In: Videoconferencing: Technology, Impact and Applications ISBN: 978-1-61668-285-9
Editor: Adam C. Rayler

Chapter 5

PROVIDING QUALITY END-TO-END VIDEOCONFERENCE SERVICES IN IP NETWORKS

Christos Bouras and Kostas Stamos*

Computer Engineering and Informatics Dept., University of Patras,
Greece and Research Academic Computer Technology Institute, Greece

ABSTRACT

The development of quality and user satisfying videoconference applications at a low cost has been hindered in the past by, among other reasons, low quality connections and a difficulty for the average user to establish end-to-end connections without hassle. In this chapter, we describe some of the latest methods and developments that overcome these problems. In particular, we discuss ways to make establishment of universal videoconference connections easier by overcoming the problem of participants that reside behind NAT routers, by deploying new protocols at the application or even the network layer. An example of the former case is the development of establishment protocols specifically designed for this purpose, while an example for the latter case is the deployment of IPv6, which aims to make NAT deployments obsolete. Furthermore, we discuss efforts to provide targeted support for quality of experience in networks that support some kind of traffic classification through the usage of dynamic mechanisms and dynamic network re-configurations. Apart from describing the state of the art in the aforementioned areas, we provide detailed insight in specific research efforts that have taken place and assess their results in the overall user experience.

1. INTRODUCTION

Videoconferencing is the live exchange of audio and video information between people that find themselves in distant locations and are connected through a telecommunications channel. In order to place phone calls over the Internet, a number of open and closed VoIP

* Corresponding author: Email: {bouras, stamos}@cti.gr.

protocols have been developed. Videoconferencing offers cost and time advantages and through encryption can offer secure communication.

Traditionally, similar protocols have been used for the establishment of both VoIP and videoconference sessions, such as the H.323 and SIP (Session Initiation Protocol) standards. With the advent of cheap end user equipment and fast, reliable broadband connections, many other services such as instant messanging and VoIP clients have also added video to their conferencing capabilities (for example ICQ [12][13][11], Google Talk [13] and Skype [11]).

However VoIP and videoconferencing applications have not always been successful in offering a trouble-free experience to the user. Two of the main problems that users face have been the difficulty in setting up calls, especially when one or both of the communicating parties reside behind firewalls or NAT routers, and the difficulty to achieve satisfying quality under all circumstances, since Internet has been designed as a best-effort network and does not typically offer guaranteed service quality. The problem is that Internet traffic consists of flows generated by different applications, and all flows typically receive the same treatment from the network. However, real-time applications, such as VoIP and videoconferencing, are sensitive on parameters such as delay, packet loss or jitter. A worsening of these parameters results in measured inferior quality, as perceived by the users of a videoconference. This worsening can translate into interrupted audio or video, distorted speech and other artifacts that significantly reduce the user experience.

A lot of research and commercial effort has been spent in order to overcome the above problems and provide the opportunity for high quality end to end videoconferencing. Furthermore, the growth of the Internet has led the IETF to propose a new Internet Protocol, IPv6 [2], for a long-term solution to the address space shortage problem. Despite being standardized for over a decade, IPv6 has not yet enjoyed wide deployment. However, due to the oncoming IPv4 address space exhaustion [3], a transition to IPv6 seems inevitable in the near future. The abundance of IPv6 address space is also expected to obsolete NAT solutions that have hindered end-to-end videoconference deployment.

The rest of this chapter provides some background and describes some solutions that have been successfully deployed. In particular, the first part of the chapter deals with the establishment of end-to-end videoconferences and the solutions in order to traverse troublesome configurations such as NATs, while the second part of the chapter deals with the provisioning of quality service in a dynamic way, as videoconferencing participants enter and leave a call.

2. ESTABLISHING END-TO-END VIDEOCONFERENCES

The scarcity of IPv4 addresses led to the development of the NAT (Network Address Translation) [1] technology for IP Masquerading, which is typically implemented in firewalls and routers in order to allow a single IP address to be used for multiple devices within a local network that wish to have network connectivity to the outside world. Each device within the local network is assigned a private IP address from a specified set of addresses that can not be globally routed (such as 192.168.x.x or 10.x.x.x), and the NAT router or firewall/router is responsible for translating the private IP addresses from the internal network to the routable public IP address. The translation is done by replacing the source address in all outgoing

packets with the public IP address, while properly directing incoming packets (which all contain the same public IP address at the destination field) with the proper internal device. The NAT router keeps a translation table in order to identify which internal device has initiated which connection and properly forward incoming packets. The translation table keeps the basic data of each active connection, i.e. the destination address and the port number.

