Journal of Computing and Information Technology - CIT 17, 2009, 2, 185–194
doi:10.2498/cit.1001196

TUKAB: An Efficient NAT Traversal Scheme on Security of VoIP Network System Based on Session Initiation Protocol

K. M. Azharul Hasan, Ifta Khirul and Kamrul Islam

Khulna University of Engineering and Technology, Bangladesh

Voice over Internet Protocol (VoIP) is subject to many security threats unique to both telephony and traditional Internet data transmission. As adoption of Session Initiation Protocol (SIP)-based telephony increases, concerns are rising over risks to system confidentiality, integrity and availability. Currently, several VoIP security tools are available to detect vulnerabilities and protect against attacks. In this paper we present various issues concerning the security of VoIP. A brief discussion of the SIP protocol is presented based on its operating principle. Finally, we proposed a solution for the Network Address Translation (NAT) traversal problem of SIP-based networks. This solution supports all types of NAT and maintains the current VoIP architecture. Based on our experiment, we examined the latency, buffer size and voice packet loss under various network conditions. We found that it is possible to establish a call from outside the NAT to inside, maintaining the quality issues of VoIP call. With this approach, it is possible to use the current network architecture by making some minor changes of the Register Server. Hence we evaluate our model, showing the QoS conditions that achieve both high efficiency and secure voice transmission. Sufficient simulation results are presented to verify our model.

Keywords: network address translation, VoIP, session initiation protocol, STUN, TURN, latency, packet loss

1. Introduction

VoIP has become very popular as it offers less cost to consumers over traditional telephone networks. By moving away from the public switched telephone networks, long distance phone calls become very inexpensive. Instead of being processed voice across conventional telecommunication line configurations, voice traffic travels on the Internet or over private data network lines. Because VoIP utilizes a compressed and packetized digital format, the potential for advanced multimedia, multi-service applications are virtually limitless. These include Web-enabled call centers, collaborative white boarding and personal productivity applications such as unified message handling.

Basic Network Address Translation or Basic NAT is a method by which IP addresses are mapped from one group to another, transparent to end users Network Address Translation (NAT) is being used by many service providers and private individuals as a way to allow many hosts to access the Internet via a small number of gateway IP addresses. An enterprise may have a block of IP addresses assigned to them, but may have many more computers than the allocated IP addresses. NAT [1][2] solves this problem by mapping internal addresses to external or public addresses. An internal IP *(address, port)* pair is mapped to an external *(address, port)* pair, and whenever the NAT receives a packet from the external *(address, port)* pair, it knows how to reroute the packet back to the internal *(address, port)* pair.The mapping is valid for some predefined mapping interval after which, in the absence of network traffic between the two communicating parties, this mapping may be expunged. In all cases, we must assume that an application will send and receive packets on the same port.

When we use Session Initiation Protocol (SIP)[7], we need to handle two aspects for making up the session:

(a) *Signaling*: Signaling is the process of "setting up the call". The endpoints exchange addressing and media information and reach agreement on the parameters of the session.

(b) *Media flow*: Media flow is established upon successful completion of call setup and involves the transfer of media packets between the endpoints.

In case of SIP, it is important to make sure that both the signaling and the media can traverse NAT's on either or both sides of the end-to-end session or dialog.

For media sessions traverse NAT, a lightweight detection protocol, STUN, allows a device inside the NAT to determine the NAT's behavior and bindings indirectly, and to modify the protocol messages appropriately. The drawback of STUN is that it cannot work with symmetric NAT, which is widely used in today's enterprise. Therefore Simple Traversal of UDP through NATs (STUN) cannot provide a complete solution. Traversal Using Relay NATs (TURN) solves this problem by relaying data through a server that resides on the public Internet. A device behind the NAT would use TURN protocol to get the address and port on the TURN server and then use them to invite its peer. It is a feasible way to pass through all kinds of NAT, but it is also expensive to the provider of the TURN server. However, there is an exception; if one party is behind a symmetric NAT and the other is not, we still choose TURN, which is not the shortest way. In this paper, a triggering method called Traversal Using Keep Alive Binding (**TUKAB**) is proposed to seek or create the shortest path for SIP messages and its associated media sessions to pass through NATs. In the following, we will call our proposed model TUKAB, for better understanding of the paper.

Unfortunately, NAT reduces the number of options for providing security. With NAT, nothing that carries an IP address or information derived from an IP address can be encrypted. While most application-level encryption should be ok, this prevents encryption of the TCP header [2]. Some other drawbacks of NAT, therefore, include sparse end-to-end traffic matrix, probability of miss-addressing, hide the identity of hosts etc. Hence NAT is still a very good area of research and further development.

