# Abstract:

For various reasons (e.g., security, lack of IPv4-addresses) the services in the home smart space only use private IP addresses. This is unfortunate in the remote service access since these addresses frequently appear in responses sent from a service in the remote smart space (e.g., your home) to the visited smart space (e.g., your friend’s home).The Internet Engineering Task Force (IETF) provides some solutions and workarounds for the problem caused by NAT.

In this project, the challenge to me is to summarize the available options, rank the options according to which one is preferred for the RA-scenario. I will come up with my practical NAT traversal techniques by testing and gathering data on the reliability of NAT traversal techniques since none of the existing ones seems to work well. A demonstration of the key features will be shown in the thesis. NAT traversal techniques apply to TCP and UDP need to be researched in advance. Handling of peers behind all kinds of NAT need to be tested and determined for the communication. The result of the paper will well improve the evaluation of specific issues on NAT and the creating of an UNSAF proposal.

# Preface

This thesis is the partial fulfillment of the two years Master of Science program in Information and Communication Technology (ICT) at University i Agder (UiA), Faculty of Engineering and Science in Grimstad, Norway. The project has been carried out from January to June 2008 and its workload equals 30 ECTS.

First and foremost, I am deeply grateful to my supervisors Prof. Frank Reichert, PH.D Andreas Häber and external supervisor Martin Gerdes for their excellent guidance and precious advices during the fulfillment of this project.

I would also like to thank Mr. Stein Bergsmark and Mrs. Sissel Andreassen for their contributions to this project and coordination of my studies and daily life in Grimstad.

Grimstad, May 2008

Li Zhu

# Table of Contents

# Table of Figures

# 1 Introduction

## 1.1 Background

Our project is based on the ONE Project [1], in which a remote service access component was produced to enable you to access the multimedia services of a remote private network. In practice, the services are restricted to public IP address for some unavoidable problems such as security and the limited amount of IPv4 addresses. Because most Internet users (both senders and receivers) are behind the private network, the extension of the services usage may be prevented by the restriction of using public IP addresses. The problem that blocks the communication of hosts behind different private networks is named as “UNilateral Self-Address Fixing (UNSAF) across Network Address Translation (NAT)”, which is defined in RFC3424 [2]. Moreover, the remote service access can be divided into three main parts: establishing connection, multimedia transmission and terminating transmission. Connections between hosts behind two different private networks can be established via session initiation protocol (SIP) [3]. The multimedia transmission should start after the establishing of connection. Proxies are used for the establishing of connection on condition that SIP registrars can keep the track of users’ whereabouts. The specific operation of remote service access is described in the ONE Project as follow:

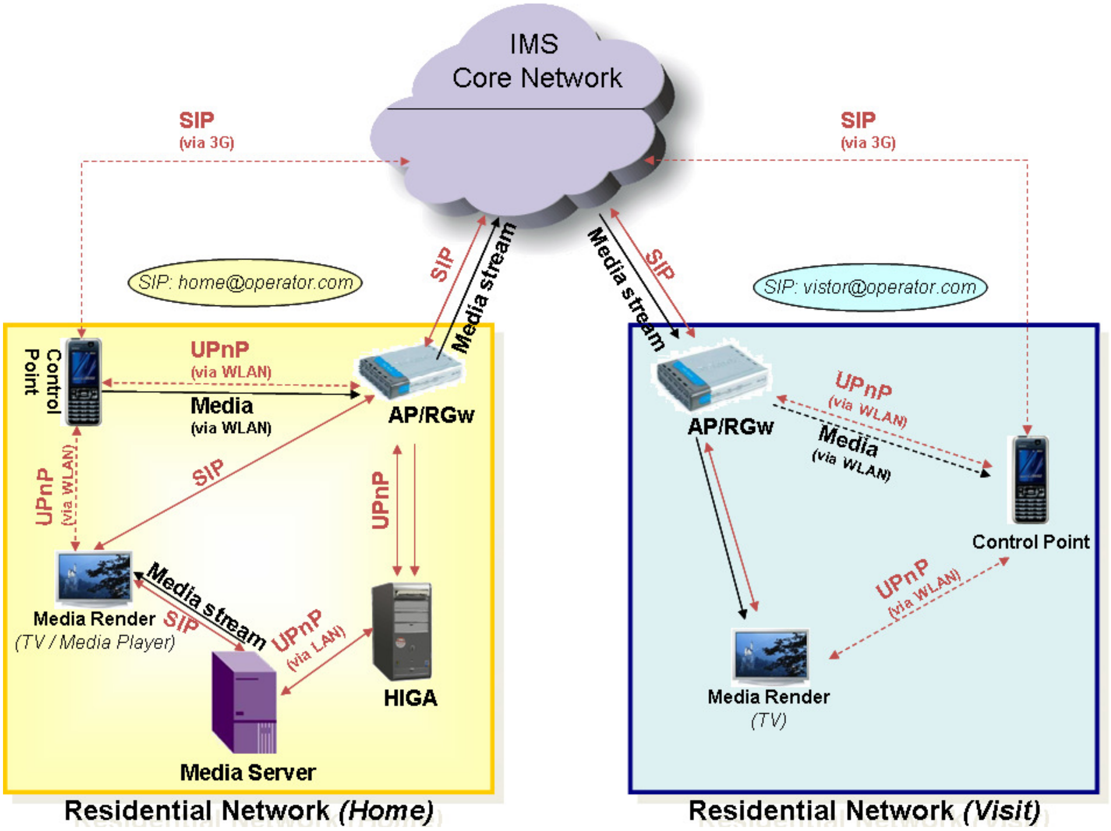

**Figure 1:** Remote Access to Home Service

In Figure 1, the control point can be located in both the home residential network and the visit residential network. Connection from home residential network or visit residential network to core network is supported by SIP. For remote access, a connection is initiated by control point in the visit residential network. Besides, the purpose of the remote access is to make a host behind the visit residential network connect to another host behind the home residential network and receive the information from the host.

In addition, if the connection is between media render and media server, the remote multimedia streaming still need to be analysed and improved. Since the multimedia streams have to go through the NAT traversal, the transmission speed of the streams may be limited. Moreover, the delay may be caused by the operation of NAT traversal.

The influence of IPv4 address depletion has drawn more and more attention on the extension of IPv4 address space in recent years [4]. Widespread acceptance has been gained by NATs for resolving the problem of IPv4 address depletion. The significant effort of NAT traversal has exhibited its ability on improving the Internet structure. The quantity of available IP addresses is efficiently extended by NATs which conserve IPv4 address space by using NAT boxes to interconnect local networks which comprise a block of private IP addresses separately to the public Internet. Because IETF missed the opportunity to standardise NAT before its wide deployment [5], there are several types of NAT traversal to serve the Internet around the world. All of them have their separate advantages and disadvantages which make them suitable for different networks.

The network structure desired in the remote access needs to be deployed with a reasonable network address translator which can apply to the remote private network. Furthermore, according to the specific demand of remote access, the multimedia transmission should be realized between hosts behind different private networks. A reasonable NAT traversal needs to be selected to satisfy the success of connection establishing, multimedia streaming and terminating transmission.

In the future, the larger address space of IPv6 [6] may reduce the need for NAT. However, in the short term IPv6 is increasing the demand for NAT, because NAT itself provides the easiest way to achieve interoperability between IPv4 and IPv6 address domains [7].

## 1.2 Problem Description

For the generalization of the result in ONE Project, UNSAF across NAT need to be carefully analyzed to settle specific problems caused by the process of communication between hosts behind different private networks. There are two cases described in RFC3424 [2]:

1. When the client initiates communication, starting the communication will make the side effect of creating an address binding in the NAT device and allocating an address in the realm outside of the NAT box.
2. A server needs to accept connections from outside. Since it does not initiate communication before, no NAT binding is created.

A mechanism is necessary for the fixing of such a binding before communication starts. The function of UNSAF server is for some originating process to determine or fix the UNSAF address and port. We can illustrate communication processes of two endpoints behind different private networks with the following Figure 2:

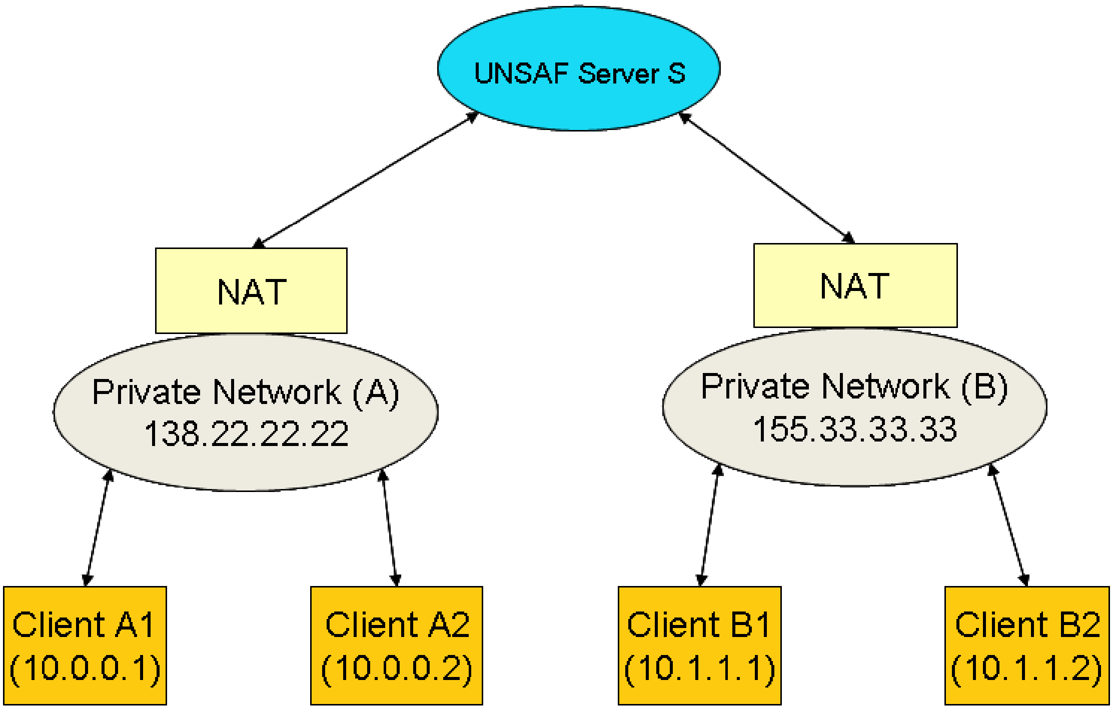

**Figure 2:** Connection of Hosts behind Different Private Networks

In the Figure, Client A1 and Client B1 are belonging to different private networks. Their respective NATs will prevent them from directly initiating the connection to each other. Before they start the establishing of connection to each other, they have to establish the Client/Server connection. By the establishing of Client/Server connection, Clients' addresses in different private networks can be gained by UNSAF Server. Client A1 then can start establishing the connection to Client B1 through Client B1's address and port number in its address realm which is fixed by the UNSAF Server.

However, a series of specific technical problem may impede the creation of standard protocol for "UNSAF across NAT". There are some architectural issues that can affect UNSAF system and these negative issues have been concluded in RFC3424 clearly. Besides, the practical issues caused by the differences of NAT box implementations are also contained in RFC 3424. All the issues are correlative with the problems of "UNSAF across NAT".

## 1.3 Project Definition

According to the final purpose of the project required by the services access in remote private networks, the project is defined as:

> "Solutions for handling the problem of UNIlateral Self-Address Fixing (UNSAF) across Network Address Translation (NAT), in the context of remote access, shall be found and described. This includes existing and proposed solutions. The solutions shall be evaluated and ranked based on criteria, such as compatibility (does the solution require modifications to the client or server?), performance (an estimate of the performance hit introduced by the solution), and more criteria to be defined within this project. If time permits, at least one of the best ranked solutions shall be prototyped"

## 1.4 Problem Premises and Temporary Assumptions

The first premise of the project is that the remote service access component can supply the services between different private networks. Fully understanding and mastery of the fundamental knowledge about remote service access can lead us to a correct entry to the suitable solution for the problem of UNSAF across NAT in remote access. Under the assumption that the selected NAT traversal technique can supply a suitable IP address translation platform, we will be able to extend the scope of multimedia services by researching the establishing and transmission operation on the application layer.

During the period of evaluation of solutions, we assume that our clients are hosts behind different private networks. There are three types of situations that should be researched and tested separately to fully settle any possible problems the remote service access may encounter. The three situations are illustrated with Figure 3 shown as follow:

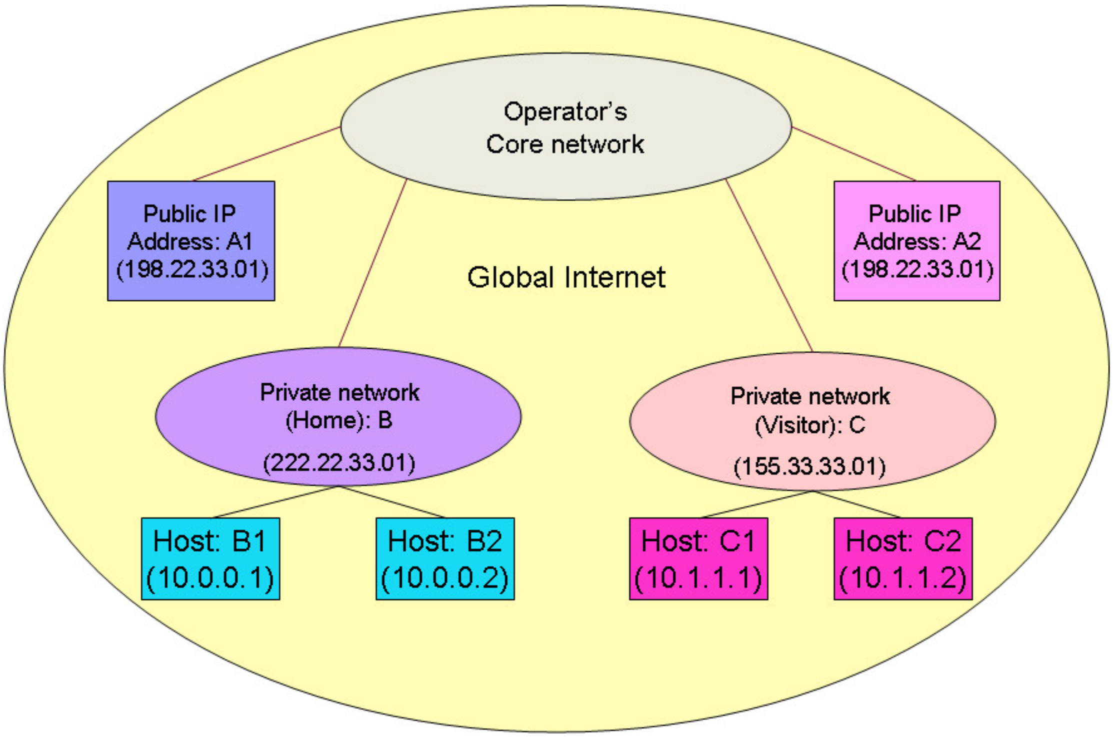

**Figure 3:** Connections of different endpoints

1. One host is public IP address, another one is behind a private network. For example, the connection between A1 and B1 belongs to the first type.
2. Both of the hosts are behind separate private networks. The connection between B1 and C1 is taken as an example.
3. Both of the hosts are using public IP addresses. The given example is connection between A1 and A2.

In the end, if time permits, we will produce a prototype with one of the best ranked solutions that can exhibit the application of our project in a convenient way.

## 1.5 Motivation

Since ONE Project's remote access service frequently sent from a remote private network to a local private network, it is reasonable for us to concentrate our attention on the powerful IP address management function of NATs. The main reason for NATs are currently attractive to us is that NATs not only offer a simple, economical way for conserving IPv4 addresses but also NATs can be installed incrementally, without changes to either routers or hosts [8]. However, we can not neglect the other side of the coin that is several inherent problems and limitation introduced by NATs, such as security limitations, increased potential for misaddressing, inability to examine and overwrite realm-specific IP addresses in many protocols etc. These existed problems are stimulating challenge to us which motivate our potential of comprehensive mastery of NATs.

During the period for preparation, much time was spent on taking cognizance of IP network structure. Solid foundation was laid for network architecture planning and design. The knowledge we learned in the period is appreciated by us because it will assist us in our future study in the field.

As a result of the project, we can gain insights into NAT technology. The project gives us good theoretical and practical experience with handling of IP address, and a chance to demonstrate our capabilities to solve complex system engineering problems, which will be very rewarding for us.

## 1.6 Importance of the Study

Our project is a necessary part for remote access. We can widely extend the application of remote access with the result of our project because the limited solution for the NAT problem in the ONE Project will be settled. The delimitation of only using private IP addresses will be transcended in our project. Multimedia services will be accessed by hosts behind different private networks. The evaluation of specific issues on NAT and the creating of an UNSAF proposal can be well improved by the result of the paper. In addition, we can change the well-know difficulties for peer-to-peer communication caused by NAT to be the great benefits produced by peer-to-peer communication based on NAT.

## 1.7 Report Outline

This report is structured as follows.
Chapter 1: introduces the master thesis that is the current chapter.
Chapter 2: provides the basic theory on IP network and NAT. This will establish a foundation for the progress of the later work described in the report.
Chapter 3: specifies three types of solutions. First solution is based on IP address swapping, second solution is achieved via SOAP intermediary and the third solution is implemented by RSIP.
Chapter 4: six types of criteria are selected to make an appropriate evaluation for the three proposal solutions.
Chapter 5: the evaluation of the three solutions is compared and ranked with the selected criteria.
Chapter 6: makes a conclusion for each of the three proposal solutions.

# 2 Background / State-of-the-art

## 2.1 IP Networks

Since the continued growth of the Internet, an ever-increasing demand on IP address space and other functional requirements that network address translation is perceived to facilitate needs to be met absolutely [9]. Internet has been forced to evolve in ways that make operation hard for many applications by the pressures of rapid growth and excessive security problems [10]. The original uniform address architecture of Internet has been recognized widely, that is every node has a globally unique IP address and can communicate with any other node directly [11]. However, for the tremendous requirements of IP address space, the original address architecture has been replaced by a new address architecture which includes a global address realm and many private address realms interconnected by network address translators [12]. The new address architecture is illustrated in Figure 4 [13].

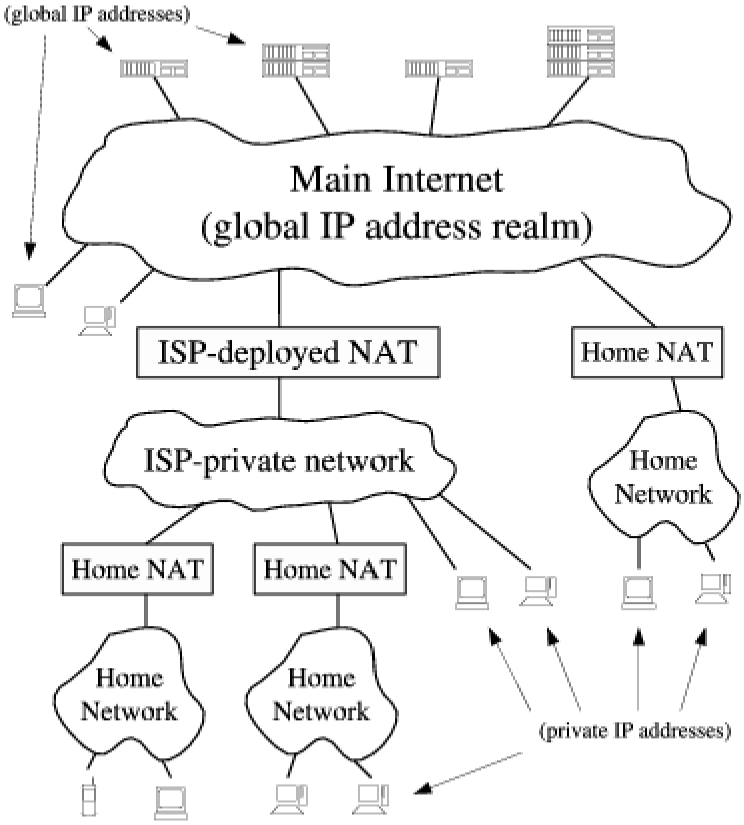

**Figure 4:** Public and private IP address domains [13]

In the new address architecture, only nodes in the main Internet (global IP address realm) have unique and globally routable IP addresses. As compared with nodes on private network, the node located in the main Internet can be easily touched from anywhere in the network.