The problem with NAT is that the NAT router may not know where it should forward traffic originating outside the local network. Several rules and refinements have been proposed and implemented in various NAT devices that allow servers to be setup within the internal network, ranging from simple rules to forwards incoming connections to a specific machine, to port forwarding, where the connection port determines the receiving device.

However, NAT fundamentally alters the end-to-end concept upon which the Internet architecture is based, since the outside world thinks it is only communicating to the NAT router and not the actual internal machine. This breaks with assumptions made by protocols that expose IP information to upper layers, such as most standard video conference protocols. SIP and H.323 deal primarily with signalling and establishment of the connection, while audio and video traffic are transferred through a separate channel. Furthermore, these protocols utilize a number of ports that cannot be easily identified and dealt with by the NAT router/firewall. Therefore, the proper signalling sequence for establishing a call and then transferring audio and video data breaks because of the changes in the IP address contained in the exchanged packets.

The establishment of video conference calls between stations behind NAT configurations and the resulting user frustration and troubles has been a major hurdle in the widespread adoption of videoconferencing from casual users. Two main approaches can overcome this problem:

- Get rid of the need to use NAT
- Design protocols that can overcome hurdles related to NAT address translation.

For the first case, the most obvious solution is to migrate to the IPv6 protocol. IPv6 has been long touted as the inevitable replacement of IPv4, and because of its enormous increase in the address space, does away with the basic rationale for NAT usage. Lately it seems that widespread IPv6 adoption is coming closer. In this chapter, we detail the efforts to make a traditional videoconferencing protocol such as H.323 operate with IPv6 and therefore permit trouble-free end to end sessions [9].

The second approach has, for the time being, been more widely deployed in the form of new, often proprietary protocols such as Skype [11], which have become successful exactly because in large part they refrain from troubling the user with network issues and “just work”. In a later section of this chapter we give a detailed description of Skype as a characteristic case. We also discuss NAT traversal using the STUN protocol [4], which in many cases allows SIP and H.323 connections to work behind NAT setups.

2.1. H.323 Voice and Video Network

2.1.1. H.323 calls

H.323 is an ITU recommendation, which defines a network architecture and the associated protocols necessary to voice and multi-media calls establishment. H.323 is defined for a packet-based network, and does not impose any network protocol, which can as well be IPv4 as IPv6 or IPX. H.323 architecture makes it possible to carry out direct calls between two multimedia phones connected on the Internet or a local area network. In this case, it is necessary to know the IP address of the called party. The main entities of an H.323 based video network are the following:

- End points: These are the H.323 clients, which are used by the end users. They can propose phone, video, fax, and application sharing functionalities.
- Gateways: Gateways can be used for the interconnection between different networks (for example an IP phone network and a traditional phone network).
- Gatekeepers: Gatekeeper makes it possible to be freed from the knowledge of called party IP address. It is then possible to call someone by his name. The gatekeeper is also able to manage the billing, and call filtering/authorization.
- Multipoint Control Units (MCUs): A Multipoint Control Unit (MCU) makes it possible to manage a conference of more than two end points. Each user connects to the MCU and then is able to discuss with all the other connected people.

Using a Gatekeeper or not, an H.323 phone call always follows the same logical order:

1. Optional registration to a gatekeeper (H.225)
2. Call setup, direct or thanks to gatekeeper (H.225)
3. Initial communication and capability exchange (H.245)
4. Establishment of audiovisual communication (H.245)
5. Call services (H.245, H.225)
6. Call termination (H.245, H.225)

The initial communication is done directly between the two phones. Those will thus have to use addresses understandable by both.

