Annales UMCS Informatica AI XI, 4 (2011) 33–47
DOI: 10.2478/v10065-011-0035-2

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Agent based infrastructure for real-time applications

Grzegorz Oryńczak[1*], Zbigniew Kotulski[2,3†]

[1] *Jagellonian University, Department of Physics, Astronomy and Applied Computer Science, Cracow, Poland*
[2] *Institute of Telecommunications, Warsaw University of Technology Warsaw, Poland*
[3] *Institute of Fundamental Technological Research of the Polish Academy of Sciences, Warsaw, Poland*

Abstract – In this paper we propose a new infrastructure for real-time applications. As a preliminary, we describe basic characteristics of the most popular real-time services like VoIP, videoconferencing, live media streaming, and network multiplayer games. We focus on the end-to-end latency, bandwidth and efficient transmission methods. Next, we present our project concepts, infrastructure model, details of implementation and our testing environment which was designed for testing many aspects of real-time services. The system combines mechanisms for ensuring best possible connection quality (QoS), load balance of servers in infrastructure and gives control over the packet routing decisions. Additionally, provided security mechanisms make it a good choice even in the environment where a high security level is required. The system is based on the Peer-to-Peer (P2P) model and data between users is routed over an overlay network, consisting of all participating peers as network nodes. This overlay can by used for application level multicast or live media stream. In the logging process each user is assigned to a specific node (based on his geographic location and nodes load). Because nodes are participating in data transmission, we have control over the data flow route. It is possible to specify the desired route, so, regardless of the external routing protocol, we can avoid paths that are susceptible to eavesdropping. Another feature of the presented system is usage of agents. Each agent acts within the single node. Its main task is to constantly control the quality of transmission. It analyzes such parameters like link bandwidth use, number of lost packets, time interval between each packet etc. The information collected by the agents from all nodes allows to build a dynamic routing table. Every node uses the Dijkstra's algorithm to find the best at the moment route to all other nodes. The routes are constantly modified as a consequence of changes found by agents or updates

*grzegorz.orynczak@uj.edu.pl
†zkotulsk@tele.pw.edu.pl

sent by other nodes. In VoD services agents also analyze popularity of streamed media, which helps build intelligent video cache. To ensure greater security and high reliability of the system, we have provided a reputation mechanism. It is used during bringing up to date the information about possible routes and their quality, given by other nodes. Owing to this solution nodes and routes which are more reliable get higher priority.

1 Introduction

Real time transmission is very important in modern communication. Owing to rapid growth of the Internet and its services, it is now possible to build systems that not only, until recently, have been reserved for telecommunications and television companies, but also give us entirely new possibilities. Internet telephony is a good example. One of the first attempts of creating a protocol for transferring human speech over computer network was the Network Voice Protocol [**1**] (NVP) made by Danny Cohen of the Information Sciences Institute from University of Southern California in 1973. NVP was used to send speech between distributed sites on the ARPANET. Since that time telephony based on Internet Protocols has become more and more popular. Further increase of speed and quality of the Internet connections made it possible to extend the scope of services also for video commutation e.g. Skype or AIM, conference systems [**2**] and even telepresence robots [**3**]. Nowadays, communication over the Internet has become a serious competitor to standard telephony, owing to the possibility of making cheap calls and a wide range of additional features it will be one of the main forms of communication in the future. However, protocol for real time data transmission over the Internet is not limited only to the use in voice and video communication. Popularity of high speed telecommunication networks made it possible to build other media content streaming applications like e.g.:

- Internet Television systems, which require the protocol for managing live transmission;
- network multiplayer games, where often real time communication between players is critical for good gaming experience;
- Video on Demand (VoD) Systems, where delay is not so important, but we still need to efficiently manage large scale multimedia data transmission.

Currently, applications that can be used for building each of the services listed here, have been developed. However, there are still some challenges that we have to face. For example a lack of mechanisms for ensuring proper quality of service (QoS) is still a significant problem. Also mechanisms for providing good security of transmission and efficient bandwidth management schemes need to be refined.