The paper is organized as follows: Section 2 presents an overview of SIP, Section 3 summarizes the existing solutions along with their limitations, in Section 4 the proposed TUKAB model is presented. The experimental results and their analysis is presented in Section 5. Finally, Section 6 outlines some concluding remarks.

2. Overview of SIP

Our proposed model is SIP-based. In this section we will discuss very shortly the idea of SIP for VoIP protocol currently used and then we will summarize the existing NAT traversal solutions.

H.323, a protocol suite defined by ITU-T, is for voice transmission over Internet (Voice over IP or VOIP). In addition to voice applications, H.323 provides mechanisms for video communication and data collaboration, in combination with the ITU-T T.120 series standards.

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions such as Internet telephony calls (VOIP). SIP can also invite participants to already existing sessions, such as multicast conferences. Media can be added to (and removed from) an existing session. SIP transparently supports name mapping and redirection services, which supports personal mobility – users can maintain a single externally visible identifier, regardless of their network location.

SIP is a simple, ASCII-based protocol that uses requests and responses to establish communication among the various components in the network and to ultimately establish a conference between two or more end points. Users in a SIP network are identified by unique SIP addresses. A SIP address is similar to an e-mail address and is in the format of sip:*userID@gateway*.com. Users register with a registrar server, using their assigned SIP addresses. When a user initiates a call, a SIP request is sent to a SIP server (either a proxy or a redirect server). The request includes the address of the caller and the address of the intended callee. Over time, a SIP end user might move between end systems. The location of the end user can be dynamically registered with the SIP server. The procedure is shown in Figure 1.

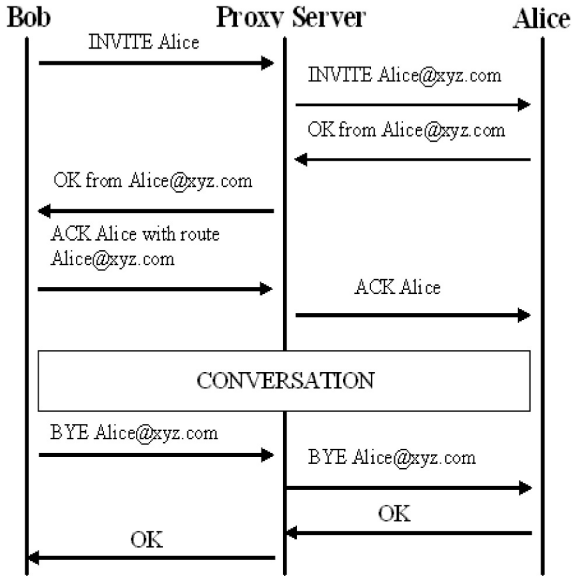

Figure 1. Establishing of SIP call.

With regard to SIP sessions, we need to address two aspects making up the session, namely signaling for call setup and the subsequent media flow. Signaling is the process of *setting up the call*. The endpoints exchange addressing and media information and reach agreement on the parameters of the session. Media flow is established upon successful completion of call setup and involves the transfer of media packets between the end points. When dealing with SIP, therefore, one must make sure that both the signaling and the media can traverse NAT's on either or both sides of the end-to-end session or dialog. NAT inhibits SIP's registration and communication mechanisms and requires innovative solutions to resolve these issues. The problems exist because in a SIP-based network, the SIP proxy is normally outside the NAT device. The major scenarios for using a SIP proxy include the followings

- The proxy is within the corporate LAN and the Teleworker connects from outside
- The proxy is at the telecom side and clients from, for instance, smaller companies connecting to this proxy for VOIP service
- Two administrative domains are connected, both have their own proxy.

So the problem is bartering communication between a proxy server that deals with global IP addresses and a machine that has been assigned a private network address. Rosenberg & Schulzrinne [8] classify three different sets of problems SIP traffic has in such architecture: originating requests, receiving requests, and handling RTP.

To initialize a session from behind the NAT, a caller can simply send an INVITE message. The outgoing port number will be preserved by the NAT, but response communication could be disturbed. If SIP is implemented over UDP (please note SIP is protocol independent) the proxy server must send the UDP response to the address and port the request arrived on. A simpler solution is to use the standard practice of routing SIP communication over TCP. With TCP, the response from the callee will come over the same channel as the original INVITE and so NAT will not cause a problem.

3. Related Works

In this Section we give a short description of the existing solutions and point out some problems of the existing solutions. UPnP [3], STUN [3], TURN [4], ICE [5] are four possible solutions for NAT traversal problem.

