Secure and Pervasive Collaborative Platform for Medical Applications

Petr HOLUB [a,c,1] Eva HLADKÁ [b,c] Michal PROCHÁZKA [a,c] Miloš LIŠKA [b,c]

[a] *Institute of Computer Science, Masaryk University, Czech Republic*
[b] *Faculty of Informatics, Masaryk University, Czech Republic*
[c] *CESNET z.s.p.o., Czech Republic*

Abstract. Providing secure, extensible, pervasive and easy to implement collaborative environment for medical applications poses significant challenge for state-of-the-art computer systems and networks. In this paper, we describe such a collaborative environment developed for Ithanet project, based on Grid authentication mechanisms. Significant effort has been put into developing a system, that is capable of deployment across tightly secured networking environments as implemented in vast majority of hospitals. The environment is extensible enough to incorporate Grid-service based collaborative systems like AccessGrid.

Keywords. Secure reliable pervasive collaborative environment, collaborative environment structure, virtual private network, latency performance, Ithanet project

Introduction

Virtual communication environments in the medicine are slowly becoming more popular with advent of reliable high-speed networking. Current trends go in two basic directions: (1) conservative commercially available technologies like voice over IP teleconferencing and H.323/SIP infrastructures, and (2) more advanced and more experimental tools like AccessGrid. While the former group is sufficient for basic communication, we focus in this paper on more advanced extensible environments, that is suitable for advanced collaboration.

The medical network are being protected heavily namely due to requirements on security of data about patients. However, such an "adverse" networking environment has usually sever consequences on collaborative technologies. Firewalls and network address translators (NATs), together with very conservative behavior of network administrators are the major obstacles for deployment of the majority of the conferencing systems, with notable exception of Skype [1], as discussed in Section 1. For collaborative environment, synchronous transmissions audio and video signals is essential. In terms of network transfers, this is usually implemented using RTP/UDP packet distribution. But UDP packet distribution in networks with NATs and firewalls with common settings is not allowed. One way to get around this problem is to change the settings; for this, users

---

have not only to ask for an approval of their administrators, but also they need them to make the configuration changes in security-sensitive devices, which is usually very hard to achieve. The second way is to use software for penetrating NATs and firewalls on application layer allowing communication without setting changes. This problem is not limited to collaborative environments, as demonstrated by a recent study by Open Grid Forum [2].

In this paper, we focus on building a secure extensible collaborative platform suitable for medical purposes, that is capable of working in such "adverse" networking environments inside hospitals with minimum requirements on support from network and system administrators. The platform uses Grid-compatible authentication based on PKI infrastructure and is ready to incorporate other Grid extensions based on Grid service oriented architecture.

The described system has been designed in Ithanet[2] project as stated in Acknowledgments at the end of this paper. The design reflects our experiences from participation in Ithanet and also EuroCareCF[3] projects, that showed how difficult it is to deploy reliable, secure, and affordable collaborative tools in multi-institutional environment across many hospitals all over the Europe. Both according to surveys we have conducted in these projects and according to our experiences with other medical communities, there is a number of problem to solve: the connectivity among the partners varies to large extent; already existing computer systems are largely based on Microsoft Windows system (often under central administration), making installation of additional software and especially remote support very difficult or virtually impossible; computer support teams at the hospitals often do not have enough experiences to work with more complex collaborative environments. Therefore we have decided to propose and implement a system, that is able to avoid or at least to mitigate these problems.

The rest of the paper is structured as follows: Section 1 summarizes relevant related work, Section 2 gives an overview of the architecture of the whole collaborative environment, Section 4 details proposed client devices and software tools. In Section 5, we give performance measurements of the VPN-based networking solution implemented for the environment and also further experiences with the system. Section 6 concludes the paper with final remarks and directions for future development of the collaborative platform.

1. Related Work

Traditionally, the tele-medicine was implemented using phone-based solutions, later migrating to H.323/SIP solutions. However, both H.323 and SIP are hard to implement in "adverse" networking environments and require serious support for network administrators side.