Many factors need to be considered when designing a system for real-time data transfer. One of them is the choice of transport layer protocol. As opposite to the circuit-switched networks such as public switched telephone network (PSTN), where packets arrive in order and have fixed delay, communication over the Internet is based on packet-switched techniques. In this kind of networks packets can arrive without

order, with different delay, or may be completely lost: these factors should be taken into account when developing a new system. For standard internet communication we use TCP or UDP transport protocols. Unfortunately, in their unaltered form, they can not be used for real-time applications, because they are not designed for this specific use, so they do not give us control over delay and jitter. TCP is a reliable, connection oriented protocol, but it is more complex and slower than UDP, and built-in retransmission mechanism is often useless for real-time transmission: it works relatively slow, so retransmitted packets can be outdated. For transferring multimedia data like voice or video, reliability for transmission is not as important as timely delivery; delays can be much more annoying than the loss of a few milliseconds of audio transmission or slight frame rate drops. For this reason, UDP is a preferable choice to base on while building custom real-time protocols. Not without significance is also the packet header length, UDP header size is 8 bytes while TCP header has 20 bytes. Although UDP has its benefits when it comes to speed, protocols based on it have to deal with a lack of some important mechanisms. First of them is a congestion control mechanism which is not present in UDP and if the sender exceeds a transmission rate that can be handled by the network, it leads to congestion problems and network overload. Protocol should also implement mechanisms for time-stamping packets to allow synchronization and minimize jitter problems. Most of the currently used protocols for transporting real-time data supports those features but QoS is still not guaranteed.

In our research we address the problems related to providing secure and efficiently managed real-time data streams. Model of our new application, that can be used for building modern real-time data sending systems, is introduced in this paper. As opposed to the standard client/server architecture used for example in SIP [**4**] or H.323 [**5**], we chose to base our system on the Peer-to-Peer (P2P) model. During last years P2P systems have become popular not only in domains like file sharing but also proven to be successful for voice and video communication (e.g. Skype). There are many benefits of using this network model; they are described in the next section. Another choice that we made was using agents for analyzing infrastructure. In many tasks agent based solutions appeared to be more efficient [**6**], this paper shows that they are also useful in real-time applications. Our goal was to design a secure system, which will ensure the best possible connection quality (QoS), which we achieved by path switching technique based on our own routing protocols assisted with reputation mechanisms. Security mechanisms that we provided and bigger control over the packet routing decisions (owing to P2P model) make this system a good choice even in the environment where a high security level is required. To make the system more efficient and reliable, mechanisms for node load balance were also provided. The paper is organized as follows. In Section 2 the most popular real-time services and their requirements are presented. In Section 3 the system architecture is described with the node model and the general communication flow. Section 4 is devoted to the security and QoS mechanisms. Finally, Section 5 concludes our work.

2 Most popular real-time services overview

Each application, regardless of their purposes, that has to transmit data in real-time, has specific requirements from the lower layers that must be fulfilled. The most important of them are those related to delay, jitter and packet loss. Depending on the nature of service it can also require different bandwidth management and data transmission method. This section overviews the most popular of these services.