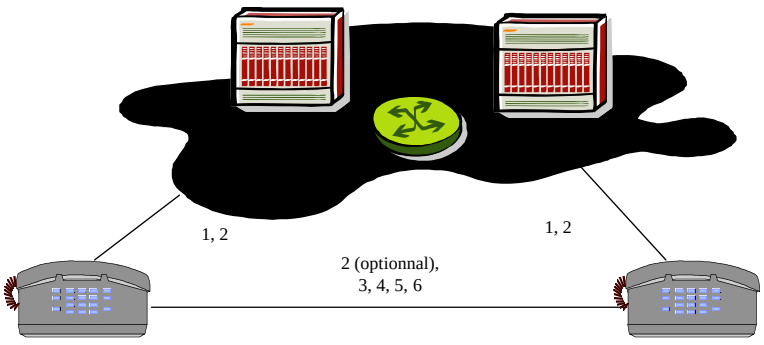

Figure 1. Call phases

	IPv4 server	**IPv6 server**
IPv4 client	Communicate using IPv4	Communicate using IPv4, server sees IPv4-mapped IPv6 address
IPv6 client	Can communicate using IPv4 if the IPv6 client uses an IPv4-mapped IPv6 address	Communicate using IPv6

Figure 2. Interoperability between IPv4 and IPv6 versions running on dual-stack hosts.

2.1.2. Impact of simultaneous IPv4 and IPv6 address on H.323

The current specification of the H.323 protocol makes it possible to manage calls independently of the network type. An H.323 transmission channel is defined by two communication terminations named endpoints: A source and a destination, which are defined by an address (IPv4, IPv6, IPX, etc.) and a Service Access Port (TCP/UDP Port).

In general, it is possible to achieve communication between IPv4 and IPv6 hosts, as shown on Figure 2.

It can however be a challenge to have an H.323 Call seamlessly using both IPv4 and IPv6 networks. When an IP phone software starts, it opens a communication port (Service Access Port) on which it could receive H.323 signaling. A dual-stack terminal can choose to open an IPv4 port, an IPv6 port or a dual protocol port. A dual port makes it able to receive both IPv4 and IPv6 call at the same time. On the other hand if it registers at a Gatekeeper, it has to make a choice and to specify an IPv4 or IPv6 address network.

As a gatekeeper maintains a list of the connected users (names or aliases), and corresponding endpoints (address and port), a dual terminal stack could be registered twice, once in IPv4 and another in IPv6.

It is possible to carry out an IPv4 and IPv6 mixed call between two dual stack terminals. Terminal A opens an IPv4 port and calls the terminal B, which opens an IPv6 port. The messages from A to B will be carried by the IPv6 network, and the messages from B to A will be in IPv4.

Problems arise if one wants to carry out a call between a pure IPv4 terminal and a pure IPv6 terminal. The use of mapped addresses is not possible, as H.225 messages at application layer clearly exchange IPv6 addresses.

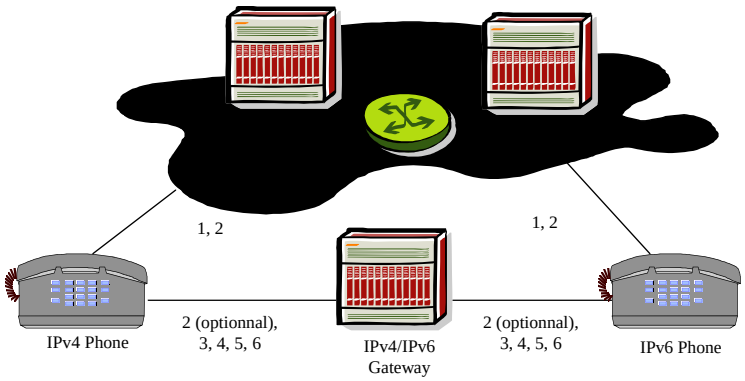

Figure 3. Call thanks to an IPv4/IPv6 Gateway.

A solution could be to use a gatekeeper connected to each network, able to detect this conflict and being used as Gateway for the transmission between the two networks.

Another solution could be to use at least one dual stack H323 client, able to detect an IPv4 only party, and sending the corresponding IPv4 address in H.225 messages instead of the mapped IPv6 address it is listening on.