The UpnP[3], pushed by Microsoft (among others), defines a protocol named Universal Plug and Play (UPnP) that allows client applications to discover and configure network components, including NATs and firewalls, which are equipped with UPnP software. Using this technology, a client queries the NAT via UPnP, asking what mapping it should use if it wants to receive on port x. The NAT responds with the (*address, port*) pair that someone on the public Internet needs to address where to reach the client on this port x. One problem with UPnP is that it will not work in the case of cascading NATs. For example, say an ISP owns a block of IP addresses, but not enough to service its user base. The ISP would use a NAT to provide IP addresses to its customers. One of those customers may require many IP addresses (for example, an Internet cafe), so it would set up its own NAT to share its one address between many computers. If a client running on one of the local computers were to use UPnP to determine its public (*address, port*) pair, then it would only get back the innermost mapping (that of the Internet café's NAT) [9], but would still have a one way voice problem. The reason is that the public Internet would still not recognize the (*address, port*) pair that the client was giving, since

a second translation occurs between the Internet café's NAT and the public Internet via the ISP's NAT. There are also security issues that have not yet been addressed with UPnP. Contrary to prevalent security policies, it is the UPnP client (and not the firewall) that dynamically controls the opening of pinholes to the outside world.

Simple Traversal of UDP through NATs (STUN) [3] is a protocol for setting up the type of NAT probe that was just described. It actually does a bit more than just returns the public (*address, port*) pair – it can also help to determine the type of NAT the client is behind. Clients are already being developed that are STUN aware and can set their Session Description Protocol (SDP) messages accordingly. Note that the STUN server does not sit in the signaling or media data path.

Traversal Using Relay NATs (TURN) [7] complements STUN and places the probe in the signaling and media path. The probe actually *terminates* the media for both ends so that vis-a-vis the client the same probe that detected its (*address, port*) pair in the first place is also the probe that is sending the client media, so the symmetric problem is taken care of. Earlier, there were some security concerns with TURN, which may have contributed to its less than stellar adoption rate.

If we examine almost all the approaches presented above, we can see that they all involve the use of an intermediary server to set up and in many cases remain in the call flow during the session. Another point to note is that, as we have discussed above, not all endpoints support the same capabilities.

ICE [5] empowers the endpoints to determine the types of NAT's that exist between them and come up with a list of IP addresses through which the endpoints can communicate. The discovery process makes use of most of the mechanisms discussed above, including STUN and TURN.

4. The TUKAB Implementation Model

TUKAB is a complete solution approach of NAT traversal for SIP. It is not visible in the model described below. We use the Location Server for TUKAB purpose. We have completed our proposed solution by implementing the TUKAB technology in the location server. Hence TUKAB becomes a logical approach for extending the location server.

SIP has two important drawbacks:

i) It is necessary to initiate sessions from public networks into a private network.

ii) It also requires modifying the address information in the SIP messages into a reachable one.

These problems are resolved in our TUKAB and the overall efficiency is improved. The details about how TUKAB achieves these goals are described as follows.

4.1. Solutions of NAT Traversal for SIP

The end device behind a NAT can transmit a SIP request to the public network. It is impossible for the device to originate a session from public network to its peer located in a private realm, except when it knows the externally assigned IP address ahead. In TUKAB, the location server is modified to solve this problem. This modified location server can tell the client whether it is behind a NAT or not.

It can also act as a relay, receiving packets at the address it provides to clients, and forwarding them to the clients. The operation is illustrated in Figure 2.

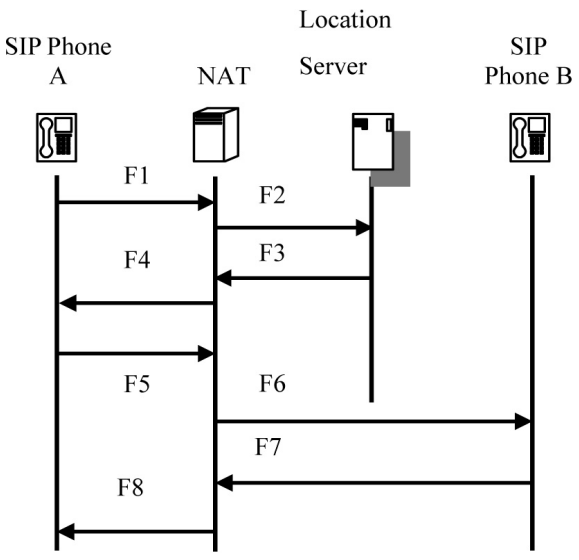

Figure 2. Registration with the public transport address provided by location server.

