAT then an NSIS signaling message will be addressed to the IP address allocated at the NAT (assuming one had already been allocated). If no corresponding NSIS NAT Forwarding State at NAT/NAPT B exists (binding IP(R-NAT B) <-> IP(R)) then the signaling message will terminate at the NAT device (most likely without generating a proper response message). The signaling message transmitted by the data sender cannot install the NAT binding or NSIS NAT Forwarding State "on-the-fly" since this would assume that the data sender knows the topology at the data receiver side (i.e., the number and the arrangement of the NAT and the private IP address(es) of the data receiver). The primary goal of path-coupled middlebox communication was not to avoid end hosts learning and preserving this type of topology knowledge. Data receivers behind a NAT must first reserve an external IP address (probably port number too).

```
       Public Internet                Private Address
                                           Space
                    Edge
    NI(DS)          NAT                    NAT                  NR(DR)
    NR+                                                         NI+
    |               |                       |                       |
    |               |                       |                       |
    |               |                       |                       |
    |               |         REA           |         REA           |
    |               |<----------------------|<----------------------|
    |               |                       |                       |
    |               |RESPONSE[Success/Error]|RESPONSE[Success/Error]|
    |               |---------------------->|---------------------->|
    |               |                       |                       |
    |               |                       |                       |

      =============================================================>
                        Data Traffic Direction
```

Figure 15: Reservation message flow

Figure 15 shows the message flow for reserving an external address/ port at a NAT. In this case the roles of the different NSIS entities are:

- The data receiver (DR) for the anticipated data traffic is the NSIS initiator (NI+) for the RESERVE-EXTERNAL-ADDRESS (REA) message, but becomes the NSIS responder (NR) for following CREATE messages.

- The actual data sender (DS) will be the NSIS initiator (NI) for later CREATE messages and may be the NSIS target of the signaling (NR+).

- The actual target of the REA message, the Opportunistic Address (OA) is an arbitrary address, that would force the message to get intercepted by the far outmost NAT in the network. The Opportunistic Address is shown as NR+.

The NI+ (could be on the data receiver DR or on any other host within the private network) sends a the REA message targeted to the Opportunistic Address (OA defined earlier). The OA selection for this message is discussed in Section 3.7. The message routing for the REA message is in the reverse direction to the normal message routing used for path-coupled signaling where the signaling is sent downstream (as opposed to upstream in this case). When establishing NAT bindings (and NSIS session state) the direction does not matter

since the data path is modified through route pinning due to the external NAT address. Subsequent NSIS messages (and also data traffic) will travel through the same NAT boxes.

NI+ may include a data sender's address information object (DSInfo) if they are aware about the data sender. The DSInfo object is used by the edge-NAT to limit the possible NI addresses to one address. A NI+ can specify a specific IP address and port from where the subsequent NSIS signaling must be originated.

The REA signaling message creates NSIS NAT session state at any intermediate NSIS NAT peer(s) encountered. Furthermore it has to be ensured that the edge-NAT device is discovered as part of this process. The end host cannot be assumed to know this device - instead the NAT box itself is assumed to know that it is located at the outer perimeter of the private network addressing realm. Forwarding of the REA message beyond this entity is not necessary, and should be prohibited as it provides information on the capabilities of internal hosts.

The edge-NAT device responds to the REA message with a RESPONSE message containing a success object carrying the public reachable IP address/port number.

Processing of REA messages is specific to the NSIS node type:

o NSLP initiator: NI+ only generate REA messages and should never receive them. When the data sender's address information is known in advance the NI+ MAY include a DSInfo object in the REA message. When the data sender's IP address is not known, NI+s MUST NOT include a DSInfo object.

o NSLP forwarder: NSLP forwarders receiving REA messages MUST first check authentication and authorization before any further processing is executed. The NF SHOULD check with its local policies if it can accept the desired policy rule given by NTLP's message routing information (MRI). Further processing depends on the middlebox type:

  * NAT: NATs check whether the message is received at the external (public in most cases) address or at the internal (private) address. If received at the external address a NF MAY generate a RESPONSE message with an error of type 'REA received from outside'. If received at the internal address, an IP address/port is reserved. In the case it is an edge-NAT, the NSLP message is not forwarded any further and a RESPONSE message with the external address and port information is generated. If it is not an edge-NAT, the NSLP message is

forwarded further with the translated IP address/port. The edge-NAT MAY reject REA messages not carrying a DSInfo object or if the address information within this object is invalid or too much wildcarded.

- Firewall: Firewalls MUST not change their configuration upon a REA message. They simply MUST forward the message and MUST keep NTLP state. Firewalls that are configured as edge-Firewalls MAY return an error of type 'no NAT here'.

- Combined NAT and Firewall: Processing at combined Firewall and NAT middleboxes is the same as in the NAT case.

o NSLP receiver: This type of message should never be received by any NR+ and it SHOULD be discarded silently.

Processing of a RESPONSE message with an external address object is different for every NSIS node type:

o NSLP initiator: Upon receiving a RESPONSE message with an external address object, the NI+ can use the IP address and port pairs carried for further application signaling.

o NSLP forwarder: NFs simply forward this message as long as they keep state for the requested reservation.

o NSIS responder: This type of message should never be received by an NR and it SHOULD be discarded silently.

o Edge-NATs: This type of message should never be received by any Edge-NAT and it SHOULD be discarded silently.

Reservations made with REA MUST be enabled by a subsequent CREATE message. Without using CREATE (Section 3.3.1 or REA in proxy mode Section 3.3.7 no data traffic will be forwarded to DR beyond the edge-NAT. REA is just taking care about enabling the forwarding of subsequent CREATE messages traveling towards the NR. Correlation of incoming CREATE messages to REA reservation states is described in Section 3.6

EDITOR's note: How long REA session must be kept alive? REA sessions are not replaced by CREATE sessions but rather handled as all other sessions too, meaning they must be either teared down or left to be timeout.

3.3.3 NATFW Session refresh

NATFW NSLP sessions are maintained on a soft-state basis. After a

specified timeout, sessions and corresponding policy rules are removed automatically by the middlebox, if they are not refreshed. Soft-state is created by CREATE, REA, and UCREATE and the maintenance of this state must be done by these messages. State created by CREATE must be maintained by CREATE, state created by REA must be maintained by REA, and state created by UCREATE must be maintained by UCREATE. Refresh messages, either CREATE/REA/UCREATE, are messages carrying the exact MRI and session ID as the initial message and a lifetime object with a lifetime greater than zero. Every refresh request message MUST be acknowledged by an appropriate response message generated by the NR. This response message is routed back towards the NI, to allow the intermediate NFs to propose a refresh period that would align with their local policies. The NI sends refresh messages destined for the NR. Upon reception by each NSIS forwarder, the state for the given session ID is extended by the session refresh period, a period of time calculated based on a proposed refresh message period. The lifetime extension of a session is calculated as current local time plus proposed lifetime value (session refresh period). Section 3.4 defines the process of calculating lifetimes in detail.

```
   NI      Public Internet        NAT    Private address       NR
      |                              |          space             |
      | CREATE[lifetime > 0]         |                            |
      |----------------------------->|                            |
      |                              |                            |
      | RESPONSE[Error] (if needed)  |                            |
      |<-----------------------------|  CREATE[lifetime > 0]      |
      |                              |--------------------------->|
      |                              |                            |
      |                              |   RESPONSE[Success/Error]  |
      |   RESPONSE[Success/Error]    |<---------------------------|
      |<-----------------------------|                            |
      |                              |                            |
      |                              |                            |
```

Figure 16: State Refresh Message Flow, CREATE as example

Processing of session refresh CREATE/REA/UCREATE messages is different for every NSIS node type:

- o NSLP initiator: The NI can generate session refresh CREATE/REA/UCREATE messages before the session times out. The rate at which the refresh CREATE/REA/UCREATE messages are sent and their

relation to the session state lifetime are further discussed in Section 3.4. The message routing information and the extended flow information object MUST be set equal to the values of the initial request message.

- NSLP forwarder: NSLP forwarders receiving session refresh messages MUST first check authentication and authorization before any further processing is executed. The NF SHOULD check with its local policies if it can accept the desired lifetime extension for the session referred by the session ID. Processing of this message is independent of the middlebox type.

- NSLP responder: NRs accepting a session refresh CREATE/REA/UCREATE message generate a RESPONSE message with response object set to success. NRs MUST check for authorization and authentication.

3.3.4 Deleting Sessions

NATFW NSLP sessions may be deleted at any time. NSLP initiators can trigger this deletion by using a CREATE, REA, or UCREATE messages with a lifetime value set to 0, as shown in Figure 17.

```
      NI      Public Internet        NAT    Private address       NR
      |                              |          space             |
      |    CREATE[lifetime=0]        |                            |
      |----------------------------->|                            |
      |                              |                            |
      |                              | CREATE[lifetime=0]         |
      |                              |--------------------------->|
      |                              |                            |
```

Figure 17: Delete message flow, CREATE as example

NSLP nodes receiving this message MUST first check for authorization and authentication and afterwards MUST delete the session immediately. Policy rules associated with this particular session MUST be deleted immediately. This message is forwarded until it reaches the final NR. The CREATE/REA/UCREATE request message with a lifetime value of 0, does not generate any response, neither positive nor negative, since there is no NSIS state left at the nodes along the path.

3.3.5 Reporting Asynchronous Events

NATFW NSLP forwarders and NATFW NSLP responders must have the ability to report asynchronous events to other NATFW NSLP nodes, especially to allow reporting back to the NATFW NSLP initiator. Such asynchronous events may be premature session termination, changes in local policies, routing change or any other reason that indicates change of the NATFW NSLP session state. Currently, asynchronous session termination, re-authorization required and route change detected are the only events that are defined, but other events may be defined in later versions of this memo. One or several events could be reported within the NOTIFY message.

NFs and NRs may generate NOTIFY messages upon asynchronous events, with a response object indicating the reason of the event and a corresponding session ID. NOTIFY messages are sent hop-by-hop upstream towards NI until they reach NI.

Processing is different for every NATFW NSLP node type and depends on the notified events:

- NSLP initiator: NIs receiving NOTIFY messages MUST first check for authentication and authorization. After successfully doing so, NIs analyze the notified event(s) and behave appropriately based on the event type. Section 4.3.4 discusses the required behavior for each notified event. NIs MUST NOT generate NOTIFY messages.

- NSLP forwarder: NFs receiving NOTIFY messages MUST first check for authentication and authorization and MUST only accept NOTIFY messages from downstream peers. After successfully doing so, NFs analyze the notified event(s) and behave based on the notified events defined in Section 4.3.4. NFs occurring an asynchronous event generate NOTIFY messages and set the response object(s) code based on the reported event(s). NOTIFY messages are sent further hop-by-hop upstream towards the NI. NFs SHOULD generate NOTIFY messages upon asynchronous events and forward them upstream towards the NI.

- NSLP responder: NRs SHOULD generate NOTIFY messages upon asynchronous events. NRs receiving NOTIFY messages MUST ignore this message and discard it. NOTIFY messages are sent hop-by-hop upstream towards NI

3.3.6 Query and diagnosis capabilities within the NATFW NSLP protocol

The NATFW NSLP provides query and diagnosis capabilities that could be used by a session(s) owner to monitor the state of those sessions.

This would be used for:

- Diagnostic purposes when no data packets were received (or the packet stream is subject to significant packet loss) and NATFW NSLP signaling was supposed to have created appropriate policy rules on the NATFW NFs along the data path.

- Discover the number of NATFW NSLP Hops between the NI and the NR (or the last NATFW NE responding to the QDRQ)

- Collecting session states owned by a specific NI, this is required in case the NI loses its sessions' information (mainly due to node system issues).