Several efforts and experiments have been made with videoconferencing applications over IPv6 [9]. The problem of porting existing applications to IPv6 has been so far addressed by several researchers. A white paper by Microsoft [18] focuses on Windows applications, but at the same time offers some general guidelines that apply to any application for any operating system. In [19], the authors emphasize more on some general knowledge that a programmer must acquire before dealing with the problem of porting applications to IPv6, than on presenting step-by-step instructions. Furthermore, a number of research projects (6NET [20], Euro6IX [23], 6INIT [21], KAME [22]) have extensively investigated the migration effort and the benefits from IPv6, and have shared their experiences. The main objective has been to produce applications that can operate in both IPv4, IPv6 and dual-stack hosts. In general, these IPv6-enabled applications enable users to completely avoid the problematic NAT configurations and enjoy the benefits of global end-to-end connectivity. However, at the time being, despite very wide hardware and software support, at the network level IPv6 support is not universal among ISPs and therefore several intermediate solutions are required in order to achieve global IPv6 connectivity. Therefore, the user burden for proper NAT traversal configuration moves to the configuration of IPv6 transition mechanisms but is not eliminated. Nevertheless, most experts agree that IPv6 is the way of the future and as soon as it becomes the dominant Internet protocol, end-to-end communication is expected to become much easier.

2.2. NAT Traversal

Usually NAT configurations only permit TCP connections to be initiated from the hosts residing within the NAT network and not the outside Internet. Therefore, NAT traversal TCP connections can be quite problematic. Two commonly used options for overcoming this problem are relaying and connection reversal. The first method, relaying, is usually the only viable method if both communicating endpoints reside behind a NAT router. Relaying means that there needs to be a host with a global IP address acting as a relay between the two communicating endpoints. This relay host has TCP connections to both parties, and is then responsible for forwarding all traffic from one connection to the other. In most cases, the parties behind the NAT configurations need to have the relay associated with them before the TCP communication takes place. This means that in that way the address of the relay can be provided to each communicating party and not the actual address used by the communicating endpoint. TURN [8] is an example of a protocol that allocates and uses relay hosts. It is most useful for hosts that reside behind symmetric NAT routers or firewalls and intend to be on the receiving end of a connection to a single peer. TURN does not allow for users to run servers on well known ports if they are behind a NAT; it supports the connection of a user behind a NAT to only a single peer. In applications such as Skype, built on the peer-to-peer model, entities called supernodes act as relays and help the endpoints that reside behind NAT routers

to communicate. In these cases, the allocation of relay hosts depends on the protocol used for the allocation of supernodes.

For UDP communications the type of the NAT configuration used is very important. A UDP packet sent through the NAT from the internal part of the network to the external Internet always creates a binding within the NAT between the internal and the external address and port pairs. Some NAT configurations forward packets sent from any address to the external address according to an active binding. However, some NATs only allow packets from the address where the original UDP packet that established the binding was sent to.

2.2.1. NAT traversal using stun

Stun is an Internet standards-track suite of methods, including a network protocol, used in NAT traversal for applications of real-time voice, video, messaging, and other interactive IP communications.

The Stun protocol allows applications operating through a network address translator (NAT) to discover the presence of a network address translator and to obtain the mapped (public) IP address (NAT address) and port number that the NAT has allocated for the application's User Datagram Protocol (UDP) connections to remote hosts. The protocol requires assistance from a 3rd-party network server (STUN server) located on the opposing (public) side of the NAT, usually the public Internet. The original version of the protocol also specified methods to ascertain the specific type of NAT, but those methods have been deprecated in the newer specification, because of the plethora of specific NAT implementation behavior in various networking equipment and the resulting intractability of the problem and the deficiencies of the method used.

Stun is not a self-contained NAT traversal solution applicable in all NAT deployment scenarios and does not work correctly with all of them. It is a tool among other methods, most notably Traversal Using Relay NAT (TURN) and Interactive Connectivity Establishment (ICE).

Stun does work with primarily three types: full cone NAT, restricted cone NAT, and port restricted cone NAT. In the cases of restricted cone or port restricted cone NATs, the client must send out a packet to the endpoint before the NAT will allow packets from the endpoint through to the client. STUN does not work with symmetric NAT (also known as bi-directional NAT) which is often found in the networks of large companies. Since the IP address of the STUN server is different than that of the endpoint, in the symmetric NAT case, the NAT mapping will be different for the STUN server than for an endpoint. TURN offers better results with symmetric NAT.