Skype [1] has thus become very attractive alternative for many users, because of its capabilities to penetrate firewalls and work behind NATs—unless explicitly forbidden by the network administrators, and also rather robust commercial-grade implementation. Providing limited multi-point conferencing support, it is also usable for small teams to communicate. When user signs up for commercial Skype services, it is possible to call also to public telephony network. In recent versions, Skype provides basic video support,

---

[2]http://www.ithanet.eu/
[3]http://www.eurocarecf.eu/

which is however much less mature than its audio counterpart. However, there are several reasons why this tool doesn't meet the needs of medical communication particularly well and becomes often explicitly forbidden by the hospital management: first, the security model is proprietary and very obscure [3,4] and may change without any prior notification, and second, users participating in the Skype network are automatically providing their computers to be used by other Skype users, thus actually supporting the business of Skype company with their own resources. Further, it is hard to block Skype on the networking layer, because of its firewall/NAT penetration techniques (e.g., even the login messages to Skype servers may be router through the super-nodes in the P2P overlay network as shown in [4]). Being a proprietary solution, Skype is hard to be integrated in more complex Grid-like infrastructures, too.

On the other side of the collaboration tools spectra, there is an experimental open extensible system called AccessGrid [5,6]. Since its version 2, it features service oriented architecture for the collaborative environment based on Grid services [7]. For deployment in the hospital environments, it has two major drawbacks though. First, the data distribution is not suitable for "adverse" networking environments—authors of Access-Grid assumed multicast to be the primary data distribution technology; since the multicast deployment is a problem even in very open academic networks, they added simple unicast bridging technology based on UDP packet reflectors later on. Second, the whole system is rather complicated and hard to deploy without knowledgeable on-site support staff. Despite these issues, AccessGrid features many interesting properties making in worth considering when planning deployment of larger collaborative systems, by at least preparing compatible equipment for its future installation.

Another well known example of collaboration environment, created for high energy physics community that is now becoming more widely used, is the Virtual Room Video-conferencing System (VRVS)[4]. VRVS is based on multicast-like schema and is provided as a service and user data traffic is managed by VRVS administrators. The successor of VRVS called Enabling Virtual Organizations (EVO)[5]—is based on self-organization of system of reflectors, again not empowering the end-user with tools to change the distribution topology. However, for medical problem it has the similar problem as AccessGrid and even worse it is a closed system with the similar consequences like for Skype.

2. Architecture of the Collaborative Stack

When designing the collaborative platform architecture, we have had the following main design principles in mind:

1. The system has to be secure to protect the communicated data. It should utilize Grid-compatible authentication, so that the users can use their existing Grid identity to join the collaborative environment.
2. Support for firewall and NAT penetration to facilitate deployment in "adverse" networking environments in hospitals. The support has to be flexible and auditable, and the network administrators must not be "cheated", (but rather just avoided unless really necessary).

---

[4]http://www.vrvs.org/
[5]http://evo.caltech.edu/

3. The system has be extensible towards service-oriented architecture compatible with Grids.

The proposed collaborative environment comprises the following layers from the bottom-up perspective: network connectivity layer, data distribution layer, central services layer, client tools and devices layer. Each of the layers is discussed in more detail below.

Network connectivity layer. The network connectivity layer takes care of interconnecting all the elements of the collaborative environment into a continuous network, so that all the element may reach one another or at least some central site or server site when all-to-all connectivity is not desirable for any reason. If the elements can't reach one another over the native network, this may be implemented as an overlay network, be it simple tunnels or more sophisticated VPNs. Assuming Internet protocol stack, the tunnels may run over UDP, TCP, HTTP (including emulation of HTTP and HTML encapsulation), TCP with HTTP proxy, and TCP with SOCKS proxy. For vast majority of the firewall and NAT protected networks, at least one of the mentioned solution works. VPNs are also useful when overlay network privacy is desired based on overlay link encryption.