2.1 Internet telephony

In telephony the callers usually notice roundtrip voice delays of 250 ms or more., sensitive people are able to detect about 200 ms latencies. If that threshold is passed the communication starts to be annoying. ITU-T G.114 [**7**] recommends maximum of 150 ms one-way latency. As it includes the entire voice path, the network transmit latency should be significantly smaller than 150 ms. Those delays also escalate the negative effects of echo. In normal PSTN telephony system echo is unnoticed because of small transmission delay, but in the case of VoIP, echo effect can be much more disturbing. One of the solutions to this problem is to implement the echo cancellation algorithm [**8**]. Another difference between standard and internet telephony is a user location service. There is a direct relation in fixed line phones between a telephone number and its physical location. In VoIP, the user can login into the system from any place where the internet connection is available, and use his telephone service. Moreover, it is also possible to register at several locations e.g. home, office, etc.; in the case of incoming call, all registered phones will be ringing at the same time or in a certain priority. Despite the advantages of this approach, it has drawbacks when it comes to emergency calls. It is difficult to locate VoIP users geographically, so emergency calls cannot be easily routed to a nearby call center. Internet telephony users are identified by their logins. For example, in SIP, Uniform Resource Identifier (URI) of the form sip:usename@host:port is used. This form of identification requires to implement in the VoIP application mechanism for user authentication, registration and location, it is also very important to provide appropriate security mechanisms for those operations. Currently ythe most popular VoIP applications are based on SIP or H.323, where H.323 is a recommendation from ITU Telecommunication Standardization Sector (ITU-T), that describes the use of several protocols for audio/video communication. SIP is an Internet Engineering Task Force (IETF) defined signaling protocol. Most of internet telephony applications for transporting real-time data use RTP in conjunction with the RTCP protocol.

2.2 Videoconferencing

The first commercial system for audio and video communication was deployed by AT&T in the early 1970s and it used trademarked Picturephone products. However, popularity of the systems capable of full duplex video and audio communication in real time increased significantly in the 2000s with the advent of free internet video

telephony services such as Skype or iChat. Currently, video communication systems able to provide high definition real-time stream (30 fps, 1280 by 720 pixels resolution) are becoming a standard. Traditionally videoconference systems differ from video telephony by their purpose, they were designed to serve the group rather than individual users. However, with technology and software improvements this distinction is becoming increasingly blurred. Despite the increasing popularity and continuous development of new features, there is still a lot of work to be done to create audio-visual communication systems that will be reliable, have good performance and scalability as well as offering a high level of security.

Link latency requirement for this kind of services is similar to those described previously for internet telephony, so one-way latency should be lower than 150 ms. Furthermore, large bandwidth is another major technical requirement for satisfactory performance. The difference can also occur in user registration and location services. As videoconference systems often require large, expensive devices that are used in large rooms or auditoriums, those systems are non-portable and can be permanently associated with a specific host or domain. In this case, additional registration and localization services are unnecessary. On the other hand, we need conference control services for allocating resources, performing data routing, adding and removing participants from a conference. Signalling layer is also needed for controlling connections and session parameters. In most cases signalling is performed using H.323 or SIP protocol. For transporting real-time audio and video data, usually RTP/RTCP protocol is used.

2.3 Media streaming services

Starting with the early 1990s personal computers have become powerful enough to handle various types of media, such as audio or video. Firstly, this media content was delivered over non-streaming channels (e.g. by downloading). However, since the early 2000s, thanks to more powerful processors and improvement of network bandwidth, streaming media services have started to become increasingly popular. Especially live media streaming services, such as internet radio stations and television, have gained much attention recently. Significant progress has been made in this area, but there are still many challenges, especially in efficient distribution of real-time stream, providing mechanisms for ensuring proper QoS, maintaining user privacy and securing streamed content.

Basically, media streams can be divided into two main categories:

- live streams e.g. internet television, radio
- non-real-time streams e.g. Video on Demand services

End-to-end latency requirements for live streaming services are also tight, but less stringent than in the services described previously. Practically, even relatively high latency links can be successfully used for most services. In particular, it is possible to build an internet live streaming service using satellite links (latency over 250ms) [**9**]. On the other hand, because these services are high-bandwidth-demanding, efficient stream distribution is a very important problem. Systems using unicast connections,

in the case of multiple users scenario, may have high bandwidth costs. Unicast is the most commonly used transmission in the Internet, but it does not scale well when many users want to view the same media content concurrently - server sends a separate copy of the media stream to each user, which may lead to network congestion. In the case of hundreds of simultaneous users IP multicasting technology [**10**] is certainly a better choice. It allows to efficiently send IP datagram to a group of interested receivers, but it also has some drawbacks. To work properly multicast services require multicast-capable routers that must be installed at all levels of the network. Due to this practical issue, it still has not been widely available. Another approach for building efficient live streaming systems is based on P2P network.