The QDRQ message can be used to query and diagnose the following session information: session id, the number of NE hops (between the NI and the last NE responding to the QDRQ) and the following session's status ordered from best to worst: up, high traffic (used to detect DOS attack or unexpected traffic rate), pending, down. The status of the policy rule will probably provide sufficient diagnostic information in most cases;if more diagnostic information is required it could be provided by the NATFW NF logs. QDRQ messages may include an optional maximum hop count number value provided by the NI, when the hop count value reaches the maximum hop count the receiving NF should stop propagating the QDRQ and generate a response message to be sent back upstream to the NI. A QDRQ message usage is shown in Figure 18 where downstream NF increments the hop counter (except when they are the responding NF) and when the session's state is not "UP" (or "UP" but a QDRQ of type LIST is sent) they insert a session status value and their IP address. The Session information could be retrieved by sending a QDRQ against a specific session id or a QDRQ type equal to LIST (this is only applicable when the NI's identity is available and identical to the one used during the session's establishment process). In the message sequences shown in Figure 18, the QDRQ message (which a QDRQ type value of SINGLE) is sent for a single session ID (provided through the NTLP API), the traversed NAT didn't have any issues to report for the session however the Firewall's (FW) traffic meters reported that the flow has exceeded the maximum number of packets provisioned against the flow, hence in addition to the session status the firewall provides its address. As the firewall is the last hop (it is configured to proxy and respond to QDRQ messages) it does not increment the hop counter and responds hop by hop back to the NI.

```
      NI  Private address  NAT                     FW
      |                     |                        |
      |QDRQ(SINGLE,HOPCNT=0)|                        |
      |-------------------->|                        |
      |                     |QDRQ(SINGLE,HOPCNT=1)  |
      |                     |--------------------->  |
      |                     |                        |
      |                     |RESPONSE(SINGLE,        |
      |                     |HOPCNT=1,               |
      |                     | [HIGH_PPS,FW@]         |
      |                     |<---------------------  |
      |RESPONSE(SINGLE,     |                        |
      |HOPCNT=1,[HIGH_PPS,  |                        |
      | FW@])               |                        |
      |<--------------------|                        |
```

Figure 18: Query and Diagnosis operation

QDRQ message processing is dependent on the NATFW NSLP node type:

- NSLP initiator: NIs only generate QDRQ messages, while inserting:

  - a HOPCNT object with a zero value

  - a QDRQ type to indicate if the QDRQ is for a single session (QDRQ type would be SINGLE) or to gather information on all the sessions initiated by the NI (QDRQ type would be LIST)

  - When required (i.e. this is optional) a maximum hop count value

  - A SID (embedded in the Bound SIP object) when the QDRQ is not related to the session handled within the Message Routing State used to route the QDRQ.

  An NI MUST discard received QDRQ messages.

- NSLP forwarder: NFs receiving QDRQ messages MUST first authenticate and authorize the message source. After successfully doing so, NFs will behave differently depending if the QDRQ is specific to one session and whether the NF co-hosts a NAT engine or not.

  - If the QDRQ is about a single session:

    1. the NF checks first if the QDRQ includes a maximum hop count, if the current hop counter is smaller (else the procedures continues as defined in 2) and the NF was

previously able to forward NSLP messages downstream for the same session (else the procedure continues as defined in 2), the NF increments the hop counter. Furthermore if the NF's session status is not "UP", the NF will insert a session status object, which includes the session's status and the node's IP address, as defined in Section 4.3.11. In case the NF was co-hosting a NAT engine, the NF needs to ensure the validity of the session status object's embedded IP address and modify the address based on the local NAT bind entry. After completing these operations the NF forwards the message downstream.

2. In addition to the conditions discussed above, this procedure is applied when the QDRQ message is scoped by the receiving NF. The NF responds, with a RESPONSE message, hop by hop back to the NI while copying the hop counter, the bound session ID if it was present and a series of session status objects.

* If the QDRQ is of type LIST then the following procedures are applied:

  1. the NF checks first if the QDRQ includes a maximum hop counter, if the current hop counter is smaller (else the procedures continues as defined in 2), the NF creates one or several QDRQ response objects which include a bound session ID (session ID created by the NI before it lost all its session states), a flow descriptor and the session status object (session state and NF's IP address). This is only performed if the NF is able to get the NI's proof of ownership on stored sessions within the node. In case the NF was co-hosting a NAT engine, the NF needs to ensure the validity of all embedded IP addresses includes in QDRQ objects and modify the addresses based on the local NAT bind entry. After completing these operations the NF increments the hop counter and forwards the message downstream, if there were no downstream nodes then the hop counter is decremented and the procedure continues as described below in step 2.

  2. In addition to the conditions discussed above, this procedure is applied when the QDRQ message is scoped by the receiving NF. The NF responds, with a RESPONSE message, hop by hop back to the NI while copying the hop counter and the series of QDRQ response objects which would include in addition to the session status objects, bound session ID as discussed in Section 4.3.13.

- NSLP responder: NRs (any node being the destination of the message) receiving QDRQ messages MUST first check for authentication and authorization. After successfully doing so, NRs must process the message as the NFs when responding with a RESPONSE message to the NI. The RESPONSE message would include a copy of all the received objects within the QDRQ message. The RESPONSE message will travel along the established reverse path given by the message routing state.

Responses to QDRQ messages are processed differently depending on theNATFW NSLP node type:

- NSLP initiator: NIs receiving RESPONSEs to QDRQ messages MUST first check for authentication and authorization. After successfully doing so, the objects within the RESPONSE messages are provided up to the application layers and the session state remains as it was unless the application triggers NATFW NSLP state changes.

- NSLP forwarder: NFs receiving RESPONSEs to QDRQ messages MUST first check for authentication and authorization. After successfully doing so, NFs forward the message upstream without any interpretation.

- NSLP responder: if an NR receives a RESPONSE to QDRQ message it MUST discard it.

QDRQ messages are mainly sent for debugging and outage recovery and hence should be sent within a trusted network infrastructure, this could either be achieved by implicitly scoping QDRQ messages at the edge of the trusted network infrastructure or using the maximum hop count counter.

3.3.7 Proxy Mode for Data Receiver behind NAT

Some migration scenarios need specialized support to cope with cases where only the receiving side is running NSIS. End-to-end signaling is going to fail without NSIS support at both data sender and data receiver, unless the NATFW NSLP also gives the NR the ability to install state on the upstream path towards the data sender for downstream data packets. The goal of the described method is to trigger the network to generate a CREATE message at the edge-NAT on behalf of the data receiver. In this case, a NR can signal towards the Opportunistic Address as is performed in the standard REA message handling scenario for NATs as in Section 3.3.2. The message is forwarded until the edge-NAT is reached. A public IP address and port number is reserved at an edge-NAT. As shown in Figure 19, unlike the standard REA message handling case, the edge-NAT is

triggered to send a CREATE message on a new reverse path which traverse several firewalls or NATs. The new reverse path for CREATE is necessary to handle routing asymmetries between the edge-NAT and DR. This behavior requires an indication to the edge-NAT within the REA message if either the standard behavior (as defined in Section 3.3.2) is required or a CREATE message is required to be sent by the edge-NAT. In addition when a CREATE message needs to be sent by the edge-NAT, the REA message may include the data sender's address (DSInfo), if available to the data receiver. Figure 19 shows this proxy mode REA as REA-PROXY.

```
      DS       Public Internet       NAT     Private address      NR
     No NI                            NF         space            NI+
      NR+
      |                               |   REA-PROXY[(DSInfo)]     |
      |                               |<------------------------- |
      |                               |  RESPONSE[Error/Success]  |
      |                               | ---------------------- >  |
      |                               |   CREATE                  |
      |                               | ------------------------> |
      |                               |  RESPONSE[Error/Success]  |
      |                               | <----------------------   |
      |                               |                           |
      |                               |                           |
```

Figure 19: REA Triggering Sending of CREATE Message on Separate Reverse Path

The processing of REA-PROXY messages is different for every NSIS entity:

- NSLP initiator (NI+): When the data sender's address information is known in advance the NI+ MAY include a DSInfo object in the REA-PROXY request message. When the data sender's address is not known, NI+'s MUST NOT include a DSInfo object. NI+ only generate REA-PROXY messages and should never receive them.

- NSLP forwarder: NSLP forwarders receiving REA-PROXY messages MUST first check authentication and authorization before any further processing is executed. The NF SHOULD check with its local policies if it can accept the desired policy rule given by NTLP's message routing information (MRI). Further processing depends on

the middlebox type:

- * NAT: NATs check whether the message is received at the external (public in most cases) address or at the internal (private) address. If received at the external address a NF MAY generate a RESPONSE message with an error of type 'REA received from outside' and stop forwarding. If received at the internal address, an IP address/port is reserved. If it is not an edge-NAT, the NSLP message is forwarded further with the translated IP address/port. In the case it is an edge-NAT, the NSLP message is not forwarded any further. The edge-NAT checks whether it is willing to send CREATE messages on behalf on NI+ and if so it checks the DSInfo object. The edge-NAT MAY reject the REA-PROXY request if there is no DSInfo object or if the address information within DSInfo is not valid or too much wildcarded. If accepted a RESPONSE message with the external address and port information is generated. When the edge-NAT accepts it generates a CREATE message as defined in Section 3.3.1. The edge-NAT MUST not generate a CREATE-PROXY message. The edge-NAT MUST refresh the CREATE message session only if a REA-PROXY refresh message has been received first.

- * Firewall: Firewalls MUST not change their configuration upon a REA message. They simply MUST forward the message and MUST keep NTLP state. Edge-Firewalls SHOULD reply with an error RESPONSE indicating 'no egde-NAT here'.

- * Combined NAT and Firewall: Processing at combined Firewall and NAT middleboxes is the same as in the NAT case.

o NSLP receiver: This type of message should never be received by any NR+ and it SHOULD be discarded silently.

Processing of a RESPONSE message with an external address object is different for every NSIS node type:

o NSLP initiator: Upon receiving a RESPONSE message with an external address object, the NI+ can use the IP address and port pairs carried for further application signaling.

o NSLP forwarder: NFs simply forward this message as long as they keep state for the requested reservation.

o NSIS responder: This type of message should never be received by an NR and it SHOULD be discarded silently.

o Edge-NATs/edge-Firewall: This type of message should never be received by any Edge-NAT/edge-Firewall and it SHOULD be discarded

silently.

The scenario described in this chapter challenges the data receiver in a way that it must make a correct assumption about the data sender's ability to use NSIS NATFW NSLP signaling. There are two cases a) DS is NSIS unaware and DR assumes DS to be NSIS aware and b) DS is NSIS aware but DR assumes DS to be NSIS unaware. Case a) will result in middleboxes blocking the data traffic, since DS will never send the expected CREATE message. Case b) will result in the DR successfully requesting proxy mode support by the edge-NAT. The edge-NAT will send CREATE messages and DS will send CREATE messages too. Both CREATE messages are handled as separated sessions and therefore the common rules per session apply. It is up to the NR's responsibility to decide whether to teardown the REA-PROXY sessions in the case of the data sender's side being NSIS aware.

EDITOR's NOTE: The NI* for the REA session has currently no means to distinguish between a CREATE message initiated by the edge-NAT and a CREATE message initiated by the other end. In the case of using REA-PROXY and having a NSIS aware data sender though, the data receivers needs a handle to decide which CREATE session to teardown. There are multiple design choices:

The NI* includes a session ID proposal in its REA-PROXY message. The edge-NAT uses this proposed session ID in its CREATE signaling as session ID. This would violate the basic semantics of NIs randomly generating their session ID on their own.

The edge-NAT could include the to be used session ID of the CREATE session in its RESPONSE message to the REA-PROXY. This leads to the question of race conditions between the arrival time of the RESPONSE to REA-PROXY and the arrival time of the CREATE. More see below within next note.

The NI* could included another handle that is only inserted in the REA-PROXY message, but not in REA messages. The CREATE would need to include this handle. This handle could be either a number or in the simple case a flag indicating that the particular CREATE is an answer to REA-PROXY.

EDITOR's NOTE: The above note mentions a possible race condition between the RESPONSE message to the REA-PROXY and the CREATE message generated by the edge-NAT. The CREATE message could arrive earlier than the RESPONSE message. This is currently neither described nor semantically clean defined.

3.3.8 Proxy Mode for Data Sender behind Middleboxes

As with data receivers behind middleboxes in Section 3.3.7 also data senders behind middleboxes require proxy mode support as well. The problem here is that there is no NSIS support at the data receiver's side and, by default, there will be no response to CREATE request messages. This scenario requires the last NSIS NATFW NSLP aware node to terminate the forwarding and to proxy the response to the CREATE message, meaning that this node is generating RESPONSE messages. This last node may be an edge-NAT/edge-Firewall, or any other NATFW NSLP peer, that detects that there is no NR available (probably through GIMPS timeouts but not limited to). This proxy mode handles data senders behind a middlebox only; for receivers behind a NAT see Section 3.3.7.

