

VIDEOCONFERENCE OVER IPV6

IPv6 Networks Advanced Developments

Carlos Friaças, José Baptista, Mónica Domingues, Paulo Ferreira
Fundação para a Computação Científica Nacional (FCCN), Avenida do Brasil n.º 101, Lisbon, Portugal

Keywords: ConferenceXP, GnomeMeeting, H.323, IPv6, OpenH323, OpenMCU, Videoconference.

Abstract: This document focuses IPv6 support on H.323 videoconference protocol and on ConferenceXP architecture. The last one was designed by Microsoft to develop collaborative tasks and videoconference applications. In this document videoconference solutions like GnomeMeeting and ConferenceXP Client (and adjacent services) that implement the analyzed protocols are also presented. This way, guidelines for the deployment of a videoconference service with IPv6 support are provided.

1 INTRODUCTION

Videoconference has been allowing people to overcome physical and political barriers, contributing for the technological, social and personal development of companies and private individuals all over the world.

The new technologies and mainly the data network concept have opened the horizon to one technology that aims to become dominant on the telecommunications world.

With videoconference's growth importance and the evolutionary usage of the IPv6 protocol, it is important to investigate and to analyze solutions with capacity to operate and to provide videoconference services in IPv6 environments.

IPv6 videoconference doesn't bring any crucial advantage to IPv4's. It only guarantees that could be possible to provide a stable and reliable video conference service such as the one currently available in IPv4.

In this document, we tried to study the H.323 videoconference protocol in order to analyze its IPv6 capability. The applications directly associated with H.323, like GnomeMeeting and OpenMCU are also studied.

Without any association to H.323, but directly connected to the videoconference world, IPv6 support on Microsoft's project, named ConferenceXP, is also a target of tests and analysis.

This document is aimed to guide the creation of videoconference services in IPv6 networks, by network managers and administrators.

2 H.323

Defined and standardized by ITU-T, H.323 is a standard that specifies a multimedia content transfer protocol over network data without guaranteed QoS. H.323 is totally independent from the network technologies it operates, allowing it to work on several protocols like IPv4, IPv6, IPX, among others, without the need to change its protocol structure. This protocol also indicates multimedia contents coding and de-coding patterns, guaranteeing interoperability between several manufacturers' applications that implement H.323.

Currently, H.323 is mostly used on videoconference and VoIP applications that allow video and audio transfers. Besides this, H.323 is also used to transfer data, such as files, written messages, among others, simultaneously, during a conference session.

In the multimedia application market, one can find different packages that implement the H.323 protocol. Just to name some: Microsoft Netmeeting, Intel VideoPhone, Netscape Conference, GnomeMeeting (freeware and open source).

2.1 Communication Elements

H.323 specifies the existence of four elements on a communication that can perform simultaneous roles, or not, depending on the utilization type and/or the network structure used. The four elements are:

Terminals – Entities that allow users an interaction with other terminals and/or with other entities specified by H.323 to enable a communication. The terminals need to support audio communication (voice), being video and data optional.

Gatekeepers – When available, they allow the management and control of network access, and perform addresses translation (through H.255.0 protocol). They execute the “role” of central point of communications on their operation zone. This is defined by the set of terminals, Gateways and MCUs (Multipoint Control Units), over the Gatekeepers’ domain.

Gateways – They allow the interconnection between resident terminals in different kinds of networks (eg: PSTN) or between terminals of different patterns (H.310, H.321, H.323, among others), doing the translation of the used types, in order to allow the communication.

MCUs – To allow sessions between three or more terminals, they are needed. Each MCU is composed by a MC (Multipoint Controller) and by one or more MPs (Multipoint Processors). It is MC’s responsibility to manipulate negotiations between all terminals, in order to determine common capacities on audio and video processing. The MP element is responsible for the treatment of contents.

The usage of all these elements allow the complete use of all the advantages and benefits regarding point-to-point and multipoint communications, on different environments, for which H.323 was projected.