In P2P application level multicast, live stream viewers behave as routers for other users. This approach has proven to be useful, scalable and efficient [**11, 12**], and has been implemented in many widely used applications e.g. Adobe Flash 10.1. Our system described in this paper also uses P2P overlay network for delivering live content.

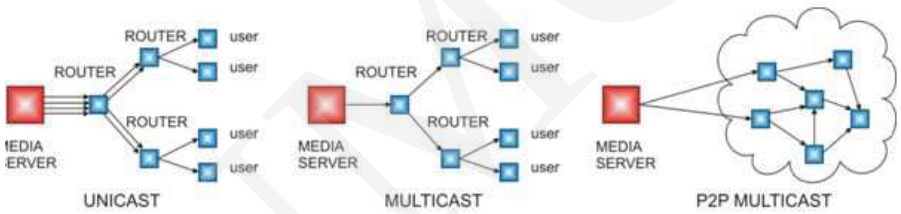

Fig. 1. Comparison of transmission methods.

On the other hand, non-real-time media streaming services, like Video-on-demand, in most cases are still based on unicast connections. Individual point-to-point connection is set up between streaming server and each user. In those systems link latency is not a big problem because local buffers are used for caching a portion of the multimedia data [**13**]. Using shared buffet for media server disk arrays also improves system efficiency [**14**]. In most cases, multimedia content is transported using the UDP or RTP protocol. For controlling streaming server, Real Time Streaming Protocol (RTSP) can be used. It provides VCR-like commands, such as play and pause. Also to avoid content piracy, the VoD content should be encrypted, additional Digital Rights Management server can be used for that purpose.

Application presented in this paper can be used for building distributed VoD systems. Distributed architecture prevents from media server overloads, minimizes request-to-server delay, and by using an intelligent caching mechanism improves overall system performance.

2.4 Multiplayer gaming

When it comes to online multiplayer gaming, it is more dependent on the type of game that is played. Basically, we can consider three types of network games:

- Real-time strategy games (RTS);
- Massively multiplayer online role- playing games (MMORPG);
- First Person Shouters (FPS);
- Non-real-time games e.g. Chess.

Usually, RTS or MMORPG games are less sensitive to delays caused by the quality of the link. In their case, test shows that often only delays bigger than 500 ms are noticeable [**15**]. On the other hand, dynamic nature of FPS type games causes, that even relatively low latency could have big impact on the performance of multiplayer online game. Often delay bigger than 200 ms can be noticeable and sometimes even make game unplayable. Some FPS game servers choose not to consider at all users with delay over 250 ms as potential players.

The end-to-end latency problem is even more important in becoming increasingly popular cloud gaming systems, like OnLive. In this type of services, games are stored and executed on remote servers, then rendered game content is streamed to the user via the Internet, so even users with less powerful computers can play games that normally would require better hardware. Because, besides streaming game video content, controller actions are forwarded to the game center, so very low latency connection must be used for good gaming experience.

3 System architecture

Application presented in this paper is based on a P2P network. As opposed to the traditional client/server architecture nodes of the P2P system (peers) act both as a client and a server and sometimes as relay for other peers in the network. With using a P2P model it is possible to implement an abstract overlay network that is built at Application Layer and consists of all participating peers as network nodes. This abstract network allows building system independent of the physical network topology, because data transport service is provided by Transport Layer considered as part of underlying network.

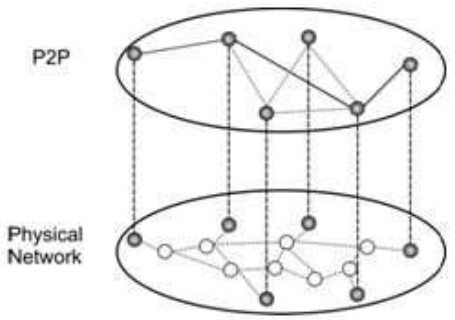

Fig. 2. Overlay network model.