NIs being aware about a NSIS unaware DR, send a CREATE message towards DR with a proxy support object. Intermediate NFs can use this additional information to decide whether to terminate the message forwarding or not. This proxy support object is an implicit scoping of the CREATE message. Termination of CREATE-PROXY request messages with proxy support object included MUST only be done by egde-NATs/edge-Firewalls; future revisions of this document may change this behavior.

```
   DS       Private Address       FW     Public Internet      NR
   NI           Space              NF                         no NR
   |                               |                           |
   |         CREATE-PROXY          |                           |
   |------------------------------>|                           |
   |                               |                           |
   |   RESPONSE[SUCCESS/ERROR]     |                           |
   |<------------------------------|                           |
   |                               |                           |
```

Figure 20: Proxy Mode Create Message Flow

The processing of CREATE-PROXY messages and RESPONSE messages is similar to Section 3.3.1, except that forwarding is stopped at the edge-NAT/edge-Firewall. The edge-NAT/edge-Firewall responds back to NI according the situation (error/success) and will be the NR for future NATFW NSLP communication.

3.3.9 Proxy Mode for Data Receiver behind Firewall

Data receivers behind firewalls would like to provide a similar sort of proxy mode operation to those behind NATs. While finding the upstream edge-NAT is quite easy, it is only required to find an edge-NAT but not a very specific one and then the data traffic is route pinned to the NAT, the location of the appropriate edge-Firewall is more difficult. Data receivers that are located behind several firewalls that are placed topology-wise in parallel (multi-homed network), must find out the one firewall the data traffic will traverse. This feature of locating the right firewall can be used for proxy mode support and for blocking certain incoming data traffic. Proxy mode support is similar to Section 3.3.7 where the DR is behind one or more NATs and installs "allow" policy rules. Blocking incoming data traffic requires that the NATFW NSLP locates the appropriate firewall in order to install a deny policy rule.

The upstream CREATE (UCREATE) message is used to locate upstream firewalls and to request installation of deny policy rules. The goal of the method described is to trigger the network to generate a CREATE message at the edge-Firewall on behalf of the data receiver. In this case, a NR can signal towards the data sender's address as in the standard REA message handling scenario for NATs Section 3.3.2. The message is forwarded until it reaches the edge-Firewall. As shown in Figure 21, the edge-Firewall is triggered to send a CREATE message on a new reverse path which could go through internal firewalls or NATs. The new reverse path for CREATE is necessary to handle routing asymmetries between the edge-Firewall and DR. UCREATE does not install any policy rule but the subsequent CREATE message initiated by the edge-Firewall does.

```
      DS       Public Internet       NAT     Private address      NR
     No NI                            NF         space            NI+
      NR+
      |                               |         UCREATE             |
      |                               |<--------------------------  |
      |                               |  RESPONSE[Error/Success]    |
      |                               | ---------------------- >    |
      |                               |   CREATE                    |
      |                               | ------------------------->  |
      |                               |  RESPONSE[Error/Success]    |
      |                               | <----------------------     |
      |                               |                             |
      |                               |                             |
```

Figure 21: UCREATE Triggering Sending of CREATE Message on Separate Reverse Path

The processing of UCREATE messages is different for every NSIS entity:

- NSLP initiator (NI+): NI+ MUST always direct UCREATE message to the address of DS. NI+ only generate UCREATE messages and should never receive them.

- NSLP forwarder: NSLP forwarders receiving UCREATE messages MUST first check authentication and authorization before any further processing is executed. The NF SHOULD check with its local policies if it can accept the desired policy rule given by NTLP's message routing information (MRI). Further processing depends on the middlebox type:

  * NAT: NATs check whether the message is received at the external (public in most cases) address or at the internal (private) address. If received at the internal interface, NATs allocated a public IP address and port and forward the message further. Edge-NATs receiving UCREATE SHOULD response with error RESPONSE indicating 'no edge-Firewall'

  * Firewall: Non edge-Firewalls simply forward the message. Edge-Firewalls stop forwarding the check for authentication and authorization. If the message is accepted, load the specified policy rule and generate CREATE messages back towards the DR as defined in Section 3.3.1.

  * Combined NAT and Firewall: Processing at combined Firewall and NAT middleboxes is the same as in the Firewall case.

- NSLP receiver: This type of message should never be received by any NR+ and it SHOULD be discarded silently.

Processing of a RESPONSE message with an external address object is different for every NSIS node type:

- NSLP initiator (NI+): Upon receiving a RESPONSE message NI+ should await incoming corresponding CREATE messages.

- NSLP forwarder: NFs simply forward this message as long as they keep state for the requested reservation.

- NSIS responder: This type of message should never be received by an NR and it SHOULD be discarded silently.

- o Edge-NATs/edge-Firewall: This type of message should never be received by any Edge-NAT/edge-Firewall and it SHOULD be discarded silently.

EDITOR's NOTE: The protocol behavior described within this section must be discussed at next IETF meeting.

3.4 Calculation of Session Lifetime

NATFW NSLP sessions, and the corresponding policy rules which may have been installed, are maintained via soft-state mechanism. Each session is assigned a lifetime and the session is kept alive as long as the lifetime is valid. After the expiration of the lifetime, sessions and policy rules MUST be removed automatically and resources bound to them should be freed as well. Session lifetime is kept at every NATFW NSLP node. The NSLP forwarders and NSLP responder are not responsible for triggering lifetime extension refresh messages (see Section 3.3.3): this is the task of the NSIS initiator.

The NSIS initiator MUST choose a session lifetime (expressed in seconds) value before sending any message (lifetime is set to zero for deleting sessions) to other NSLP nodes. The session lifetime value is calculated based on:

- o The number of lost refresh messages that NFs should cope with

- o The end to end delay between the NI and NR

- o Network vulnerability due to session hijacking ([6]). Session hijacking is made easier when the NI does not explicitly remove the session.

- o The user application's data exchange duration, in terms of seconds, minutes or hours and networking needs. This duration is modeled as M x R, with R the message refresh period (in seconds) and M a multiplier for R.

As opposed to the NTLP Message Routing state [1] lifetime, the NSLP session lifetime is not required to have a small value since the NSLP state refresh is not handling routing changes but security related concerns. [11] provides a good algorithm to calculate the session lifetime as well as how to avoid refresh message synchronization within the network. [11] recommends:

1. The refresh message timer to be randomly set to a value in the range [0.5R, 1.5R].

2. To avoid premature loss of state, L (with L being the session lifetime) must satisfy L >= (K + 0.5)*1.5*R, where K is a small integer. Then in the worst case, K-1 successive messages may be lost without state being deleted. Currently K = 3 is suggested as the default. However, it may be necessary to set a larger K value for hops with high loss rate. Other algorithms could be used to define the relation between the session lifetime and the refresh message period, the algorithm provided is only given as an example.

This requested lifetime value is placed in the 'lifetime' object of the NSLP message and messages are forwarded to the next NATFW NSLP node.

NATFW NFs processing the request message along the path MAY change the requested lifetime to fit their needs and/or local policy. If an NF changes the lifetime value it must also indicate the corresponding refresh message period. NFs MUST NOT increase the lifetime value; they MAY reject the requested lifetime immediately and MUST generate an error response message of type 'lifetime too big' upon rejection. The NSLP request message is forwarded until it reaches the NSLP responder. NSLP responder MAY reject the requested lifetime value and MUST generate an error response message of type 'lifetime too big' upon rejection. The NSLP responder MAY also lower the requested lifetime to an acceptable value (based on its local policies). NSLP responders generate their appropriate response message for the received request message, sets the lifetime value to the above granted lifetime and sends the message back hop-by-hop towards NSLP initiator.

Each NSLP forwarder processes the response message, reads and stores the granted lifetime value. The forwarders SHOULD accept the granted lifetime, as long as the value is within the tolerable lifetime range defined in their local policies. They MAY reject the lifetime and generate a 'lifetime not acceptable' error response message.
Figure 22 shows the procedure with an example, where an initiator requests 60 seconds lifetime in the CREATE message and the lifetime is shortened along the path by the forwarder to 20 seconds and by the responder to 15 seconds.

```
   +-------+ CREATE(lt=60s)  +-----------+ CREATE(lt=20s)  +---------+
   |       |---------------->|    NSLP   |---------------->|         |
   |  NI   |                 |           |                 |  NR     |
   |       |<----------------| forwarder |<----------------|         |
   +-------+ RESPONSE(lt=15s +-----------+ RESPONSE(lt=15s +---------+
                      MRR=3s)                        MRR=3s)

      lt  = lifetime
      MRR = Message Refresh Rate
```

Figure 22: Lifetime Calculation Example

3.5 Message Sequencing

NATFW NSLP messages need to carry an identifier so that all nodes along the path can distinguish messages sent at different points of time. Messages can be lost along the path, delayed, or duplicated. So all NATFW NSLP nodes should be able to identify either old messages that have been received before (duplicated), or the case that messages have been lost before (loss). Furthermore, for message replay protection it is necessary to keep information about already received messages. Therefore, it is required that every NATFW NSLP message MUST carry a message sequence number (MSN), see also Section 4.3.6.

The MSN MUST be only set by the NI and MUST no be set or modified by any other node. The initial value for the MSN MUST be generated randomly and MUST be only unique within the used session. The NI MUST increment the MSN for every message sent. Once the MSN has reached the maximum value, the next value it takes is zero.

NSIS forwarders and the responder store the MSN received with the initial packet as start value. NFs and NRs include the received MSN value in their response messages if generated.

When receiving a message, a NATFW NSLP node uses the MSN given in the message to determine whether the state being requested is different to the state already installed. The message MUST be discarded if the received MSN value is lower or equal than the stored MSN value. This received MSN value can indicate a delayed message or replayed message. If the received MSN value is greater than the already stored MSN value, the NATFW NSLP MUST update its stored state accordingly, if permitted by all security checks, and store the updated MSN value accordingly.

This semantics applied to a CREATE message exchange means that the first CREATE (initial CREATE to setup the path and session) carries the initial, randomly generated, MSN. All nodes along the path store this value and the responder includes the received value in its response (assuming that the CREATE message reaches the responder). Subsequent CREATE messages, updating the request policy rule or lifetime, carry an increment MSN value, so that intermediate nodes can note the requested update.

3.6 De-Multiplexing at NATs

Section 3.3.2 describes how NSIS nodes behind NATs can obtain a public reachable IP address and port number at a NAT and how it can be activated by using CREATE messages (see Section 3.3.1)". The information about the public IP address/port number can be transmitted via an application level signaling protocol and/or third party to the communication partner that would like to send data toward the host behind the NAT. However, NSIS signaling flows are sent towards the address of the NAT at which this particular IP address and port number is allocated and not directly to the allocated IP address and port number. The NATFW NSLP forwarder at this NAT needs to know how the incoming NSLP requests are related to reserved addresses, meaning how to de-multiplex incoming NSIS requests.

The de-multiplexing method uses information stored at NATs (such as mapping of public IP address to private, transport protocol, port numbers), information given by NTLP's message routing information and further authentication credentials.

3.7 Selecting Opportunistic Addresses for REA

As with all other message types, REA messages need a reachable final destination IP address. But as many applications do not provide a destination IP address in the first place, there is a need to choose a destination address for REA messages. This destination address can be the final target, but for applications which do not provide an upfront address, the destination address has to be chosen independently. Choosing the 'correct' destination IP address may be difficult and it is possible there is no 'right answer'. [15] shows choices for SIP and this section provides some hints about choosing a good destination IP address.

1. Public IP address of the data sender:

   * Assumption:

        - The data receiver already learned the IP address of the data sender (e.g., via a third party).

    - Problems:

        - The data sender might also be behind a NAT. In this case the public IP address of the data receiver is the IP address allocated at this NAT.

        - Due to routing asymmetry it might be possible that the routes taken by a) the data sender and the application server b) the data sender and NAT B might be different, this could happen in a network deployment such as in Figure 14. As a consequence it might be necessary to advertise a new (and different) external IP address within the application (which may or may not allow that) after using NSIS to establish a NAT binding.

2. Public IP address of the data receiver:

    - Assumption:

        - The data receiver already learned his externally visible IP address (e.g., based on the third party communication).

    - Problems:

        - Communication with a third party is required.

3. IP address of the Application Server:

    - Assumption:

        - An application server (or a different third party) is available.

    - Problems:

        - If the NSIS signaling message is not terminated at the NAT of the local network then an NSIS unaware application server might discard the message.

        - Routing might not be optimal since the route between a) the data receiver and the application server b) the data receiver and the data sender might be different.

4. NATFW NSLP Message Components

A NATFW NSLP message consists of a NSLP header and one or more objects following the header. The NSLP header is common for all NSLPs and objects are Type-Length-Value (TLV) encoded using big endian (network ordered) binary data representations. Header and objects are aligned to 32 bit boundaries and object lengths that are not multiples of 32 bits must be padded to the next higher 32 bit multiple.