2.2.2. Proprietary protocols (Skype)

One of the most attractive features of Skype [11] and probably an important factor to its success, has been its ability to “just work”, which means that users have no trouble making Skype calls even when they reside behind NAT routers and firewalls. Skype uses a proprietary, non-standardized and non-interoperable protocol that has not been made publicly available, however many important details about its operation are widely known. Specifically, it is based on a peer-to-peer model, and its architecture consists of two types of nodes, the supernodes and the ordinary nodes, as well as a central entity, the login server. A supernode is selected among ordinary Skype clients with the requirements of a public address and

sufficicient CPU, memory and network bandwidth, in order to improve the overall availability of the system. Skype forms an overlay network: Supernodes maintain an overlay network among themselves, while ordinary nodes pick one or more supernodes to associate with. Supernodes are grouped into slots and slots are grouped into blocks. They relay communications to other clients behind a firewall. Ordinary nodes issue queries through the supernodes they are associated with. Each client builds and refreshes a list of reachable nodes known as the host cache. The host cache contains IP address and port numbers of supernodes. TCP is the protocol used for signaling.

It is believed [15] that Skype uses its own version of the STUN protocol in order to determine the type of NAT or firewall it is behind. Furthermore, it seems that there is no server containing this information, because experiments have not found this type of network exchange of kind of information. The information about the type of NAT or firewall is dynamically determined and then locally stored and periodically refreshed in the Windows registry. A Skype client can traverse NAT routers and firewalls and needs no explicit NAT or firewall traversal server. The main characteristics that enable this are:

- The random selection of sender and listener ports by Skype: The client randomly chooses a port number upon installation, which can be configured in its connection dialog box.
- The usage of TCP as the voice streaming protocol when UDP can not be used: Specifically, Skype is based on the detection of the type of NAT or firewall. If both communicating endpoints use global IP address, then media traffic flows directly between them using UDP. If either one or both the endpoints are behind port-restricted NAT, they sent voice traffic to a node that acts as media proxy using UDP. If both endpoints reside behind port-restricted NAT and UDP-restricted firewalls, then the TCP protocol is used for transferring voice traffic over from another online Skype node.
- The peer-to-peer nature of the Skype network.
- The usage of the UDP hole punching technique: After an initial contact with the intermediate host in the public address space, the endpoints switch to direct communication hoping that the NAT devices will keep the states despite the packets coming from a different host.

Although Skype has achieved admirable NAT traversal in most cases and worldwide adoption, its lack of interoperability and lack of openness about the details of its architecture can be a limiting factor especially when interoperability with standard protocols and solution from different vendors or legacy equipment is a requirement.

3. PROVIDING TARGETED QOS

Setting up the videoconference successfully only solves half the problem, as many users find the experience to be underwhelming in terms of perceived quality. Videoconferencing presents a difficult challenge for best effort networks, because it requires a large amount of bandwidth and its real-time nature imposes strict requirements in terms of network parameters

such as delay and jitter. Many networks operate with a constant heavy load in order to maximize resource utilization, which increases buffer queues in the network's routing devices and degrades the user experience because of lost or delayed packets.

Therefore, this section deals with the issue of providing strict guarantees and ensuring high quality videoconferences in networks that implement the logic suitable to enforce these guarantees.

3.1. Dynamic Qos Provisioning on the Fly

This section describes a solution for dynamic QoS provisioning that was experimentally implemented at the greek research network GRNET [10]. GRNET provides network services for research and academic organizations (institutes and universities) in Greece, which are its clients.

Figure 4 shows a typical setup for a videoconference between three users. GRNET uses a central MCU in order to facilitate videoconferences with multiple participants. All users have to connect to this MCU, with which they exchange traffic. The MCU is responsible for receiving audio and video streams from the participants and then multiplexing the streams and transmitting it back to all the participants. This means that whenever the network experiences congestion, the quality of the whole videoconference can be affected. This applies even in the case where a single network link is congested. If the packets originating from a single participant traverse the congested link, all the participating users will experience degraded quality, because the audio and video stream will be multiplexed by the MCU and sent to everybody.