We chose the P2P model for our application for several important reasons. First of all, owing to overlay network it is possible to make our own routing decisions and be more independent of external routing protocols. It is important because widely used routing protocols are often not real-time traffic friendly [**16**], but now by monitoring path quality between peers it is possible to choose the best route for packets transmission and quickly respond to any quality changes. Another P2P feature, that has proven to be useful in the services described in the previous sections, is self-organization, which implies that any peer can enter or leave network at any time without a risk of overall system stability degradation. Owing to self-organization, the system can be easily extended and is more reliable and less vulnerable to failures and attacks. A chosen model has also benefits in efficient streaming media distribution. Peers are used for application level multicast when streaming live media, and for implementing distributed VoD service, they and cache recant streamed movies which, combined with implemented nodes load-balancing mechanisms, increase overall performance and stability.

Another benefit of this system is an automatic elimination of problems with clients that are behind NATs. In normal circumstances, when both clients are behind NATs they are unable to establish direct connection. Although there are techniques like Session Traversal Utilities for NAT (STUN) [**17**] that can detect the presence of a network address translator and obtain the port number that the NAT has allocated for the applications UDP connection, but they are often ineffective [**18**]. In this case, additional server for traversal transmission between clients is required. In most cases that additional relay server is not on optimal path between those two clients, so it imposes additional delay in real-time communication. On the other hand, in P2P network peers are used for data routing, so no additional server is required, and because we chose peers that form the optimal path between clients transmission delay is minimized. Finally, owing to overlay network architecture and own routing protocol, we were also able to implement additional security mechanisms; they are described in Subsection 4.2.

3.1 Application components

Our applications consists of three main elements:

Login Server

As the name suggests, the main task of the server is to provide services for authentication and authorization. Each user that wants to connect to the system previously must be logged into this server. Registration of new users and account management is also supported by the server. If needed, it can be used for charge calculation as well. Also every node must previously bypass the authorization check before it can be attached to the infrastructure. In the user logging process each user is assigned to a specific node, selection is based on a geographic location and load information. Server contains actual information about every user's state, its current IP address, port number and assigned node ID, so it is used to determine user's current location by other users, which is necessary for example in communication service to establish a call. Because server has full knowledge about a current state of nodes and path qualities it

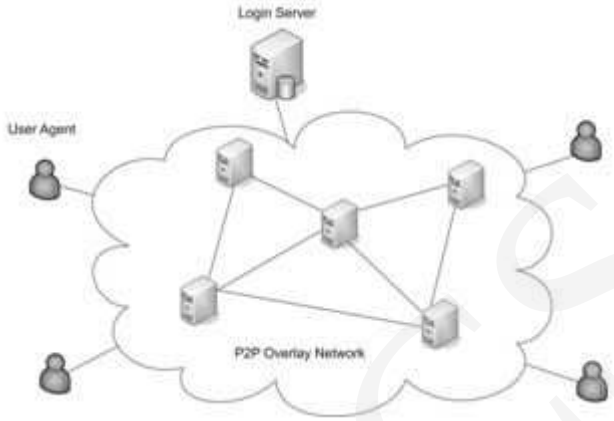

Fig. 3. Infrastructure components.

plays a main role in informing other nodes about any changes, so nodes can quickly recalculate their routing tables. As can be noticed, the presence of the server is critical for this application, so it is important to provide appropriated hardware resources for its operation. For big systems, it is also possible to split these services between several servers.