Data distribution layer. This layer takes care of multi-point data distribution to the connected clients. Assuming the Internet protocol stack, this is usually implemented either using multicast, which is more efficient but much harder to deploy properly esp. in network spanning multiple administrative domains, or using UDP packet reflectors. While less efficient, the UDP packet reflectors are much less error-prone and provide also possibility of data processing even on per-user basis (something theoretically impossible in multicast).

UDP packet reflectors may also be modified to provide the network connectivity layer to the clients and between the reflectors directly, thus merging Network connectivity and Data distribution layers.

Central services layer. This layer comprises services provided to the client on some "server" basis—though the servers may be largely distributed and not limited to one physically central location. The services may include monitoring, virtual rooms or venues (for creating separate virtual spaces for communication of different user groups), wikis, persistent data storage, etc.

Client tools layer. The client layer comprises of tools and hardware devices on client side. The software tools primarily incorporate audio, video, and a chat service, and may include other tools like shared presentation, shared desktop or application window, or shared text editor.

While software tools are traditional when looking on computer based collaborative environments, we have included also the hardware part, as the quality of hardware and a level of its software support is critical for the successful experience with collaborative environments. Often this creates a point of failure for the collaborative environments deployment. The collaborative tools are much harder to deploy compared, e.g., to SETI@Home or similar computation-heavy tasks that are relatively easily distributed because they are only dependent on CPU and very basic OS services.

3. Preliminary Implementation of the Collaborative Stack

3.1. Network Connectivity Layer

Secure communication together with ability of firewall and NAT penetration is achieved using the OpenVPN software. It makes the VPN on the application layer from the ISO/OSI perspective and supports the whole range of methods as discussed in the previous section. It means that there is no need to modify configuration of network elements on the path from the client to the VPN server. Also whole communcation between client can be encrypted.

All client workstations are connected to a VPN server in a point-to-point mode. The set up of the VPN network guarantees that only the traffic belonging to the collaboration services is sent into the VPN tunnel. Originally, we wanted to use one of the private IP address ranges as defined by RFC 1918 for internal VPN addressing. After the partner networking survey we have found it very complicated to avoid conflicts with internal address ranges used at various institutions, especially as new institutions may join. The whole overlay network is therefore addressed using a public IP address range assigned by RIPE, but the addresses are treated as internal address and not distributed outside of the VPN overlay network. As there is no direct traffic between any two partners and thus all the traffic may be filtered on the VPN server.

The OpenVPN server runs in two modes—either over UDP or TCP. The UDP mode is preferred due to better performance, as the VPN is not limited by the TCP congestion control algorithm [8]. The TCP mode can also run over HTTP or SOCKS proxy.

The client side needs OpenVPN software to be able to connect to the OpenVPN server. This software makes the virtual network adapter and sets appropriate routing table records for the client. Clients are authenticated using their personal X.509 certificates—the Grid users may user their existing ones, while others are given new certificates from a dedicated certificate authority. Client software is able to work with certificates stored in the file or on the secure smart card. Second option is strongly preferred because the client certificate can be delivered and kept by the client in a secure way.

3.2. Data distribution layer

The data distribution layer is implemented using modular user-empowered UDP packet reflector [9], which is known to work very well with the taget client media tools—MBone Tools[6]. It is highly configurable with modules loadable in run-time, supporting sophisticated access control policing and even data transcoding for some data formats. Media streams may be encrypted by the client software tools using symmetric encryption in case that the data replication site is not considered trusted enough. Communication based on UDP packet reflector is communication with central replication unit and number of communicated client is limited by capacity of this unit. Solution of this problem is to decentralize reflector by network of reflectors [10].

---

[6] http://www-mice.cs.ucl.ac.uk/multimedia/software/

3.3. Central services layer

Currently, there are UDP packet reflector administration and monitoring run as a central services. When the system is extended to full AccessGrid support, Venue Server could be an example of central service. Another centrally run service is the OpenVPN server supporting users as described in section 3.1. To support the user communication we furthermore provide an IRC daemon (currently IRCD-hybrid[7] run as a central services. A special IRC client daemon which run on the same machine was developed to store and provide the chat history. A Jabber instant messaging server may be provided in the same way as IRC.