2.2 H.323 and IPv6

A crucial point to implement videoconference over IPv4 is the Network Address Translation (NAT) mechanism. Sometimes its use can originate connections problems and compromise videoconference sessions. Efforts were made to solve this issue, but it is proved that a videoconference usage without NAT is better.

IPv6 solves this problem by eliminating NAT usage and providing global IP addresses to everyone and every device.

According to H.323 specifications, this protocol could be used on IPv6 networks without any problem, using TCP and UDP.

With RTP and RTCP protocols, problems shouldn’t also arise, because they are implemented over UDP. This way, it is clear that the specification was conceived in an intelligent way, to avoid future protocol changes on H.323, so that it can work with different kinds of networks.

3 OPENH323

The OpenH323 project aims to implement without any costs and in a public approach, all the characteristics and functionalities specified by the ITU-T’s H.323 videoconference protocol. This implementation aims to be used both by commercial entities and private individuals, without any counterparts.

Built over the MPL (Mozilla Public License), OpenH323 project provides backup and orientation to build simplified private and commercial projects that are directly and indirectly related with videoconference data services.

To use this platform, one needs to install the `pwlib` and `openh323` packages.

In the scope of OpenH323 project, some open source applications (terminals and specified services on H.323) had been created having as development basis the OpenH323 framework. This demonstrates the versatility/complementarily of the H.323 protocol through the OpenH323 platform.

3.1 OpenMCU

OpenMCU is an Open Source implementation of a MCU (specified by H.323 protocol), created on the scope of the OpenH323 project. It has the standard basic functionalities, allowing the establishment of videoconferences between two or more terminals in a H.323 communication.

In a practical perspective, OpenMCU establishes unicast connections with each communication intervenient (H.323 terminals). Through these connections, it receives data individually and later it replies the data to the existing terminals in the same “conference room”.

3.2 GnomeMeeting

GnomeMeeting is an application built over the GNU/GPL license to perform videoconference or

VoIP calls through H.323. This application allows a remote interaction with other applications or hardware devices that support the same protocol. Built and implemented over OpenH323, it integrates the characteristics specified by the H.323 protocol, and also supports a Gatekeeper service, to calls using external MCUs, PC-to-Phones calls using Gateways, among others.

3.2.1 GnomeMeeting and IPv6

The OpenH323 framework is the responsible for the H.323 protocol implementation on GnomeMeeting. This is also responsible for the network protocol management on H.323 communications, which presents support for the last two standardized versions of IP protocol (IPv4 and IPv6).

However, it's important to point out that the communication performed between the GnomeMeeting and the Gatekeeper is not yet implemented for IPv6, which prevents the full utilization of this service, on IPv6, to GnomeMeeting users.

3.3 H.323 Tests

On the first GnomeMeeting usage (versions 1.0.2, 1.1.0, 1.2.1 and 1.2.2), a wizard is presented, allowing for parameters' configuration (for example, audio and/or video capture devices, identification data, among others). Later, accessing *Preferences* in the *Edit* menu, it is possible to change the previous configurations and/or others (audio and video devices, Gatekeeper, Gateway, etc.). Notice that it could be necessary to activate video reception (and sending), when it exists, on the codec's section.

3.3.1 Point-to-Point Unicast Connection

In one unicast connection between GnomeMeeting terminals (or others H.323 implementations) with IPv6 support, it is necessary to insert the other end's IPv6 address (Figure 1). Notice that it's absolutely necessary to use brackets to limit an IPv6 address (for example: [2001:690:900::1]).

Figure 1 represents a simultaneous videoconference and data exchange.

Besides using the IPv6 address, it is possible to use the fully-qualified domain name, or the addresses defined in one Gatekeeper (example: paulo@gatekeeper.pt). On the GnomeMeeting current version (1.2.2) this functionality isn't available for IPv6 yet.



Figure 1: GnomeMeeting Point-to-Point Connection.

3.3.2 Multipoint Unicast Connection

The realization of this test is only possible with a MCU in the network. This must be accessible by all H.323 terminals participants on the IPv6 "multipoint" videoconference.