Nodes

Nodes are an essential part of our application and every P2P network. Their main task is to handle end-user support. As it was written before, during the logging process each user is assigned to a specific node and that node is used to route signaling messages and real-time data between other users and nodes. They also have a built-in mechanism allowing them to mediate between the user and the Logging Server (e.g. in the logging process), owing to it infrastructure is more resistant to blocking (e.g. by the Internet Service Provider). Every node in our infrastructure has knowledge about other nodes and qualities of paths between them - this knowledge is used for building route tables. Nodes cooperate with the Logging Server by exchanging information about paths. They report states of path between neighbours and receive information about rest of the infrastructure. Because in our system nodes have to perform many tasks, we decided to use agents that will take over some of them. That allowed us to decompose code, so it becomes more transparent. System gains flexibility - nodes can be easily upgraded just by changing agents (even remotely). Additionally, different kinds of agents can be used, e.g. intelligent agents with ability to learn and adapt to different network conditions and by communicating with each other they can share knowledge and make routing more efficient. Each agent acts within the single node. Its main task is to constantly control the quality of transmissions relayed by this node. It analyzes such parameters as link bandwidth usage, a number of lost packets, time interval between each packets etc. Agents also test state of the other temporarily not being used links for

detecting any changes. In the VoD service they analyze popularity of streamed media, which helps build dynamic cache with those most popular at the moment. More about agents and routing mechanism is written in the next section.

From a security perspective, we distinguish two types of nodes: standard and trusted. Every machine that has required resources can join our network and become a standard node. Nodes that had been previously verified as trusted can be used for routing data that requires a higher level of security. Besides, to ensure greater security and high reliability of the presented VoIP application, we provide it with a reputation mechanism. Every node has its own reputation index assigned by Logging Server, based on node reliability and long time behaviour. Those reputation indices are used for supporting mechanisms for building routing tables.

End terminals

Depending on the selected service, end terminals may be applications installed on computers (or smartphones) that are used for making and receiving calls, plugging for software/hardware media player etc. In standard configuration they use only two ports: TCP for signalling and UDP for real-time data transfer, so it is easy to configure firewall to work with them. After logging in to the Logging Server (directly or if direct connection is unenviable/blocked by any of nodes belonging to the infrastructure – it has a list of trusted nodes in memory) end terminal connects to a node assigned by Server and it is ready for use.

In the VoIP configuration, the system can support availability status for contact list. When the user is logged in, or has just changed his status, other users that have him on their contact lists are informed about this change. This mechanism works due to bilateral relations between the users stored in the Login Server database. After being added to someone's contact list the user is asked for permission to check his status. If permission is granted, the relation between those users is stored on server so they can be immediately informed about status changes.

Other elements

Besides the mentioned elements, the presented application can be easily extended with the additional specialized nodes, like public switched telephone network (PSTN) gateway or other IP telephony standards (e.g. SIP or Skype) gateways, media databases, TV broadcast server etc.

4 Design and implementation details

In this section we give an overview of design issues based on our implementation.

4.1 Quality of Service

As it was said before standard best effort Internet is not real time traffic friendly. Although there are techniques for providing QoS like Intserv [**19**] or Diffserv [**20**] that reserve certain network resources for handling real-time traffic; they still depend on service providers policies and are often unable to ensure required end-to-end quality.

The method proposed in this paper can be used to complement those existing mechanisms. Our application combines the traffic flow adjustment method and the path switching technique to ensure best possible connection quality. The traffic flow adjustment method is popular and widely used in controlling real-time traffic. The essence of this method is to adjust codec configuration parameters (output rate, voice frame size, etc.) and play buffer size to adapt to current network state. To make proper adjustments it is necessary to determine actual quality so feedback information is needed. It can be provided by using additional feedback channel (like in RTCP) or added into real-time traffic flow: into audio/video(e.g. using watermarking techniques) or into packet header [**21**].