Before registration, the client SIP phone A sends a request (messages F1 and F2) using UDP to the location server. The application payload of the request contains the address information about where it comes from. Server compares this information with that in its own IP and UDP header. If they are different, it means there is at least one NAT in the path between the client and the server. In TUKAB, the server allocates a public transport address, Host Dummy Address (HDA), for relaying SIP messages to the client in the future. Afterward, a response is sent to the client (message F3); the response includes HDA and the network configuration. After the client learns it is in a private realm, it will register its location using Host Dummy Address (HDA), instead of its local address (message F5).

Hereafter, SIP proxy will deliver SIP requests to SIP phone A's Host Dummy Address (Message F9 in Figure 3). When location server receives packets on the HDA, it relays them towards SIP phone A (Message F10). Since the request has triggered an address binding on the NAT, and location server will periodically send dummy packets to keep the binding alive, the relayed packets can pass through the NAT and finally reach SIP phone A. Afterwards, SIP phone A sends SIP responses to the location server, and the server will forward these SIP messages to the SIP proxy (See messages F12, F13, and F14).

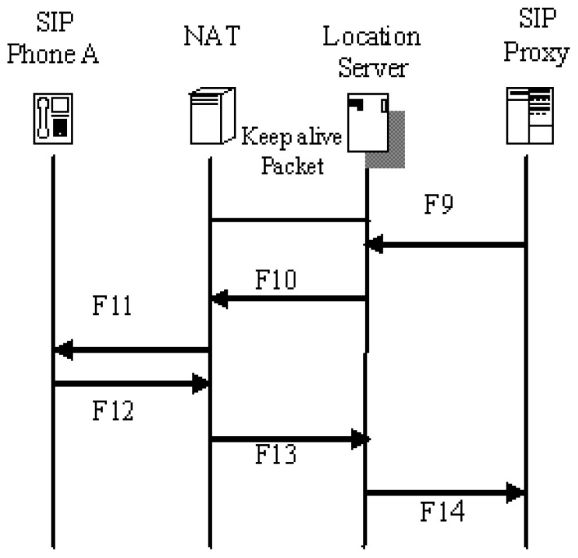

Figure 3. Session originated outside the NAT.

In Figure 4, SIP phone A is behind a NAT, but SIP phone B is not. Hence B can transmit RTP traffic to A's Media Dummy Address, and then location server will relay the packets to A. As the path is not the shortest one, A sends a Trigger Packet to B for triggering an address binding on the NAT. Now, B is aware of the shortest path, so it will send a message to inform location server to close A's Media Dummy Address.

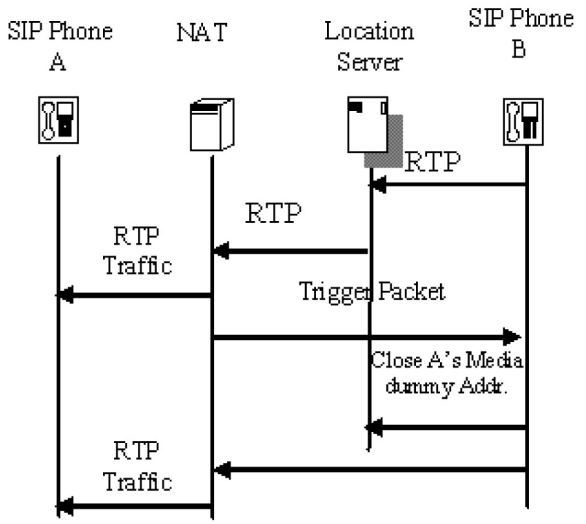

Figure 4. Triggering the shortest path.

Finally, B sends all the traffic to the source transport address of the Trigger Packet.

When both of the participants are behind NAT(s), they need to do a Connectivity Check to verify whether they are in the same private realm. The Connectivity Check is a UDP packet sent form one's local transport address to the other one's. If the check is succesful, they can communicate with each other directly (see Figure 5). If not, they will send traffic to the Media Dummy Address of the other party for relay (see Figure 6).

First, the client has to send a request to get a Host Dummy Address. If the client is in a private network, it should register its address with the SIP proxy, using the Host Dummy Address. Before inviting or answering its peer, it sends another request to get a Media Dummy Address. Then it inserts the Media Dummy Address and its local address into the SIP message. After exchanging the SIP messages, the SIP client knows whether its peer is behind a NAT or not. If the peer is in the public network, the SIP

client will use the Trigger Packet to establish the shortest communication channel. If not, it will do a Connectivity Check.