On a Linux system, after a `pwlib`, `openh323` and `openmcu` (versions 1.9.1, 1.17.2 and 1.1.9) source code installation and compilation, the MCU is started by the following command:

```
./openmcu -n -v -i 2001:690:900::1
```

The flag `-n` guarantees that the MCU won't look for a nearby Gatekeeper (this search is done in IPv4). The flag `-v` allows the use of video on conferences, and the `-i` flag specifies the interface that will receive the connection requests from H.323 terminals.

Beyond the used parameters, it is also possible to change the connections' listening port (`--listenport`) and change the default "conference room" name (`--defaultroom`), among others.

When the `--save` flag is used, at the end of OpenMCU's execution, a configuration file is created on the `.pwlib_config` directory (inside the home directory, with the name `openmcu.ini`). This file avoids using flags again on the next OpenMCU executions.

To establish a connection between a terminal and a MCU, it's enough to place the MCU's address on the terminal and make a call to that address, as if it was one point-to-point call (unicast).

Figure 2 illustrates the establishment of one connection to an MCU, where already exist other

connected terminals. In this case, a connection to the MCU through port 9999 was established.

Other MCU feature is the possibility to create “conference rooms”. For one room creation, it’s necessary to define it on the terminals that will use it. When this procedure is executed, the room is automatically created on the MCU on the moment that the first terminal connects. The “conference rooms” definition on a terminal has the following syntax:

```
h323:FCCN_ROOM[2001:690:900::1]
```

In this case, when the terminal connects to the MCU (using 2001:690:900::1), it enters one room called FCCN_ROOM. It is important to keep in mind that dialogues are only possible between terminals on the same room.

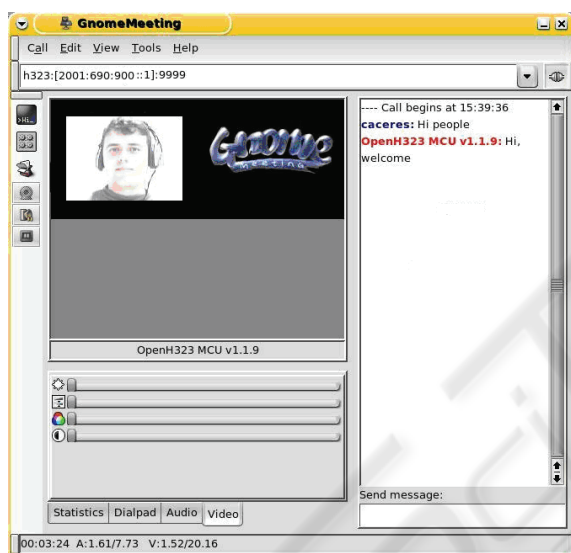


Figure 2: GnomeMeeting and OpenMCU Multipoint Connection.

Moreover, OpenMCU contains a default room (room101) for terminals that don’t specify any (Figure 2). It is possible to remove the default room in order to stop any connection attempt by the terminals that don’t specify any room (--no-defaultroom flag). It is also possible the change of the default room name (--defaultroom).

OpenMCU’s newest versions (newer than version 1.1.9) have flags completely different from the presented version, for example the -i flag has been removed. This way, on OpenMCU’s recent versions, it’s impossible to define the interface we wish to use for communications. This makes impossible the usage of IPv6, because the interface automatically selected will only work by default in IPv4.

4 CONFERENCEXP

ConferenceXP is a Microsoft’s investigation group initiative. This project aims the implementation of a technologic platform that allows the development of new applications that helps enable distance learning and distributed collaboration tools, among others.

Through ConferenceXP, Microsoft wants, to promote the research and development of a new framework to enable distributed applications that can benefit of its advantages, as well as some other new technologies (Tablet PCs, wireless networks).

With the contribution of research organizations and universities, the ConferenceXP project combines the academic community’s knowledge, with Microsoft’s technologic experience. The project focuses on the collaborative implementation of applications that don’t confine the information and knowledge shared to the each place’s physical limits. This is materialized through the high quality and low latency audio and video contents on real time.