The whole NSLP message is carried as payload of a NTLP message.

Note that the notation 0x is used to indicate hexadecimal numbers.

4.1 NSLP Header

The NSLP header is common to all NSLPs and is the first part of all NSLP messages. It contains two fields, the NSLP message type and a reserved field. The total length is 32 bits. The layout of the NSLP header is defined by Figure 23.

```
     0                               16                             31
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   NSLP message type           |       reserved                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 23: Common NSLP header

The reserved field MUST be set to zero in the NATFW NSLP header before sending and MUST be ignored during processing of the header. Note that other NSLPs use this field as a flag field.

4.2 NSLP message types

The message types identify requests and responses. Defined messages types are:

- 0x0101 : CREATE

- 0x0102 : RESERVE-EXTERNAL-ADDRESS(REA)

- 0x0104 : UCREATE

- 0x0108 : QDRQ

- 0x0201 : RESPONSE

- 0x0301 : NOTIFY

4.3 NSLP Objects

NATFW NSLP objects use a common header format defined by Figure 24. The object header contains two fields, the NSLP object type and the object length. Its total length is 32 bits.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |A|B|r|r|   Object Type         |r|r|r|r|   Object Length       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 24: Common NSLP object header

The length is the total length of the object without the object header. The unit is a word, consisting of 4 octets. The particular values of type and length for each NSLP object are listed in the subsequent sections that define the NSLP objects. The two leading bits of the NSLP object header are used to signal the desired treatment for objects whose treatment has not been defined in this memo (see [1], Section 3.2), i.e., the Object Type has not been defined. NATFW NSLP uses a subset of the categories defined in GIMPS:

- AB=00 ("Mandatory"): If the object is not understood, the entire message containing it must be rejected with an error indication.

- AB=01 ("Optional"): If the object is not understood, it should be deleted and then the rest of the message processed as usual.

- AB=10 ("Forward"): If the object is not understood, it should be retained unchanged in any message forwarded as a result of message processing, but not stored locally.

The combination AB=11 ("Refresh") MUST NOT be used since the NATFW NSLP refreshes its state end-to-end and not locally. Fields marked with 'r' are reserved for future use.

The following sections do not repeat the common NSLP object header, they just state the type and the length.

4.3.1 Session Lifetime Object

The session lifetime object carries the requested or granted lifetime of a NATFW NSLP session measured in seconds. The Message refresh rate value is set by default to 0xFFFF and only set to a specific value when an intermediate node changes the message lifetime and informs the upstream node about the recommended message refresh rate.

Type: NATFW_LT

Length: 1

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                  NATFW NSLP session lifetime                  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                  NATFW NSLP message refresh rate              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 25: Lifetime object

4.3.2 External Address Object

The external address object can be included in RESPONSE messages (Section 4.4.3) only.

Type: NATFW_EXT_IPv4

Length: 2

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |         port number           |           reserved            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                           IPv4 address                        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 26: External Address Object for IPv4 addresses

Type: NATFW_EXT_IPv6

Length: 5

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |         port number           |          reserved             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     +                                                               +
     |                                                               |
     +                          IPv6 address                         +
     |                                                               |
     +                                                               +
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 27: External Address Object for IPv6 addresses

Please note that the field 'port number' MUST be set to 0 if only an IP address has been reserved, for instance, by a traditional NAT. A port number of 0 MUST be ignored in processing this object.

4.3.3 Extended Flow Information Object

In general, flow information is kept at the NTLP level during signaling. The message routing information of the NTLP carries all necessary information. Nevertheless, some additional information may be required for NSLP operations. The 'extended flow information' object carries this additional information about action to be taken on the installed policy rules.

Type: NATFW_EXT_FLOW

Length: 1

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |           rule action         |       reserved                |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 28: Extended Flow Information

These fields are defined for the policy rule object:

- o Rule action: This field indicates the action for the policy rule to be activated. Allowed values are 'allow' (0x01) and 'deny' (0x02)

4.3.4 Response Code Object

This object carries the response code, which may be indications for either a successful request or failed request depending on the value of the 'response code' field.

Type: NATFW_RESPONSE

Length: 1

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         response code                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 29: Response Code Object

TBD: Define response classes, success codes and error codes. Possible error classes are:

- o Policy rule errors
- o Authentication and Authorization errors
- o NAT

Currently errors defined in this memo are:

- o lifetime too big
- o lifetime not acceptable
- o no NAT here
- o no reservation found
- o requested external address from outside

- re-authorization needed

- routing change detected

4.3.5 Proxy Support Type Object

This object indicates that proxy mode support is required. Either in a REA message or CREATE message.

Type: NATFW_RESP_TYPE

Length: 0, only object header

EDITOR's NOTE: This is quite a short object and probably better moved to a flag somewhere.

4.3.6 Message Sequence Number Object

This object carries the MSN value as described in Section 3.5.

Type: NATFW_RESP_MSN

Length: 1

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    message sequence number                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 30: Message Sequence Number Object

4.3.7 Bound Session ID Object

This object carries a session ID and is used for QUERY messages only.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       bound session ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 31: Bound Session ID Object

This object is used when a session owner queries multiple session, every session would be indicated with the bound session ID object.

4.3.8 Data Sender Information Object

Type: NATFW_DSINFO_IPv4

Length: 1

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |         port number           |           reserved            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                           IPv4 address                        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 32: Data Sender's IPv4 Address Object

Type: NATFW_DSINFO_IPv6

Length: 1

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |         port number           |          reserved             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     +                                                               +
     |                                                               |
     +                          IPv6 address                         +
     |                                                               |
     +                                                               +
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 33: ata Sender's IPv6 Address Object for IPv6 addresses

4.3.9 NATFW NF Hop Count Object

Type: NATFW_NF_HOPCNT

Length: 1

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                     NATFW NF HOP COUNT                        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 34: NATFW NF Hop Count Object

Editor note next revision will include Hop count, maximum hops and QDRQ type in the same object to minimize the overhead since 4 bits would be sufficient for the counters.

4.3.10 Maximum Hops Object

Type: NATFW_NF_MAX_HOPCNT

Length: 1

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                     NATFW NF MAX HOP COUNT                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 35: NATFW NF Maximum Hop Count Object

4.3.11 Session Status object

The session status object is inserted within the QDRQ message and copied in a response message. It embeds the current local node's session status and the node's IP address

Type: SESSION_STS

Length: 2 or 5

SESSION STATUS:

Length 1, Possible values: UP(0),HIGH_PPS(1), PENDING(2), DOWN(3)

Reserved bits: RRR

Length 3 bits

IP version: V bit

Length 1 bit: IPv4(0), IPv6(1)

NODE IP ADDRESS:

Length 1 or 4

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    SESSION STATUS                     |R|R|R|V|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                    NODE IP ADDRESS (1 or 4 words)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 36: Session Status Object

4.3.12 QDRQ type

Type: QDRQ_TYPE

Length: 1

Possible values: SINGLE(0), LIST(1)

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                           QDRQ TYPE                           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 37: QDRQ TYPE object

4.3.13 QDRQ Response object

TBC for next version: includes an optional bound session id, an

optional flow descriptor (used when a LIST QDRQ type is used) and a mandatory session status

4.4 Message Formats

This section defines the content of each NATFW NSLP message type. The message types are defined in Section 4.2. First, the request messages are defined with their respective objects to be included in the message. Second, the response messages are defined with their respective objects to be included.

Basically, each message is constructed of NSLP header and one or more NSLP objects. The order of objects is not defined, meaning that objects may occur in any sequence. Objects are marked either with mandatory [M] or optional [O]. Where [M] implies that this particular object MUST be included within the message and where [O] implies that this particular object is OPTIONAL within the message.

Each section elaborates the required settings and parameters to be set by the NSLP for the NTLP, for instance, how the message routing information is set.

4.4.1 CREATE

The CREATE request message is used to create NSLP sessions and to create policy rules. Furthermore, CREATE messages are used to refresh sessions and to delete them.

The CREATE message carries these objects:

- Lifetime object [M]

- Extended flow information object [M]

- Message sequence number object [M]

- Proxy support object [O]

The message routing information in the NTLP MUST be set to DS as source address and DR as destination address. All other parameters MUST be set according the required policy rule.

4.4.2 RESERVE-EXTERNAL-ADDRESS (REA)

The RESERVE-EXTERNAL-ADDRESS (REA) request message is used to target a NAT and to allocated an external IP address and possibly port number, so that the initiator of the REA request has a public reachable IP address/port number.

The REA request message carries these objects:

- o Lifetime object [M]

- o Message sequence number object [M]

- o Extended flow information object [M]

- o Proxy support object [O]

- o Data sender information object [O]

The REA message needs special NTLP treatment. First of all, REA messages travel the wrong way, from the DR towards DS. Second, the DS' address used during the signaling may be not the actual DS (see Section 3.7). Therefore, the NTLP flow routing information is set to DR as initiator and DS as responders, a special field is given in the NTLP: The signaling destination.

4.4.3 RESPONSE

RESPONSE messages are responses to CREATE, REA, UCREATE, and QUERY messages.

The RESPONSE message carries these objects:

- o Lifetime object [M]

- o Message sequence number object [M]

- o Response code object [M]

- o External address object [O]([M] for success responses to REA)

This message is routed upstream.

EDITOR's note: Text says that this section is defining the behavior depending on the response type.

4.4.4 QDRQ

QDRQ messages are used for query and diagnosis purposes.

The QDRQ message carries these objects:

- o Response object [M]

- o Message sequence number object [M]

- o NATFW NSLP Hop Count [M]

- o Maximum Hop Count value [O]

- o Bound session ID [O]

- o Session status [O]

- o QDRQ type [O]

This message is routed downstream.

4.4.5 NOTIFY

The NOTIFY messages is used to report asynchronous events happening along the signaled path to other NATFW NSLP nodes.

The NOTIFY message carries this object:

- o Response code object with NOTIFY code [M].

The message routing information in the NTLP MUST be set with the NI's address being the destination address and the node's address as source address. The message is forwarded upstream hop by hop using the existing upstream node address entry within the node's Message Routing State table. The session id object must be set to the corresponding session that is effected by this asynchronous event.

4.4.6 UCREATE

TBD: XYX.

5. NATFW NSLP NTLP Requirements

The NATFW NSLP requires the following capabilities from the NTLP:

- Ability to detect that the NSIS Responder does not support NATFW NSLP. This capability is key to launching the proxy mode behavior as described in Section 3.3.7 and [13].

- Detection of NATs and their support of the NSIS NATFW NSLP. If the NTLP discovers that the NSIS host is behind an NSIS aware NAT, the NR will send REA messages to the opportunistic address. If the NTLP discovers that the NSIS host is behind a NAT that does not support NSIS then the NSIS host will need to use a separate NAT traversal mechanism.

- Message origin authentication and message integrity protection

- Detection of routing changes

- Protection against malicious announcement of fake path changes, this is needed to mitigate a threat discussed in Section 7 of [6]

6. NSIS NAT and Firewall Transition Issues

NSIS NAT and Firewall transition issues are premature and will be addressed in a separate draft (see [13]). An update of this section will be based on consensus.

7. Security Considerations

Security is of major concern particularly in case of Firewall traversal. This section provides security considerations for the NAT/Firewall traversal and is organized as follows:

Section 7.1 describes the framework assumptions with regard to the assumed trust relationships between the participating entities. This subsection also motivates a particular authorization model.

Security threats that focus on NSIS in general are described in [6] and they are applicable to this document. Within Section 7.2 we extend this threat investigation by considering NATFW NSLP specific threats. Based on the security threats we list security requirements.

Finally we illustrate how the security requirements that were created based on the security threats can be fullfilled by specific security mechanisms. These aspects will be elaborated in Section 7.3.

7.1 Trust Relationship and Authorization

The NATFW NSLP is a protocol which may involve a number of NSIS nodes and is, as such, not a two-party protocol. This fact requires more thoughts about scenarios, trust relationships and authorization mechanisms. Trust relationships and authorization are very important for the protocol machinery and they are closely related to each other in the sense that a certain degree of trust is required to authorize a particular action. For any action (e.g. create/delete pinholes), authorization is very important due to the nature of middleboxes. More problematic scenarios are described in Appendix B.

Different types of trust relationships may affect different categories of middleboxes. As explained in [21], establishment of a financial relationship is typically very important for QoS signaling, whereas financial relationships are less directly of interest for NATFW middlebox signaling. It is therefore not particularly surprising that there are differences in the nature and level of authorization likely to be required in a QoS signaling environment and in NATFW middlebox signaling. Typically NATFW signaling requires authorization to configure firewalls or to modify NAT bindings. The outcome of the authorization is either allowed or disallowed whereas QoS signaling might just indicate that a lower QoS reservation is allowed.

Different trust relationships that appear in middlebox signaling environments are described in the subsequent sub-sections. As a comparison with other NSIS signaling application it might be

interesting to mention that QoS signaling relies on peer-to-peer trust relationships and authorization between neighboring nodes or neighboring networks. These type of trust relationships turn out to be simpler for a protocol. However, there are reasons to believe that this is not the only type of trust relationship found in today's networks.

7.1.1 Peer-to-Peer Trust Relationship

Starting with the simplest scenario, it is assumed that neighboring nodes trust each other. The required security association to authenticate and to protect a signaling message is either available (after manual configuration), or has been dynamically established with the help of an authentication and key exchange protocol. If nodes are located closely together, it is assumed that security association establishment is easier than establishing it between distant nodes. It is, however, difficult to describe this relationship generally due to the different usage scenarios and environments. Authorization heavily depends on the participating entities, but for this scenario, it is assumed that neighboring entities trust each other (at least for the purpose of policy rule creation, maintenance, and deletion). Note that Figure 38 does not illustrate the trust relationship between the end host and the access network.