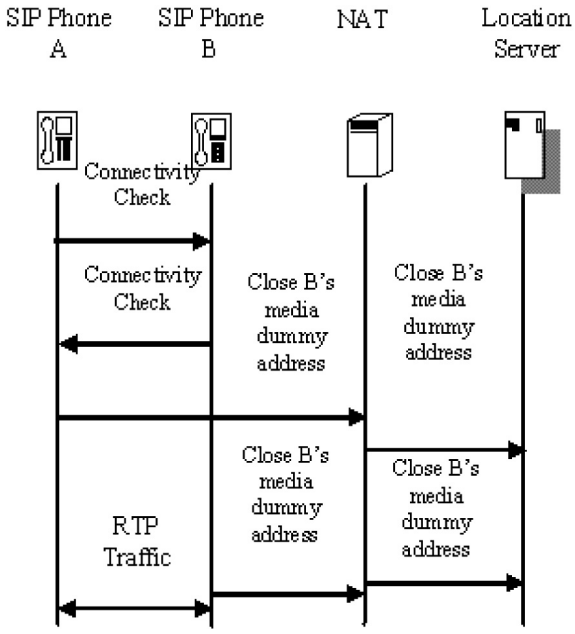

Figure 5. Behind the same NAT.

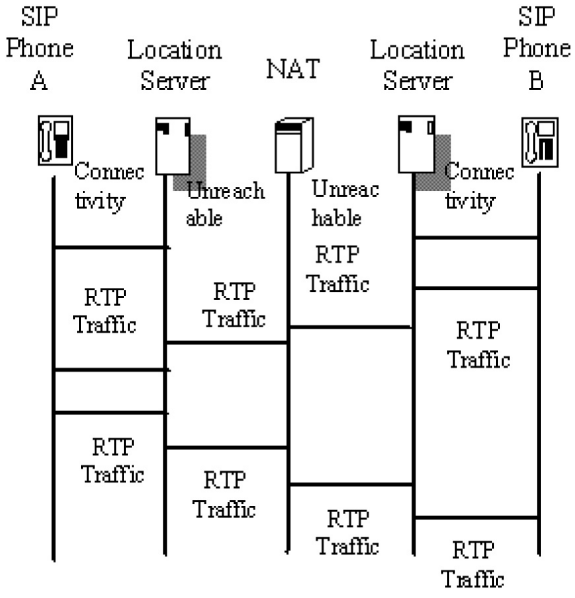

Figure 6. Behind different NATs.

5. Performance Analysis

5.1. Experimental Setup

To verify our traversal using Keep Alive Binding approach, we construct a SIP-based experimental platform, as shown in Figure 7. The devices are linked with 100-Mbps Ethernet connections. This platform comprises four main components, namely, SIP phones or hosts, NATs, location servers and routers. We chose JAVA as a language to implement the simulation. It offers multithreading which is extensively used in our simulation.

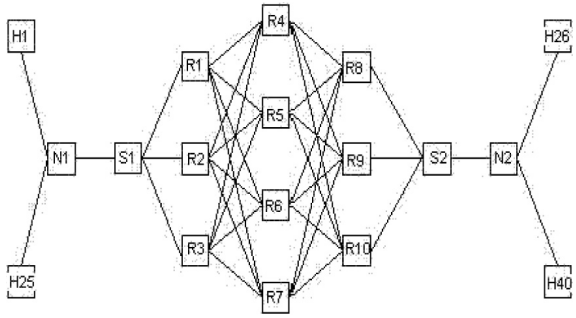

Figure 7. Experimental setup.

We assume that the location server, after initial Host Dummy Address and Media Dummy Address allocation, works like NAT since it merely does address translation, like NAT, while media stream passes through it.

In the simulation experiment, we employed two LANs, both behind NATs (N1 and N2), 40 Hosts (H1 – H40), 3 layers of Routers (R1 – R10), which are fully connected and two location servers (S1, S2). The Registrar Server which is integrated in the location server is attached to every Host.

We used the same number (termed as uniform) of Hosts/Clients and different number of (termed as non uniform) Hosts/Clients as shown in Table 1 and Table 2 respectively.

H1	H2
20	20
30	30
40	40
50	50
...	...
500	500

Table 1. Variation of Hosts/Clients (uniform distribution of Hosts).

The main concern is the bandwidth consumption. The dummy packets which are sent every minute, consume significant amounts of bandwidth. For the packet size of 40 bytes, the bandwidth consumption is as follows:

Bandwidth Consumption = (Number of Hosts * 40*8/60) bps.

H1	H2
25	15
35	25
45	35
55	45
...	...
505	495

Table 2. Variation of Hosts/Clients (nonuniform distribution of Hosts.

Generally, there are 100 $\sim$ 200 hosts per LAN in most of the situations, so in a 10Mbps backbone network (currently used) it does not add too much congestion.

Here, location server is used to determine whether the user is behind a NAT or not. The Registrar sends dummy packets before the timeout interval. As a result, the binding is kept alive. It is now possible to establish a call to both private realms, even after the connection is turned down.