4.1 ConferenceXP Client

The ConferenceXP Client is an application of interaction and cooperation with other applications in virtual collaborative spaces called Venues. These Venues are terminal points defined by an IP address (IPv4 or IPv6) and a port.

When one connection is made with ConferenceXP Client to one Venue, this starts automatically the ability of video and audio multipoint send and receive, also allowing the audio and video parameters definition that will be used.

This application was projected to work over high capacity networks (at least at 2 Mbps), allowing the transfer of full-screen videos at 30 frames per second rate and supporting five different transfer rates:

High – 640 x 480 pixel resolution at 1.5Mbps compression;

Medium High – 320 x 240 pixel resolution at 1Mbps compression;

Medium – 320 x 240 pixel resolution at 512Kbps compression;

Low – 320 x 240 pixel resolution at 256Kbps compression;

Very Low – 160 x 120 pixel resolution at 50Kbps compression.

Beyond the video and audio transfer ability, ConferenceXP allows its users to transmit presentations (MS Powerpoint and/or Whiteboard)

concurrently with real time written conversation (chat).

The content transaction is made simultaneous between the different ConferenceXP Clients involved without any need of a server. This way, the transfers occur between the several clients on the network with a possible lower volume of traffic (to take advantage of the maximum available resources). For that, ConferenceXP Client appeals to multicast usage allowing the sending of streams to all involved clients.

It is also possible to use one unicast connection between two ConferenceXP Clients, making a classic point-to-point connection.

4.2 ConferenceXP Services

The ConferenceXP architecture allows the creation of several services to the Network Transport layer with the goal of adding and improving Microsoft's investigation set of services.

4.2.1 Venue Service

A Venue Service allows users to create and manage virtual and collaborative spaces, where they can participate. These spaces (Venues), indicate to the ConferenceXP Clients info about applications that try to connect to it (what ports and IP addresses they should use).

After clients learned the parameters to the connection from Venue Service, they can start multicast connections between them.

4.2.2 Archive Service

The Archive Service allows users to save a specific session on a server using SQL Server, so that it's possible to reproduce it later.

This service doesn't have any kind of IPv6 support yet. Its use (save conferences) is only possible over IPv4.

4.2.3 Reflector Service

The Reflector Service allows the operation of the ConferenceXP Clients over unicast only networks (without multicast support). This server guarantees functionality, making in a transparent way a virtual tunnel between the multicast network and the clients in the unicast network (Figure 3).

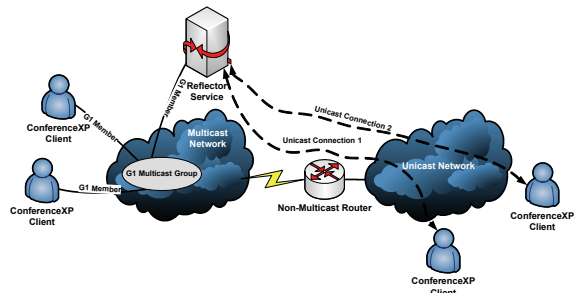


Figure 3: Reflector Service Functioning.

4.3 ConferenceXP Services

In this section, it is our aim to demonstrate the functioning of videoconference sessions using the ConferenceXP Client and some of the available services.

4.3.1 Unicast Point-to-Point Connection

In one unicast point-to-point connection with ConferenceXP Client (version 3.2), two computers with .NET 1.1 framework (with its service pack 1) installed are necessary. These computers need to have IPv6 connectivity between them.

Satisfied the previous criteria, it is necessary to proceed to the clients' configuration. It's possible in this configuration to specify the sending audio and video capture devices, the window layout to present, on the Settings -> Audio/Video menu.

Through the Actions menu, on Start a Two-Way Unicast Conference, the addresses needed to directly connect to another client can be configured (Figure 4).