```
   +------------------------+              +-------------------------+
   |Network A               |              |                Network B|
   |              +---------+              +---------+               |
   |        +-///-+ Middle- +---///////----+ Middle- +-///-+         |
   |        |     |  box 1  |   Trust      |  box 2  |     |         |
   |        |     +---------+ Relationship +---------+     |         |
   |        |   Trust       |              |      Trust    |         |
   |        | Relationship  |              |  Relationship |         |
   |        |               |              |               |         |
   |     +--+---+           |              |            +--+---+     |
   |     | Host |           |              |            | Host |     |
   |     |  A   |           |              |            |  B   |     |
   |     +------+           |              |            +------+     |
   +------------------------+              +-------------------------+
```

Figure 38: Peer-to-Peer Trust Relationship

7.1.2 Intra-Domain Trust Relationship

In larger corporations, often more than one middlebox is used to protect or serve different departments. In many cases, the entire enterprise is controlled by a security department, which gives

instructions to the department administrators. In such a scenario, a peer-to-peer trust-relationship might be prevalent. Sometimes it might be necessary to preserve authentication and authorization information within the network. As a possible solution, a centralized approach could be used, whereby an interaction between the individual middleboxes and a central entity (for example a policy decision point - PDP) takes place. As an alternative, individual middleboxes could exchange the authorization decision with another middlebox within the same trust domain. Individual middleboxes within an administrative domain should exploit their trust relationship instead of requesting authentication and authorization of the signaling initiator again and again. Thereby complex protocol interactions are avoided. This provides both a performance improvement without a security disadvantage since a single administrative domain can be seen as a single entity. Figure 39 illustrates a network structure which uses a centralized entity.

```
    +-----------------------------------------------------------+
    |                                               Network A   |
    |                     +---------+                +---------+
    |      +----///--------+ Middle- +------///------++ Middle- +---
    |      |               |  box 2  |                |  box 2  |
    |      |               +----+----+                +----+----+
    | +----+----+               |                          |    |
    | | Middle- +---------+     +---------+                |    |
    | |  box 1  |         |               |                |    |
    | +----+----+         |               |                |    |
    |      |              |          +----+-----+          |    |
    |      |              |          | Policy   |          |    |
    |   +--+---+          +----------+ Decision +----------+    |
    |   | Host |                     | Point    |               |
    |   |  A   |                     +----------+               |
    |   +------+                                                |
    +-----------------------------------------------------------+
```

Figure 39: Intra-domain Trust Relationship

7.1.3 End-to-Middle Trust Relationship

In some scenarios, a simple peer-to-peer trust relationship between participating nodes is not sufficient. Network B might require additional authorization of the signaling message initiator. If authentication and authorization information is not attached to the initial signaling message then the signaling message arriving at Middlebox 2 would result in an error message being created, which indicates the additional authorization requirement. In many cases the signaling message initiator is already aware of the additionally

required authorization before the signaling message exchange is executed. Replay protection is a requirement for authentication to the non-neighboring middlebox, which might be difficult to accomplish without adding additional roundtrips to the signaling protocol (e.g., by adding a challenge/response type of message exchange).

Figure 40 shows the slightly more complex trust relationships in this scenario.

```
    +--------------------+              +---------------------+
    |          Network A | Trust        |Network B            |
    |                    | Relationship |                     |
    |          +---------+              +---------+           |
    |    +-///-+ Middle- +---///////----+ Middle- +-///-+     |
    |    |     |  box 1  |      +-------+  box 2  |     |     |
    |    |     +---------+      |       +---------+     |     |
    |    |Trust          |      |       |   Trust       |     |
    |    |Relationship   |      |       |   Relationship|     |
    |    |               |      |       |               |     |
    | +--+---+           |      |       |            +--+---+ |
    | | Host +----///----+------+       |            | Host | |
    | |  A   |           |Trust         |            |  B   | |
    | +------+           |Relationship  |            +------+ |
    +--------------------+              +---------------------+
```

Figure 40: End-to-Middle Trust Relationship

7.2 Security Threats and Requirements

This section describes NATFW specific security threats and requirements.

7.2.1 Attacks related to authentication and authorization

The NSIS message which installs policy rules at a middlebox is the CREATE message. The CREATE message travels from the Data Sender (DS) toward the Data Receiver (DR). The packet filter or NAT binding is marked as pending by the middleboxes along the path. If it is confirmed with a success RESPONSE message from the DR, the requested policy rules on the middleboxes are installed to allow the traversal of a data flow.

```
    +-----+               +-----+               +-----+
    | DS  |               | MB  |               | DR  |
    +-----+               +-----+               +-----+
       |                     |                     |
       |       CREATE        | CREATE              |
       |-------------------->+-------------------->|
       |                     |                     |
       |   Succeeded/Error   |   Succeeded/Error   |
       |<--------------------+<--------------------|
       |                     |                     |
        ==========================================>
                      Direction of data traffic
```

Figure 41: CREATE Mode

In this section we will consider some simple scenarios for middlebox configuration:

o Data Sender (DS) behind a firewall

o Data Sender (DS) behind a NAT

o Data Receiver (DR) behind a firewall

o Data Receiver (DR) behind a NAT

A real-world scenario could include a combination of these firewall/ NAT placements, such as, a DS and/or a DR behind a chain of NATs and firewalls.

Figure 42 shows one possible scenario:

```
        +-------------------+                  +--------------------+
        |    Network A      |                  |       Network B    |
        |                   |                  |                    |
        |    +-----+        |    //-----\\     |        +-----+     |
        |    | MB2 |--------+----|  INET  |----+--------| MB3 |     |
        |    +-----+        |    \\-----//     |        +-----+     |
        |       |           |                  |           |        |
        |    +-----+        |                  |        +-----+     |
        |    | MB1 |        |                  |        | MB4 |     |
        |    +-----+        |                  |        +-----+     |
        |       |           |                  |           |        |
        |    +-----+        |                  |        +-----+     |
        |    | DS  |        |                  |        | DR  |     |
        |    +-----+        |                  |        +-----+     |
        +-------------------+                  +--------------------+

        MB: Middle box (NAT or Firewall or a combination)
        DS: Data Sender
        DR: Data Receiver
```

Figure 42: Several middleboxes per network

7.2.1.1 Data Sender (DS) behind a firewall

```
                   +------------------------------+
                   |                              |
                   |   +-----+     create      +-----+
                   |   | DS  | --------------> | FW  |
                   |   +-----+                 +-----+
                   |                              |
                   +------------------------------+
```

DS sends a CREATE message to request the traversal of a data flow.

It is up to network operators to decide how far they can trust users inside their networks. However, there are several reasons why they should not.

The following attacks are possible:

- o DS could open a firewall pinhole with a source address different from its own host.

- o DS could open firewall pinholes for incoming data flows that are not supposed to enter the network.

- o DS could request installation of any policy rules and allow all traffic go through.

SECURITY REQUIREMENT: The middlebox MUST authenticate and authorize the neighboring NAT/FW NSLP node which requests an action. Authentication and authorization of the initiator SHOULD be provided to NATs and Firewalls along the path.

7.2.1.2 Data Sender (DS) behind a NAT

The case 'DS behind a NAT' is analogous to the case 'DS behind a firewall'.

Figure 44 illustrates such a scenario:

```
                   +-------------------------------+
                   |                               |
                   |   +------+     CREATE         |
                   |   | NI_1 | ------\         +-----+ CREATE  +-----+
                   |   +------+        \------> | NAT |-------->| MB  |
                   |                            +-----+         +-----+
                   |   +------+                    |
                   |   | NI_2 |                    |
                   |   +------+                    |
                   +-------------------------------+
```

Figure 44: Several NIs behind a NAT

In this case the middlebox MB does not know who is the NSIS Initiator since both NI_1 and NI_2 are behind a NAT (which is also NSIS aware). Authentication needs to be provided by other means such as the NSLP or the application layer.

SECURITY REQUIREMENT: The middlebox MUST authenticate and ensure that the neighboring NAT/FW NSLP node is authorized to request an action. Authentication and authorization of the initiator (which is the DR in this scenario) to the middleboxes (via another NSIS aware middlebox) SHOULD be provided.

7.2.1.3 Data Receiver (DR) behind a firewall

In this case a CREATE message comes from an entity DS outside the network towards the DR inside the network.

```
                                 +------------------------------+
                                 |                              |
       +-----+    CREATE      +-----+    CREATE      +-----+    |
       | DS  | -------------> | FW  | -------------> | DR  |    |
       +-----+ <------------- +-----+ <------------- +-----+    |
               success RESPONSE  |     success RESPONSE         |
                                 |                              |
                                 +------------------------------+
```

Since policy rules at middleboxes must only be installed after receiving a successful response it is necessary that the middlebox waits until the Data Receiver DR confirms the request of the Data Sender DS with a success RESPONSE message. This is, however, only necessary

- if the action requested with the CREATE message cannot be authorized and

- if the middlebox is still forwarding the signaling message towards the end host (without state creation/deletion/modification).

This confirmation implies that the data receiver is expecting the data flow.

At this point we differentiate two cases:

1. DR knows the IP address of the DS (for instance because of some previous application layer signaling) and is expecting the data flow.

2. DR might be expecting the data flow (for instance because of some previous application layer signaling) but does not know the IP address of the Data Sender DS.

For the second case, Figure 46 illustrates a possible attack: an adversary Mallory M could be sniffing the application layer signaling and thus knows the address and port number where DR is expecting the data flow. Thus it could pretend to be DS and send a CREATE message towards DR with the data flow description (M -> DR). Since DR does not know the IP address of DS, it is not able to recognize that the request is coming from the "wrong guy". It will send a success RESPONSE message back and the middlebox will install policy rules that will allow Mallory M to inject its data into the network.

```
                    Application Layer signaling
              <------------------------------------>
             /                                      \
            /                      +-----------------\------------+
           /                       |                  \           |
       +------+                +------+               +------+   |
       | DS   |             -> | FW   |               | DR   |   |
       +------+            /    +------+               +------+   |
                  CREATE  /       |                               |
       +------+          /        +-------------------------------+
       | M    |----------
       +------+
```

Figure 46: DR behind a firewall with an adversary

Network administrators will probably not rely on a DR to check the IP address of the DS. Thus we have to assume the worst case with an attack such as in Figure 46. Many operators might not allow NSIS signaling message to traverse the firewall in Figure 46 without proper authorization. In this case the threat is not applicable.

SECURITY REQUIREMENT: A binding between the application layer and the NSIS signaling SHOULD be provided.

7.2.1.4 Data Receiver (DR) behind a NAT

When a data receiver DR behind a NAT sends a RESERVE-EXTERNAL-ADDRESS (REA) message to get a public reachable address that can be used as a contact address by an arbitrary data sender if the DR was unable to restrict the future data sender. The NAT reserves an external address and port number and sends them back to DR. The NAT adds an address mapping entry in its reservation list which links the public and private addresses as follows:

(DR_ext <=> DR_int) (*).

The NAT sends a RESPONSE message with the external address' object back to the DR with the address DR_ext. DR informs DS about the public address that it has recently received, for instance, by means of application layer signaling.

When a data sender sends a CREATE message towards DR_ext then the message will be forwarded to the DR. The data sender might want to update the NAT binding stored at the edge-NAT to make it more restrictive.

We assume that the adversary Mallory M obtains the contact address

(i.e., external address and port) allocated at the NAT possibly by eavesdropping on the application layer signaling and sends a CREATE message. As a consequence Mallory would be able to communicate with DR (if M is authorized by the edge-NAT and if the DR accepts CREATE and returns a RESPONSE.