5.2. QoS Issues and Analysis

Most of the security measures implemented in state of the art data networks could be used in VOIP networks. But, because of the time-critical nature of VOIP, and its low tolerance of disruption and packet loss, many security measures implemented in traditional data networks just are not applicable to VOIP in their current form. The main QoS issues associated with VOIP that affect security are presented in this section.

A. Determination of Packet Loss

Packet Loss occurs when packets do not arrive at their destination or arrive too late to be processed. Packet loss is usually perceived as gaps in the communication. Defense against packet loss may include sending redundant information and can be reduced with encoding schemes. Carrier VoIP networks avoid congestion by means of traffic engineering. VOIP is exceptionally intolerant of packet loss. Packet loss can result from excess latency. Compounding the packet loss problem is VOIP's reliance on RTP, which uses the unreliable UDP for transport, and thus does not guarantee packet delivery [5].

By varying the buffer size of NAT, ROUTER and SERVER, we calculate packet loss ratio and dummy packet loss ratio which is depicted in Figure 8.

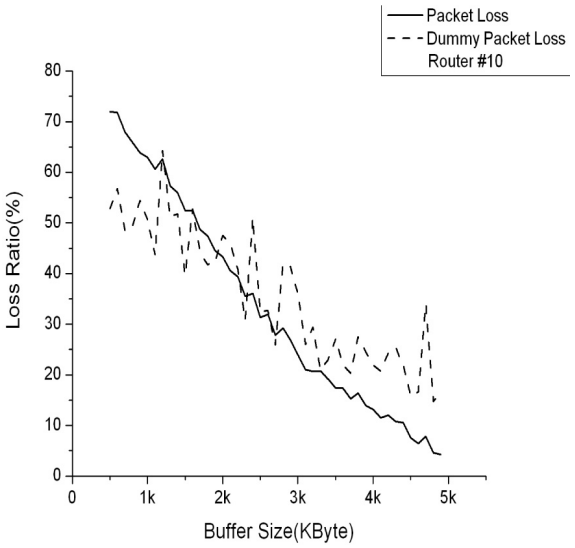

Figure 8. Buffer size vs. loss ratio curve.

Here we have used 10 routers arranged in three layers. We have found from the experimental result that when the buffer size reaches to about 4.5KB, the packet loss ratio drops to 10 percent and dummy loss to 15 percent. So, for such a system, if the buffer size is larger than 5KB, the packet loss ratio can be kept at acceptable minimum. Dummy packet is sent by the registrar periodically (usually in one minute's intervals). So after a certain amount of time, when the dummy packet arrives at the buffer and if it is full with packets at that time, dummy packet loss increases. This nature is shown by the irregular (Zigzag) pattern of the curve of the dummy packet loss.

We have also varied the number of routers in each layer. Figure 9 shows the result of buffer size and loss ratio when 16 routers were used.

By comparing with the previous result of Figure 8, we found that increasing the number of routers in each layer reduces packet loss. Figure 10 shows the comparison of Loss Ratio for varying routers. From this graph we investigate that increasing the number of routers decreases packet loss ratio. But the impact is not so significant.

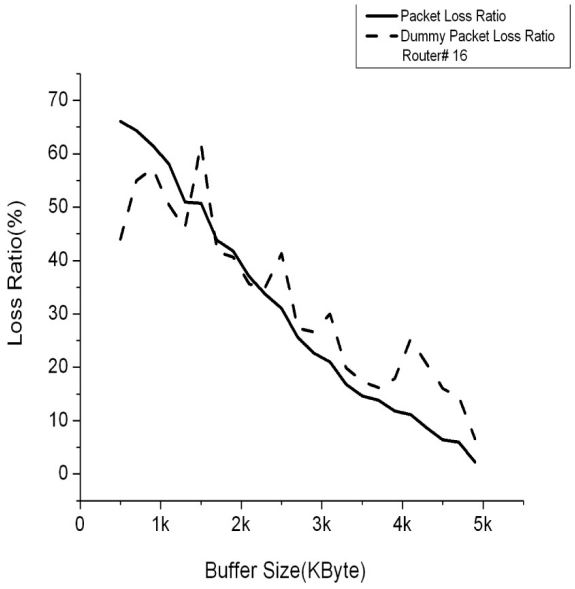

Figure 9. Buffer size vs. loss ratio curve.