4. Client Platform

The collaborative platform client is developed as a mixed HW/SW solution. The client is based on well defined and tested HW with preinstalled operating system and set of collaborative and especially videoconferencing tools. The choice of operating system was done with emphasis on simple modifications of the system, remote management, wide hardware support, security and last but not least user friendly environment. As a result we chose Ubuntu Linux to be the base for the client SW.

The collaborative tools depend on many dedicated hardware devices like sound cards, sound acquisition devices, video capture cards, USB cameras, etc., whose quality and level of support may vary to a large extent. Therefore, we have opted for suggesting a set of hardware: widely available low-cost HP Compaq dc7700 Ultra-Slim Desktop PC with known-to-work and well tested sound and graphics card and video capture cards, together with some other devices like headsets and USB cameras. This provides well defined starting point for efficient distribution of operating system and collaborative tools.

Security and personal configuration of the platform is based on combined USB security and storage token. The token is used for following purposes:

- identity storage which is based on PKI for authentication purposes, especially to connect to VPN server
- customized configuration storage for each users configuration, allowing user to store his contact informations for videoconferencing purposes and start the videoconference in specific mode (e.g., specific reflector address and port or VPN configuration)

Basic videoconferencing capabilities are provided by MBone Tools. Robust Audio Tool (RAT) is used for audio transmission and playback. RAT supports variety of audio codecs and allows to fine tune the audio stream according to quality or bandwidth limitations where necessary. Video communication is provided by Videoconferencing Tool (VIC) providing transmissions of video acquired from video capture card or USB cameras.

Besides VIC and RAT other we provide audio and videoconferencing tools like Ekiga, WengoPhone and even Skype which allow audio and video communication with number of other videoconferencing platforms.

---

[7]http://ircd-hybrid.com/

While the individual tools are rather user-friendly when started, the initialization of the conference itself is not very intuitive step. In order to facilitate this, we are developing an integrating Graphical User Interface (GUI) (see fig. 1 for the platform, that supports easy setup of the conference. The default settings may be stored for future reuse, so in the production state, the user just pushes a single button to start the whole conference. The GUI also monitors all the applications, so that if any part of the system crashes, the user is immediately informed and it also provides a very detailed information for the remote user support.
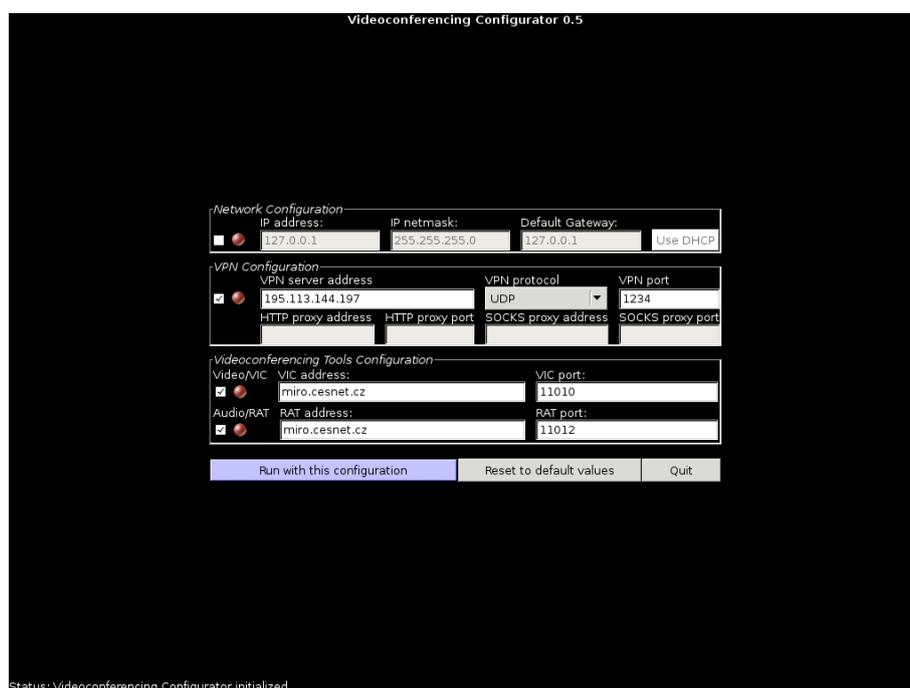

Figure 1. GUI for the collaborative platform client.