More attention should be paid on nodes in private networks because they represent the extension of IP address space. These nodes can easily connect to other nodes behind the same private network. Moreover, well-known nodes in the global IP address realm can also be connected with them by using TCP [14] or UDP [15]connections. Outgoing connections from the private network need nodes act as public endpoints which are temporarily allocated by NATs. Besides, the addresses and port numbers in packets comprising these session are translated by NATs.

In the context of remote access, we find it is extremely important to establish the peer-to-peer communication when the two nodes are on different private networks. However, the new IP address architecture is only suitable for client/sever communication in terms of the client is on a private network while the server is in the global address realm. It is difficult for the new address architecture to operate the situation directly when two nodes are on different private networks. An effective method of establishing peer-to-peer communication between hosts on different private networks need to be produced undoubtedly to cooperate with NAT.

### 2.1.1 Address Allocation for Private Internets

Hosts within enterprises that use IP addresses can be partitioned into three categories according to the scope of hosts' requirements. With the definition of the categories' difference in RFC1918 [16], we summarize the three categories as follow:

- Category 1: hosts do not need to access hosts in other enterprises; hosts use unambiguous IP addresses within an enterprise, but maybe ambiguous IP addresses between enterprises.
- Category 2: hosts need to access to a limited set of outside services; hosts use unambiguous IP addresses within an enterprise, but maybe ambiguous IP addresses between enterprises.
- Category 3: hosts need network layer access outside the enterprise; hosts require globally unambiguous IP addresses.

The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

1. 10.0.0.0 - 10.255.255.255 (10/8 prefix)
2. 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
3. 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

IP addresses out of the address space defined in RFC1918 can be used by an enterprise without any coordination with IANA and making an Internet registry. The address space can thus be used by many enterprises. Addresses within this private address space must be unique within the enterprise. If some enterprises which choose to cooperate over this space, they may communicate with each other in their own private internet. An enterprise that needs globally unique address space is required to obtain such addresses from an Internet registry. The three blocks of IP addresses defined above will never be assigned when an enterprise requests IP addresses for external connectivity.

Private hosts can not connect with any host outside of the enterprise besides accessing to external services via mediating gateways. Public hosts can communicate with any hosts inside the enterprise and public hosts outside the enterprise, but they can not connect with private hosts of other enterprises.

Using private address space can conserve the globally unique address space of Internet through not using it on condition that global uniqueness is not required. A lot of flexibility in network design has been gained by enterprises by owning more address space at their disposal than they can obtain from the globally unique pool.

An enterprise’s flexibility to access the Internet may actually be reduced through using private address space because one is committing to renumber part or all of an enterprise when one commits to using a private address. The cost of renumbering can usually be measured by counting the number of hosts that have to transition from private to public. Renumbering may be required when merging several private internets into a single private internet because these privates prior to the merge could maintain their uncoordinated internets using private address space.

### 2.1.2 Network Address Translation (NAT) and Traversal

Due to the unexpected growth of the Internet in recent years, not only the danger of IP address space exhaustion was encountered but also the instant demand of IP addresses needs to be met. Because the regular IP address allocation process could not meet the demand, NAT was chosen to meet that instant massive demand [17]. As compared with the great influence of success made by NAT, emphasis will not be placed on the reason for IETF missed the opportunity to standardize NAT. Since there is no traversal technique that works with all existing NATs, there are many standards are created for NAT behavior. The principal objective of this chapter is to describe how NAT works and analyze existing solutions of NAT problem over the assumed network configuration.

#### 2.1.2.1 Operation Processes of NAT

The issue of NAT working process can be presented in a simple scenario according to the analysis of a retrospective view of NAT. A NAT box is the focus of the communication's attention which is between public IP address and private IP address. As the follow Figure 5 [13] shows, 155.99.25.11 is the public IP address to left NAT box and 10.0.0.1 is the private IP address to it. 62000 is the port number reserved for Client A. The right NAT box is between public IP address: 138.76.29.7 and private IP address: 10.1.1.3. 31000 is the private number reserved for Client B.

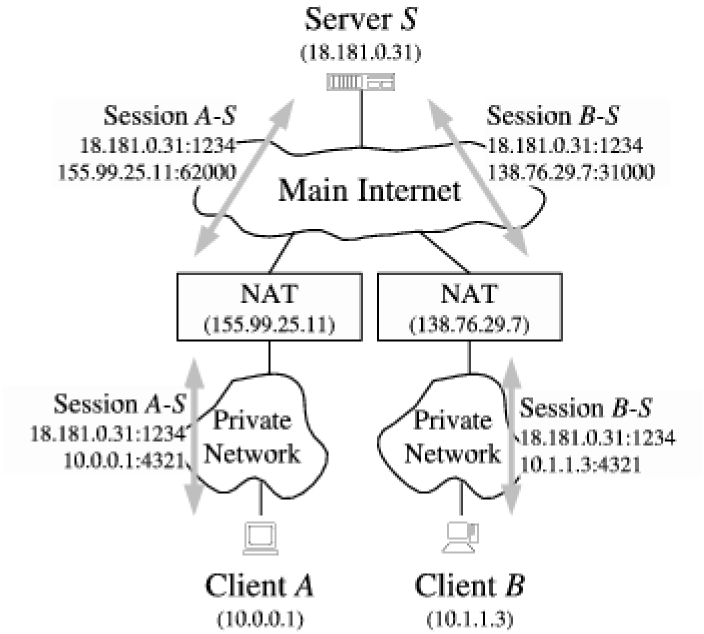

**Figure 5:** NAT Traversal by Relaying [13]

The NAT box has a public IP address for its interface connecting to the global Internet (Main Internet) and a private address facing the internal network (Private Network). If an internal host (Client A) sends an IP packet to a public IP destination address (138.76.29.7) in the global Internet, the packet has to be routed to the NAT box. The box translates the private source IP address in the packet's header to its public IP address and an entry leads to its internal table that keeps the track of the mapping between the internal host and the outgoing packet which should be added. The mapping table is preserved in the NAT box. The entry confirms a function that enables all subsequent packet exchanges between the internal host and the destination address. After a certain period of idleness which is normally set to a specific value by vendor, the mapping entry will be times out.

When a packet sent back from the destination address (138.76.29.7) as a response to the packet, it will have to arrive at the left NAT box first. The corresponding entry then can be found from the mapping table of the NAT box. The destination's own public IP address (155.99.25.11) is replaced with the real destination address of the host (10.0.0.1) and then the response packet is delivered to the original host. During the process of changing the IP address carried in the IP header of every transferring packet, the IP header checksum and the transport protocol' checksum must be recalculated on condition that it is calculated based on IP address.

#### 2.1.2.2 NAT Traversal for SIP

According to the demand of remote service access, a reasonable technique for NAT traversal should be chosen for the private network. The process of selecting a suitable NAT traversal is complicated by not only the network topology but also many details in the communication processes. Six types of proposed NAT traversal for SIP [18] are simply introduced as follow:

- UPnP Internet gateway device (IGD) depends on the respond from NAT to the discovery request. Pinholes can be opened under the dynamic control of the UPnP client. The user agent and NAT need to be UPnP enabled for this situation. Simple configurations are allowed by UPnP, however, some disadvantages need to be overcome such as the failure to cascaded NATs and massive task of upgrading UPnP with the currently deployed NATs and SIP entities.

- The STUN makes a SIP client to check if it is behind a NAT and confirm the type of NAT [19]. An exploratory message is sent to the external STUN server by STUN-enabled client to determine the details of the public side of the NAT. After the information received by the server, the public IP address and ports that were used by NAT will be known by client. Then the client can construct a SIP request containing public IP address and port numbers. The STUN server does not participate in the signaling and media flows. The assumption that binding in NAT is oblivious to the destination IP address and port number makes STUN fails with symmetric NAT which is the most common NAT type.

- The TURN protocol can make media traversal through symmetric NAT. A TURN server is inserted either in the customer's demilitarized zone or the service provider network to response the TURN-enabled SIP client with the public IP address and port used by NAT which will be used for this session. Symmetric NATs can be used for there is no change in the destination address detected by NAT. However, TURN needs existing SIP user agents to be upgraded by client vendors.

- As compared with ALG solution, MIDCOM solution inserts application layer protocol intelligence into a MIDCOM agent that can dynamically control MIDCOM-enabled NATs to punch holes for signaling and media flows [20]. Although the NATs and firewalls are not required to continually upgrade to support diverse application layer protocols, upgrading existing NATs for supporting MIDCOM is still a hard role [21].

- The VPN solution is differentiated from other solutions by using tunneling technique [22]. It makes tunnels for both media and signaling to go through the NAT and firewall installations to a public address space server. SIP signaling is modified by tunnel termination server to reflect its outbound port information, thus either the incoming or outgoing connections can be established. The clear disadvantages of VPN are the latencies in signaling and media flows and the external server that is pregnable to purposive attacks [23].

- Though ICE is designed to work with SIP and its companion protocol, the Session Description Protocol (SDP) [24], it can provide NAT and firewall traversal capabilities for any other type of session-oriented protocol [25]. STUN and TURN is used by ICE and a unifying framework is provided around them. ICE may provide traversal under even the most complex topologies. ICE will make use of intermediate relays (the TURN server) only when nothing else works. ICE also supports Transmission Control Protocol media sessions. ICE has not reach RFC states.

#### 2.1.2.3 Evaluation of NAT

Nowadays, NAT offers a lot of advantages beyond the modest claim in RFC1918. We conclude the following features associated with the advantages of NAT from the retrospective view of NAT [5]. Firstly, NAT can unilaterally be deployed by any end site [26]. Secondly, one can use 16 million private IP addresses without asking for any permission from IANA, and the rest of the Internet can be connected by using only a single allocated public IP address [27]. Thirdly, one never needs to worry about renumbering the internal network when changing providers besides renumbering the NAT box because of one level of indirection [28]. Fourthly, A NAT box also makes multi-homing easy. One NAT box can be connected to multiple providers and use one IP address from each provider. Not only does the NAT box shelter the connectivity to multiple ISPs from all the internal hosts, but also it does not require any of its providers to punch a hole in the routing announcement. If the multi-homed site takes an IP address block from one of its providers and asks the other providers to announce the prefix, the hole punching would be needed [29]. Last but not least, for external hosts cannot directly initiate communication with hosts behind a NAT and figure out the internal topology. This one level of indirection can also be perceived as one level of protection [30].

At the same time, we should not neglect the disadvantages of NAT that can be identified immediately. First of all, hosts behind a NAT must go through the NAT to reach others host on the Internet and only internal host can initiate communication first for establishing the mapping entries [31]. Additionally, if the NAT box crashes, all of the existing state may be lost and data exchange between all of the internal and external hosts will have to be restarted because the ongoing data exchange depends on the mapping entry kept at NAT box [32]. It is a clear deviation from the goal of original IP that promise the successful delivering of packets to their destination as long as any physical connectivity exists between the source and destination. Moreover, all protocols dependent on IP address are affected because the IP addresses carried in a packet are altered by NAT [33].

Although the existence of NAT in today’s architecture has gain widely acceptance, we still can not simply adopt the existing NAT traversal solutions as given. Instead, handling of peers behind all kinds of NAT still needs to be tested and determined for the better communication. The NAT traversal design space needs to be fully explored to promote the solution development to be unanimous in the model of Internet architecture.

## 2.1.3 Realm Specific Internet Protocol (RSIP)

The main purpose of producing Realm Specific Internet Protocol is to make an alternative to NAT. From the working process of NAT, we can notice that the NAT router must examine and change the network layer, and even the transport layer. Moreover, the header of each packet which is crossing the addressing domain connected to the NAT router is also need to be examined and changed. This is the reason for why the mechanism of NAT offends the end-to-end nature of the Internet connectivity. In addition, the operation of NAT router also destroys protocols that requiring or enhancing end-to-end integrity of packets.

When RSIP takes the place of NAT, a RSIP gateway will replace the NAT router. RSIP-aware hosts on the private network will be regarded as RSIP hosts. According to the viewpoint of RFC3102 [34], we can take RSIP as a necessary supplement to NAT because it can reduce the influence of some IP address shortage problems in some scenarios without some limitations of NAT. However, RSIP is not a long-term solution for the problem of IP address shortage. The main content of RSIP will be summarized from RFC3102 to give a clear precondition for my analysis in the remote access.

The degree of address realm transparency which is defined by RSIP is allowed to be achieved between two different scopes, even two completely different addressing realms by RSIP. It is useful for establishing a network structure which realizes end-to-end packet transparency between different addressing realms. In order to satisfy the requirement of increasing number on IP address, RSIP is expected to be deployed on private IPv4 networks and can be permitted to access to public IPv4 networks.

We need to pay attention on an important character of RSIP that is no requirement for the modification of applications. All the modifications to RSIP host associated with RSIP should be made at network layer and transport layer. Although RSIP allows end-to-end packet transparency, it might not appear transparently to all applications.

### 2.1.3.1 Terminology of RSIP

Before the introduction to the architecture of RSIP, we need to list the terminology of RSIP firstly [34].

- Private Realm: The routing realm which uses private IP addresses from the scope of 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16 [7] or addresses those are non-routable from Internet.

- Public Realm: The routing realm consisted of globally unique network addresses.

- RSIP Host: The host locates in an addressing realm and obtains addressing parameters from another addressing realm via an RSIP gateway.

- RSIP Gateway: A router or gateway which is located on the boundary of two addressing realms. The router or gateway should be assigned at least one IP address in at least one of the two realms. RSIP gateway should be able to manage parameters and assign it from one realm to RSIP hosts in another realm. If hosts located in a realm are not RSIP enabled, the RSIP gateway will act as a normal NAT router.

- RSIP Client: The application process that act as the client portion of the RSIP client/server protocol. The RSIP client application must exist in all RSIP hosts, and may exist in RSIP gateways.

- RSIP Server: The application process that act as the server portion of the RSIP client/server protocol. The RSIP client application must exist in all RSIP gateways.

- RSA-IP (Realm Specific Address IP): A RSIP method which is used to allocated each RSIP host a unique IP address from the public realm.

- RSAP-IP (Realm Specific Address and Port IP): A RSIP method which is used to allocated each RSIP host an IP address (may be shared with other RSIP hosts) and some unique ports for each address from the public realm.

- Demultiplexing Fields: The set of packet's header or payload fields which is used by a RSIP gateway to route an incoming packet to a RSIP host.

#### 2.1.3.2 Two Types of RSIP

According to the definition of RFC3102 [34], we divide RSIP into two types: RSA-IP and RSAP-IP. RSIP hosts and gateways with different capabilities may support RSA-IP, RSAP-IP or both of them.

RSA-IP allows an RSIP gateway to maintain a pool of IP addresses that can be leased by RSIP hosts. If the RSIP gateway receives a request from a host, it will allocate an IP address to the host and prevent other hosts from using the IP address until the IP address is returned to the pool from the host. If an address is not assigned to a host, the host will not be able to use the address. A host can lease at least one address from the same or different RSIP gateway. Information about the IP address which is leased to the host must be recorded in the demultiplexing fields of the RSA-IP session. Host can use any TCP/UDP port which is associated with their assigned address and run gateway applications at the port. These applications will be available to public network even though there is no help from RSIP gateway.

RSAP-IP allows an RSIP gateway to maintain a pool of IP addresses and pools of port number for each address. At least one port can be leased with an IP address to be used together. If an address with ports is assigned to a host, it can only be used by the host until it is released to the pool. If address and port combinations are not specifically allocated to a host, the host will not be able to use them. Gateway applications bound may be operated on an assigned address/port combination by hosts. If the RSIP gateway agrees to route all traffic destined to the combination for the host, these application will be available to the public network. In addition, information about the address/port combination which is leased to the host also must be recorded in the demultiplexing fields of the RSAP-IP session. As compared with the host of RSA-IP, a RSAP-IP host can at least lease one address/port combination from the same or different RSIP gateways.

#### 2.1.3.3 Format of RSIP Message

There are three mandatory fields contained in a RSIP message. They are “version”, “message type” and “overall length”. Besides, there are one or more required parameters following them. Moreover, zero or more optional parameters are defined to follow the required parameters in turn. The order for all required parameters is exactly specified in the RFC 3103. However, optional parameters can be located in any order.

We can specify the format of RSIP message by the following Figure 6.

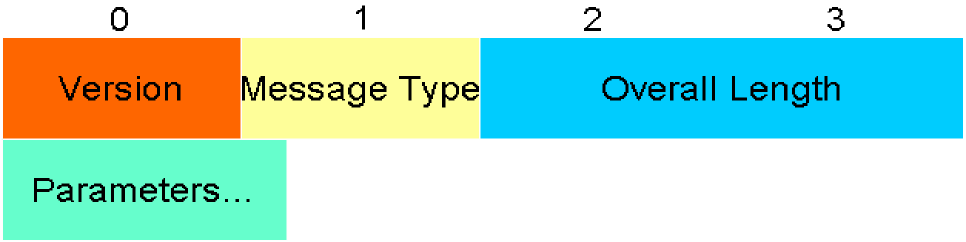

**Figure 6:** Format of RSIP Message

In Figure 6, the “version” field is a single byte which is used to specify the version number the RSIP message. The “message type” field is also a single byte. We use this field to specify the message contained in the packet. There are two bytes for “overall length” field which is used to contain the number of bytes in the RSIP message.

The types of RSIP message are defined in BNF. Different types of RSIP messages can be used to achieve different functionalities in the operation processes of RSIP. However, not all message types need to be used to make the compliance for RSIP.

### 2.1.3.4 Formats of RSIP Parameters

According to definition in RFC 3103 [35], there are one or more parameters that encode the information transmitted between the RSIP host and RSIP gateway can be contained by a RSIP message. All RSIP parameters follow a format of "type-length-value" which is formed by 1-byte type followed by 2-bytes length followed by the value of length in byte. We can divide the "value" field into some other fields in accordance with the types of the parameter. Moreover, the length field only encodes the number of bytes in value field in stead of the overall number of bytes in the parameter.

There are 12 types of parameters about the RSIP message. Each of them has been defined with a unique format. Format of address, port and flow policy are illustrated here as examples.

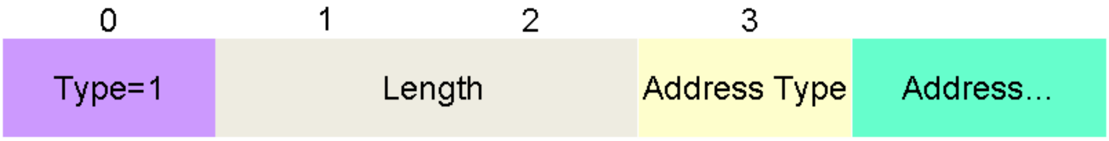

**Figure 7:** Address Format of RSIP message

In Figure 7, "Type=1" represents the format of RSIP message's address. "Address Type" field is used to indicate the type of the address and its length is one byte.

Similarly, we can illustrate the format of RSIP message port in Figure 8 shown as follow:

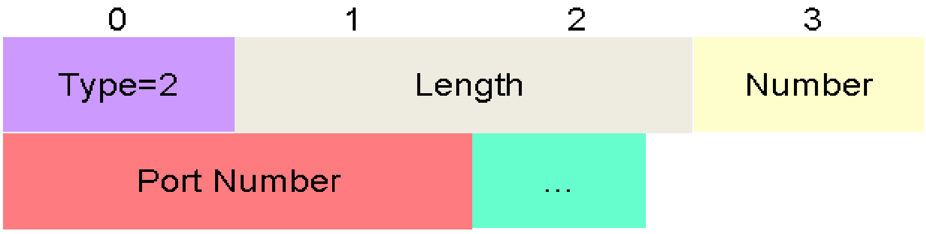

**Figure 8:** Port Format of RSIP message

In Figure 8, the port parameter can encode zero or more TCP or UDP ports. If we specify a single port, the value of "Number" should be one. It means there is only one port field following the field of "Number". However, if there is more than one port specified, the field of "Number" will indicate the total number of ports included in the RSIP message. The number of port fields should be equal to the value of the "Number" field. In addition, if a given ports parameter is applied by "micro-flow" based policy, the ports parameter must only contain one port field.

The parameters of flow policy can be used to specify both the local flow policy and remote flow policy. We detail the format of flow policy in the following Figure 9.

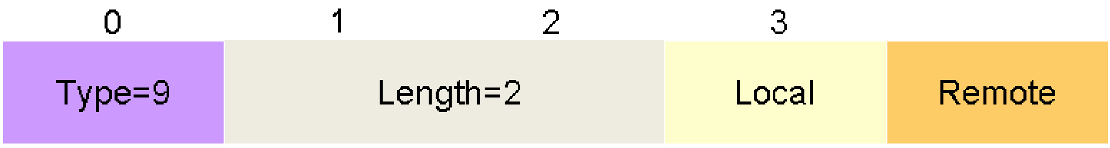

**Figure 9:** Flow Policy Format of RSIP message

Because of the definition our project, we only are only interested in the remote flow policy. If the value of “remote” field is two, we will get the micro flows.