Grnet does implement Quality of Service (QoS), which guarantees traffic parameters for high quality connections. The architecture model is based on DiffServ [5] which is based on the concept of traffic classes. Each traffic class is mapped to a Per-Hop Behavior (PHB), and PHBs are implemented at routers by means of queuing and scheduling at congestion points, where queues are formed. By mapping different traffic types into different PHBs, routers are able to ensure service guarantees. The provisioning model at GRNET requires only policing at the network perimeter, while core routers implement priority queuing mechanism, and admission control based solely on the availability of IP premium bandwidth at the access links. User requests for QoS reservations are made through a management tool developed by GRNET, called ANStool [7].

The ANStool therefore allows the creation of permanent (or long-term) QoS configurations that would treat traffic from the MCU to every destination as priority traffic and treat traffic from every source to the MCU as priority traffic. This approach is however not attractive in the current setting of dynamic and short-term videoconferences. First of all, such a static configuration should be applied at the network perimeter to make sure traffic is marked as priority throughout its whole journey though the network. This means that there is no way to know whether incoming traffic corresponds to one or more users from the same source (client institute to GRNET), and therefore no way to calculate how much traffic should be prioritized. Allowing the maximum possible traffic would mean that the admission control architecture of the QoS provisioning service would be completely bypassed.

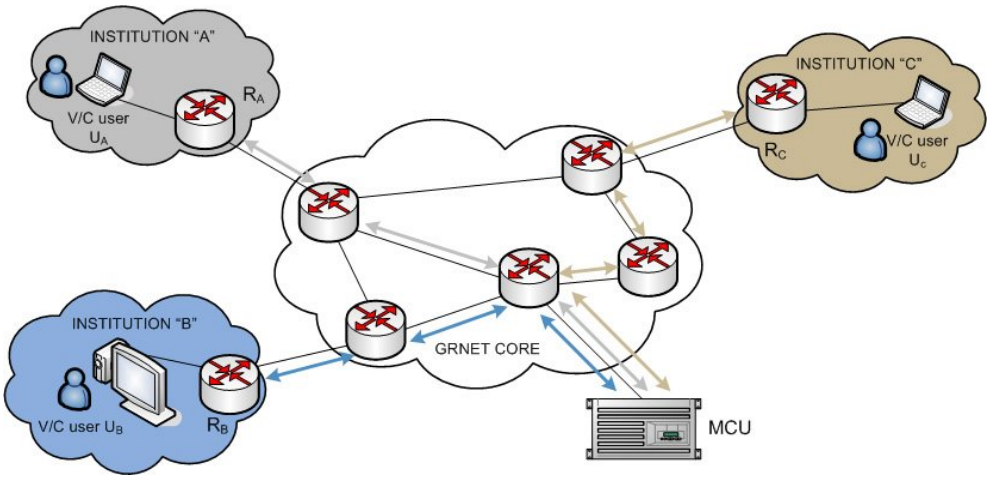

Figure 4. Typical videoconference setup [10]

A related problem is that the bypassing of admission control leads to the possibility of DOS (denial of service) attacks, by injecting malevolent traffic towards the MCU (illegitimate RTP packets cannot easily told apart and filtered out by ordinary router mechanisms such as access-lists). This traffic would be treated with priority throughout the network, with the danger of summing up to high bandwidths and disrupting the normal operation of the network, the MCU or both.

Therefore, GRNET decided to implement a mechanism that creates, propagates, and installs on-the-fly rules for admission control and marking of videoconference traffic at the network perimeter (where traffic is marked and policed for conformance to contracted-to traffic profiles). In this manner, only traffic from legitimate users is prioritized. No other users may inject priority traffic in the network. Moreover, this method performs admission control, so that if multiple users from a given institution participate in a videoconference, it is possible that only some of them will be allowed to inject priority traffic, while others will not.

The mechanism operates by taking into account a maximum amount of bandwidth that each client institute has allocated for videoconferencing. Once every minute, a script queries the MCU and retrieves the IP addresses of all users participating at selected videoconferencing sessions. The script then checks which of these participants are authorized for prioritized traffic and queries the network routers for their FIB (Forwarding Information Base) paths towards the QoS-authorized participants. The path information is stored in a database, where it is retrieved by a dedicated router daemon, that then advertises it to all routers as /32 addresses over a specially-crafted BGP (Border Gateway Protocol) [6] session. Therefore, the database serves as the intermediate between the script and the routing daemon, and contains an up-to-date view of the participants that require preferential (QoS) treatment in a dynamic way. The /32 routes advertised to the routers take precedence over the normal BGP route and the routers are configured to insert these routes in their FIB with an Expedited Forwarding tag, for QoS treatment. This configuration at GRNET is currently Cisco-specific, called by Cisco QoS Propagated Policy via BGP (QPPB). This tag signals that matching packets should be classified as priority traffic and sent to a low latency queue.