To support the collaboration beyond scope of just audio or videoconferencing we provide bidirectional sharing of whole desktop or particular applications between video-conferencing platform clients. Desktop and application sharing is based on VNC proto-col [11] and related tools, namely shared-app-vnc[8] and x11vnc[9]. A secondary intent is to provide user with remote control of his videoconferencing machine in case the machine has no display and keyboard or the user wants to control the videoconference from his/her laptop computer. However, using VNC as a software display is considered as an emergency solution only because of high demands on the videoconferencing machine and the latency introduced video displaying.

The client platform is based on SW that is stable yet under development and some new and desired functionality may appear. In a worse case a security hole may be dis-

---

[8] http://shared-app-vnc.sourceforge.net/
[9] http://www.karlrunge.com/x11vnc/

covered in one or more platform SW components. Thus it is necessary to update the SW base of the platform regularly. The individual videoconferencing tools as well as other platform SW may be updated automatically based on the operating system SW repositories.

Underlying operating system updates are more complicated and failed update may turn whole machine unusable. There may be no system administrator available or skilled enough to perform the operating system update on end users site. That is why we opt for prepare the operating system update as a black box solution. The update is based on bootable CD/DVD with an image of well tested and working updated operating system and platform SW. It is not necessary to distribute the CDs among the end users. More comfortable approach is to create and ISO image of the update CD and make it available for download. The update procedure is performed automatically right after the CD is inserted into the machine. End users don't need to reconfigure their box after each operating system update because all custom configuration is stored on the USB token and thus is not affected by the update.

5. Experiences with Preliminary Implementation

In order to evaluate influence of incorporation of OpenVPN into the collaborative platform, we have measured a number of parameters critical for real-time multimedia communication using different VPN modes. The measurement testbed comprised one client and one VPN server, interconnected with high-speed backbone network link with capacity above 1 Gbps spanning about 250 km. The results of measurements are summarized in Table 1. We can conclude that UDP based VPN is very safe and has minimum impact on the traffic. Slight CPU requirements increase for the UDP based VPN compared to TCP based VPN is due to application-level packet loss recovery and congestion control, which is marginally less CPU efficient compared to kernel-based TCP implementation. TCP-based VPNs also perform very well provided they are on low-latency network with very low packet loss, so that congestion control algorithm doesn't influence the data flow significantly. If the HTTP proxy is of good performance, it has minimum impact on performance, too.

Table 1. Measured comparison between direct communication and communication through various VPN modes as implemented by OpenVPN.

	no VPN	UDP VPN	TCP VPN	TCP VPN + HTTP proxy
pchar latency [ms]	3.51	3.69	3.94	3.93
iperf jitter [μs]	6	6	9	13
pchar capacity est. [Mb/s]	39.8	35.2	20.1	19.8
iperf packet loss @ 30 Mb/s [%]	0.0	0.0	0.0	0.0
iperf CPU idle @ 30 Mb/s [%]	48.9±0.2	41.7±0.4	44.5±0.4	42.6±0.4

When evaluating the performance of this solution subjectively, the media streams are fine and the overall quality is very good. The only problem we were facing is that the users are sometimes very reluctant to buy a new hardware for the client platform, even though it is very cost effective compared to dedicated videoconferencing solutions. However, they are also unhappy about performance of videoconferencing tools self-installed

on their existing desktop computers, as these were not performing well without substantial tweaking because of rather complex interactions with undefined or poor hardware components and existing software. Thus the deployment requires significant work in order to explain principles of the system to the users as described above.