### 2.1.3.5 Remote Flow Policy and State of RSIP

As the remote access we defined for NAT, the RSIP should also own the ability to definitely control which local addresses and ports are used to communicate with remote addresses and ports. Moreover, we can implement the remote flow policy in both ingress and egress directions of the RSIP gateway. Remote flow policy can define the level of detail that a RSIP host should specify addressing parameters of a remote RSIP host before the RSIP gateway allows one RSIP host to communicate with another RSIP host. With the different level of detail, we can also divide the remote flow policy of RSIP into three forms [35].

First of all, if there is no policy for the flow of RSIP, the RSIP host will be able to use the allocated parameters to communicate with any remote host without notifying the RSIP gateway.

The second form of remote flow policy of RSIP is named “macro-flow” policy. When this kind of policy is realized, the RSIP host will identify the address of remote RSIP host that it wants to communicate with as part of its request for local addressing parameters. When the request is accepted, the RSIP host that sent the request must use the specified local parameters only with the remote address. In addition, the RSIP host must not communicate with the remote address by any local parameters except the allocated parameters. However, we can use the RSIP host to connect with any port number of the same remote RSIP host.

If the RSIP host can identify the address and port of the remote host that it wants to communicate with as part of its request for local addressing parameters, we will be able to name this kind of remote flow policy “micro-flow” policy. When the request is accepted by RSIP gateway, the RSIP host that sent the request must use the specified local parameters only with only with the remote address and port.

Additionally, this kind of RSIP host must not communicate with the remote address and port by any local parameters except the allocated parameters.

On the other hand, additional flow state is also necessary if “micro-flow” or “macro-flow” based remote policy is used. Thus, if “macro-flow” based remote policy is used, the state of remote host’s address must be maintained. If “micro-flow” based remote policy is used, the state of remote host’s address and remote host’s port must be maintained.

## 2.2 Session Initiation Protocol (SIP)

Session Initiation Protocol is a signalling protocol for Internet. It is used to establish interactive communication session with reasonable extensibility and flexibility. SIP is a smart and general-purpose tool for creating, modifying and terminating sessions without any dependency on the established session's type [36]. These sessions always work independently of those underlying transport protocols. For many protocols have been authorized to transmit various types of real-time multimedia session data such as voice and video, SIP will be forced to work together with these protocols by enabling user agents to discover other endpoints and to negotiate with a session for a characterization which they want to share. An infrastructure of proxy servers to which user agents may send registrations, invitations and other request need to be created for the purpose of locating prospective session participants and other relevant functions.

The content of SIP covers many aspects concerned with the establishing of communication. We summarize it in the chapter from RFC3261 [3] and separate it into three main parts.

### 2.2.1 Functionality of SIP

Nowadays, SIP is a mature technology in each aspect of IP. It can be widely used in various equipments which are provided by different vendors. SIP is a control protocol and operates on application layer. It can be used to establish, modify and terminate multimedia sessions such as Internet telephone calls. Participants may be invited to join the already existing session for multicast conferences. SIP may add media to an existing session or remove media from an existing session. SIP can support name mapping and redirection services transparently. Users can maintain a unique and externally observable identifier no matter where they are located with the aid of personal mobility which is supported by SIP.

There are five terms of operation on multimedia communication which are supported by SIP. We conclude them simply as follow: User location is used to determine the end system used for communication. User availability is used to determine if the called users are willing to join in communications. User capabilities are used to determine the media and its parameters. Session setup is used to establish session parameters at both endpoints of the call. Session management used to control the transfer and termination of sessions, modify session parameters and invoke services.

A complete multimedia architecture may be established by SIP together with other IETF protocols. These architectures will comprise RTP [37] (Real-time Transport

Protocol), RTSP [38] (Real-Time Streaming Protocol), MEGACO [39] (Media Gateway Control Protocol) and SDP [24] (Session Description Protocol). Although SIP need to be used together with other protocols to provide complete services for users, the fundamental functionality and operation of SIP is independent of any other protocols.

SIP can provide primitives which are used to implement different services instead of providing services. SIP can initiate session that uses some other conference control protocol instead of providing conference control service or regulating the management of a conference. SIP does not reserve capabilities for network resource because SIP message and the sessions made by SIP can go through any different networks. SIP also provides a group of security services which comprise denial-of-service prevention, authentication, integrity protection and encryption for privacy services. In addition, we should not neglect that SIP may be used for both IPv4 and IPv6.

### 2.2.2 Operation of SIP

In this chapter, an example is used to display the basic functions of SIP. Location of an endpoint, a requirement signal for communication, negotiation of session parameters for establishing of the session and disassembly of the session are totally shown in Figure 10 as follow:

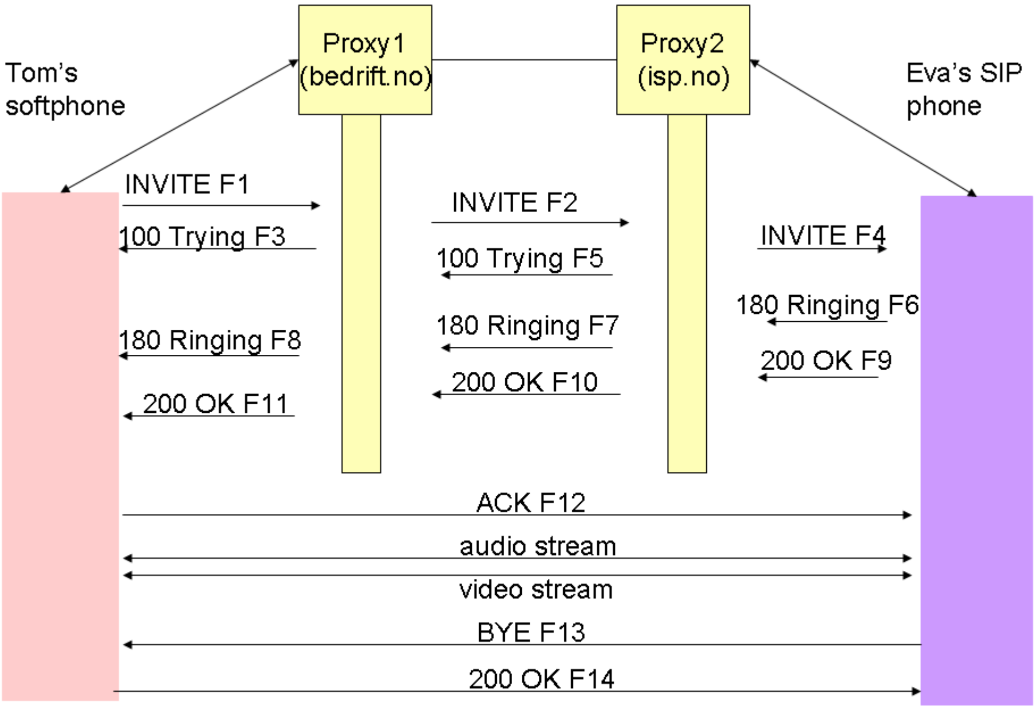

**Figure 10:** Establishing Process of SIP Session based on Fig. 1 of [RFC3261]

As the Figure 10 shows, there is a SIP message exchange between Tom and Eva. Tom uses a SIP application on his computer to call Eva on her SIP phone on the Internet. Proxy1 and Proxy2 can represent Tom and Eva respectively to improving the establishment of the session.

SIP Uniform Resource Identifier (URI) functions like the original IP address, however it looks like an email address. In addition, because the SIP URI can identify a user or a process rather than just a machine, another person or computer can use the SIP URI to find a user on multiple devices and follow when you move. Eva’s SIP URI is used by Tom to make the call. In Figure 10, Eva’s SIP URI is sip:eva@isp.no where isp.com means the domain of Eva’s SIP service provider. Tom’s SIP URI is sip:tom@bedrift.no. Tom can type in Eva’s URI or click on a hyperlink or an entry in an address book. A secure URI is also provided by SIP to make sure that secure and encrypted transport is used to take all SIP messages from Tom to the domain of Eva. The security mechanisms depend on the policy of the domain of Eva.

A form of request/response transaction model which is similar as HTTP is used by SIP. A request that invokes a particular method or function on the server and at least one response will be comprised in a transaction. The transaction in Figure 8 starts with Tom’s INVITE request addressed to Eva’s SIP URI. INVITE is a SIP method that specifies the action that Tom wants Eva to make. A few of header fields which provide additional information about a message are included in INVITE request. An unique identifier for the call, Eva’s address, Tom’s address and information about the type of session that Tom want to transmit with Eva is exhibited in an INVITE. The text-encoded message which reflects the specific process of SIP session establishment is shown as follow:

```
INVITE sip: eva@isp.no SIP/2.0
Via: SIP/2.0/UDP bedrift.no:5060
From: Tom<sip:tom@bedrift.no>;tag=111
To: Eva<sip:eva@isp.no>
Call-ID:12345678@bedrift.no
CSeq: 1 INVITE
Subject: Hello
Contact: Tom<sip:tom@bedrift.no>
Content-Type: application/sdp
Content-length: 111
V=0
O=Tom 2873397496 2873404696 IN IP4 bedrift.no
S=Session SDP
C=IN IP4 195.35.78.162
t=0 0
M=audio 49100 RTP/AVP 0
A=rtpmap: 0 PCMU/8000
M=video 52154 RTP/AVP 21
A=rtpmap: 21 H263-1998/90000
```

“INVITE” is the method name and the other part of the first line is destination’s SIP address. “Via” include the address at which Tom is expecting to receive responses to this request and the branch parameter. “From” comprise the addresses of Tom. “To” comprises the address of Eva. Tag parameter contained in third line is used for identification purposes. A globally unique identifier for this call is contained in “Call-ID”. The “Call-ID” is generated by the combination of a random string and Tom’s name or IP address. An integer and a method name are contained in “CSeq”. The “CSeq” number represents the increment of each new request within a dialog. “Contact” is composed of a username at a fully qualified domain name (FQDN). It always contains a SIP or SIP secure URI which can represent a direct route to touch Tom. A description of the message body should be contained in “Content-Type”. An octet count of message body needs to be comprised in “Content-Length” to represent the size of the content. The complete set of SIP header fields along with the notes on syntax, meaning, and usage is list in RFC3261.

### 2.2.3 Real-time Transport Protocol (RTP)

RTP will be used for IP based multimedia communication after the connection is established by SIP. Multimedia stream is transmitted using RTP which is separate from SIP. Data packets do not follow the same path as the SIP packet. Audio and video data are digitized and compressed by RTP, and then sent out in the form of UDP packets. Limitation of human eyes or ears is used by compression schemes to save bandwidth [40].

RTP is standardized by IETF and used by ITU-T. It can be used to transmit real-time data such as voice and video [41]. RTP is design to separate data from control mechanisms. It has reasonable scalability and flexibility. The UDP-based RTP can prevent crippled voice quality which is caused by the attacking controlled by the flow of TCP. The packet transmitted with RTP is expressed simply in the following Figure 11:

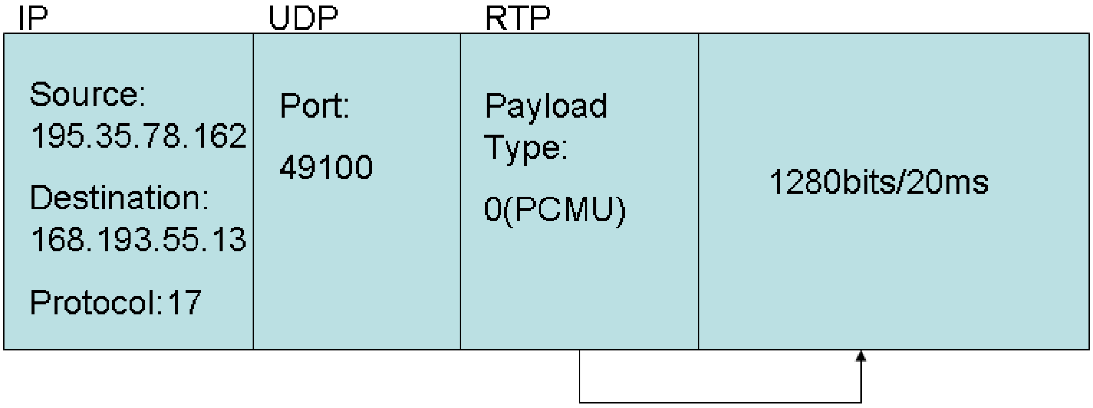

**Figure 11:** RTP Packet

From Figure 11, we can find source address, destination address and protocol type are included in IP header. Port number is stored in UDP header. Payload type is specified in RTP and can be divided into two types that are audio and video. The transmission speed is also shown in the payload. Except IP header and UDP header, all parts of the packet belong to payload.

### 2.2.4 Real-time Transport Control Protocol (RTCP)

RTCP can separate packets that are sent on different port numbers. It may help the end systems to exchange information about losses and delays between them [42]. RTCP is able to make packets to be sent in intervals which are determined by the number of end systems and the available bandwidth.

There are five types of useful information contained in RTCP. Sender Report includes information about sent data and synchronization timestamp. Receiver Report comprises information about received data, losses, jitter and delay. Source Description describes name, email, phone and identification. Bye is used to express leave indication. Application Defined Parts represent the parts that have experimental functions.

## 2.3 Simple Object Access Protocol (SOAP)

SOAP is a communication protocol that uses both XML and HTTP to provide a Web-based messaging and a mechanism of remote procedure call [43]. The function of XML is the expression of the messages contents. The main purpose of using HTTP is for transmitting the messages to the destination. SOAP is designed to support communication between applications via Internet. Because SOAP is language independent, it owns the excellent extensibility which makes it easy for users to take up. SOAP message is the fundamental unit which is transmitted between peers. We write SOAP messages in XML that makes SOAP platform-independent. Any system can send and receive SOAP message on condition that it has the capability of admitting XML documents. The content of SOAP will be summarized from SOAP Version 1.2 recommended by W3C [44].

### 2.3.1 SOAP Namespaces

XML namespaces are used to confirm the uniqueness among different XML elements and prevent elements that have the same name but come from different sources from collisions. Four namespaces used in SOAP are listed as follow:

- http://www.w3.org/2001/06/soap-envelope
  This is SOAP envelope namespace. We take the envelope as the outermost container for SOAP messages.

- http://www.w3.org/2001/06/soap-encoding
  This is SOAP encoding namespace. The rules of encoding are used to specify how to encode data types in SOAP messages.

- http://www.w3.org/2001/XMLSchema-datatypes
  This is XML schema for data types. We consider it as the basic set of types for SOAP messages.

- http://www.w3.org/2001/XMLSchema-instance
  This is XML schema for instances. Several attributes used in any XML documents are defined here.

### 2.3.2 SOAP Terminology

According to the principle of concepts, The SOAP terminology is divided into three different types as follow:

#### 2.3.2.1 Protocol Concepts

- SOAP: The protocols that are used to govern the format and processing rules of a SOAP message. Interactions among SOAP nodes are comprised in these protocols. The generating and accepting of messages that are used to exchange information along a SOAP message path are realized by SOAP.

- SOAP node: According to the set of protocols defined by the recommendation, SOAP node realizes the processing logic and makes it necessary for the transmitting, receiving, processing and/or relaying of a SOAP message. SOAP nodes should enforce the rules that are used to control the exchange of SOAP messages. At least one SOAP binding needs to be used when the SOAP nodes need the services from the underlying protocols.

- SOAP role: The function that a SOAP receiver wants to supply in processing a message. At least one role can be acted by a SOAP receiver.

- SOAP binding: The rules used to transfer a SOAP message within or on top of another underlying protocol in order to make the exchange. SOAP binding can carry a SOAP message with in a HTTP entity-body or over a TCP stream.

- SOAP feature: The SOAP message framework’s extension such as Reliability and Security.

- SOAP module: The specification that includes combined syntax and semantics of SOAP header blocks. A SOAP module can realize several SOAP features.

- SOAP message exchange pattern (MEP): The moulding board that contains the exchange of SOAP messages between SOAP nodes which are enabled by at least by one underlying SOAP protocol binding. It is also an example of a SOAP feature.

- SOAP application: The entity that can produce, consume and even act upon SOAP messages with the method according with the SOAP processing model.

#### 2.3.2.2 Data Encapsulation Concepts

- SOAP message: The basic unit transmitted between SOAP nodes.
- SOAP envelope: The outermost container for SOAP messages.
- SOAP header: The collection of SOAP header blocks that may be targeted at any SOAP receiver that follows the SOAP message path.
- SOAP header block: The information element that can be used to constrain data that constitutes a single countable unit within the SOAP header. Name of head block expanded by XML is used to identify the type of a SOAP header block.
- SOAP body: The collection of information elements that are sent to an ultimate SOAP receiver in the SOAP message path.
- SOAP fault: The information item that stores the fault information generated by a SOAP node.

#### 2.3.2.3 Message Sender and Receiver Concepts

- SOAP sender: The SOAP node used to transmit a SOAP message.
- SOAP receiver: The SOAP node used to accept a SOAP message.
- SOAP message path: A group of SOAP nodes through which a SOAP message may pass. It can be formed by an initial SOAP sender, some SOAP intermediaries and a ultimate SOAP receiver.
- Initial SOAP sender: The starting point of a SOAP message path where the SOAP sender originates a SOAP message.
- SOAP intermediary: A SOAP intermediary can be both a SOAP sender and SOAP receiver. It is used to transmitted SOAP header blocks which targeted it and forward a SOAP message to an ultimate SOAP receiver finally.
- Ultimate SOAP receiver: The final destination of a SOAP message. It is used to receive the contents of the SOAP body and any SOAP header blocks targeted at it. An ultimate SOAP receiver can act as a SOAP intermediary for the same SOAP message.

### 2.3.3 SOAP Binding

A SOAP binding represents the formal set of rules that are used to carry a SOAP message within or over another protocol for the purpose of exchange. The general rules for the specification of protocol bindings are provided by the SOAP protocol binding framework. The framework also includes the relationship between bindings and SOAP nodes that implement these bindings. Five areas of a SOAP binding specification is listed as follow:

- Features provided by a binding.
- Method about using underlying protocol services to transmit SOAP message information sets.
- Method about using underlying protocol services to support the agreement made by the features which are supported by the binding.
- The handling of all potential failures that may be anticipated within the binding.
- The requirements for making a consistent implementation of the binding that is specified.

Beside the specification of a SOAP binding, the main goals of binding framework needs to achieved are list as follow:

- The requirements and concepts that are common to all binding specifications can be set out.
- Similar description of situations where multiple binding support common features needs to be improved by the binding framework in order to promote reuse across bindings.
- Consistency in the specification of optional features can be enhanced by the binding framework.

A given optional feature such as reliable delivery using different means can be provided by two bindings. One of them may exploit an underlying protocol that facilitates the feature directly and the other one can supply the necessary logic by itself such as reliability which is achieved through logging and retransmission.

### 2.3.4 SOAP Message Construct

A SOAP message can be specified an XML information set. The comment, element, attribute, namespace and character information items of the XML information set may be serialized as XML 1.0. The SOAP envelope element information item must be included in a SOAP message information set to act as the children property. The content that is not directly serially using XML is permitted by the information set recommendation. The XML information set of a SOAP message must be correspond to an XML 1.0 serialization but must not include a document type declaration information item. Processing instruction information items must not be contained in the SOAP message transmitted by initial SOAP senders. During the process of relaying, processing instruction information items are not allowed to be inserted in SOAP messages by SOAP intermediaries. A SOAP fault may be generated when a SOAP message containing a processing instruction information item is accepted by SOAP receivers.

Besides the element information items which are defined as allowable members of their children property, element information items can also own zero or more character information item children. The character code belonging to the character information item should be amongst the white space characters which are defined by XML 1.0. Comment information items can act as children and/or descendants of the element information item instead of before or after the element information item. The occasion for adding and/or removing the comment information items is selected according some restrictions in the processing model.