More technical and implementation details for this service can be found at [10].

4. CONCLUSION

We have seen in this chapter how the emergence of quality and user satisfying videoconference applications can be achieved by advanced techniques for overcoming past problems of NAT penetration and network quality guarantees. We have described recent methods for the establishment of universal quality videoconference connections and we have discussed how this application field will be affected by the upcoming migration of the Internet to IPv6.

Videoconference applications seem to have overcome many of the obstacles that hindered their widespread adoption in the past. Future problems might shift from technical challenges to challenges related to the end user perception of the services, for example the preference of audio communication over video and audio communication, usability features as well as the level of compatibility and interoperability of future videoconferencing solutions.

5. REFERENCES

[1] RFC, (20001). 3022, "*Traditional IP Network Address Translator* (Traditional NAT)", P. Srisuresh, & K. Egevang, (January).

[2] RFC, (1998). 2460, "*Internet Protocol,* Version 6 (IPv6) Specification", S. Deering, & R. Hinden (December).

[3] American Registry for Internet Numbers, "*ARIN Board Advises Internet Community on Migration to IPv6*", May 2007, http://lists.arin.net/pipermail/info/2007-May/000111.html

[4] RFC, (2003). 3489, "*STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*", J., Rosenberg, J., Weinberger, C. Huitema, & R. Mahy, The Internet Society (March)

[5] RFC, (1998). 2475, "*An Architecture for Differentiated Services*", S., Blake, D., Black, M., Carlson, E., Davies, Z. & Wang, W. Weiss, (December).

[6] RFC, (2006). 4271, "*A Border Gateway Protocol 4 (BGP-4)*", Y., Rekhter, T. Li, & S. Hares, (January)

[7] Varvitsiotis, A., Siris, V., Primpas, D., Fotiadis, G., Liakopoulos, A. & Bouras, C. (2005). "Techniques for DiffServ - based QoS in Hierarchically Federated MAN Networks - the GRNET Case", *The 14th IEEE Workshop on Local and Metropolitan Area Networks* (LANMAN 2005), Chania. Island of Crete, Greece, 18-21 September.

[8] Internet-Draft, (2005). "*Traversal Using Relay NAT (TURN)*", Rosenberg, J., Mahy, R. & Huitema, C. September, (work in progress)

[9] Bouras, C., Josset, S., Gkamas, A. & Stamos, K. (2003). "Adding IPv6 support to H323 Gnomemeeting/openH323 port" *11th International Conference on Software,* Telecommunications and Computer Networks (SoftCOM 2003), Croatia, Italy, October 7 - 10 2003, 458-462.

[10] http://rts.grnet.gr/vc-qos.php

[11] Skype software, www.skype.com.

[12] ICQ, http://www.icq.com/

[13] Google Talk, http://www.google.com/talk/
[14] Desclaux, F. & Kortchinsky, K. (2006). "*Vanilla Skype*" (parts 1 and 2), June.
[15] Baset, S. A. & Schulzrinne, H. G. (2006). "*An analysis of the skype peer-to-peer internet telephony protocol*," in INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, 1-11.
[16] Sven Ehlert and Sandrine Petgang, "*Analysis and signature of skype voip session traffic*", Fraunhofer FOKUS Technical report NGNI-SKYPE-06b, 2006.
[17] Markus Isomäki,. "*Peer-to-Peer Communication Services in the Internet*", Nokia Research Center, 2003.
[18] "*Adding IPv6 capability to Windows Socket Applications*", Microsoft Corporation.
[19] "*Porting Networking Applications to the IPv6 APIs*", Sun Microsystems.
[20] 6NET project, http://www.sixnet.org.
[21] 6INIT project, http://www.6init.org/presentations.html.
[22] KAME project, http://www.kame.net/
[23] Euro6IX project, http://www.euro6ix.net