6. Conclusions and Future Work

In this paper, a secure and pervasive collaborative platform for medical applications has been introduced to provide flexible multi-point Grid-compliant collaborative environment. The design and implementation was targeted to create remotely supported system, that is scalable, robust and flexible allowing to collaborate to tens of people. The system has been implemented in year 2006 to the state described in this paper. In 2007, we plan to gather experiences based on wider scale deployment and modify the system accordingly.

The first version of the system, described in this paper, has risen a number of new problems and ideas. The first ideas for future work are concerned with reflector functionality. In the future we plan to utilize per-user processing on the reflector for solving the problem when a single client with a very limited network connectivity limits the quality of the collaboration for the whole group. Currently we are using OpenVPN to traverse NATs and firewalls but in the future, we plan to implement this directly in the reflectors. Such an approach allows for more aggressive failure detection and faster problem recovery. Also as the reflectors may be deployed as a network, it naturally avoids the single point of failure currently imposed by a central VPN server.

Grids and Grid-based systems are now widely developed and used in many areas. We plan to utilize the Grid-services based approach and to incorporate AccessGrid services. On the other side, AccessGrid needs to be modified to work with our advanced reflectors, firewall and NAT penetration techniques and reflector networks for better scalability and robustness. This will be discussed with AccessGrid developers and we will offer our tools to that project. We are at the beginning of practical usage of the proposed platform and its routine operation will definitely bring other new ideas and requirements for the future.

References

[1] Zennström, N., Friis, J.: Skype (2003-2007) `http://www.skype.com/`.

[2] Niederberger, R., Allcock, W., Gommans, L., Grünter, E., Metsch, T., Monga, I., Volpato, G.L., Grimm, C.: Firewall issues overview. Technical Report GFD-I.083, Open Grid Forum (2006)

[3] Biondi, P., Desclaux, F.: Silver needle in the skype. In: BlackHat Europe. (2006) `http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-biondi/bh-eu-06-biondi-up.pdf`.

[4] Baset, S.A., Schulzrinne, H.: An analysis of the skype peer-to-peer internet telephony protocol. In: INFOCOM 2006, Barcelona, Spain (2006) `http://www1.cs.columbia.edu/~salman/publications/skype1_4.pdf`.

[5] Childers, L., Disz, T., Hereld, M., Hudson, R., Judson, I., Olson, R., Papka, M.E., Paris, J., Stevens, R.: ActiveSpaces on the Grid: The construction of advanced visualization and interaction environments. In Engquist, B., ed.: Simulation and visualization on the grid: Parallelldatorcentrum, Kungl. Tekniska Högskolan, seventh annual conference, Stockholm, Sweden, December 1999: proceedings. Volume 13 of Lecture Notes in Computational Science and Engineering., New York, NY, USA, Springer-Verlag Inc. (2000) 64–80

[6] Childers, L., Disz, T., Olson, R., Papka, M.E., Stevens, R., Udeshi, T.: Access grid: Immersive group-to-group collaborative visualization. In: Proceedings of Immersive Projection Technology, Ames, Iowa (2000)

[7] Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The physiology of the grid: An open grid services architecture for distributed systems integration (2002)

[8] Jacobson, V.: Congestion avoidance and control. In: ACM SIGCOMM '88, Stanford, CA (1988) 314–329

[9] Hladká, E., Holub, P., Denemark, J.: An active network architecture: Distributed computer or transport medium. In: 3rd International Conference on Networking (ICN'04), Gosier, Guadeloupe (2004) 338–343

[10] Holub, P., Hladká, E., Matyska, L.: Scalability and robustness of virtual multicast for synchronous multimedia distribution. In: Networking - ICN 2005: 4th International Conference on Networking, Reunion Island, France, April 17-21, 2005, Proceedings, Part II. Volume 3421/2005 of Lecture Notes in Computer Science., La Réunion, France, Springer-Verlag Heidelberg (2005) 876–883

[11] Richardson, T., Stafford-Fraser, Q., Wood, K.R., Hopper, A.: Virtual network computing. IEEE Internet Computing **2** (1998) 33–38