#### 2.3.4.1 SOAP Envelope

The SOAP envelope contains a local name of the envelope and a namespace name which is defined in "http://www.w3.org/2001/06/soap-envelope". The attributes property contains zero or more namespace-qualified attribute information items. SOAP envelope is a well-formed XML document that follows the standard HTTP headers in the body of the HTTP message. The envelope element is comprised of two child elements. They are header element and body element. The SOAP header element is optional, but the SOAP body element is mandatory. Both the SOAP header element and the SOAP body element contain SOAP blocks which are formed by valid XML data. The block within the SOAP header is named a header block and the block within a SOAP body is named a body block. The structure of SOAP envelope is illustrated in Figure 12 shown as follow:

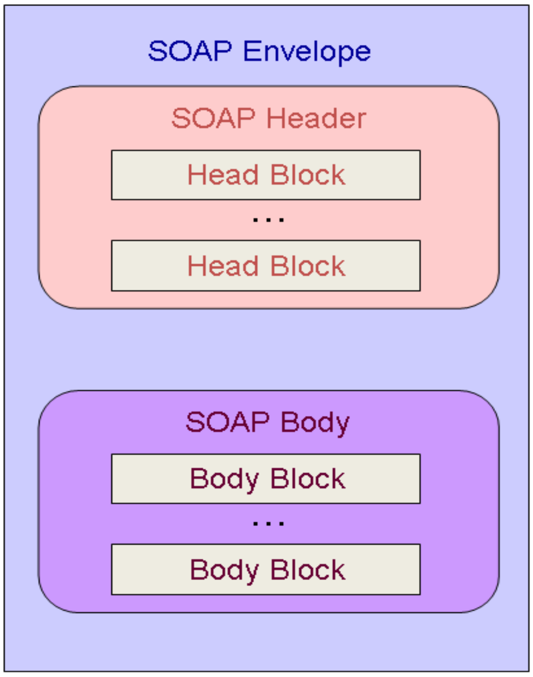

**Figure 12:** Composition of a SOAP Envelope

#### 2.3.4.2 SOAP Header

The SOAP header is an optional element which can be used to take supplementary for authentication, exchange and payments. It provides a mechanism for the extension of a SOAP message in a decentralized and modular way. The SOAP header contains a local name of the header and a namespace name which is defined in "http://www.w3.org/2001/06/soap-envelope". The attributes property contains zero or more namespace-qualified attribute information items. The children property is comprised of zero or more namespace-qualified element information items. Each of these children element information items is named a SOAP header block.

#### 2.3.4.3 SOAP Body

We can take SOAP body as a collection of zero or more SOAP blocks. It provides a mechanism which is used to send information to an ultimate SOAP receiver.

The SOAP body contains a local name of the body and a namespace name which is defined in "http://www.w3.org/2003/05/soap-envelope". The attributes property contains zero or more namespace-qualified attribute information items. The children property is comprised of zero or more namespace-qualified element information items. The body element information item can own any number of character information item children. The code of these character information items should be constrained amongst the white space characters which are defined by XML 1.0. The SOAP body element includes the core of the message that is a remote method call and its associated arguments, a method response, or error information for failed calls [18].

#### 2.3.4.4 SOAP Fault

The function of a SOAP fault is to carry error information within a SOAP message. The fault element information item contains a local name of the fault and a namespace name which is defined in "http://www.w3.org/2003/05/soap-envelope". It always has at least two element information items in its children property. We list five types of element information item as follow:

1. The mandatory code element information item.
2. The mandatory reason element information item.
3. The optional node element information item.
4. The optional role element information item.
5. The optional detail element information item.

If a SOAP message is allowed to carry SOAP error information, it should be forced to include a single SOAP fault element information item which acts as the only child element information item of the SOAP body. Additional element information items are not allowed to exist in the SOAP body when a fault is generated because they can make the message have no SOAP-defined semantics.

## 2.4 Universal Plug and Play (UPnP)

UPnP technology is used by devices to connect with each other and establish a peer-to-peer network. It is based on UPnP device architecture that can be used to connect networked devices, such as PCs, entertainment equipments, and intelligent appliances [43] together. UPnP defines the conventions for the description of devices with adhered standards and these devices' services. UPnP architecture is based on the mechanism that leverages existing standards such as TCP/IP, HTTP, and XML. Each UPnP technology-enabled device can implement the standardized protocols provided by UPnP to realize the services of discovery, control, and data transfer between UPnP devices. Any common OS and hardware platform can provide services for UPnP. Additionally, it can cooperate with almost any type of physical networking media which may be wired or wireless.

### 2.4.1 Technical Foundation of UPnP

#### 2.4.1.1 Uniform Resource Identifiers

The information owned and linked by the World Wide Web is in the form of resources. The resources can either be physical resources or abstract resources. A Uniform Resource Identifier (URI) can be taken as a compact string of characters which is used to identify a resource on the Web by a conceptual mapping from an identifier to a resource on the Web. URIs can also be divided into Uniform Resource Locators and Uniform Resource Names [45].

- Uniform Resource Locators (URL): URL is used to identify a resource by specifying its location instead of identifying the resource by name or some other attribute. URI schemes are always named after protocols.

- Uniform Resource Names (URN): URN is used to mark a resource that needs to remain globally unique and persistent even when the resource becomes useless. Once a URN is used to name a resource, it will never be used again to name other resources. The name of resource which is provided by URN can be drawn from one of defined namespaces. Each of the namespaces has its own structure and procedures for allocating names.

Both URL and URN have many uses within UPnP devices. URLs are used by UPnP devices to define locations which are used by control points to send requests to the UPnP devices. URNs are always used by UPnP devices to uniquely identify both the type of the device and the particular instance of device. UPnP services also use URNs to identify both the type and name of the service.

#### 2.4.1.2 IP Multicast

IP multicast can provide an efficient means to implement a group communication model which is necessary to some phases of UPnP. When the multicast is used, a host can transmit data to many other hosts simultaneously on the network without transmitting a copy of the data to each host separately [46]. The components of IP multicast are listed as follow:

- A group of hosts that can receive data.
- A mechanism that controls hosts' joining and leaving of the group.
- Multicast-capable routers used for managing and relaying group membership information and forwarding multicast traffic efficiently.
- Protocols and APIs for the applications of creating, sending and managing data.

IP multicast is used as transport to send messages to many recipients simultaneously by the UPnP architecture. Control points can use an IP multicast-based discovery mechanism to find out which devices are on the local network. State change event messages from services can also be received by registered control points over IP multicast. Devices can announce their presence on the network by the messages carried over IP multicast.

#### 2.4.1.3 Hypertext Transfer Protocol (HTTP) 1.0

The HTTP which starts with a simple text-based request and response protocol designed to transmit various web-based resources has been a ubiquitous transport protocol. The commands of HTTP permit a web browser to make simple interaction with web servers such as retrieving web pages and transmitting data to a server. The use of HTTP grew rapidly because of its simplicity and the ability of pass through most network administrators' routers. HTTP has become a generic, stateless, extensible and object-oriented protocol. It can be found in many tasks that are from name servers to distributed object management systems.

A simple request/response model of communication is used in HTTP [47]. Firstly, a connection between a HTTP client and a HTTP server is established by the HTTP client that sends a request message to the HTTP server. Then a response message which usually includes the resource requested by the client is sent from the HTTP server to the HTTP client. When the response is delivered, the HTTP 1.0 server will close the connection. The HTTP server maintains no connection state and makes HTTP a stateless protocol. However, we can maintain the state at the HTTP client in the form of cookies.

In principle, we can divide the structure of HTTP transactions into request message part and response message part. But actually the format of HTTP request and response messages looks similar.

#### 2.4.1.4 Hypertext Transfer Protocol (HTTP) 1.1

With the gradual evolvement of HTTP, HTTP 1.1 was produced to satisfy the new needs of the transfer and overcome the shortcoming of HTTP 1.0. We can take HTTP 1.1 as an extension of HTTP 1.0 [48]. It improves the efficiency of server response with the following methods [49]:

- Multiple transactions are permitted to be made over a single persistent connection.
- Cache is supported to reduce required bandwidth.
- Chunked encoding technique that allows a response to be sent before its total length is known has been used to support dynamically-generated pages.
- The efficiency of using IP addresses is improved by HTTP 1.1 via allowing a single IP address to serve multiple domains.

#### 2.4.1.5 HTTP over UDP (HTTPU) and HTTP over Multicast UDP (HTTPMU)

HTTP is a protocol carried over TCP which is a stream-oriented protocol between two communicating peers. Many other kinds of communication may not be modelled efficiently such as transmitting a single message to many recipients. A host has to send the same message to all recipients separately.

HTTPU achieves the benefits of HTTP along with the simplicity of UDP. A host may transmit an HTTP formatted message to another host without the expense of establishing a new TCP connection.

HTTPMU allows transmitting HTTP messages to many recipients simultaneously. HTTPMU may enable a group communication model which uses HTTP-style request/response messages.

#### 2.4.1.6 Extensible Markup Language (XML)

Customized markup languages are allowed to be developed by XML which specifies the structure of data and the relationship of various elements. Because XML documents include self-describing information about the rules which are used to compose them, data exchange is simple to XML. The W3C provides specification of XML 1.0 which describes the details of XML.

Markup and character data are used to compose XML documents. Markup part of the XML document can supply the structure such as the start and end tags which can delimit the elements of an XML document to the XML document. Markup part also includes the comments used to describe the document and the processing instructions that are used to provide the XML processor with the processing method of the document. All of the data that are not belonging to markup part can be taken as character data.

XML documents must be formed syntactically. A well-formed XML document should follow the syntax established for it by the W3C. According to the definition of XML, we can divide XML document into three main parts: a prolog, a root element and a miscellaneous part. A document prolog is the beginning part of an XML documents. It may include an XML declaration (optional), processing instructions, comments, white space and document type declarations. The root element comes directly after the document’s prolog. We can use it to hold all of the other elements of the document.

UPnP device architecture may use XML to describe UPnP devices and the services provided by these UPnP devices. UPnP action requests and responses should be carried by XML. In addition, we can use XML to specify the format of event that is sent from services to control points [50].

## 2.4.2 UPnP Terminology

### 2.4.2.1 UPnP device

We can define an UPnP device as an entity on the network that can implement the required protocols of UPnP architecture. An UPnP device can either be dedicated physical device or a logical device. The main function of an UPnP device is to provide information on the description of itself such as the model name and manufacturer. An UPnP device can also contain other devices. In UPnP terminology, the top-level device is defined as the root device and the contained device is defined as embedded device. A root device may contain a number of embedded devices. In addition, an UPnP-enabled device can invoke more than one root device.

### 2.4.2.2 Service

A service is formed by a unit of functionality which is executed by a device. An UPnP device can contain zero or more services. There are a group of actions for each service and the service is used to group the actions provided by UPnP devices. Each action owns a name, optional input and output arguments and a return value that provides the result of the action. In addition, each service can own a state table which is used to group its state variables. There are a name, a type and a value for each state variable. The specifics of a service can be found in UPnP Forum Working Committee [20]. The UPnP Forum also standardizes the set of services that particular types of devices should support. A device can only implement the services that are determined by the type of the device. The structure concerning the relationship of

UPnP devices, services and actions can be illustrated in the following figure:

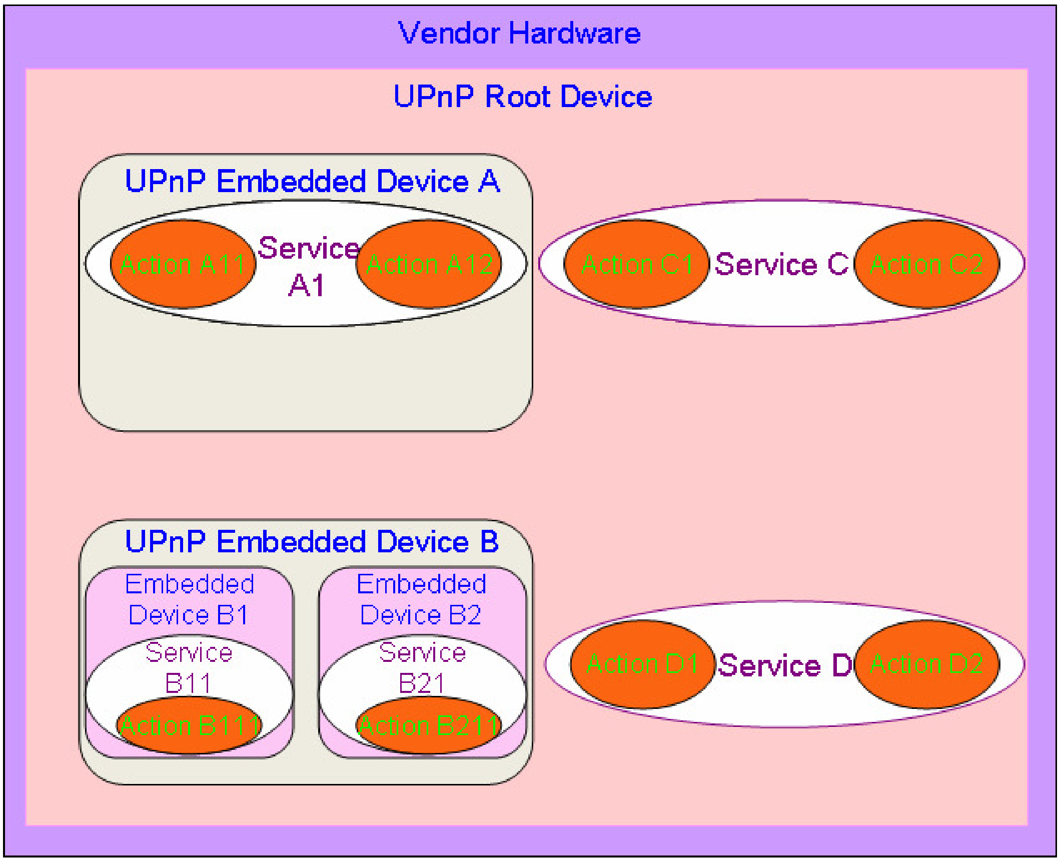

**Figure 13:** Relationship of UPnP Devices, Services and Actions

### 2.4.2.3 Control Point

A control point can be defined as an entity on the network that cooperates with the functionality of a device. We can take any entity that invokes the services from an UPnP device as a control point. Control points are used to invoke actions on services, provide input parameters and receive output parameters and a return value. In practice, we can build control point functionality in UPnP technology-enabled devices in order to make them invoke the services or monitor the state changes of other devices. A peer-to-peer network where devices make use of each other's services can be established with the function of built in control point. We can exhibit the invoking of service from a built in control point with Figure 14.

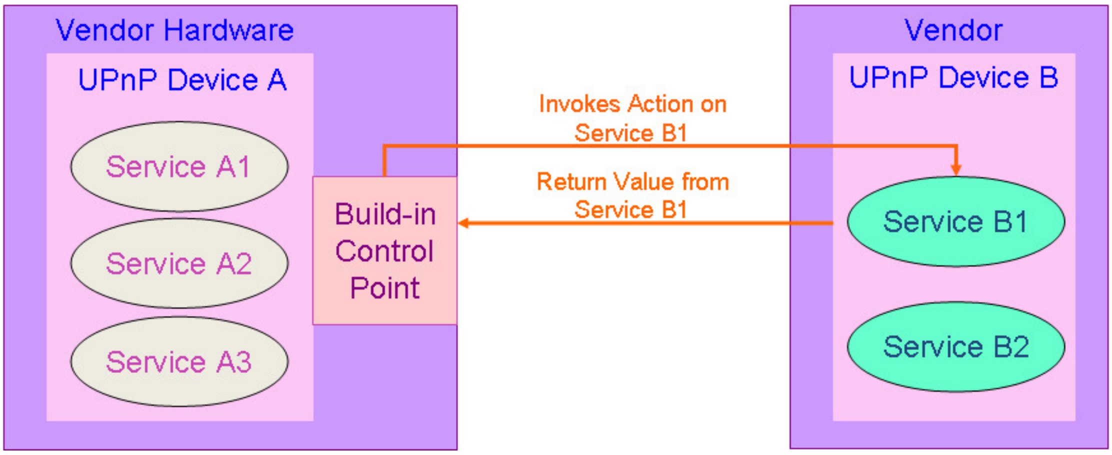

**Figure 14:** Build-in Control Point Invoking an Action

Besides, control point can make a request for the notification of the state variable changes from the service. After a change to one of the state variables is detected by a service, the service will notify the change to all of the registered control points. The processes for achieving the notification of state variable changes can be detailed in Figure 15.

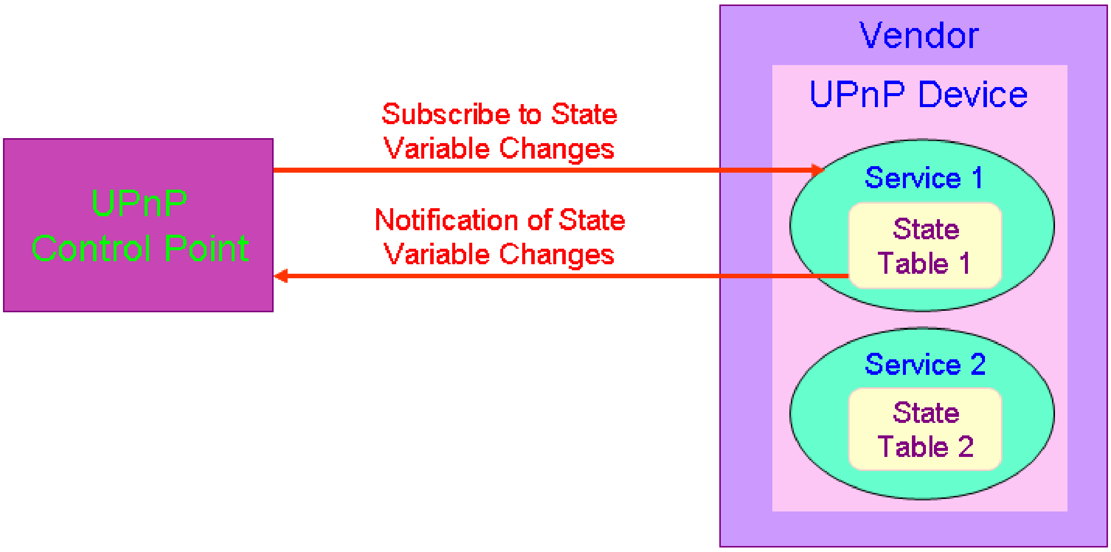

**Figure 15:** Notification of State Variable Changes

### 2.4.3 UPnP Protocols

#### 2.4.3.1 Addressing

Addressing is the foundation of UPnP networking. With the addressing process, a device can automatically acquire its IP address. Addressing is the first step of UPnP operation. Devices are allowed to join the network and connect with other UPnP devices in the addressing process. We can build addressing protocols into UPnP devices. These UPnP devices will be able to join an IP network dynamically and get an address without the configuration of the user.

Another main function of addressing is to evaluate whether a device is operating in an unmanaged or managed network. An unmanaged network can be taken as Ad hoc network. There are no pre-existing infrastructure devices in unmanaged network. The network is made up by the network nodes themselves. A managed network owns its infrastructure devices. Devices can get an IP address from a DHCP server on the network.

#### 2.4.3.2 Discovery

With the discovery process, control points can find devices and services. The related information can be retrieved from them by the control points. When a device gets an IP address, the device will advertise itself and its related services on the network. In devices' advertisements and discovery response, there is a URL for their device description document. The URL can provide control points with the information that is used to get the device and service descriptions by control points. The control points can also learn all about the device and the services which are offered by URL.
If a control point discovers a device and get its device and service description documents, the control point will be able to control the device and subscribe to the events that are sourced by the device's services or get the presentation page of the device.

#### 2.4.3.3 Description

Description is used by devices to list the functionality provided by them. The description of devices and their services are included in XML-based description documents. These devices description documents are used to store device information such as manufacturer, make, model and serial number. A group of services provided by the device and a group of embedded devices are also stored in these documents. Besides the device description document, there is a service description document. The detailed information about the service is contained in the service description document. The return value, the actions provided by the service, and the parameters of the actions are also included in the service description document.

#### 2.4.3.4 Control

Control in the UPnP operation is used by control points to invoke the actions provided by a device's services. If a control message is received by a device's service, the device's service will follow the command of the control message. The operation may change the state of the device and lead to another UPnP phase named eventing.

#### 2.4.3.5 Eventing

The main function of eventing is making control points to monitor state change in devices. A publisher/subscriber model where control points can order a service provided by a device is used by the UPnP architecture. The information about changes in state variables can be sent to all of the registered control points by the device's service. The method of the responding to state changes makes a dynamic, responsive and event-driven system for an UPnP network of devices.

#### 2.4.3.6 Presentation

Presentation is the final step in UPnP networking. If a device has a URL for presentation, the device will be able to present a browser-based user interface for manual user control and the viewing of device status. Each device should contain a web server and be able to send a web page to browser-based clients. The web page can act as the manual interface for device which is different from the programmatic control interface of the device. We can use the browser-based interface to control the device, change operational parameters and scan information of device and service.