```
                    Application Layer signaling
              <------------------------------------->
             /                                      \
            /                      +-----------------\------------+
           /                       |       REA        \           |
       +-----+                  +-----+  <-----------  +-----+    |
       | DS  |              ->  | NAT |  ----------->  | DR  |    |
       +-----+             /    +-----+  rtn_ext_addr  +-----+    |
                  CREATE  /       |                               |
       +-----+           /        +-------------------------------+
       | M   |----------
       +-----+
```

SECURITY REQUIREMENT: The DR MUST be able to specify which data sender are allowed to traverse the NAT in order to be forwarded to DRs address.

7.2.1.5 NSLP Message Injection

Malicious hosts, located either off-path or on-path, could inject arbitrary NATFW NSLP messages into the signaling path, causing several problems. These problems apply when no proper authorization and authentication scheme is available.

By injecting a bogus CREATE message with lifetime set to zero, a malicious host could try to teardown NATFW NSLP session state partially or completely on a data path, causing a service interruption.

By injecting a bogus responses or NOTIFY message, for instance, timeout, a malicious host could try to teardown NATFW NSLP session state as well. This could affect the data path partially or totally, causing a service interruption.

SECURITY REQUIREMENT: Messages, such as TRIGGER, can be misused by malicious hosts, and therefore need to be authorized.

7.2.2 Denial-of-Service Attacks

In this section we describe several ways how an adversary could launch a Denial of service (DoS) attack on networks running NSIS for middlebox configuration to exhaust their resources.

7.2.2.1 Flooding with CREATE messages from outside

7.2.2.1.1 Attacks due to NSLP state

A CREATE message requests the NSLP to store state information such as a NAT binding or a policy rule.

The policy rules requested in the CREATE message will be installed at the arrival of a confirmation from the Data Receiver with a success RESPONSE message. A successful RESPONSE message includes the session ID. So the NSLP looks up the NSIS session and installs the requested policy rules.

An adversary from outside could launch a DoS attack with arbitrary CREATE messages. For each of these messages the middlebox needs to store state information such as the policy rules to be loaded, i.e., the middlebox could run out of memory. This kind of attack is also mentioned in [6] Section 4.8.

SECURITY REQUIREMENT: A NAT/FW NSLP node MUST authorize the 'create-session' message before storing state information.

7.2.2.1.2 Attacks due to authentication complexity

This kind of attack is possible if authentication is based on mechanisms that require computing power, for example, digital signatures.

For a more detailed treatment of this kind of attack, the reader is encouraged to see [6].

SECURITY REQUIREMENT: A NAT/FW NSLP node MUST NOT introduce new denial of service attacks based on authentication or key management mechanisms.

7.2.2.1.3 Attacks to the endpoints

The NATFW NSLP requires firewalls to forward NSLP messages, a malicious node may keep sending NSLP messages to a target. This may consume the access network resources of the victim, drain the battery

of the victim's terminal and may force the victim to pay for the received although undesired data.

This threat may be more particularly be relevant in networks where access link is a limited resource, for instance in cellular networks, and where the terminal capacities are limited.

SECURITY REQUIREMENT: A NATFW NSLP aware firewall or NAT MUST be able to block unauthorized signaling message, if this threat is a concern.

7.2.2.2 Flooding with REA messages from inside

Although we are more concerned with possible attacks from outside the network, we need also to consider possible attacks from inside the network.

An adversary inside the network could send arbitrary RESERVE-EXTERNAL-ADDRESS messages. At a certain point the NAT will run out of port numbers and the access for other users to the outside will be disabled.

SECURITY REQUIREMENT: The NAT/FW NSLP node MUST authorize state creation for the RESERVE-EXTERNAL-ADDRESS message. Furthermore, the NAT/FW NSLP implementation MUST prevent denial of service attacks involving the allocation of an arbitrary number of NAT bindings or the installation of a large number of packet filters.

7.2.3 Man-in-the-Middle Attacks

Figure 48 illustrates a possible man-in-the-middle attack using the RESERVE-EXTERNAL-ADDRESS (REA) message. This message travels from DR towards the public Internet. The message might not be intercepted because there are no NSIS aware middleboxes.

Imagine such an NSIS signaling message is then intercepted by an adversary Mallory (M). M returns a faked RESPONSE message whereby the adversary pretends that a NAT binding was created. This NAT binding is returned with the RESPONSE message. Malory might insert it own IP address in the response, the IP address of a third party or the address of a black hole. In the first case, the DR thinks that the address of Mallory M is its public address and will inform the DS about it. As a consequence, the DS will send the data traffic to Mallory M.

The data traffic from the DS to the DR will re-directed to Mallory M.

M will be able to read, modify or block the data traffic (if the end-to-end communication itself does not experience protection). Eavesdropping and modification is only possible if the data traffic is itself unprotected.

```
     +-----+          +-----+               +-----+
     | DS  |          |  M  |               | DR  |
     +-----+          +-----+               +-----+
        |                |                     |
        |                |       REA           |
        |                | <------------------ |
        |                |                     |
        |                |      RESPONSE       |
        |                | ------------------> |
        |                |                     |
        |  data traffic  |                     |
        |===============>|        data traffic |
        |                |====================>|
```

Figure 48: MITM attack using the RESERVE-EXTERNAL-ADDRESS message

SECURITY REQUIREMENT: Mutual authentication between neighboring NATFW NSLP MUST be provided. To ensure that only legitimate nodes along the path act as NSIS entities the initiator MUST authorize the responder. In the example in Figure 48 the firewall FW must perform an authorization with the neighboring entities.

7.2.4 Message Modification by non-NSIS on-path node

An unauthorized on-path node along the path towards the destination could easily modify, inject or just drop an NSIS message. It could also hijack or disrupt the communication.

SECURITY REQUIREMENT: Message integrity, replay protection and data origin authentication between neighboring NAT/FW NSLPs MUST be provided.

7.2.5 Message Modification by malicous NSIS node

Message modification by a NSIS node that became malicious is more serious. An adversary could easily create arbitrary pinholes or NAT bindigs. For example:

- NATs need to modify the source/destination of the data flow in the 'create session' message.

- Each middlebox along the path may change the requested lifetime in the CREATE message to fit their needs and/or local policy.

SECURITY REQUIREMENT: None. Malicous NSIS NATs and Firewalls will not be addressed.

7.2.6 Session Modification/Deletion

The Session ID is included in signaling messages as a reference to the established state. If an adversary is able to obtain the Session Identifier for example by eavesdropping on signaling messages, it would be able to add the same Session Identifier to a new signaling message and effect some modifications.

Consider the scenario described in Figure 49. Here an adversary pretends to be 'DS in mobility'. The signaling messages start from the DS and go through a series of routers towards the DR. We assume that an off-path adversary is connected to one of the routers along the old path (here Router 3). We also assume that the adversary knows the Session ID of the NSIS session initiated by the DS. Knowing the Session ID, the adversary now sends signalling messages towards the DR. When the signaling message reaches Router3 then existing state information can be modified or even deleted. The adversary can modify or delete the established reservation causing unexpected behavior for the legitimate user. The source of the problem is that the Router 3 (cross-over router) is unable to decide whether the new signaling message was initiated from the owner of the session. In this scenario, the adversary need not even be located in the DS-DR path. This problem and the solution approaches are described in more detail in [23].

```
                                   Session ID(SID-x)
                                          +--------+         +--------+
                        +-------->--------+ Router +-------->+   DR   |
       Session ID(SID-x)|                 |   4    |         |        |
                    +---+----+            +--------+         +--------+
                    | Router |
             +------+   3    +*******
             |      +---+----+      *
             |                      *
             | Session ID(SID-x)    * Session ID(SID-x)
         +---+----+             +---+----+
         | Access |             | Access |
         | Router |             | Router |
         |   1    |             |   2    |
         +---+----+             +---+----+
             |                      *
             | Session ID(SID-x)    * Session ID(SID-x)
        +----+------+          +----+------+
        |    DS     |          | Adversary |
        |           |          |           |
        +-----------+          +-----------+
```

Figure 49: State Modification by off-path adversary

As a summary, an off-path adversary's knowledge of Session-ID could cause session modification/deletion.

SECURITY REQUIREMENT: The initiator MUST be able to demonstrate ownership of the session it wishes to modify.

7.2.6.1 Misuse of mobility in NAT handling

Another kind of session modification is related to mobility scenarios. NSIS allows end hosts to be mobile, it is possible that an NSIS node behind a NAT needs to update its NAT binding in case of address change. Whenever a host behind a NAT initiates a data transfer, it is assigned an external IP and port number. In typical mobility scenarios, the DR might also obtain a new address according to the topology and it should convey its new IP address to the NAT. The NAT is assumed to modify these NAT bindings based on the new IP address conveyed by the endhost.

```
     Public                       Private Address
     Internet                     space

                   +----------+                  +----------+
        +----------|  NAT     |------------------|End host  |
                   |          |                  |          |
                   +----------+                  +----------+
                            |
                            |                    +----------+
                            \--------------------|Malicious |
                                                 |End host  |
                                                 +----------+
                         data traffic
                    <========================
```

Figure 50: Misuse of mobility in NAT binding

A NAT binding can be changed with the help of NSIS signalling. When a DR moves to a new location and obtains a new IP address, it sends an NSIS signalling message to modify the NAT binding. It would use the Session-ID and the new flow-id to update the state. The NAT updates the binding and the DR continues to receive the data traffic. Consider the scenario in Figure 50 where an the endhost(DR) and the adversary are behind a NAT. The adversary pretending that it is the end host could generate a spurious signaling message to update the state at the NAT. This could be done for these purposes:

Connection hijacking by redirecting packets to the attacker as in Figure 51

Third party flooding by redirecting packets to arbitrary hosts

Service disruption by redirecting to non-existing hosts

```
       +----------+        +----------+          +----------+
       |  NAT     |        |End host  |          |Malicious |
       |          |        |          |          |End host  |
       +----------+        +----------+          +----------+
            |                    |                      |
            | Data Traffic       |                      |
            |--------->----------|                      |
            |                    |      Spurious        |
            |                    | NAT binding update   |
            |---------<----------+--------<------------|
            |                    |                      |
            | Data Traffic       |                      |
            |--------->----------+-------->------------|
            |                    |                      |
```

Figure 51: Connection Hijacking

SECURITY REQUIREMENT: A NAT/FW signaling message MUST be authenticated, authorized, integrity protected and replay protected between neighboring NAT/FW NSLP nodes.

7.2.7 Misuse of unreleased sessions

Assume that DS (N1) initiates NSIS session with DR (N2) through a series of middleboxes as in Figure 52. When the DS is sending data to DR, it might happen that the DR disconnects from the network (crashes or moves out of the network in mobility scenarios). In such cases, it is possible that another node N3 (which recently entered the network protected by the same firewall) is assigned the same IP address that was previously allocated to N2. The DS could take advantage of the firewall policies installed already, if the refresh interval time is very high. The DS can flood the node (N3), which will consume the access network resources of the victim forcing it to pay for unwanted traffic as shown in Figure 53. Note that here we make the assumption that the data receiver has to pay for receiving data packets.