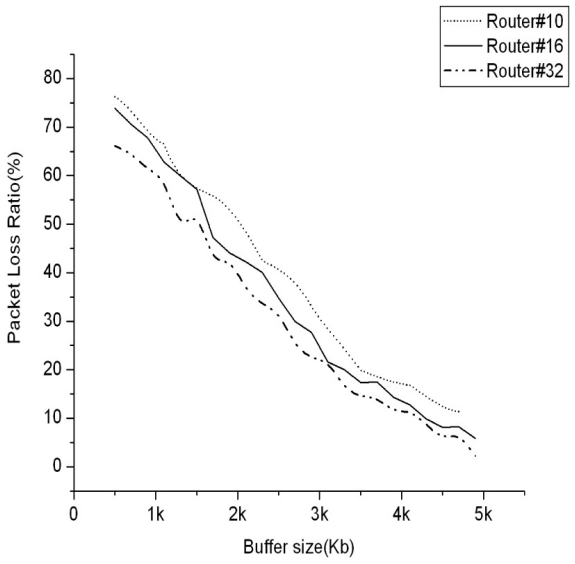

Figure 10. Comparison of loss varying routers.

B. Determination of Latency

Latency is the time it takes for the data to get from the source to the destination. The source is the person speaking into the phone and the destination is the listener at the other end. This is termed as one-way latency. Round trip time is the summation of one-way latency (for source) and one-way latency back (for destination). PSTNs have a round trip latency of less than 150 ms[3]. The 1996 ITU Recommendation for one-way end-to-end transmission limits is as follows:

- Under 150 ms: acceptable for most user applications
- 150 to 400 ms: acceptable, provided that administrators are aware of the transmission time impact on the transmission quality of user applications
- Over 400 ms: unacceptable for general networking purposes.

This gives us a tolerable range of latency of 75 ms to 400 ms for one-way. Since users of telephone systems have grown to expect less than 150 ms, in this section we assume 150 ms as the maximum cumulative latency.

In this experiment, we keep the number of host nonuniform under each NAT. Then we vary the number of hosts in both side and count the latency. Our experimental result is shown in Figure 11. It shows that if the number of Hosts is below 920 under each NAT, then the system preserves the quality of service. Initially, when the system is idle (no. of clients is very small) all the buffers are empty. At that time, packets need not wait to be assigned in the buffer for further processing. Queuing delay is almost zero.

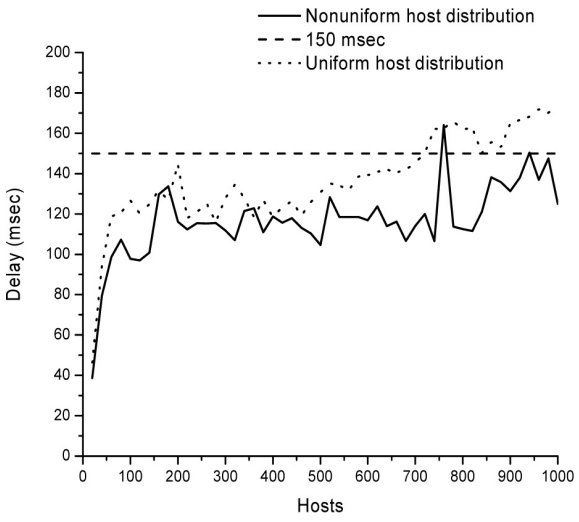

Figure 11. Delay curve for uniform and nonuniform distribution(s) of clients.

The latency for a particular packet in the system is low. The fact corresponds to the portion of the curve when the number of Hosts/clients is less than 100. The latency lies in an acceptable range for at the most 900 Hosts. The network congestion is at a moderate rate.

If the number of Hosts/clients crosses this limit, then quality of service cannot be maintained. This is because the latency rises above the 150ms threshold. Further increase in the number of clients makes the situation worse. And it is clearly evident from the curve (high rising portion of the curve).

Now we keep the number of hosts uniform under each NAT. Then we vary hosts in both sides and count the latency. Figure 11 shows the result. For nonuniform host distribution after 920 hosts, latency may cross the threshold value again, but we did not consider that case. Because at that point, quality of service deteriorated for some users. Due to sudden huge availability of network components, the threshold may be touched again, but it is not a general case.

TUKAB can find the shortest paths for the media sessions with only a trivial overhead. This means that the end-to-end system delay for our solution is shortest. It does not take much longer than 150 ms. This way, we have said that TUKAB can find the shortest paths for media sessions.

The result shows that if the number of hosts is below 710 under each NAT, the system maintains the quality of service. From the two results we observe that when the Hosts are distributed uniformly then number of Hosts that the system can support preserving the QoS decreases, due to overall network congestion. This extra latency is added due to equal contribution of clients at both ends of the network.