### 2.4.4 UPnP Audio/Video (UPnP A/V)

According to the standard UPnP technology model, UPnP A/V can use a single control point to coordinate activities between multiple logical devices. Users can communicate only with the UPnP A/V control point which is used to discover and configure the other UPnP A/V devices on the network instead of interacting with and configuring many other distinct devices to make them communicate with each other.

The basic UPnP A/V architecture defines two device types: the Media Server and the Media Render. They are associated with control point to form a triangle of interacting devices shown in Figure 16 as follow:

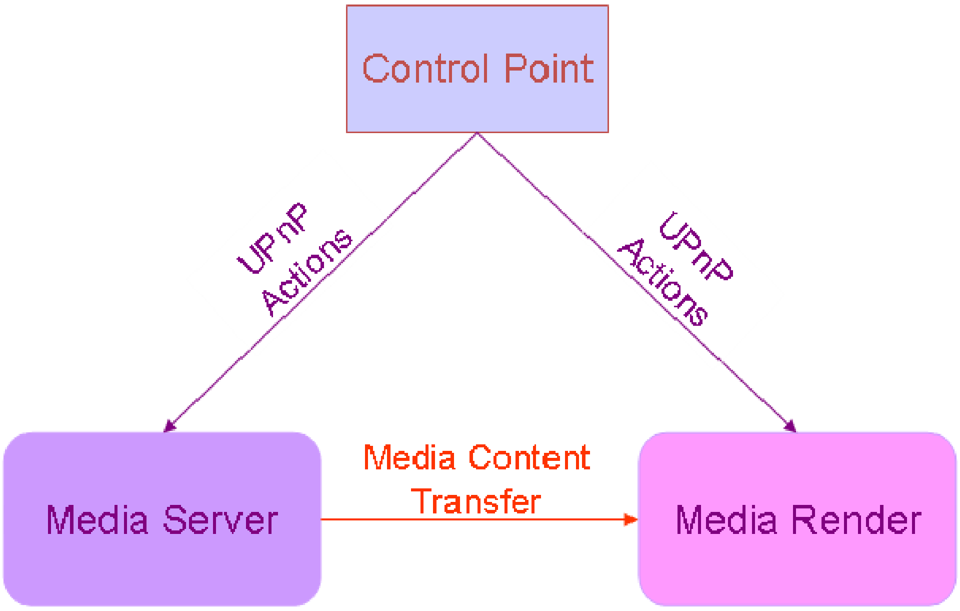

**Figure 16:** The Architecture of UPnP A/V

In Figure 16, the Media Server is used to store the content and the Media Render is used to render the content received from the Media Server. The UPnP A/V Control Point is used to discover Media Servers and Media Renders on the network. With the defined UPnP Actions, a Media Server and a Media Render can be connected by the Control Point. The Media Server can transmit media content to the Media Render directly without going through the UPnP A/V Control Point.

There are three main goals that UPnP A/V was designed to achieve. The first one is media and transfer protocol agnostic. It is satisfied by the standard UPnP A/V actions in Media Server and Media Render devices. The second one is direct source-to-sink transfer of content. According to the three-way architecture of UPnP A/V, the renders of content can connect directly to the source of the content. The final one is the ability to support A/V devices of all complexities. It is achieved by a minimal set of services and actions that each device must support.

With the definition in "http://www.upnp.org/", there are a set of UPnP A/V specifications and each of them defines a set of required and optional state variables and actions. We list the set of UPnP A/V specifications as follow:

- UPnP AV Architecture.
- Media Server Device Template.
- Media Render Device Template.
- Rendering Control Service Template.
- Connection Manager Service Template.
- AV Transport Service Template.
- Content Directory Service Template.

### 2.4.5 Remote Access of UPnP A/V

The operation processes of UPnP A/V in remote access are concluded in Figure 17 shown as follow:

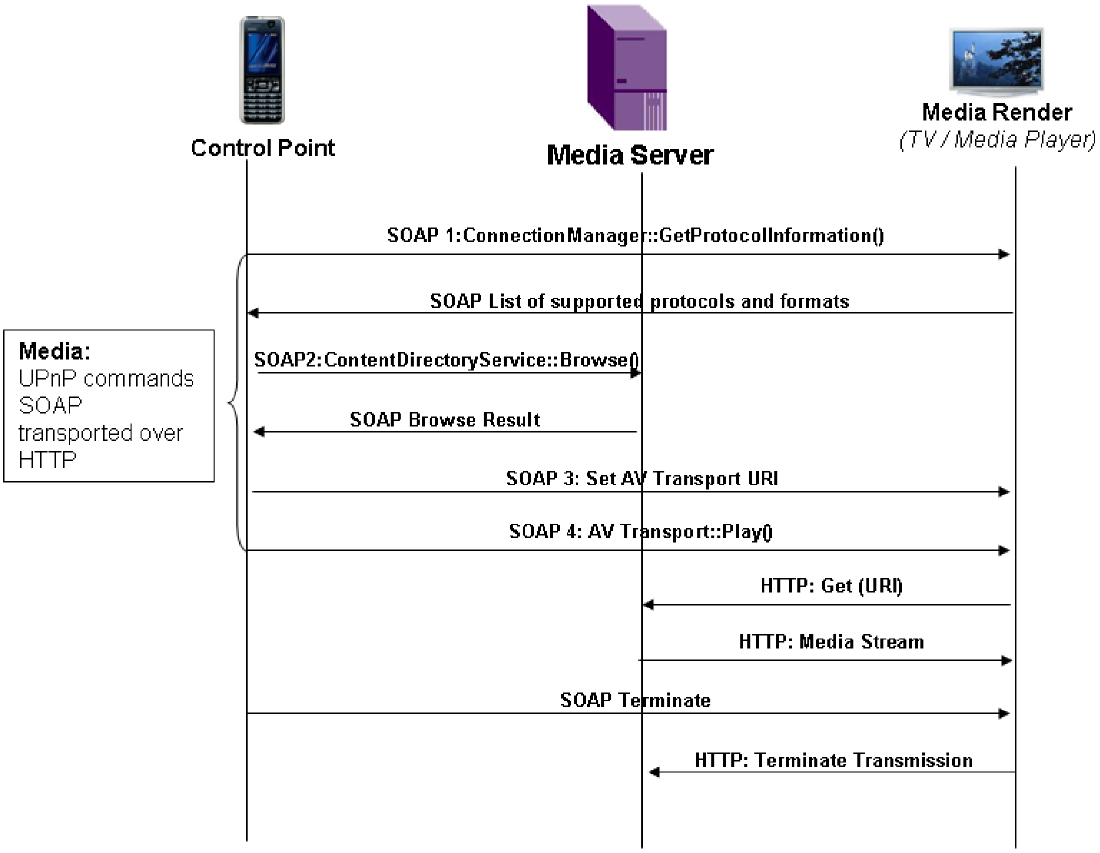
Control Point
Media Server
Media Render
(TV / Media Player)
Media:
UPnP commands
SOAP
transported over
HTTP
SOAP 1:ConnectionManager::GetProtocolInformation()
SOAP List of supported protocols and formats
SOAP2:ContentDirectoryService::Browse()
SOAP Browse Result
SOAP 3: Set AV Transport URI
SOAP 4: AV Transport::Play()
HTTP: Get (URI)
HTTP: Media Stream
SOAP Terminate
HTTP: Terminate Transmission

**Figure 17:** Remote Access Procedure

We assume the media render and the media server in Figure 17 are located in different private networks while the control point and the media render are located in the same private network. The main part of the remote access procedure should be concentrated on the operation of control point. It realizes all the steps of the connection establishment between the media render and media server.

# 3 Solution Proposals

According to the problem description and project definition, we can conclude the problem that UNSAF encounters into how a host behind a private network discovers the correct address of another host behind a different private network. These hosts can be assumed as some functional nodes such as a media render or a media server. From the sequence diagram described in Figure 17, we can find that the essential ability for a media render to connect with a media server is from the address information received at a control point. Moreover, the information receiving of the control point is after the browse for content directory service. The address information should be gained by the control point within the browsing process. If the address information can make the media render connect to the media server located in a different private network, the control point will send it directly to the media render via WLAN. In addition, if the address information enables the media render to communicate with the media server located in a different private network, the solution proposals for the problem defined in chapter one will be achieved.

## 3.1 Solution with IP Address Swapping

### 3.1.1 Overview

As the Figure 1 shows before, we introduce SIP into the communication processes of remote access. According to the functionality of SIP we described before, we can establish a new multimedia architecture with SIP. The main advantages of SIP used in our solutions are its long-term stable identifier named SIP uniform resource identifier (URI) and the separateness of the signaling and media planes. Although UNSAF is also applies to SIP, UNSAF with regards to SIP is partly solved via TURN, STUN and ICE etc. Therefore we can establish a suitable model as a solution for handling the problem of UNSAF across NAT with the assistance of SIP. Additionally, although SIP does not require any proxy to be part of the flow to work, SIP proxy can be very helpful and when it is added into the SIP networks.

### 3.1.2 Introduction of SIP RHN

Since media server is an UPnP device and can not identify the SIP invitation from control point, we need to introduce a remote helper node (RHN) into the private network where the media server located. We can take the RHN as a SIP user agent which can establish SIP connection with outside control point for these different nodes behind the same private network as the RHN. Because the RHN is inside a private network, it can get the port mapping of the nodes behind the same private network from the residential gateway. According to the content of port mapping information, each node behind the residential network has its unique port number. Each node behind the residential has its unique port mapping. We can take a media server as a common node with a NAT port and its corresponding local port. If the binding of the NAT port and the local port is known by the control point, the control point will be able to find the required node through its residential gateway. All in all, we can describe the operation processes of the solution with a logic model shown in Figure 18.

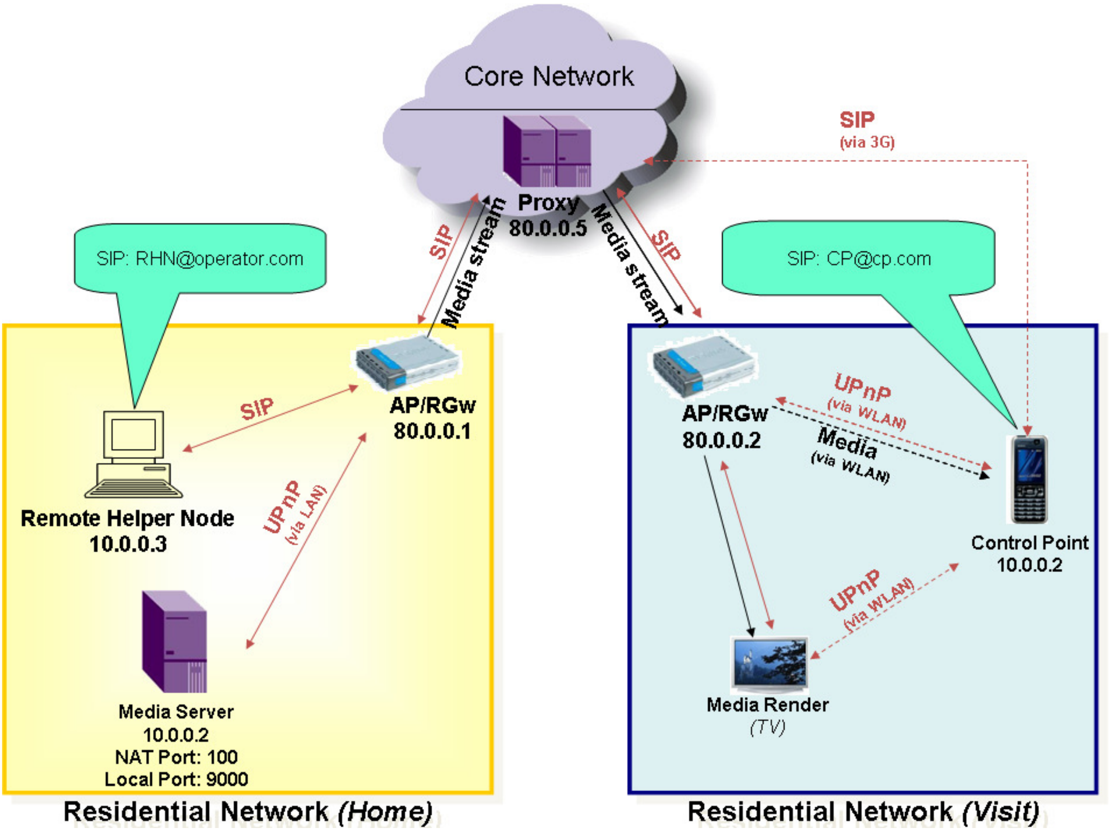

**Figure 18:** A Logic Model of the Solution with IP Address Swapping

In Figure 18, we can find the most obvious change from Figure 1 is the appearance of a remote helper node. We assume the RHN with its unique SIP address is behind a home residential network. A media server is assumed to exist in the same private network as the RHN. In addition, we assume the media server owns a local port number and a NAT port number. A proxy is added because it is the necessary condition for the SIP utilization in the logic model. In the visit residential network, we can also need to assign an IP address to the control point which has its corresponding SIP address. All the devices that are behind the residential networks are UPnP devices and can be connected via WLAN or LAN.

Moreover, the control point needs to send a request to the RHN for the port mapping information. When the INVITE for the port mapping information arrives at the RHN, the RHN needs to obtain the port mapping information from the residential gateway. Because the RHN is supported by both SIP and UPnP, the control point can get the port mapping information from the RHN via SIP. The SIP session establishment and the operation processes between two peers have been described clearly in chap 2.2 with figure and code.

### 3.1.3 Remote Access Procedure with SIP RHN

With the introduction of SIP RHN, we can detail the connection processes between the media render and media server in the following Figure 19.

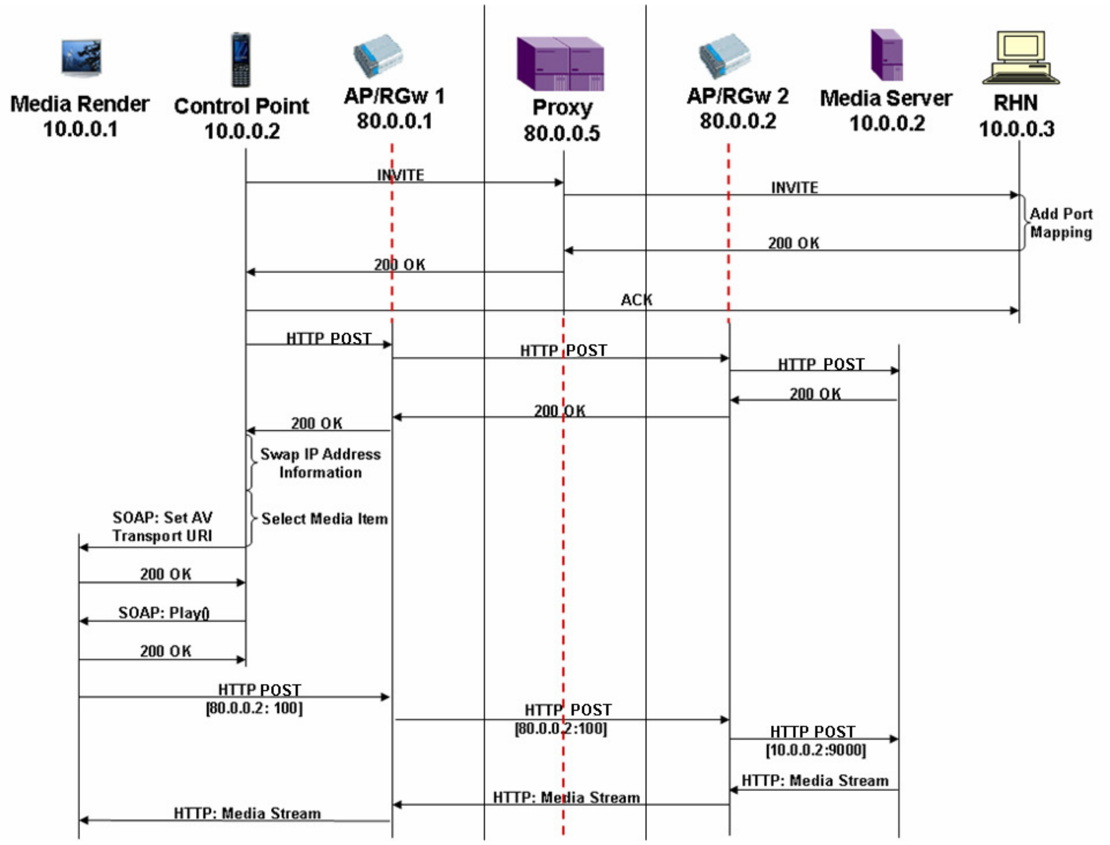

**Figure 19:** Connection Flow between Media Render and Media Server with IP Address Swapping

In Figure 19, the media render and the control point locate in the same residential network while the media server and the RHN locate in another residential network. We can find the operation process of adding port mapping is realized by the RHN when the RHN receives the INVITE from the control point. Since the control point supported SIP and UPnP, an INVITE commend can be initiated by the control point. Moreover, the processes of request and response have to go through the proxy. After the INVITE is received by the proxy, the proxy will retransmit it to the RHN which also has a unique SIP address. The port mapping information will be sent back to the control point from the RHN via SIP messages. When the response is received by the control point, ACK will be sent to the RHN to finish the successful processes of obtaining port mapping information. Additionally, information about both the device we need to use and its IP address is stored in the SIP messages. In the processes of obtaining port mapping, the RHN can be taken as a SIP user agent and provide the control point with visit residential network's public address and port number that is NAT bound for a request device.

After the port mapping is obtained by the control point via SIP, the control point will be able to arrive at the media server to get protocol information via browse. Additionally, according to the port mapping information from the RHN, the control point will swap the IP address information of the media server. When the swapped IP address information of the media server is sent to the media render, the media render will put it into the body of HTTP POST message. With the swapped IP address information, the HTTP POST message will be able to connect to the media server correctly. Besides, the private IP addresses included in the body of HTTP POST messages can be handled as the description in section 3.1.4.

Figure 19 also exhibits a sequence of the connection processes between the media render and the media server. According to the logic model shown in Figure 16, we define an IP address for each of these network devices. To avoid the interference between hosts that own same sub-addresses in the logic model, we can assume the control point and the media server own the same private IP address but they are behind different private network. Moreover, the media render and the control point are behind the same private network to achieve the function of remote access.

### 3.1.4 Swapping of IP Address in Transmission Processes

As the processes described in Figure 17, the control point can start to get protocol information via HTTP to match appropriate protocol formats between the content we want to play and what the media render supports. Then, the control point will swap the IP address information between the private IP address and the public IP address that include the port information of NAT bound for local requested device. The swapping of IP address information is based on the information of port mapping added by the RHN. We can detail the realization of swapping IP address processes with Figure 20.

In Figure 20, we can assume the HTTP POST is sent from client will go through the proxy with IP address. Because the swapping of IP address information should be achieved by the media server's residential gateway, we will concentrate on the view of RGw 2 shown in Figure 18. From the view of RGw 2, the message of HTTP POST is sent from the proxy and its source address is: 80.0.0.5 with a default port number. Since the message's final destination is the media server behind RGw 2 and the port mapping information of RGw 2 has been received by the control point, the destination address should be the address of RGw 2 with an assumed port number that is NAT bound for the media server: 100. The RGw 2 then will relay the message to the media server behind it. From the view of the media server, the source address of the message should be the address of access point of the RGw 2 with another assumed default port number. The destination address should be the media server's sub-address with its assumed local port number: 9000. With the relaying of RGw 2, the HTTP message can finally arrive at the media server correctly.

After the HTTP POST message is received by the media server, a response message should be sent back to the client. From the view of RGw 2, the response's source address should be the media server's sub-address with local port number 9000 and the destination address should be the address of access point of RGw 2 with another default port number. RGw 2 then relays the response to the proxy with the address of RGw 2 as source address and the media server's NAT port number. Its destination address should be the address of the proxy with a default port number.