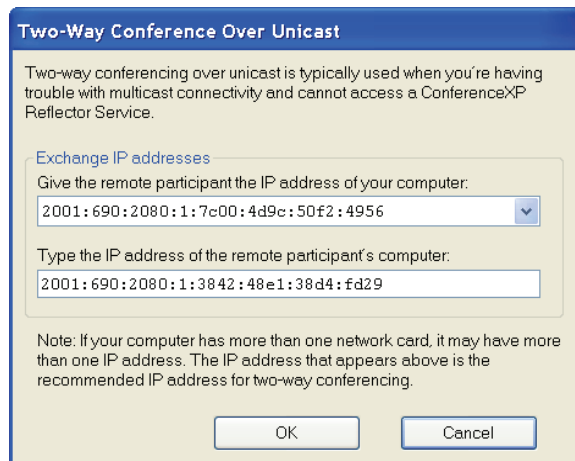


Figure 4: Point-to-Point Unicast Connection Establishment.

The source and destination addresses configuration procedure (Figure 4) is needed on both clients.



Figure 5: ConferenceXP Client Videoconference.

4.3.2 Multicast Multipoint Connections using Venues

The use of Venues implies the acquisition and installation of a Venue Service (version 2.1) on MS Windows XP or MS Windows 2003 Server with IIS available. IIS server plays an important role on the Venues publicity; therefore, it is through this web server that the clients will request/receive the information about the existing Venues (IP addresses and ports) through SOAP (Simple Object Access Protocol).

During the Venue Service installation process, the necessary steps to a correct installation are presented. One of the steps is the definition of a virtual directory to IIS, and the respective access port (Figure 6).

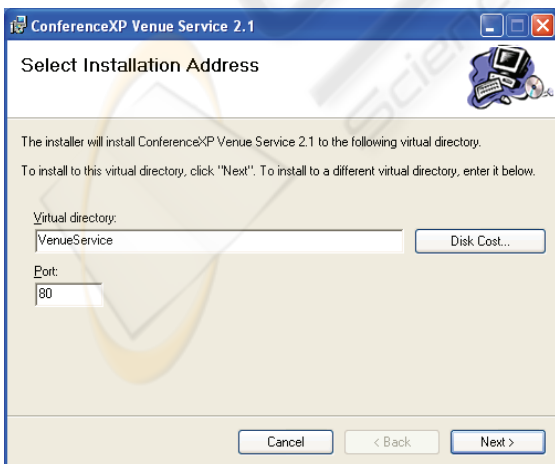


Figure 6: Venue Service Parameters Installation.

On the creation of a new Venue, it is necessary to define its naming, creator identification, multicast address and multicast port, used by the clients on the communications (Figure 7).

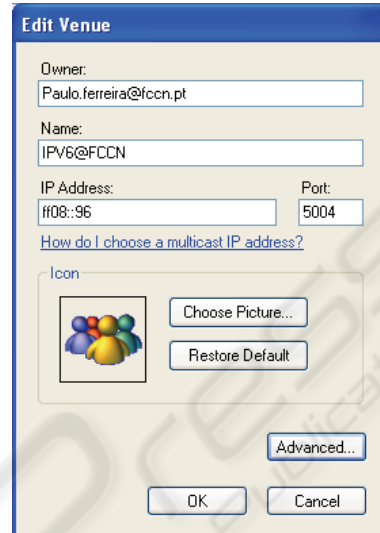


Figure 7: A Venue Creation.

On each Venue Advanced Properties, it is possible to define expressions to avoid its visualization by users that don't have the same expression in their identifications.

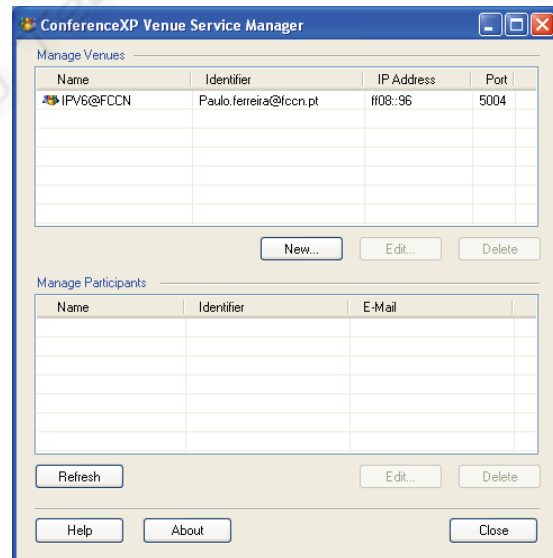


Figure 8: Venues Management and their Clients.