```
       Public Internet
                                         +--------------------------+
                                         |                          |
       +-------+    CREATE           +---+-----+        +-------+   |
       |       |-------------->------|         |---->---|       |   |
       |  N1   |--------------<------|   FW    |----<---|  N2   |   |
       |       |  success RESPONSE   |         |        |       |   |
       |       |==============>======|         |====>===|       |   |
       +-------+    Data Traffic     +---+-----+        +-------+   |
                                         |                          |
                                         +--------------------------+
```

Figure 52: Before mobility

```
    Public Internet
                                      +--------------------------+
                                      |                          |
    +-------+                     +---+-----+        +-------+   |
    |       |                     |         |        |       |   |
    |  N1   |==============>======|   FW    |====>===|  N3   |   |
    |       |    Data Traffic     |         |        |       |   |
    +-------+                     +---+-----+        +-------+   |
                                      |                          |
                                      +--------------------------+
```

Figure 53: After mobility

Also, this threat is valid for the other direction as well. The DS which is communicating with the DR may disconnect from the network and this IP address may be assigned to a new node that had recently entered the network. This new node could pretend to be the DS and send data traffic to the DR in conformance with the firewall policies and cause service disruption.

SECURITY REQUIREMENT: Data origin authentication is needed to mitigate this threat. In order to allow firewalls to verify that a legitimate end host transmitted the data traffic data origin authentication is required. This is, however, outside the scope of this document. Hence, there are no security requirements imposed by this section which will be addressed by the NATFW NSLP.

7.2.8 Data traffic injection

In some environments, such as enterprise networks, it is still common to perform authorization for access to a service based on the source IP address of the service requestor. There is no doubt that this by

itself represents a security weakness.Hence by spoofing a connection, an attacker is able to reach the target machines, using the existing firewall rules.

The adversary is able to inject its own data traffic in conformance with the firewall policies simultaneously along with the genuine DS.

SECURITY REQUIREMENT: Since IP spoofing is a general limitation of non-cryptographic packet filters no security requirement needs to be created for the NAT/FW NSLP. Techniques such as ingress filtering (described below) and data origin authentication (such as provided with IPsec based VPNs) can help mitigate this threat. This issue is, however, outside the scope of this document.

Ingress Filtering: Consider the scenario shown in Figure 54. In this scenario the DS is behind a router (R1) and a malicious node (M) is behind another router (R2). The DS communicates with the DR through a firewall (FW). The DS initiates NSIS signaling and installs firewall policies at FW. But the malicious node is also able to send data traffic using DS's source address. If R2 implements ingress filtering, these spoofed packets will be blocked. But this ingress filtering may not work in all scenarios. If both the DS and the malicious node are behind the same router, then the ingress filter will not be able to detect the spoofed packets as both the DS and the malicious node are in the same address range.

```
       +-----------------------------------+
       | +------------------+              |
       | |  +-------+   +---+---+          |
       | |  |  DS   +>--+  R1   +->+       |
       | |  |       |   |       |  |       |
       | |  +-------+   +---+---+  |       |
       | |                  |      |       |
       | +------------------+      |   +---+---+     +-------+
       |                           |   |       |     |       |
       |                           +---+  FW   +-->--|  DR   |
       | +------------------+      ****|       |*****|       |
       | |                  |      *   +---+---+     +-------+
       | |  +-------+   +---+---+  *       |
       | |  |   M   |   |  R2   |  *       |
       | |  |       |***|       |***       |
       | |  +-------+   +---+---+          |
       | +------------------+              |
       +-----------------------------------+

   ---->---- = genuine data traffic
   ********* = spoofed data traffic
```

Figure 54: Ingress filtering

7.2.9 Eavesdropping and traffic analysis

By collecting NSLP messages, an adversary is able to learn policy rules for packet filters and knows which ports are open. It can use this to inject its own data traffic due to the IP spoofing capability as already mentioned in Section 7.2.8.

An adversary could learn authorization tokens included in CREATE messages and use them to launch replay-attacks or to create a session with its own address as source address. (cut-and-paste attack)

As shown in Section 4.3 of [23] one possible solution for the session ownership problem is confidentiality protection of signaling messages

SECURITY REQUIREMENT: The threat of eavesdropping itself does not mandate the usage of confidentiality protection since an adversary can also eavesdrop on data traffic. In the context of a particular security solutions (e.g., authorization tokens) it might be necessary to offer confidentiality protection. Confidentiality protection also needs to be offered to the refresh period.

7.3 Security Framework for the NAT/Firewall NSLP

Based on the identified threats a list of security requirements has been created.

7.3.1 Security Protection between neighboring NATFW NSLP Nodes

Based on the analyzed threats it is necessary to provide, between neighboring NATFW NSLP nodes, the following mechanism: provide

- data origin authentication

- replay protection

- integrity protection and

- optionally confidentiality protection

To consider the aspect of authentication and key exchange the security mechanisms provided in [1] between neighboring nodes MUST be enabled when sending NATFW signaling messages. The proposed security mechanisms at GIMPS provide support for authentication and key exchange in addition to denial of service protection. Depending on the chosen protocol, support for flexible authentication protocols could be provided. The mandatory support for security, demands the usage of C-MODE for the delivery of data packets and the usage of D-MODE only to discover the next NATFW NSLP aware node along the path.

7.3.2 Security Protection between non-neighboring NATFW NSLP Nodes

Based on the security threats and the listed requirements it was noted that some scenarios also demand authentication and authorization of a NATFW signaling entity (including the initiator) towards a non-neighboring node. This mechanism mainly demands entity authentication. Additionally, security protection of certain payloads MAY be required between non-neighboring signaling entities and the Cryptographic Message Syntax (CMS) [17] SHOULD be used. CMS can be used

- This might be, for example, useful to authenticate and authorize a user towards a middlebox and vice versa.

- If objects have to be protected between certain non-neighboring NATFW NSLP nodes.

Details about the identifiers, replay protection and the usage of a dynamic key management with the help of CMS is for further study. In

some scenarios it is also required to use authorization token. Their purpose is to associate two different signalling protocols (e.g., SIP and NSIS) and their authorization decision. These tokens are obtained by non-NSIS protocols, such as SIP or as part of network access authentication. When a NAT or Firewall along the path receives the token it might be verified locally or passed to the AAA infrastructure.

Examples of authorization tokens or assertions can be found in RFC 3520 [29] and RFC 3521 [30]. More recent work on authorization token alike mechanisms is Security Assertion Markup Language (SAML). For details about SAML see [31], [32] and [33]. Figure 55 shows an example of this protocol interaction. An authorization token is provided by the SIP proxy, which acts as the assertion generating entity and gets delivered to the end host with proper authentication and authorization. When the NATFW signalling message is transmitted towards the network, the authorization token is attached to the signalling messages to refer to the previous authorization decision. The assertion verifying entity needs to process the token or it might be necessary to interact with the assertion granting entity using HTTP (or other protocols). As a result of a successful authorization by a NATFW NSLP node, the requested action is executed and later a RESPONSE message is generated.

```
    +----------------+   Trust Relationship    +----------------+
    | +------------+ |<.......................>| +------------+ |
    | | Protocol   | |                         | | Assertion  | |
    | | requesting | |    HTTP, SIP Request    | | Granting   | |
    | | authz      | |------------------------>| | Entity     | |
    | | assertions | |<------------------------| +------------+ |
    | +------------+ |    Artifact/Assertion   |  Entity Cecil  |
    |       ^        |                         +----------------+
    |       |        |                          ^     ^|
    |       |        |                          .     || HTTP,
    |       |        |              Trust       .     || other
    |   API Access   |              Relationship.     || protocols
    |       |        |                          .     ||
    |       |        |                          .     ||
    |       |        |                          v     |v
    |       v        |                         +----------------+
    | +------------+ |                         | +------------+ |
    | | Protocol   | |  NSIS NATFW CREATE +    | | Assertion  | |
    | | using authz| |  Assertion/Artifact     | | Verifying  | |
    | | assertion  | | ----------------------- | | Entity     | |
    | +------------+ |                         | +------------+ |
    |  Entity Alice  | <---------------------- |  Entity Bob    |
    +----------------+   RESPONSE              +----------------+
```

Figure 55: Authorization Token Usage

Threats against the usage of authorization tokens have been mentioned in [6] and also in Section 7.2. Hence, it is required to provide confidentiality protection to avoid allowing an eavesdropper to learn the token and to use it in another session (replay attack). The token itself also needs to be protected against tempering.

7.3.3 End-to-End Security

As part of the threat analysis we concluded that end-to-end security is not required and, if used, would be difficult to deploy. Furthermore, it might be difficult to use the suitable identifiers and to establish the necessary infrastructure for this propose.

The only reasonable end-to-end security protection needed within NSIS seems to be a binding between an NSIS signaling session and application layer session. This aspect is, however, for further study.

In order to solicit feedback from the IETF community on some hard security problems for path-coupled NATFW signaling a more detailed description in [20] is available.

8. Open Issues

The NATFW NSLP has a series of related documents discussing several other aspects of path-coupled NATFW signaling, including security [20], migration (i.e., traversal of nsis unaware NATs) [13], intra-realm signaling [14], and inter-working with SIP [15]. Summaries of the outcomes from these documents may be added, depending on WG feedback, to a later version of this draft.

A more detailed list of open issue can be found at:
https://kobe.netlab.nec.de/roundup/nsis-natfw-nslp/index

It is intended to add an overview figure for all NATFW NSLP building blocks into the next version of this memo. Figure 56 sketches the overview