6. Concluding Remarks

The need for IP Address translation arises when a network's internal IP addresses cannot be used outside the network, either for privacy reasons or because they are invalid for use outside the network. We have proposed a solution for establishing sessions using SIP through NAT. An efficient approach called Traversal Using Keep Alive Binding is designed for implementation at both the end devices and of modifying the location server. This method did not modify the design of NAT, so there was no need to upgrade the existing NAT products. Moreover, this method is applicable for all kinds of NATs and compatible with traditional SIP devices located in public network. Despite the network topology and deployment configuration, Traversal Using Keep Alive Binding can find the shortest paths for the media sessions with only a trivial overhead. While there are a number of VoIP solutions available today, most of them have limitations of one kind or another. We believe that, the Traversal Using Keep Alive Binding is committed to providing a next generation network that offers both full multi-vendor interoperability and support for a full featured, secure PSTN service.

References

[1] P. SRISURESH, K. EGEVANG, Traditional IP Network Address Translator, RFC3022, 2001.

[2] K. EGEVANG, P. FRANCIS, The IP Network Address Translator (NAT), RFC 1631, 1994.

[3] J. ROSENBERG, J. WEINBERGER, J. HUITEMA, R. MAHY, STUN-Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs), RFC 3489, March 2003.

[4] J. ROSENBERG, Traversal Using Relay NAT (TURN), Draftrosenberg-midcom-turn-04, February 2004.

[5] J. ROSENBERG, Interactive Connectivity Establishment (ICE), IETF Draft, July 2005.

[6] D. R. KUHN, T. J. WALSH, S. FRIES, Special Publication: Security Considerations for Voice Over IP Systems, NIST, Jan 2005.

[7] J. ROSENBERG, G. CAMARILLO, Examples of Network Address Translation (NAT) and Firewall Traversal for the Session Initiation Protocol (SIP), draft-rosenbergsipping-nat scenarios-02, December 2003.

[8] P. MEHTA, S. UDANI, Overview of Voice over IP. Technical Report MS-CIS-01-31, Department of Computer Information Science, University of Pennsylvania, February 2001.

[9] J. ROSENBERG, H. SCHULZRINNE, An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing, RFC 3581, August 2003.

Received: January, 2008
Revised: February, 2009
Accepted: April, 2009

Contact addresses:
Dr. K. M. Azharul Hasan
Assistant Professor
Department of Computer Science
and Engineering (CSE)
Khulna University of Engineering
and Technology (KUET)
Khulna-9203, Bangladesh
Tel: +880 1714087273
Fax: +880 41 774403
e-mail: azhasan@gmail.com
azhasan@cse.kuet.ac.bd

Ifta Khirul
Software Engineer
Institute of Information
and Communication Technology (IICT)
Bangladesh University of Engineering
and Technology (BUET)
Dhaka-1000, Bangladesh
Tel: +880 1190547558
e-mail: ifticse_kuet@hotmail.com

Md. Kamrul Islam
Flat-902, Building-23
Japan Garden City, Ring road
Mohammadpur, Dhaka-1207, Bangladesh
Tel: +880 1615473618
e-mail: md.kamrul.islam@hotmail.com
md.kamrul.islam@ericsson.com

K. M. AZHARUL HASAN received his B.Sc. (Engg.) from Khulna University, Bangladesh in 1999 and M. E. from Asian Institute of Technology (AIT), Thailand in 2002, both in computer science. He received his Ph.D. in information science from the Graduate School of Engineering, University of Fukui, Japan in 2006. He is now Assistant Professor at the Department of Computer Science and Engineering at Khulna University of Engineering and Technology (KUET), Bangladesh. His research interests include database, VoIP system, wireless sensor network, data warehousing, parallel algorithms, information retrieval, multidimensional databases, and software engineering.

IFTA KHIRUL received his B.Sc. (Engg.) in computer science and engineering from the Khulna University of Engineering and Technology (KUET) in 2007 and is pursuing his M.Sc. (Engg.) from Bangladesh University of Engineering and Technology (BUET) in information and communication technology. He has been working as a software engineer in IICT, BUET since January, 2008. His research interests include bioinformatics computing, parallel algorithms, graph theory, VoIP system, software engineering, database and data warehousing.

MD. KAMRUL ISLAM received his B.Sc. (Engg.) in computer science and engineering from Khulna University of Engineering and Technology (KUET) in 2007. He was a Lecturer at the Department of Computer Science and Engineering in Eastern University, Dhaka, Bangladesh. Recently, he is working as an integration engineer in L.M. Ericsson Bangladesh LTD. His research interests include bioinformatics computing, parallel algorithms, graph theory, VoIP system, network security, wireless sensor network, performance evaluation of 802.11 NAC protocol in wireless network, software engineering, database and data warehousing.