With the Venues creation, they start to be available to the clients connected to the Venue Service. It is also possible to watch and manage the

connected clients (for example, cancel a client connection) to the Venue Service (Figure 8).

For clients connected to the Venue Service, it is necessary to configure them by adding and choosing the computer address where the service and the virtual directory are installed (Settings menu in Services - Figure 9).

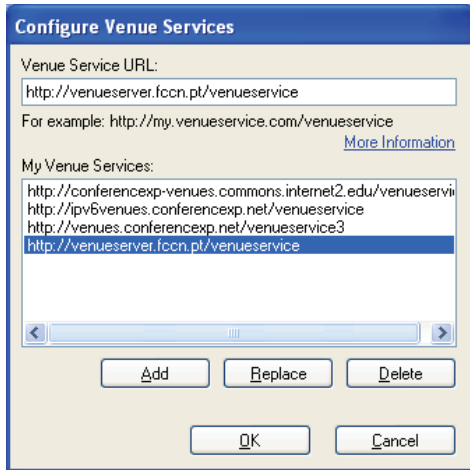


Figure 9: Venue Service Definition on the Client Side.

Once configured the new Venue Service on the clients' side, the venues are presented (Figure 10).



Figure 10: Available Venues on the Venue Service.

In this case, after finishing all the configurations, when one client enters the Venue called IPV6@FCCN, it starts a communication with all the clients who chose the same Venue, through ff08::96 multicast address on port 5004. All

clients in this Venue receive and send contents in multicast, making use of multipoint connectivity.

Besides the IPv6 Venues support by the Venue Service, the connectivity between the clients and the Venue Services is made through IPv4, because this IPv6 feature is still to be implemented by Microsoft.

4.3.3 Connections to Multicast Conferences through Unicast

In this section we hope to demonstrate the solution for a common problem: The lack of multicast support on commercial networks.

A client in a network without multicast support, could participate on a conference made on a network with multicast support, by establishing an unicast connection to a Reflector Service (Figure 3).

It is a smart option to install Microsoft's Reflector Service (version 1.2), on the same system than the Venue Service, to simplify management. It is necessary to proceed with the configuration of this service, so that it can accept IPv6 requests (disabled by default). On the service installation directory, it is necessary to edit the ReflectorService.exe.config file and change the following instruction:

```
<add
key="MSR.LST.Reflector.IPv6Support"
value="false"/>
to:
<add
key="MSR.LST.Reflector.IPv6Support"
value="true"/>
```

After changing the configuration file, and rebooting the service, it is ready to receive IPv4 and IPv6 requests.

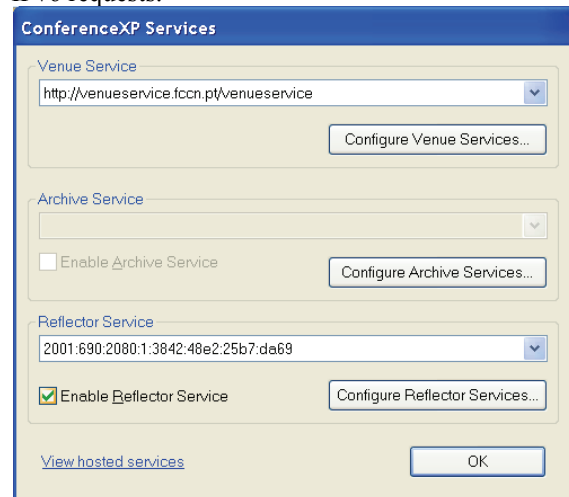


Figure 11: Reflector Service Definition on the Client Side.

For networks without multicast support, ConferenceXP clients can participate on multicast conferences, only by configuring connections to a Reflector Service (Figure 3).

This configuration is possible by adding the Reflector Service address through the *Settings* menu on the *Services* option (Figure 11).