```
                                 +------------------+
                                 |Security Policies |
                                 |   Server         |
                                 +--------^---------+
                                          |
         +--------------------------------|----------------------+
         | +---------+        +-----------V----+        +-------+|
         | |Firewall |<-----> |                |<------>| NAT   ||
         | |Engine   |        | Security policy|        | Engine||
         | +----^----+        | Table/Cache    |        +-^-----+|
         |      |             |             ^  |          |      |
         |      |             +---- --------|--+          |      |
         |   +--|---------------------------|-------------|--+   |
         |   |  V               NATFW NSLP  V             V  |   |
         |   |                                               |   |
         |   +-----------------------------------------------+   |
         |   +--------------------------------------------------+|
         |   |                       GIMPS                      ||
         |   |                                                  ||
         |   +--------------------------------------------------+|
         |   +---------+ +-------+  +------+  +-------+  +------+|
         |   | TCP     | |  UDP  |  | DCCP |  |  SCTP |  | ICMP ||
         |   +---------+ +-------+  +------+  +-------+  +------+|
         |  +-----------------------------+ +--------------------|
         |  |      IPv4                   | |     IPv6           |
         |  +-----------------------------+ +--------------------|
         +-------------------------------------------------------+
```

Figure 56: NATFW NSLP Building Blocks

9. Contributors

We would like to thank the following individuals for their contributions to this document:

- Marcus Brunner and Henning Schulzrinne for work on work on IETF drafts which lead us to start with this document.

- Miquel Martin for his help on the initial version of this document.

- Srinath Thiruvengadam and Ali Fessi work for their work on the NAT/Firewall Threats draft.

- Elywn Davies for his help to make this document more readable.

10. References

10.1 Normative References

[1] Schulzrinne, H., "GIMPS: General Internet Messaging Protocol for Signaling", draft-ietf-nsis-ntlp-05 (work in progress), February 2005.

10.2 Informative References

[2] Hancock, R., "Next Steps in Signaling: Framework", draft-ietf-nsis-fw-07 (work in progress), December 2004.

[3] Brunner, M., "Requirements for Signaling Protocols", RFC 3726, April 2004.

[4] Bosch, S., Karagiannis, G., and A. McDonald, "NSLP for Quality-of-Service signaling", draft-ietf-nsis-qos-nslp-06 (work in progress), February 2005.

[5] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

[6] Tschofenig, H. and D. Kroeselberg, "Security Threats for NSIS", draft-ietf-nsis-threats-06 (work in progress), October 2004.

[7] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[8] Tsirtsis, G. and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", RFC 2766, February 2000.

[9] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[10] Srisuresh, P., Tsirtsis, G., Akkiraju, P., and A. Heffernan, "DNS extensions to Network Address Translators (DNS_ALG)", RFC 2694, September 1999.

[11] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[12] Yadav, S., Yavatkar, R., Pabbati, R., Ford, P., Moore, T., Herzog, S., and R. Hess, "Identity Representation for RSVP", RFC 3182, October 2001.

[13] Aoun, C., Brunner, M., Stiemerling, M., Martin, M., and H. Tschofenig, "NAT/Firewall NSLP Migration Considerations", draft-aoun-nsis-nslp-natfw-migration-02 (work in progress), July 2004.

[14] Aoun, C., "NATFirewall NSLP Intra-realm considerations", draft-aoun-nsis-nslp-natfw-intrarealm-01 (work in progress), July 2004.

[15] Martin, M., "SIP NSIS Interactions for NAT/Firewall Traversal", draft-martin-nsis-nslp-natfw-sip-01 (work in progress), July 2004.

[16] Tschofenig, H., "Extended QoS Authorization for the QoS NSLP", draft-tschofenig-nsis-qos-ext-authz-00 (work in progress), July 2004.

[17] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3369, August 2002.

[18] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[19] Ohba, Y., "Problem Statement and Usage Scenarios for PANA", draft-ietf-pana-usage-scenarios-06 (work in progress), April 2003.

[20] Tschofenig, H., "Path-coupled NAT/Firewall Signaling Security Problems",
DRAFT draft-tschofenig-nsis-natfw-security-problems-00.txt, July 2004.

[21] Tschofenig, H., Buechli, M., Van den Bosch, S., and H. Schulzrinne, "NSIS Authentication, Authorization and Accounting Issues", March 2003.

[22] Adrangi, F. and H. Levkowetz, "Problem Statement: Mobile IPv4 Traversal of VPN Gateways",
DRAFT draft-ietf-mobileip-vpn-problem-statement-req-02.txt, April 2003.

[23] Tschofenig, H., "Security Implications of the Session Identifier", draft-tschofenig-nsis-sid-00 (work in progress), June 2003.

[24] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through

Network Address Translators (NATs)", RFC 3489, March 2003.

[25] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[26] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J., and S. Waldbusser, "Terminology for Policy-Based Management", RFC 3198, November 2001.

[27] Rosenberg, J., "Traversal Using Relay NAT (TURN)", draft-rosenberg-midcom-turn-07 (work in progress), February 2005.

[28] Tschofenig, H., "Using SAML for SIP", draft-tschofenig-sip-saml-02 (work in progress), December 2004.

[29] Hamer, L-N., Gage, B., Kosinski, B., and H. Shieh, "Session Authorization Policy Element", RFC 3520, April 2003.

[30] Hamer, L-N., Gage, B., and H. Shieh, "Framework for Session Set-up with Media Authorization", RFC 3521, April 2003.

[31] Maler, E., Philpott, R., and P. Mishra, "Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1", September 2003.

[32] Maler, E., Philpott, R., and P. Mishra, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1", September 2003.

[33] Maler, E. and J. Hughes, "Technical Overview of the OASIS Security Assertion Markup Language (SAML) V1.1", March 2004.

Authors' Addresses

Martin Stiemerling
Network Laboratories, NEC Europe Ltd.
Kurfuersten-Anlage 36
Heidelberg 69115
Germany

Phone: +49 (0) 6221 905 11 13
Email: stiemerling@netlab.nec.de
URI: http://www.stiemerling.org

Hannes Tschofenig
Siemens AG
Otto-Hahn-Ring 6
Munich 81739
Germany

Phone:
Email: Hannes.Tschofenig@siemens.com
URI: http://www.tschofenig.com

Cedric Aoun
Nortel/ENST Paris
France

Email: cedric.aoun@nortel.com

Appendix A. Firewall and NAT Resources

The NATFW NSLP carries, in conjunction with the NTLP's Message Routing Information (MRI), the policy rules to be installed at NATFW peers. This policy rule is an abstraction with respect to the real policy rule to be installed at the respective firewall or NAT, since it only conveys the initiator's request and must be mapped to the possible configuration on the particular used NAT and/or firewall. For pure firewalls a filter rule must be created and for pure NATs a NAT binding must be created. In mixed firewall and NAT boxes, the policy rule must be mapped in filter rules and bindings observing the ordering of the firewall and NAT engine. Depending on the ordering, NAT before firewall or vice versa, the firewall rules must carry private or public addresses. However, the exact mapping depends on the implementation of the firewall or NAT and is very different per vendor. The remainder of this section gives thus only an abstract mapping of NATFW NSLP policy rules to firewall rules and NAT bindings, without going into the specifics on single configuration parameter possibilities.

A policy rule consists out of the message routing information (MRI), carried in the NTLP, and information available in the NATFW NSLP. The information of the NSLP is stored in the extend flow information object and the message type, in particular the flow direction. Additional information, such as the external IP address and port number, stored in the NAT or firewall will be used as well.

EDITOR's NOTE: This section needs to describe how to map flow routing information to middlebox policy rules. Further, this section must clarify wildcarding.

EDITOR's NOTE: Should this section describe how NATFW NSLP messages are handled in twice-NATs?

A.1 Mapping to Firewall Rules

EDITOR's NOTE: This section is to be done (CREATE, UCREATE).

A.2 Mapping to NAT Bindings

EDITOR's NOTE: This section is to be done (CREATE, REA).

A.3 Mapping for combined NAT and Firewall

EDITOR's NOTE: This section is to be done.

Appendix B. Problems and Challenges

This section describes a number of problems that have to be addressed for NSIS NAT/Firewall. Issues presented here are subject to further discussions. These issues might be also of relevance to other NSLP protocols.

B.1 Missing Network-to-Network Trust Relationship

Peer-to-peer trust relationship, as shown in Figure 38, is a very convenient assumption that allows simplified signaling message processing. However, it might not always be applicable, especially between two arbitrary access networks (over a core network where signaling messages are not interpreted). Possibly peer-to-peer trust relationship does not exist because of the large number of networks and the unwillingness of administrators to have other network operators to create holes in their Firewalls without proper authorization.

```
   +----------------------+              +--------------------------+
   |                      |              |                          |
   |          Network A   |              |              Network B   |
   |                      |              |                          |
   |            +---------+   Missing    +---------+                |
   |      +-///-+ Middle- |    Trust     | Middle- +-///-+          |
   |      |     |  box 1  |   Relation-  |  box 2  |     |          |
   |      |     +---------+     ship     +---------+     |          |
   |      |               |     or       |               |          |
   |      |               | Authorization|               |          |
   |      |               |              |               |          |
   |      |   Trust       |              |      Trust    |          |
   |      | Relationship  |              |  Relationship |          |
   |      |               |              |               |          |
   |      |               |              |               |          |
   |      |               |              |               |          |
   |   +--+---+           |              |            +--+---+      |
   |   | Host |           |              |            | Host |      |
   |   |  A   |           |              |            |  B   |      |
   |   +------+           |              |            +------+      |
   +----------------------+              +--------------------------+
```

Figure 57: Missing Network-to-Network Trust Relationship

Figure 57 illustrates a problem whereby an external node is not allowed to manipulate (create, delete, query, etc.) packet filters at a Firewall. Opening pinholes is only allowed for internal nodes or

with a certain authorization permission. Hence the solution alternatives in Section 3.3.2 focus on establishing the necessary trust with cooperation of internal nodes.

B.2 Relationship with routing

The data path is following the "normal" routes. The NAT/FW devices along the data path are those providing the service. In this case the service is something like "open a pinhole" or even more general "allow for connectivity between two communication partners". The benefit of using path-coupled signaling is that the NSIS NATFW NSLP does not need to determine what middleboxes or in what order the data flow will go through.

Creating NAT bindings modifies the path of data packets between two end points. Without NATs involved, packets flow from endhost to endhost following the path given by the routing. With NATs involved, this end-to-end flow is not directly possible, because of separated address realms. Thus, data packets flow towards the external IP address at a NAT (external IP address may be a public IP address). Other NSIS NSLPs, for instance QoS NSLP, which do not interfere with routing - instead they only follow the path of the data packets.

B.3 Affected Parts of the Network

NATs and Firewalls are usually located at the edge of the network, whereby other signaling applications affect all nodes along the path. One typical example is QoS signaling where all networks along the path must provide QoS in order to achieve true end-to-end QoS. In the NAT/Firewall case, only some of the domains/nodes are affected (typically access networks), whereas most parts of the networks and nodes are unaffected (e.g., the core network).

This fact raises some questions. Should an NSIS NTLP node intercept every signaling message independently of the upper layer signaling application or should it be possible to make the discovery procedure more intelligent to skip nodes. These questions are also related to the question whether NSIS NAT/FW should be combined with other NSIS signaling applications.

B.4 NSIS backward compatibility with NSIS unaware NAT and Firewalls

Backward compatibility is a key for NSIS deployments, as such the NSIS protocol suite should be sufficiently robust to allow traversal of none NSIS aware routers (QoS gates, Firewalls, NATs, etc).

NSIS NATFW NSLP's backward compatibility issues are different than the NSIS QoS NSLP backward compatibility issues, where an NSIS

unaware QoS gate will simply forward the QoS NSLP message. An NSIS unaware Firewall rejects NSIS messages, since Firewalls typically implement the policy "default to deny".

The NSIS backward compatibility support on none NSIS aware Firewall would typically consist of configuring a static policy rule that allows the forwarding of the NSIS protocol messages (either protocol type if raw transport mode is used or transport port number in case a transport protocol is used).

For NATs backward compatibility is more problematic since signaling messages are forwarded (at least in one direction), but with a changed IP address and changed port numbers. The content of the NSIS signaling message is, however, unchanged. This can lead to unexpected results, both due to embedded unchanged local scoped addresses and none NSIS aware Firewalls configured with specific policy rules allowing forwarding of the NSIS protocol (case of transport protocols are used for the NTLP). NSIS unaware NATs must be detected to maintain a well-known deterministic mode of operation for all the involved NSIS entities. Such a "legacy" NAT detection procedure can be done during the NSIS discover procedure itself.

Based on experience it was discovered that routers unaware of the Router Alert IP option [RFC 2113] discarded packets, this is certainly a problem for NSIS signaling.

B.5 Authentication and Authorization

For both types of middleboxes, Firewall and NAT security is a strong requirement. Authentication and authorization means must be provided.

For NATFW signaling applications it is partially not possible to do authentication and authorization based on IP addresses. Since NATs change IP addresses, such an address based authentication and authorization scheme would fail.

B.6 Directional Properties

There two directional properties that need to be addressed by the NATFW NSLP:

- Directionality of the data

- Directionality of NSLP signaling

Both properties are relevant to NATFW NSLP aware NATs and Firewalls.

With regards to NSLP signaling directionality: As stated in the previous sections, the authentication and authorization of NSLP signaling messages received from hosts within the same trust domain (typically from hosts located within the security perimeter delimited by Firewalls) is normally simpler than received messages sent by hosts located in different trust domains.

The way NSIS signaling messages enters the NSIS entity of a Firewall (see Figure 2) might be important, because different policies might apply for authentication and admission control.

Hosts deployed within the secured network perimeter delimited by a Firewall, are protected from hosts deployed outside the secured network perimeter, hence by nature the Firewall has more restrictions on flows triggered from hosts deployed outside the security perimeter.

B.7 Addressing

A more general problem of NATs is the addressing of the end-point. NSIS signaling message have to be addressed to the other end host to follow data packets subsequently sent. Therefore, a public IP address of the receiver has to be known prior to sending an NSIS message. When NSIS signaling messages contain IP addresses of the sender and the receiver in the signaling message payloads, then an NSIS entity must modify them. This is one of the cases, where a NSIS aware NATs is also helpful for other types of signaling applications e.g., QoS signaling.

B.8 NTLP/NSLP NAT Support

It must be possible for NSIS NATs along the path to change NTLP and/or NSLP message payloads, which carry IP address and port information. This functionality includes the support of providing mid-session and mid-path modification of these payloads. As a consequence these payloads must not be reordered, integrity protected and/or encrypted in a non peer-to-peer fashion (e.g., end-to-middle, end-to-end protection). Ideally these mutable payloads must be marked (e.g., a protected flag) to assist NATs in their effort of adjusting these payloads.

B.9 Combining Middlebox and QoS signaling

In many cases, middlebox and QoS signaling has to be combined at least logically. Hence, it was suggested to combine them into a single signaling message or to tie them together with the help of some sort of data connection identifier, later on referred as Session ID. This, however, has some disadvantages such as:

- NAT/FW NSLP signaling affects a much small number of NSIS nodes along the path (for example compared to the QoS signaling).

- NAT/FW signaling might show different signaling patterns (e.g., required end-to-middle communication).

- The refresh interval is likely to be different.

- The number of error cases increase as different signaling applications are combined into a single message. The combination of error cases has to be considered.

B.10 Inability to know the scenario

In Section 2 a number of different scenarios are presented. Data receiver and sender may be located behind zero, one, or more Firewalls and NATs. Depending on the scenario, different signaling approaches have to be taken. For instance, data receiver with no NAT and Firewall can receive any sort of data and signaling without any further action. Data receivers behind a NAT must first obtain a public IP address before any signaling can happen. The scenario might even change over time with moving networks, ad-hoc networks or with mobility.

NSIS signaling must assume the worst case and cannot put responsibility to the user to know which scenario is currently applicable. As a result, it might be necessary to perform a "discovery" periodically such that the NSIS entity at the end host has enough information to decide which scenario is currently applicable. This additional messaging, which might not be necessary in all cases, requires additional performance, bandwidth and adds complexity. Additional, information by the user can provide information to assist this "discovery" process, but cannot replace it.

Appendix C. Object ID allocation for testing

TBD.

Appendix D. Acknowledgments

We would like to acknowledge: Vishal Sankhla and Joao Girao for their input to this draft; and Reinaldo Penno for his comments on the initial version of the document. Furthermore, we would like thank Elwyn Davies for his valuable help and input.

Intellectual Property Statement

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.