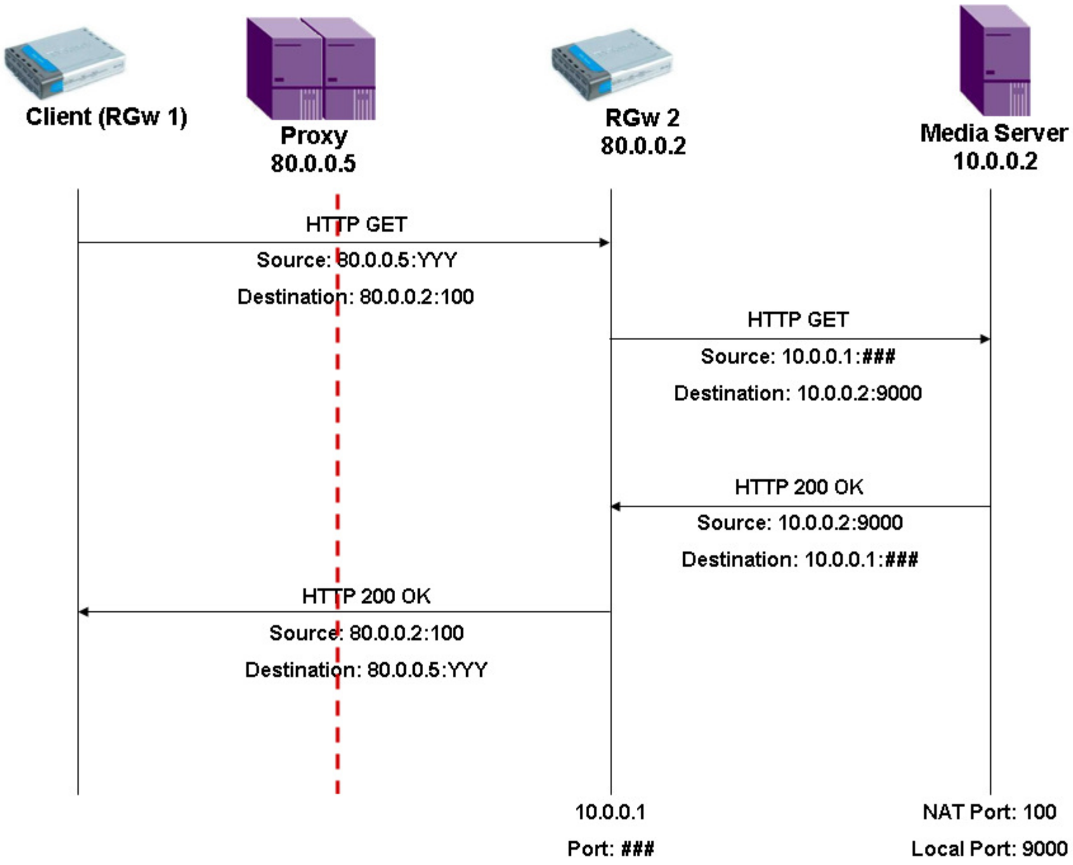

**Figure 20:** Swapping of IP address in Remote Access Procedure

With the operation of swapping IP address information in the control point, the control point will be able to select media item and set AV transport URI and send the command of play. The media render will send the request selected by the control point to the media server behind different networks correctly with the assistance of swapped IP address information. Media stream will also be sent back the media render correctly with the swapped IP address information.

## 3.2 Solution with SOAP Intermediary

### 3.2.1 Overview

Compared with solution with IP address swapping, we can also settle the problem caused by UNSAF across NAT with the help of SOAP Intermediary. As the introduction of SOAP in background part, we find the SOAP 1.2 can realize the same function as IP address swapping [51].

In addition, the IP feature of this solution is still NAT. We apply it to the connection of hosts behind different private networks. An SOAP intermediary is needed to be added in the private network for the transmission via SOAP. If the IP address mapping information about the remote device (media server) is known by the SOAP intermediary, the request from the control point will be relayed to the remote device simply. Relaying of SOAP message is achieved by the SOAP intermediary that is similar to a proxy in SIP and HTTP.

With the assistance of the SOAP intermediary, we will be able to decouple the remote access functionality from the control point. Thus, we are allowed to support a normal control point with no idea about remote access to realize same functionality as solution with IP address swapping.

Notwithstanding UPnP devices can not directly use SOAP 1.2, we can make the control point and the RHN to be able to use it. Communication between the control point and the RHN can be realized by SOAP 1.2. However, when the control point and the RHN are communicating with the UPnP devices, the transmission protocol will be transform to SOAP 1.1.

As the description of SOAP terminology, there is a concept of SOAP headers. It is a suitable place to put IP address mapping information that can be used by the SOAP intermediary. When the request from the control point is sent to the remote device, the IP address mapping information will be adopted by the SOAP intermediary to swap the IP address in the body of the message.

Compared with solution in chapter 3.1, we can detect the main difference between the two solutions is the place where the IP address swapping happens and the application of SOAP. We are able to move the IP address swapping functionality out of the control point and into a SOAP intermediary. With the assistance the SOAP intermediary, we will be able to support standard UPnP control points. Although the SOAP intermediary acts as a virtual device, it will look like a real device for UPnP control points. Moreover, all requests will be relayed to the remote device via the virtual device instead of the control point.

### 3.2.2 Relaying of SOAP Header Blocks

According to the functionality of SOAP 1.2, a SOAP message can be originated at initial SOAP sender and sent to an ultimate SOAP receiver via SOAP intermediaries. If the SOAP header blocks are processed by an intermediary SOAP node, the relaying of SOAP header blocks will go through the intermediary SOAP node. In addition, the reinsertion of SOAP header blocks is also depending upon when the processing of the SOAP header blocks determines that the SOAP header block needs to be reinserted into the forwarded message. As long as a SOAP header block is targeted at a role played by the SOAP intermediary, the specification for the SOAP header block will be able to call for the header block to be relayed in the forwarded message.

A relay attribute information item may be carried by a SOAP header block. If the value of the relay attribute information item is “true”, we can believe the header block is relayable. Besides, when the header block carries a “mustUnderstand” attribute information item with a value of “true”, the item has no influence on the SOAP processing model. Moreover, we should also notice that the relay attribute information item has no effect on the processing of SOAP messages by the SOAP ultimate receiver [51].

### 3.2.3 SOAP Forwarding Intermediary

Based on the relaying of SOAP header blocks, we can introduce the SOAP forwarding intermediary into the procedure of remote access. If there are one or more SOAP header blocks in a SOAP message, the SOAP message may be forwarded to another SOAP node. The SOAP node will act as the initiator of the inbound SOAP message. And then, we can find that the processing SOAP node can realize the function as the role of SOAP forwarding intermediary. Besides, when the SOAP forwarding intermediary processes the SOAP message, it will follow the SOAP processing model.

When a SOAP message is generated for the purpose of forwarding, we must remove all processed SOAP header blocks and non-relayable SOAP header blocks. Those non-relayable SOAP header blocks may be targeted at the forwarding node but were ignored during processing. Meanwhile, we also need to retain all relayable SOAP header blocks that may be ignored during processing but were targeted at the forwarding node.

In addition, there is a specification that defines the SOAP forwarding features for SOAP forwarding intermediaries. Each feature of the specification needs to describe the required the semantics. The rules that describe the constructing processes of the forwarded message should be included in the semantics. Besides, the placement of inserted or reinserted SOAP header blocks may also be contained by the rules.

According to the rules for SOAP messages processing [52], the contents of the SOAP envelope may determine the set of roles in which the node should act. Besides, all header blocks targeted at the mandatory node need to be identified by SOAP messages. When one or more SOAP header blocks identified in the preceding step can not be understood by the node, a single SOAP fault will be generated with code value “env:MustUnderstand”. The fault will stop any further processing. Furthermore, all mandatory SOAP header blocks targeted at the node should be processed by the node. If the node is a SOAP intermediary, the SOAP message will be sent along the SOAP message path by the relaying of the SOAP intermediary.

Since it may be hard to distinguish inserted SOAP header blocks from those header blocks removed by the intermediary, re-inserting header blocks is necessary to be defined in processing. With the definition of these re-inserting header blocks, processing will be able to concentrate on the requirement of processing at each SOAP node of the SOAP message route.

Because media server is an UPnP device and can not directly use SOAP 1.2, we have to introduce a SOAP intermediary in the home residential network where the media server located. The SOAP intermediary is used to connect with another SOAP intermediary located in a visit residential network. As the description of obtaining port mapping information in solution one, if the port mapping information behind the home residential network is gained by the control point via SIP, the control point will be able to start the browse with the help of SOAP intermediaries .

### 3.2.4 Remote Access with SOAP Intermediary

Due to the specific situation in remote access via SOAP intermediaries, we need to create a logic model to explain the solution with SOAP intermediaries in the transmission procedure. The main difference between solution with IP address swapping and solution with SOAP intermediary is the move of the IP address swapping functionality. We move the functionality of IP address swapping out of the control point and into SOAP intermediary nodes. We also need to add a RHN into the home residential network to realize the function of collecting port mapping information. Meanwhile, we can assume the RHN on home residential network acts as remote SOAP intermediary.

Connection between the control point and the RHN is supported by SOAP 1.2. However, because SOAP 1.2 can not be used by other UPnP devices, it needs to be transformed to SOAP 1.1 when the control point and the RHN are communicating with other UPnP devices.

We assume the SIP address of the RHN is known by the SOAP intermediary located in the same private network. Besides, all the other devices information is assumed to be the same as the information in solution with IP address swapping. Then we can illustrate the logic model of solution with SOAP intermediaries in Figure 21 shown as follow.

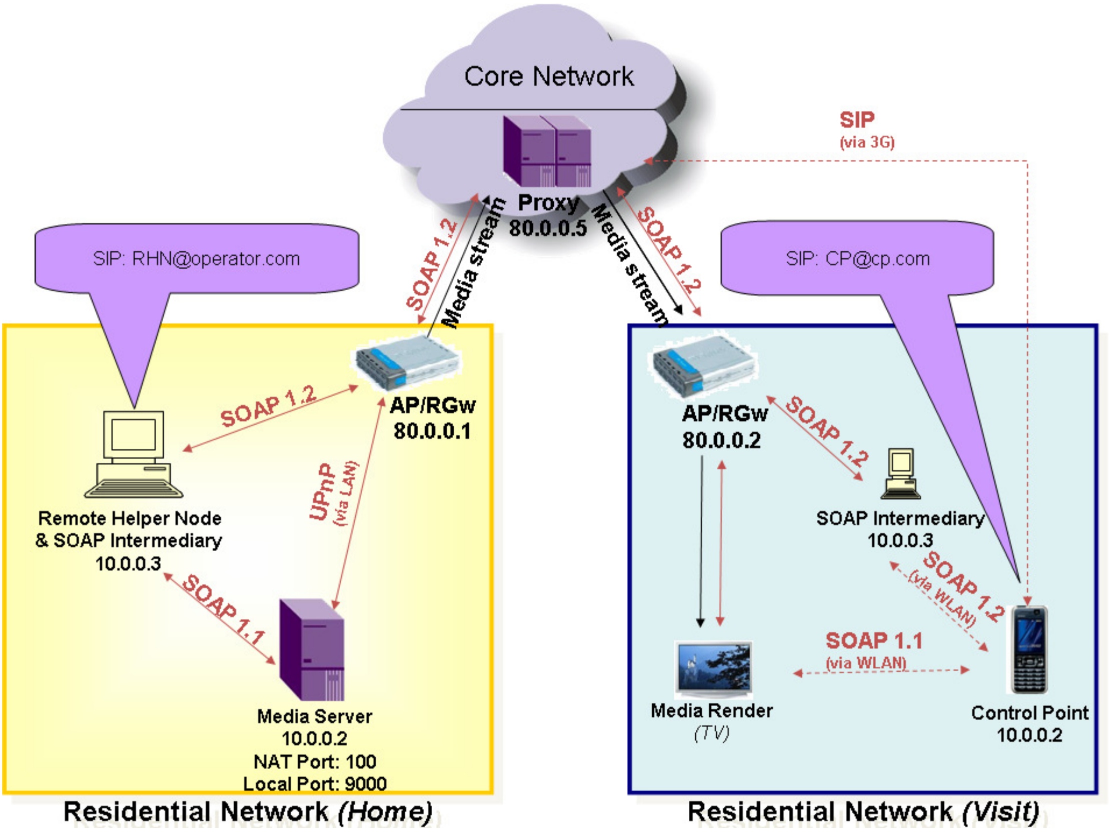

**Figure 21:** A Logic Model with SOAP Intermediary for the Solution of UNSAF across NAT

In Figure 21, we also need to define unique SIP address for the RHN and the control point separately. Moreover, there is sub-address assigned to the SOAP intermediary in visit residential network which is shown in Figure 21. SIP is also used in this solution for the session establishment and the creating of port mapping. A SIP proxy is kept to realize the same function as the proxy shown in solution one.

Compared with the logic model of solution one, the main difference between Figure 18 and Figure 21 is the application of SOAP intermediary. A SOAP intermediary is added into visit residential network to achieve the relaying SOAP messages for the control point. At the same time, another SOAP intermediary is added into home residential network to support the relaying of SOAP messages for media server.

All the steps of communication between the RHN and the control point are supported by SOAP 1.2. However, the control point uses SOAP 1.1 to communicate with media render. SOAP 1.1 is also used by the RHN to connect with the media server. In addition, we assume the functionality of the SOAP intermediary in home residential network is achieved by the RHN.

According description of SOAP version 1.2, SOAP does not detail how a message path is determined and followed. What we concern is how a SOAP node should do when it receives a SOAP message for which it is not the ultimate receiver. Although there are two types of intermediaries: forwarding intermediaries and active intermediaries, what our logic model needs to apply are the forwarding intermediaries. The forwarding intermediaries we apply for the logic model is based on the semantics of a header block in a received SOAP message. It is used to forward the SOAP message to another SOAP node.

The main function of SOAP intermediaries in the logic model is the relaying of SOAP messages. If an incoming SOAP message with a port mapping header block that describes a message path feature is processed, the SOAP message will be forwarded to another SOAP node identified by port mapping information in its header block. Both of the SOAP intermediaries shown in our logic model can realize the functionality of forwarding according to the port mapping information of the media server. Besides, the format of the SOAP header of the outbound SOAP message depends on the overall processing at the forwarding intermediary.

Through the external specification of the header blocks' semantics, the SOAP intermediary node will have the knowledge of what to do. Various details of the SOAP message needs to be recorded at every node that receives the message. Meanwhile, all the SOAP intermediaries should make sure the SOAP message that is relayed along the message path unchanged. In order to keep the accuracy of the transmitted SOAP message, the header blocks need the same header blocks to be reinserted in the outbound message. Then, we can confirm the SOAP node acts as a forwarding intermediary and perform the successful relaying of SOAP messages in our logic model.

Based on the logic model illustrated in Figure 21, we can detail the remote access procedure with the help of SOAP intermediaries by a flow chart. We assume the parameters of UPnP devices in the flow chart are the same as the parameters in Figure 17. SIP RHN is also acting as a SOAP intermediary. Media render and the control point are located in the same residential network while media server and the RHN are located in the same residential network. Address of the RHN should be known by the control point firstly. In addition, the IP address information of the media server also needs to be collected by the RHN in advance. The detailed procedure with SOAP intermediaries is described in Figure 22 shown as follow:

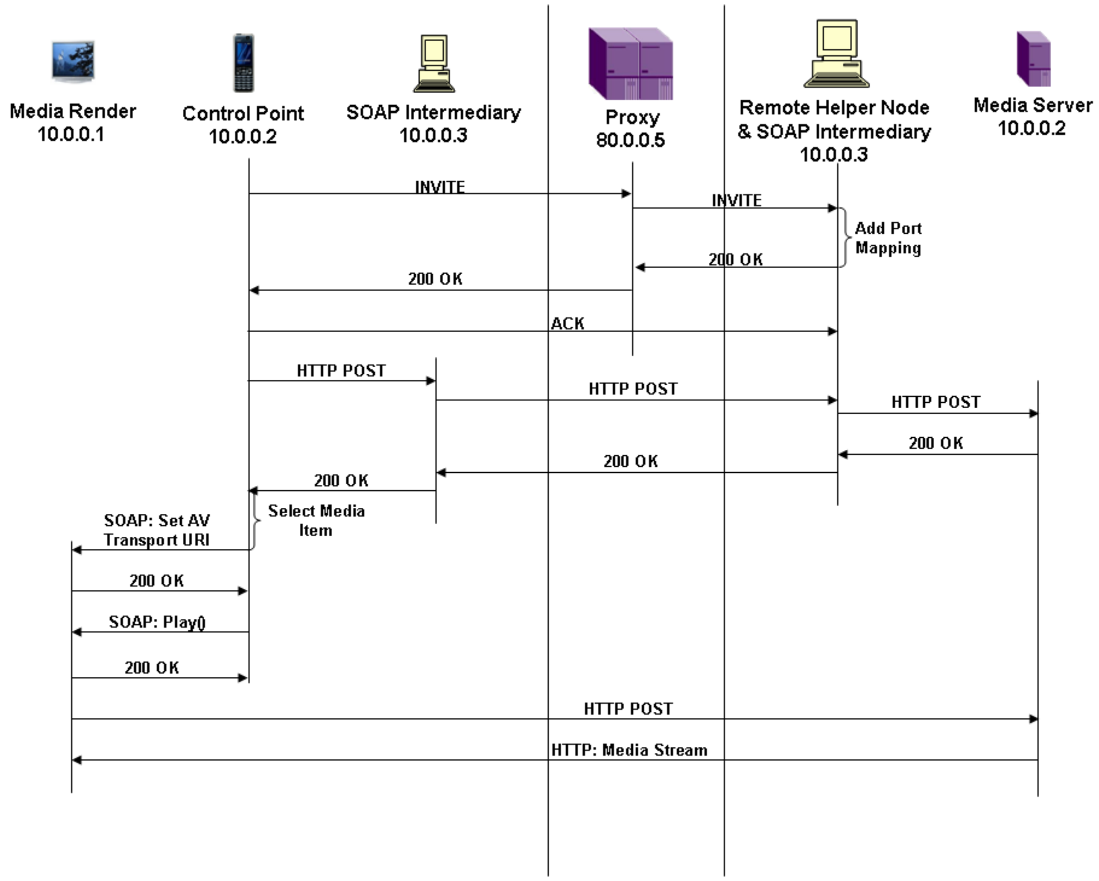

**Figure 22:** Connection Flow between Media Render and Media Server with SOAP Intermediary

In Figure 22, compared with the connection flow in solution one, the SOAP request for device and service description is transmitted through a different way. The message path is composed of the control point and two SOAP intermediaries. SOAP messages are forwarded from a SOAP intermediary in visit residential network to another SOAP intermediary in home residential network. What the SOAP intermediaries need to do is relaying the SOAP messages according to the processing semantics defined for SOAP forwarding intermediaries. With this different path of transmission, we can obtain the same result described in solution one. Swapping of IP address is moved out of the control point. We use the two added SOAP intermediaries to achieve the functionality of IP address swapping.

SOAP request message is initiated by the control point when the port mapping information is received by it. Because both the control point and the RHN are supported by SOAP 1.2, the SOAP request message can be transmitted directly to the SOAP intermediary in visit residential network via SOAP 1.2. Besides, in Figure 20, HTTP GET is also used to realizing the forwarding of SOAP request message. There is no doubt that the SOAP request message is relayed from the SOAP intermediary in visit residential network to another SOAP intermediary in home residential network via SOAP 1.2. The media server acts as an ultimate receiver of the SOAP request message path. However, the connection from the SOAP intermediaries in home residential network to the media server has to be finished by SOAP 1.1. Similarly, all the portions of SOAP response message path are realized by SOAP 1.2 except the connection between the media server the SOAP intermediaries in home residential network.

Finally, we can express the content of the relayed SOAP message with codes for the convenience of understanding. The codes are detailed in Appendix 2.

From the codes about a SOAP message shown above, we can note that the header blocks “publicaddressofmediaserver” and “NATportnumberofmediaserver” are intended for the nodes that assume the role is “next”. It is means the SOAP message is targeted at next SOAP node that receives the message in the message path. The “mustUnderstand” attribute is set to “true” to make sure the header blocks are mandatory and have knowledge about operation.

Port mapping information about the media server is put in the header of the SOAP message and it will be used by the SOAP intermediaries for relaying. In addition, two times of relaying is necessary for the SOAP request message to arrive the ultimate receiver. Enough information of port mapping has been added into the SOAP message to ensure the request will be transmitted to the media server successfully.

## 3.3 Solution with RSIP

### 3.3.1 Overview

RSIP is the solution operated on the application layer. However, the use of a transport layer protocol is also necessary for the end-to-end delivery of packets. In fact, solution with RSIP is the same as the explanation on how RSIP works in remote access. We have given sufficient description in terms of fundament of RSIP. Therefore, the key point of the solution with RSIP should be the method used for assigning parameters to an RSIP host from an RSIP gateway. Based on the introduction about RSIP in background part, we will go directly for the specified components that make up of the remote access with the help of protocol specification of RSIP.

According to definition of RSIP, we can make the address and other routing parameters in home private network to be directly used by the host in remote private network. It makes the address sharing of RSIP is more transparency than NAT. An RSIP gateway needs to assign one address from home residential network to the RSIP host located in visit residential network. With the assigned address, the RSIP host located in visit residential network will be able to establish end-to-end connectivity to a host or other entities in the home residential network. Moreover, both the home residential network and visit residential network can act as the routing realm. Although a routing realm is not directly accessible from a RSIP host in different routing realm, the integrity of packets from the RSIP host to their destination can still be well maintained. Before we specify operation processes of RSIP in remote access, we need to exhibit the relationship of RSIP host and RSIP gateway in advance.