When the client does a Venue selection, a unicast connection to the Reflector Service is established. Meanwhile, the Reflector Service will make a multicast connection to the specified conference according to the Venue chosen (Figure 12).

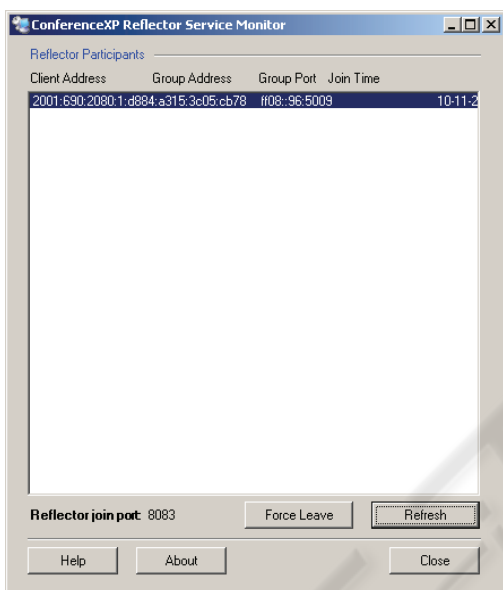


Figure 12: Reflector Service Connections Management.

It is possible for the Reflector Service administrator to remove a client from a conference through the Reflector.

5 CONCLUSIONS

The analysis made to the OpenH323 project demonstrates some poor focus on IPv6 support implementation. As an example, OpenMCU's newest versions reveal flaws on IPv6 support. This fact, forced us to use an older version to perform the presented tests.

Directly connected to OpenH323 project's IPv6 support stagnation, functioning problems on the GnomeMeeting software were verified. The main problem was the impossibility of establishing a connection to Gatekeepers through IPv6.

In summary, despite the identified problems on the OpenH323 project (applications and services) and on GnomeMeeting, these tools are able to the build up of a basic IPv6 videoconference service/environment.

The ConferenceXP project, due to its embryonic state, still presents some instability. On the other hand, there is a concern about perfection, by considering itself like a project with "legs to walk" on the IPv6 world.

The ConferenceXP Reflector Service shows the adaptation capability of this project, allowing videoconferences between clients, independently of multicast availability. This way, this project can be seen as a serious competitor to the existing videoconference software tools.

With IPv6 transition, we are certain that new projects, applications and frameworks with abilities to satisfy the videoconferences service needs in IPv6 environments will rise.

REFERENCES

- OpenH323 Project, 2005. URL available on: <http://www.openh323.org/>.
- Draft revised Recommendation H.323 V5, May 2003. "Packet-based multimedia communications systems"; "H.323: Um padrão para sistemas de comunicação multimídia baseado em pacotes", 2005. URL available on: <http://www.rnp.br/newsgen/0111/h323.html>.
- Rodrigues, N., Costa, C., Vale, S., September 2005. "VoIP@IPB-Piloto de Telefonia IP numa Instituição de Ensino Superior". Actas da CRC 2005, "A Internet e a Adaptabilidade das Redes e Aplicações às Comutações de Infra-estruturas".
- Silva, R., Sargento, S., September 2005. "Segurança do H.323 - a Norma H.235". Actas da CRC 2005, "A Internet e a Adaptabilidade das Redes e Aplicações às Comutações de Infra-estruturas".
- "H.323 Protocols Suite", 2005. URL available on: <http://www.protocols.com/pbook/h323.htm>.
- Srisuresh, P., Egevang, K., January 2001. "Traditional IP Network Address Translator (Traditional NAT)". RFC 3022.
- Thomson, S., Narten, T., December 1998. "IPv6 Stateless Address Autoconfiguration", RFC 2462.
- OpenH323 Gatekeeper - The GNU Gatekeeper, 2005. URL available on: <http://www.gnugk.org/>.
- ConferenceXP Official Page, 2005. "Microsoft Research, ConferenceXP", URL available on: <http://www.conferencexp.net/>.
- GnomeMeeting Official Page, 2005. "GnomeMeeting.org", URL available on: <http://www.gnomemeeting.org/>.