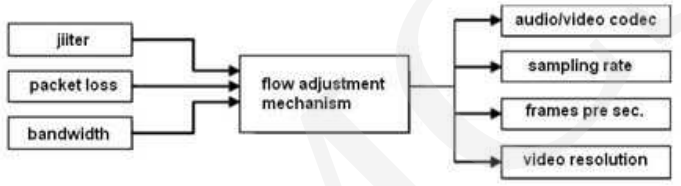

Fig. 4. Flow adjustment mechanism.

For simplicity, in this application we chose to add feedback information about quality into real-time traffic packets header, so if quality falls below desired level end user terminal will modify audio parameters. Also every node on the path is informing its neighbour about quality of the links between them. It is done by inserting the additional information like a number of sent and received packets, average delay and jitter into header. By analyzing that data, agent within the node has knowledge about the current quality of the link, and if it detects any changes, it may decide to re-route traffic by choosing other nodes to relay data. Logging Server is also informed about these changes and it passes this information to other nodes. Additionally, frequent changes in link quality affect this link reputation by decreasing it. Temporary not being used links are also regularly tested by agents, they are sending (with the desired time interval) series of test packets to simulate real-time traffic and analyze the responses. For performing routing decisions every node is building a graph that represents the current network state, then Dijkstra's shortest path algorithm [**22**] is applied, but instead of shortest path counting, paths with the best end-to-end quality are chosen. It is done by assigning to each edge in the graph its cost index, which is calculated by multiplying the corresponding link quality index by its reputation. If any link state has changed, graph needs to be updated and Dijkstra's algorithm applied again.

4.2 Security

In the case of designing security mechanism for real-time traffic, it is very import to select appropriate security level. It must be chosen so that it ensures safety of transmission but also is not too demanding for resources (additional bandwidth and

CPU power). If too many security mechanisms are applied it can affect QoS, so call quality may be degraded. It is also possible that users may choose to disable these mechanisms to get better connection quality. In this application we used the fallowing security schema:

User logging: for the logging process TLS connection is established. The user verifies the authenticity of the Logging Server using his CA certificate (with was previously delivered with a client program, or downloaded from WWW page). Next, Digest method is used for user authentication. Afterward, server chooses node that will handle this client, and with server assist (server-node connection is also secure) node and client exchange their public keys, client updates information about his contact list, connection ends.

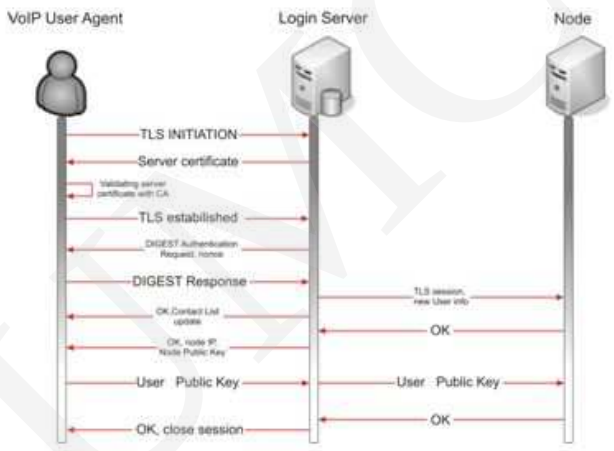

Fig. 5. User logging schema.

User to node connection: secure TLS connection is established, for two-way authentication previously exchanged with Login Sever assist keys are used.

Signalling and real-time traffic. Nodes are used to assist in signalling between end-users. A main reason for choosing this signalling method is willingness to provide mechanism for maintaining secrecy of end-users location, so the IP address needs to be hidden from other users. For example in order to establish a phone-to-phone call, only user's names and indices of nodes to which they are attached are needed (indices are not necessary - they can be retrieved for the Login Server, but in this schema server load and time needed to establish connection is reduced). Node, to which the calling user is connected establishes TLS connection with the destination user node, then they forward signalling data between users. The Diffie-Hellman key exchange protocol is used to establish the encrypting key for real-time transmission. To avoid problems related with maintaining the Public Key Infrastructure (PKI) users do not use certificates for authentication. But in the case additional security is needed, we provide a

mechanism for to-way user authentication: Login Server as a trusted intermediary is used.