### 3.3.2 Relationship of RSIP Host and RSIP Gateway

RSIP host and RSIP gateway are two basic elements that form the configuration of RSIP. Different relationships about them decide how much information can be transmitted between them. These are three types of fundamental relationships for RSIP host and RSIP gateway can be summarized from RFC 3103 [35]. In addition, different relationships of the RSIP gateway and the RSIP host can represent the results of different operation processes between them.

Firstly, we need to introduce the loosest relationship named “unregistered” between them. It means the RSIP gateway does not know of the existence of the RSIP host and the RSIP gateway will not forward or deliver globally addressed packets to the RSIP host. Then, the only action that RSIP host can perform is to make a request for registration with an RSIP gateway.

The second type relationship of RSIP host and RSIP gateway we need to introduce is named “registered”. It means the RSIP host should be known by RSIP gateway and can obtain a client ID from the RSIP gateway. Moreover, the flow policies required by the RSIP host should be specified by the RSIP gateway. With this type of relationship, no resources can be allocated to the RSIP host and the RSIP gateway will still not forward or deliver globally addressed packets to the RSIP host. Besides, there is an associated lease time for each registration. RSIP host will automatically return to the “unregistered” state when the lease time expires.

Relationship named “assigned” is the tightest relationship for RSIP host and RSIP gateway. With the relationship of “assigned”, the RSIP host can obtain at least one binding from the RSIP gateway. Additionally, the gateway will forward and deliver globally addressed packets to the RSIP host. An associated lease time also defined for the binding. When the lease time expires, the RSIP gateway will automatically revoke the binding.

### 3.3.3 Architecture of RSIP in Remote Access

After the introduction to the relationship of RSIP host and gateway, we now can concentrate on the remote access architecture of RSIP. The typical scenario that RSIP is deployed consists of some hosts within one addressing realm connected to another addressing realm by the RSIP gateway. We illustrate the typical architecture of RSIP in remote access with a Figure as follow:

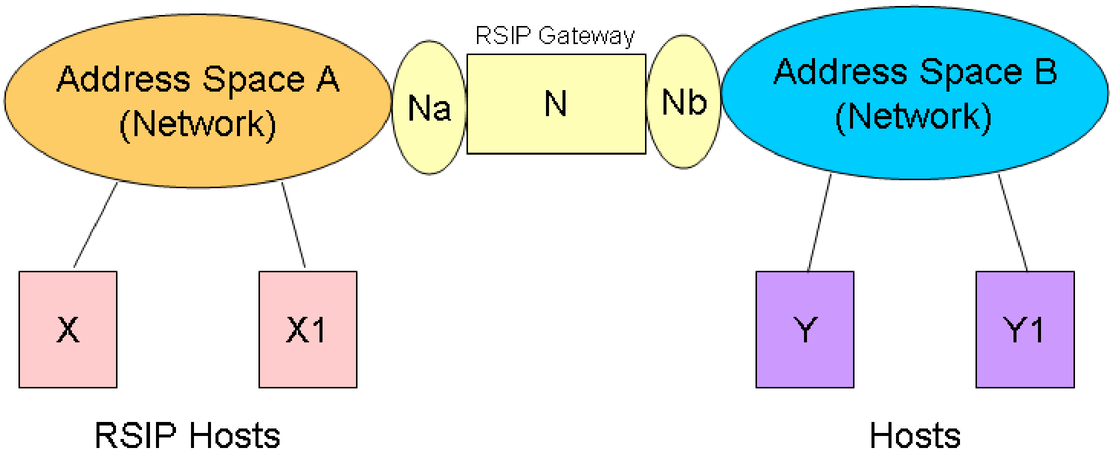

**Figure 23:** Model of RSIP Architecture

From Figure 23, we can clearly find two different addressing realms A and B. Hosts X and Y belong to them respectively. N represents the RSIP gateway and may also perform the functions of NAT. There are two interfaces for N: Na is on address space A and Nb is on address space B. N can have a few of addresses in address space B such as Y and Y1. These addresses can be assign to or lend to X and other hosts in address space A such as X1.

RSIP host (X, X1) must be able to maintain at least one virtual interface for IP address which is leased from the RSIP gateway. Host (Y, Y1) must support tunneling and can serve as an endpoint for at least one tunnel to RSIP gateways. If the RSIP host supports RSAP-IP, it must be able to maintain at least one port assigned by the RSIP gateway from which chooses temporary source ports. When there is no free port in host's pool and the host has to establish a new communication session with a public host, it must be able to dynamically request at least one extra port through its RSIP mechanism.

The RSIP gateway should be able to route packets between two or more realms and act as a boundary router for two or more administrative domains to achieve remote access. It must support tunneling and can act as an endpoint for tunnels to RSIP hosts. The RSIP gateway should also be a policy enforcement point which is always be required to act as the firewall and packet filter for RSIP. All the incoming packet fragments from the public network must be reassembled for the purpose of routing and tunneling them to the suitable host. Fragments that are not fully reassembled must be stopped by RSIP gateway because reassembly is necessary for fragments.

Hosts on the private network may still establish communication with the public network because of the NAT functionality comprised in the function of RSIP gateway. All the resources that are allocated to RSIP hosts must be managed by RSIP gateway. The management is always be influenced by the local network's policy.

### 3.3.4 Operation Processes of RSIP in Remote Access

In most of the time, the hosts within address space A may use private addresses on condition that the RSIP gateway is multi-homed with at least one private address from address space A. We can separate the total realm into two portions. Private realm means the area RSIP host resides in while public realm means the area from where RSIP host can leases addressing parameters. Sometimes, the distinction is presented for convenience because these realms may both be public or private. The special case is that address space A is an IPv6 realm or a non-IP address space.

If host X wants to have an end-to-end connection to host Y located in address space B, it needs to negotiate with RSIP gateway and gains the assignment of resources such as addresses and other routing parameters of address space B from it. Service location protocol (SLP) can be used by RSIP clients to find the RSIP server [53]. With the assignment of these parameters, a mapping which binds X's addressing information and the assignment of resources can be created. The RSIP gateway then can exactly demultiplex inbound traffic information made by Y and forward it to X. Multiple such bindings which is associated with the lease time can be created on the same RSIP gateway and even across several RSIP gateways with the permission of RSIP gateway for remote access.

RSIP hosts can tunnel data packets across address space A to RSIP gateway which acts as the endpoint of the tunnel on condition that public parameters are assigned to these RSIP hosts by RSIP gateway. The RSIP gateway will strip off the outer headers of inner packets and routing them onto the public realm. RSIP gateway also maintains a mapping of assigned public parameters and RSIP host's private addresses. These public parameters need to be regarded as demultiplexing fields for matching them to RSIP host addresses in private network. If a packet arrives at RSIP gateway from the public realm and can match a given demultiplexing field, it will be tunneled by RSIP gateway for the correct RSIP host. For X have been assigned Nb, the tunnel headers of outbound packets is from X to Y. The processes of address assignment in the packet are described in Figure 24 as follow:

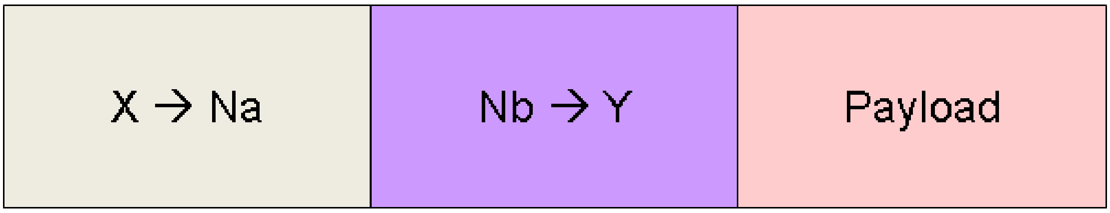

**Figure 24:** Address Assignment in a Packet

From the illustrated packet in Figure 24, we can distinctly find the movements of RSIP host addresses. It is sent to interface Na of the RSIP gateway firstly. RSIP gateway then transmitted it to Y through interface Nb which is connected to the public realm.

# 4 Evaluation of Solutions

## 4.1 Overview

As we have seen in chapter three, three kinds of solutions were proposed to settle the problem described in project definition. The further job for us is to demonstrate that our solutions can actually solve the defined problem. Through the evaluation, we can prove these solutions indeed work. Furthermore, with the assistance of the evaluation, the advantages and disadvantages of each solution will be specified exactly. The results of the evaluation will also provide the necessary evidence for the selection of best proposed solution.

In order to make a comprehensive evaluation of these solutions in handling the problem of UNSAF across NAT in remote access, we select five types of qualified criteria that need to be taken in consideration. Besides, we will make an introduction to each of these criteria. Each solution will be evaluated with these criteria individually.

## 4.2 Availability

This is the basic type of criteria used to evaluate the three solutions. Availability is concerned with the failure situation of the solutions and the consequences associated with the failure. If the proposal solution can not handle the problem of UNSAF across NAT in the context of remote access, we will define it as the occurrence of failure. Both network systems and its users can observe this kind of failure. Besides, when a failure occurs, we will concentrate on how the failure is detected and prevented. The required notifications about the occurrence of failure will also be listed for evaluation.

In addition, we should pay more attention on the difference between failures and faults. If a fault is not corrected, it may become a failure. Thus, the fault may not be observable until it becomes the failure. In other words, if a fault does become observable, it will become a failure.

### 4.2.1 Solution with IP Address Swapping

As we have described in the connection flow between a local host and a remote host with IP address swapping, the availability of IP address swapping is focused on the obtaining of port mapping information. Because the port mapping information is obtained via SIP, the availability of SIP in obtaining the port mapping information will determine whether the solution is available or not.

As we detailed in the solution, SIP proxy is added to enhance the reliability of the SIP messages transmission. The SIP proxy can not only glue SIP components but also provide the maintenance of central role in SIP network such as the security and routing. With the assistance of SIP proxy, the host can find the right recipient of an invitation no matter where it is located. The SIP proxy can route the host in remote private network via DNS lookup which is similar as searching with email.

Besides, seven types of NAT traversal for SIP have been introduced in chapter two. Each of them has its own advantages. As the comparison in chapter two, we may select UPnP to achieve the NAT traversal for SIP. Because we have assumed all devices in private networks are UPnP enabled in this solution. Moreover, there are no cascaded NATs that may cause the failure of SIP's NAT traversal.

With respect to the firewall traversal of SIP, we can realize it from three specified spaces. The first space is formed by the education of administrators. That means if we want to get our users over a firewall, we will need to evaluate these administrators to open up proper port ranges and coordinate those with used ports. Deployment of ALGs is the second space for firewall traversal. ALGs can be used to punch the needed holes. However, this space is not perfect for frequent application. The last but not the least, we can tunnel all the traffic in HTTP to realize circumvent firewall policy. Although it is not a good method, it can still work for ASPs.

### 4.2.2 Solution with SOAP Intermediary

Our concern in this section is making an evaluation for the availability of solution with SOAP intermediary. According to the solutions we detailed before, the main difference between solution with IP address swapping and solution with SOAP intermediary is the move of the IP address swapping functionality. The functionality of IP address swapping is achieved via SOAP intermediary nodes instead of the control point. In solution with SOAP intermediary, the port mapping information is also obtained via SIP. Thus, we will only concentrate on the evaluation of availability to swap IP address in SOAP intermediaries.

Based on the messaging framework of SOAP version 1.2, we can add one or more SOAP intermediaries into the transmission routing. There are two types of SOAP intermediaries. One of them is named SOAP forwarding intermediaries which has been detailed in solution with SOAP intermediary. The other type of SOAP intermediary is named SOAP active intermediary which is used to undertake some additional processing that can modify the outbound SOAP message in ways not described in the inbound SOAP message. As the new functionality of SOAP 1.2, both of the SOAP intermediaries have been definitely accepted and applied. However, what we concern is only the SOAP forwarding intermediaries. Furthermore, version transition from SOAP 1.1 to SOAP 1.2 has been detailed with version management rules. If these rules are implemented by a SOAP node, the SOAP node will be able to support versioning from SOAP 1.1 to SOAP 1.2.

### 4.2.3 Solution with RSIP

RSIP can solve the problem in project definition via architecture that allows the hosts within a routing realm to directly use addresses and other routing parameters from another realm. We are interested in how the hosts within a routing realm obtain the addresses and other routing parameters from another realm. In this section, we will evaluate the availability of the method of obtaining the required address information from a different routing realm.

Although RSIP has been defined as a method for address sharing, we still need to confirm whether the address information in remote realm can be correctly obtained or not. With the description of the relationship between the RSIP host and RSIP gateway, we find the “registration” and “assignment” are used to achieve the address sharing. Besides, lease times for registration and binding are managed with a scheme defined in RFC 3103. With this defined scheme, a registration will never expire as long as any lease of binding is valid. However, if this lease time expires, the binding will be automatically revoked.

In addition, with the specified parameters and message types in registration and assignment processes, we believe the remote flow policy will be exactly followed to maintain the specified remote host’s address and port.

## 4.3 Modifiability

Modifiability of a solution can be expressed with focus on the cost of modification. We can divide it into two related aspects. The first aspect is about what can be modified. A modification may occur to any aspect of the solution such as the protocols and devices used to realize the functionality. Another aspect is related to who makes the modification. It may be made by a device, an end user or a solution designer.

Besides, we will make an evaluation of modifiability for each of the three solutions respectively. Once modifiability has been specified, the new solution will be detailed with more suitable technique. The improved solution may take extra time and money, both of which can be calculated.

### 4.3.1 Solution with IP Address Swapping

Port mapping information collected by the RHN can be modified when the address information of the media server is changed. It must be modified before adding port mapping. Once a port mapping is sent the control point via SIP, no more modification will be made for the port mapping. This kind of modification should be implemented by the RHN automatically. Besides, with the swapped IP address, the address information in HTTP POST can be modified by the home residential gateway. It is necessary for the HTTP request to be sent to the correct device. Both of the modifications described above do not need any cost.

### 4.3.2 Solution with SOAP Intermediary

Modification of port mapping information in solution with SOAP intermediary is the same as the solution with IP address swapping. Because the SOAP messages is relayed by SOAP intermediaries, the address information added into the header blocks the SOAP messages needs to be modified. SOAP forwarding intermediaries will implement these modifications. Besides, more SOAP intermediaries may be used for the relaying of SOAP messages. Of course, the added intermediaries need extra cost.

### 4.3.3 Solution with RSIP

Compared with the other two solutions, solution with RSIP requires the modification to RSIP hosts. Some number of network layer, transport layer or other values assigned by the RSIP gateway need to be put into each of the packets. These packets have been bound for other routing realm.

## 4.4 Performance

In this chapter, we need to make an estimation of the performance hit introduced by the three solutions. More attention should be paid for how completely our solution can handle the problem of UNSAF across NAT in the context of remote access. When a host needs to connect to another host in remote private network, the solution must make a response to the host’s request and realize the connection for it.

### 4.4.1 Solution with IP Address Swapping

Solution with IP address swapping is relatively easiest solution among the three solutions. Introduction of the RHN is the only change to the initial system. With this easiest solution, we can still solve the problem of UNSAF across NAT. Besides, if there is any SIP request to the RHN, the RHN must be able to collect a correct port mapping and make a corresponding response via SIP. The swapped IP address information is enough for a host to connect to another host in remote private network. However, because IP address swapping is a text search and replacement, this solution is not a very quick operation.

### 4.4.2 Solution with SOAP Intermediary

A relatively advanced technique is utilized for solution with SOAP intermediary. As a new feature of SOAP 1.2, SOAP intermediaries are used to swap IP address and relay SOAP messages. A more reliable route is formed by these SOAP intermediaries. In addition, with the address mapping information put into the SOAP header, these SOAP intermediaries can realize the same functionality as solution with IP address swapping. Moreover, compared with solution with IP address swapping, the transmission efficiency of this solution can be improved obviously.

### 4.4.3 Solution with RSIP

As long as the address of a host in remote residential network is assigned to the host located in home residential network by RSIP gateway, the host located in home residential network must be able to establish an end-to-end connectivity to the host in remote residential network. With the end-to-end connectivity, RSIP will make no change to the transmitted packets. Thus, RSIP also provide an appropriate maintenance for the packets.

## 4.5 Security

Security of a solution is a measure of the solution’s consistence. We can characterize security as a system that provides confidentiality, client authorization and message integrity.

### 4.5.1 Solution with IP Address Swapping

Security of solution with IP address swapping is determined by the security of SIP. Best Current Practices (BCP) [54] of SIP has provided a reasonable mix of SIP security [55]. With the mixed SIP security, a proper and viable service access control can be achieved. Besides, the confidentiality of SIP can be realized by encryption. Message integrity check is used to make sure no man in the transmission process can temper with the messages.

### 4.5.2 Solution with SOAP Intermediary

Security of solution with SOAP intermediary is related to the security of both SIP and SOAP. Because a mechanism used to dealing with access control, confidential and message integrity has been specified in the SOAP extensibility model, the relaying of SOAP message must own a high level security. Besides, when a SOAP message is received by a SOAP node, the node will be able to evaluate what level it can trust the sender of the SOAP message and the contents of the SOAP message.

### 4.5.3 Solution with RSIP

The floating of port numbers may cause problems for some applications. It will prevent an RSIP host from interoperating obviously with existing applications [53]. Moreover, some significant operational complexities are associated with using RSIP [34]. There is no security provided in RSIP. Although we can hide a private address space for security, security of solution with RSIP can still only be ensured by the reasonable use of security protocol and other related cryptographic techniques [56].

Besides, the RSIP gateway may take all necessary measures to prevent its hosts from requesting large sets of resources. The RSIP message transmitted between a gateway and a host is also allowed to be authenticated.

## 4.6 Compatibility

Compatibility of a solution is related to whether the solution can cooperate with the existing communication system. We need to focus on whether modification should be made for the existing components or not. Besides, the compatibility of protocols used by the solutions also needs to be evaluated. The degree of the compatibility of solutions is determined by both of the factors listed above.

### 4.6.1 Solution with IP Address Swapping

When we evaluated the compatibility of solution with IP address swapping, the introduction of RHN should be the first component we need to concern. Because the RHN is added to act as a SIP user agent and SIP is supported by the control point, we do not need to make any modification on the client or the server. The RHN can collect the port mapping information of the media server without any modification to the media server. Meanwhile, because the control point is support by UPnP, it will be able to communicate with the media render directly without any change to the media render.

Protocols used by the initial system also need no change. Although SIP is used by the connection between the RHN and the control point, the initial protocols are still adopted to support communication between the media render and the media server. In fact, the introduction of SIP is only used for obtaining port mapping information and serves the initial communication system.

### 4.6.2 Solution with SOAP Intermediary

Compatibility of solution with SOAP intermediary is related to the acceptance of SOAP intermediaries. With the assistance of RHN, no change needs to be made to the initial communication system. The media server can communicate with the RHN directly with SOAP 1.1, while the control point can also communicate with the media render directly via SOAP 1.1. Besides, the connection with SOAP forwarding intermediaries also needs no modification to the initial components.

Compared with the introduction of SIP in solution with IP address swapping, SOAP 1.2 is used to realize the functionality of IP address swapping and serve the initial communication system.

### 4.6.3 Solution with RSIP

In general, an arbitrary host is not allowed to start public gateways by the RSIP gateway. According to the definition of RFC 3103, when the remote micro-flow based policy is used, an RSIP gateway will only allow public gateways on RSIP hosts via administrative override [35]. Besides, we can only identify RSIP hosts with their local IP address or MAC address.

# 5 Discussion

## 5.1 Introduction

In this chapter, we will make a discussion for the evaluation of the three proposal solutions. The evaluation of the three solutions will be compared for the ranking of these solutions.

## 5.2 Availability

Availability of solution with IP address swapping is mainly determined by the availability of SIP in obtaining the address information. There is no doubt that SIP can be used to transmit the address information between two SIP user agents.

Availability of solution with SOAP intermediaries is determined by the functionality of both SIP and SOAP in the transmission processes. SOAP forwarding intermediaries have been clearly defined in SOAP 1.2 to relaying SOAP message. Thus, the SOAP message's relaying handled by SOAP intermediary makes this solution more available.

As long as the RSIP is well deployed for a network, the availability of solution with RSIP will be beyond the shadow of a doubt. However, the deployment of RSIP may cause much change to the existing network.

In a word, both the solution with IP address swapping and the solution with SOAP intermediaries have high-level availability.

## 5.3 Modifiability

Modification to the solution with IP address swapping does not need any cost. Besides, no modification is required to be made for the initial system.

Modifiability of the solution with SOAP intermediaries is determined by the cost of the added intermediaries. More intermediaries can be used to realize the relaying of SOAP message.

If RSIP is used to make the solution, modification to RSIP hosts will be necessary.

Thereby, all of the three solutions own their required modifiability to ensure the functionality of their service.

## 5.4 Performance

The solution with IP address swapping provides a simple method for solving the defined problem properly.

We find the solution with SOAP intermediaries can exhibit a better performance with the introduction of SOAP intermediaries.

Solution with RSIP gives the best performance among the three solutions. Besides, no change can be made to the transmitted packet.

All the solutions perform their duty on solving the defined problem. The better performance, the more cost is required. However, RSIP is unlikely to be deployed as a new technology.

## 5.5 Security

Security of the solution with IP address swapping is ensured by the reasonable mix of SIP security provided by BCP of SIP. Related parameters of security such as confidentiality and integrity have been well realized.

Security of the solution with IP address is not only determined by SIP security but also determined by the security of SOAP. Because of the comprehensive definition for SOAP security in SOAP 1.2, the security level of solution with SOAP intermediaries should be as high as solution with IP address swapping.

According to the specification of RSIP, security of the solution with RSIP is mainly depending on the reasonable use of security protocol and some related cryptographic techniques [35].

Accordingly, both the solution with IP address swapping and solution with SOAP intermediary own the high security level while the solution with RSIP owns the low security level.

## 5.6 Compatibility

Because SIP is only used to obtain port mapping information and has no influence on the initial system, the solution with IP address swapping should own the best compatibility.

Although the initial UPnP devices are only supported by SOAP 1.1, SOAP 1.2 can still be used by the control point and the RHN in solution with SOAP intermediaries. With the transformed version of SOAP, the control point and the RHN will be able to communicate with the initial UPnP devices directly.

Compatibility of solution with RSIP is strictly limited by RSIP gateway.

Therefore, the compatibility of the solution with IP address swapping and the solution with SOAP intermediaries has much better perform than the solution with RSIP in terms of compatibility.

## 5.7 Rank of Solutions

All in all, we can conclude the rank of the three solutions with the following Table 1.

**Table 1** Rank of Three Solutions

| | Solution with IP address swapping | Solution with SOAP intermediary | Solution with RSIP |
|---|---|---|---|
| Availability | First choice | First choice | Second choice |
| Modification | Second choice | First choice | Second choice |
| Performance | Second choice | First choice | Third choice |
| Security | First choice | First choice | Second choice |
| Compatibility | First choice | Second choice | Third choice |

With the content of Table 1, we can illustrate the rank of solutions with Figure 25.

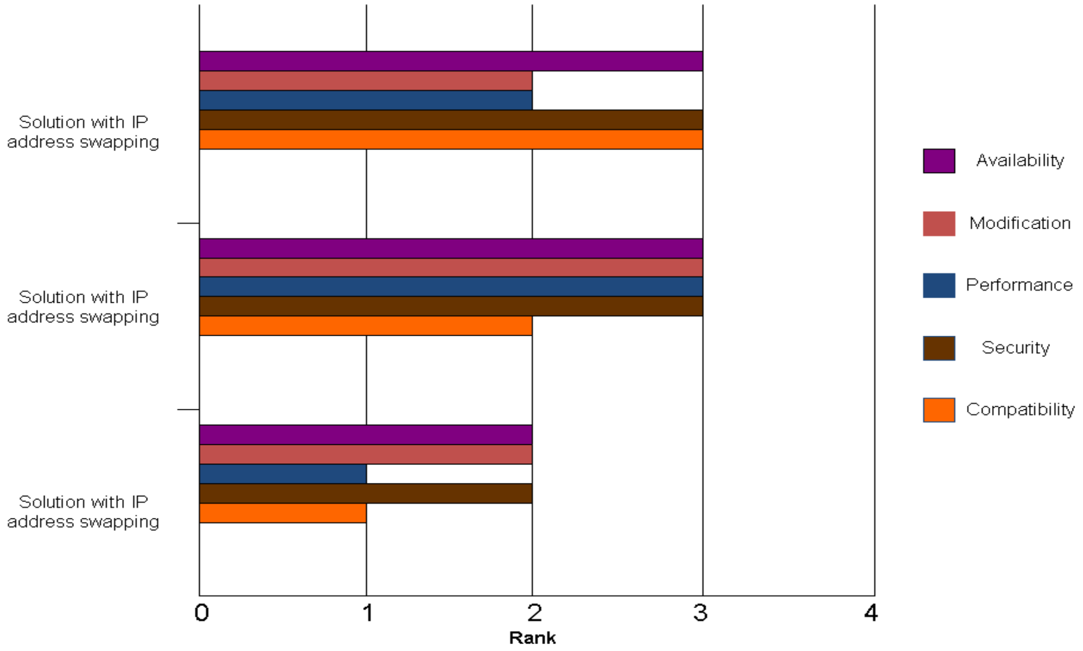

**Figure 25:** Rank of Solutions

# 6 Conclusion

Based on the results of the discussion, we can find that most of the evaluation criteria are realized appropriately by the three proposal solutions. In this chapter, we will make a conclusion for each of the three proposal solutions. The conclusions are based on the discussion above and the processes of designing the solutions.

## 6.1 Conclusion of Solution with IP Address Swapping

Solution with IP address swapping is the easiest one among the three proposal solution. Its high-level availability is determined by the availability of SIP in obtaining port mapping information. Moreover, there is no cost required for its modifiability. Compared with the other two solutions, its performance is not good enough. However, this solution owns the best compatibility and the highest security level. Besides, SIP URI is used by this solution to ensure the veracity of the transmission.

## 6.2 Conclusion of Solution with SOAP Intermediary

We also use SIP to obtain the port mapping for this solution. IP address swapping functionality is moved out of the control point and into SOAP forwarding intermediaries. This solution is more available than the other two solutions. Its modifiability is determined by the cost of the intermediaries. Furthermore, this solution owns a better performance than the solution with IP address swapping. However, the security and compatibility of this solution is similar as the solution with IP address swapping.

## 6.3 Conclusion of Solution with RSIP

This solution needs an RSIP gateway to replace the NAT router. Hosts on private network should be RSIP-aware. With the assigned address from an RSIP gateway, the RSIP host located in a private network will be able to establish end-to-end connectivity to another host located in a different network. The availability, security and compatibility of this solution are not as good as the other two solutions. Its modifiability is constrained to RSIP host. However, its performance is as good as the solution with SOAP.

## 6.4 Overall

According to the evaluation and discussion of the three proposal solution, we find all of these solutions have their own advantages and disadvantages in terms of different criteria. As long as the requirements and limitations are specified for the solutions, we will be able to select the most appropriate solution for the defined problem. However, from the experiences of designing these solutions and the general consideration of remote access, we recommend the solution with SOAP intermediary to be the first choice for handling the problem of UNSAF across NAT in the context of remote access.

# 7 References

1. Andreas Häber, et al., *REMOTE SERVICE USAGE THROUGH SIP WITH MULTIMEDIA ACCESS AS A USE CASE.* IEEE, 2007.
2. Daigle, L., *IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation*, in *RFC 3424*. November 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3424.txt.
3. J. Rosenberg, et al., *SIP: Session Initiation Protocol*, in *RFC 3261*. June 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3261.txt.
4. P. Nesser and A. Bergstrom, *Survey of IPv4 Addresses in Currently Deployed IETF Transport Area Standards Track and Experimental Documents*, in *RFC 3794*. June 2004, ftp://ftp.rfc-editor.org/in-notes/rfc3794.txt.
5. Zhang, L., *A Retrospective View of NAT*, in *IETF Journal*. October 2007, http://www.isoc.org/tools/blogs/ietfjournal/?p=157#more-157.
6. S. Deering and R. Hinden, *Internet protocol, version 6 (IPv6) Specification*, in *RFC 2460*. December 1998, ftp://ftp.rfc-editor.org/in-notes/rfc2460.txt.
7. G. Tsirtsis and P. Srisuresh, *Network address translation - protocol translation (NAT-PT)*, in *RFC 2766*. February 2000, ftp://ftp.rfc-editor.org/in-notes/rfc2766.txt.
8. P. Srisuresh and K. Egevang, *Traditional IP Network Address Translator (Traditional NAT)*, in *RFC 3022*. January 2001, ftp://ftp.rfc-editor.org/in-notes/rfc3022.txt.
9. D. Wing and T. Eckert, *IP Multicast Requirements for a Network Address Translator (NAT) and a Network Address Port Translator (NAPT)*, in *RFC 5135*. February 2008, ftp://ftp.rfc-editor.org/in-notes/rfc5135.txt.
10. Hain, T., *Architectural Implications of NAT*, in *RFC 2993*. November 2000, ftp://ftp.rfc-editor.org/in-notes/rfc2993.txt.
11. Siyan, K., *An IP Address Extension Proposal*, in *RFC 1365*. September 1992, ftp://ftp.rfc-editor.org/in-notes/rfc1365.txt.
12. P. Srisuresh, et al., *DNS extensions to Network Address Translators (DNS_ALG)*, in *RFC 2694*. September 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2694.txt.
13. Bryan Ford, P. Srisuresh, and D. Kegel, *Peer-to-Peer Communication Across Network Address Translators.* USENIX Annual Technical Conference, February 2005.
14. S. Floyd, et al., *Quick-Start for TCP and IP*, in *RFC 4782*. January 2007, ftp://ftp.rfc-editor.org/in-notes/rfc4782.txt.
15. Postel, J., *User Datagram Protocol*, in *RFC 768*. August 1980, ftp://ftp.rfc-editor.org/in-notes/std/std6.txt.
16. Y. Rekhter, et al., *Address Allocation for Private Internets*, in *RFC 1918*. February 1996, ftp://ftp.rfc-editor.org/in-notes/rfc1918.txt.
17. P. Srisuresh and M. Holdrege, *IP Network Address Translator (NAT)*

*Terminology and Considerations*, in *RFC 2663*. August 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2663.txt.

18. Vijay K. Gurbani and R. Jain, *Contemplating Some Open Challenges in SIP.* Bell Labs Technical Journal 9(3),, 2004.
19. J. Rosenberg, et al., *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, in *RFC 3489*. March 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3489.txt.
20. R. P. Swale, et al., *Middlebox Communications (midcom) Protocol Requirements*, in *RFC 3304*. August 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3304.txt.
21. M. Stiemerling, J. Quittek, and T. Taylor, *Middlebox Communication (MIDCOM) Protocol Semantics*, in *RFC 5189*. March 2008, ftp://ftp.rfc-editor.org/in-notes/rfc5189.txt.
22. B. Fox and B. Gleeson, *Virtual Private Networks Identifier*, in *RFC 2685*. September 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2685.txt.
23. F. Adrangi and H. Levkowetz, *Problem Statement: Mobile IPv4 Traversal of Virtual Private Network (VPN) Gateways*, in *RFC 4093*. August 2005, ftp://ftp.rfc-editor.org/in-notes/rfc4093.txt.
24. M. Handley, V. Jacobson, and C. Perkins, *SDP: Session Description Protocol*, in *RFC 4566*. July 2006, ftp://ftp.rfc-editor.org/in-notes/rfc4566.txt.
25. Rosenberg, J., *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols.* draft-ietf-mmusic-ice-18, November 2006.
26. Senie, D., *Network Address Translator (NAT)-Friendly Application Design Guidelines*, in *RFC 3235*. January 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3235.txt.
27. IANA, *Special-Use IPv4 Addresses*, in *RFC 3330*. September 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3330.txt.
28. P. Ferguson and H. Berkowitz, *Network Renumbering Overview: Why would I want it and what is it anyway?*, in *RFC 2071*. January 1997, ftp://ftp.rfc-editor.org/in-notes/rfc2071.txt.
29. T. Bates and Y. Rekhter, *Scalable Support for Multi-homed Multi-provider Connectivity*, in *RFC 2260*. January 1998, ftp://ftp.rfc-editor.org/in-notes/rfc2260.txt.
30. R. Moskowitz, et al., *Host Identity Protocol*, in *RFC 5201*. April 2008, ftp://ftp.rfc-editor.org/in-notes/rfc5201.txt.
31. P. Srisuresh, B. Ford, and D. Kegel, *State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)*, in *RFC 5128*. March 2008, ftp://ftp.rfc-editor.org/in-notes/rfc5128.txt.
32. F. Audet and C. Jennings, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*, in *RFC 4787*. January 2007, ftp://ftp.rfc-editor.org/in-notes/rfc4787.txt.
33. H. Levkowetz and S. Vaarala, *Mobile IP Traversal of Network Address Translation (NAT) Devices*, in *RFC 3519*. April 2003,

ftp://ftp.rfc-editor.org/in-notes/rfc3519.txt.

34. M. Borella, et al., *Realm Specific IP: Framework*, in *RFC 3102*. October 2001, ftp://ftp.rfc-editor.org/in-notes/rfc3102.txt.

35. M. Borella, J. Lo, and K. Taniguchi, *Realm Specific IP: Protocol Specification*, in *RFC 3103*. October 2001, ftp://ftp.rfc-editor.org/in-notes/rfc3103.txt.

36. M. Garcia-Martin, et al., *The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp)*, in *RFC 3485*. February 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3485.txt.

37. H. Schulzrinne, et al., *RTP: A Transport Protocol for Real-Time Applications*, in *RFC 3550*. July 2003, ftp://ftp.rfc-editor.org/in-notes/std/std64.txt.

38. H. Schulzrinne, A. Rao, and R. Lanphier, *Real Time Streaming Protocol (RTSP)*, in *RFC 2326*. April 1998, ftp://ftp.rfc-editor.org/in-notes/rfc2326.txt.

39. F. Cuervo, et al., *Megaco Protocol Version 1.0*, in *RFC 3015*. November 2000, ftp://ftp.rfc-editor.org/in-notes/rfc3015.txt.

40. Jiri Kuthan and D. Sisalem, *SIP: More Than You Ever Wanted To Know About.* Tekelec, March 2007.

41. H. Schulzrinne, et al., *RTP: A Transport Protocol for Real-Time Applications*, in *RFC 1889*. January 1996, ftp://ftp.rfc-editor.org/in-notes/rfc1889.txt.

42. Wing, D., *Symmetric RTP / RTP Control Protocol (RTCP)*, in *RFC 4961*. July 2007, ftp://ftp.rfc-editor.org/in-notes/rfc4961.txt.

43. Michael Jeronimo and J. Weast, *UPnP Design by Example: A Software Developer's Guide to Universal Plug and Play*. Intel Press. April 2003.

44. *UPnP Forum*, http://www.upnp.org/default.asp.

45. T. Berners-Lee, R. Fielding, and L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, in *RFC 3986*. January 2005, ftp://ftp.rfc-editor.org/in-notes/rfc3986.txt.

46. Deering, S., *Host Extensions for IP Multicasting*, in *RFC 1112*. August 1989, ftp://ftp.rfc-editor.org/in-notes/rfc1112.txt.

47. T. Berners-Lee, R. Fielding, and H. Frystyk, *Hypertext Transfer Protocol -- HTTP/1.0*, in *RFC 1945*. May 1996, ftp://ftp.rfc-editor.org/in-notes/rfc1945.txt.

48. J. C. Mogul, et al., *Use and Interpretation of HTTP Version Numbers*, in *RFC 2145*. May 1997, ftp://ftp.rfc-editor.org/in-notes/rfc2145.txt.

49. R. Fielding, et al., *Hypertext Transfer Protocol -- HTTP/1.1*, in *RFC 2616*. June 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt.

50. O. Levin, R. Even, and P. Hagendorf, *XML Schema for Media Control*, in *RFC 5168*. March 2008, ftp://ftp.rfc-editor.org/in-notes/rfc5168.txt.

51. N. Mitra and Y. Lafon, *SOAP Version 1.2 Part 0: Primer (Second Edition)*, in *W3C Recommendation*. April 2007, http://www.w3.org/TR/2007/REC-soap12-part0-20070427/.

52. Martin Gudgin, et al., *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*, in *W3C Recommendation*. April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/.

53. J. Kempf and G. Montenegro, *Finding an RSIP Server with SLP*, in *RFC 3105*.

October 2001, ftp://ftp.rfc-editor.org/in-notes/rfc3105.txt.

54. J. Postel, T. Li, and Y. Rekhter, *Best Current Practices*, in *RFC 1818*. August 1995, ftp://ftp.rfc-editor.org/in-notes/rfc1818.txt.

55. A. Vemuri and J. Peterson, *Session Initiation Protocol for Telephones (SIP-T): Context and Architectures*, in *RFC 3372*. September 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3372.txt.

56. G. Montenegro and M. Borella, *RSIP Support for End-to-end IPsec*, in *RFC 3104*. October 2001, http://tools.ietf.org/html/rfc3104.

# 8 Appendices

## 8.1 Glossary & Abbreviations

| | |
|---|---|
| **ALG** | Application Layer Gateway |
| **ASP** | Active Server Page |
| **BNF** | Backus-Naur Form |
| **CP** | Control Point |
| **DHCP** | Dynamic Host Configuration Protocol |
| **DNS** | Domain Name Server |
| **FQDN** | Fully Qualified Domain Name |
| **HTTP** | Hypertext Transfer Protocol |
| **HTTPMU** | HTTP over Multicast UDP |
| **HTTPU** | HTTP over UDP |
| **IANA** | Internet Assigned Numbers Authority |
| **ICE** | Interactive Connectivity Establishment |
| **IETF** | Internet Engineering Task Force |
| **IP** | Internet Protocol |
| **IPv4** | Internet Protocol version 4 |
| **IPv6** | Internet Protocol version 6 |
| **ISP** | Internet Service Provider |
| **ITU** | International Telecommunications Union |
| **MAC** | Media Access Control |
| **MEGACO** | Media Gateway Control Protocol |
| **MIDCOM** | Middlebox Communication |
| **MR** | Media Render |
| **MS** | Media Server |
| **NAT** | Network Address Translation |
| **RFC** | Request For Comments |
| **RGw** | Residential Gateway |
| **RHN** | Remote Helper Node |
| **RSA-IP** | Realm Specific Address IP |
| **RSAP-IP** | Realm Specific Address and Port IP |
| **RSIP** | Realm Specific IP |
| **RTP** | Real-time Transport Protocol |
| **RTCP** | Real-time Transport Control Protocol |
| **RTCP** | Real-Time Streaming Protocol |
| **SDP** | Session Description Protocol |
| **SIP** | Session Initiation Protocol |
| **SOAP** | Simple Object Access Protocol |

| | |
|---|---|
| **STUN** | Simple Traversal of User Datagram Protocol |
| **TCP** | Transmission Control Protocol |
| **TURN** | Traversal Using Relay NAT |
| **UDP** | User Datagram Protocol |
| **UNSAF** | UNnilateral Self-Address Fixing |
| **UPnP** | Universal Plug and Play |
| **UPnP A/V** | UPnP Audio/Video |
| **URI** | Uniform Resource Identifier |
| **URL** | Uniform Resource Locators |
| **URN** | Uniform Resource Names |
| **VPN** | Virtual Private Network |
| **WLAN** | Wireless LAN |
| **XML** | eXtensible Markup Language |

## 8.2 The Codes of the Relayed SOAP Message

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
 <env:Header>
  <m:publicaddressofmediaserver
xmlns:m="http://publicaddress.mediaserver.com"
    env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
       env:mustUnderstand="true">
   <m:dateAndTime>2008-05-29T13:20:00.000-05:00</m:dateAndTime>
  </m:publicaddressofmediaserver>
  <n:NATportnumberofmediaserver
xmlns:n="http://NATportnumber.mediaserver.com"
    env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
  </n: NATportnumberofmediaserver >
 </env:Header>
 <env:Body>
  <p:messagepath
    xmlns:p="http://SOAPmessagepath.com/description">
   <p:SOAPintermediary1>
     <p:location>visit residential network</p:location>
     <p:publicIPaddress>public   IP   address   of   SOAP   intermediary
1</p:publicIPaddress>
     <p:NATportnumber>NAT    port    number    of    SOAP    intermediary
1</p:NATportnumber>
   </p:SOAPintermediary1>
   <p:SOAPintermediary2>
     <p:location>home residential network</p:location>
     <p:publicIPaddress>public   IP   address   of   SOAP   intermediary
2</p:publicIPaddress>
     <p:NATportnumber>NAT    port    number    of    SOAP    intermediary
2</p:NATportnumber>
   </p:SOAPintermediary2>
  </p:messagepath>
 </env:Body>
```