For real-time transfer TLS cannot be used because it is based on TCP, so it can cause additional delays. In this application we used AES in an integer counter mode [**23**] (with the key agreed within the signalling process) as a stream cipher. Bits from cipher are XORed with sound data, and SHA-1 hash function is used to ensure packet integrity.

Besides, because nodes are participating in data transmission, we have bigger control over the data flow route. If higher security level is required, it is possible to specify the desired route, so regardless of the external routing protocol we can avoid paths that are vulnerable to eavesdropping.

4.3 Implementation and testing

We chose VoIP service for the testing environment. It was written in C# and uses DirectSound to access the sound device. Also we created a simple agent platform for our needs: agents can be run as separate threads and communicate with each other using sockets.

For testing our infrastructure in many different network configurations the additional simulation software was written. This simulator allows to graphically create desired network infrastructure by adding nodes and connections between them, real-time data flow is created by streaming audio files, then links parameters and nodes behaviours can be changed in order to simulate different cases. For simplicity, simulator is using the same software that is running on nodes: it is running them as threads, and configuring by assigning different port numbers on the same IP. Node state, link delay, jitter, and packet dropping percentage can be set.

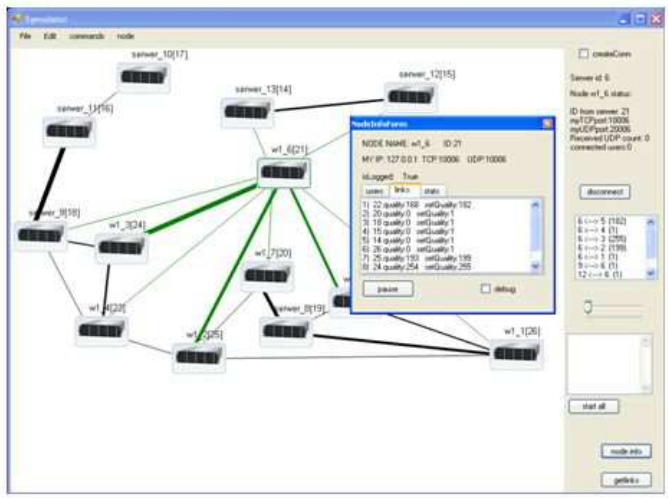

Fig. 6. Infrastructure simulator.

5 Conclusions and future work

In this paper we presented a new, based on Peer-to-Peer network model, application for real-time data transfer. As a preliminary, the most popular real-time services with their basic characteristics and requirements from network infrastructure were presented. Next, out system model, infrastructure elements and few implementation details were described. Also benefits of using P2P networks for building real-time infrastructures have been mentioned. Additionally, by placing an agent on each infrastructure node and indicating the advantages of this approach, we showed that agent based programming could be a valuable tool for designing this kind of systems. In our application we implemented mechanisms for ensuring the highest possible connection quality, and security in particular, that for improving QoS through continuous measurement of audio/video quality at each node in the overlay network, building dynamic routing tables and path switching technique. System security is guaranteed by using secure connection with authentication for the login process, AES encryption of real-time data and SHA-1 hash function for packets integrality. Additionally, owing to the P2P model, maintaining the secrecy of users location is possible, and by using internal routing protocols we can avoid unsafe paths.

So far, we have built working implementation of presented VoIP system, but it is still in its early testing phase. Many changes and improvements are still being made, so many elements like e.g. reputation mechanism behaviour or quality drops tolerance before path switch occurs still needs to be tweaked and validated by simulations. Also, besides software simulations, we are planning to build real infrastructure and test system behaviour in real environment. Work on implementing other services is also in progress, in particular live audio broadcast and video communication are in the advanced stage of testing.

In parallel, we are testing new features, like for example fast retransmission mechanism for real-time packets, that will be controlled by agents and will work between each two nodes that participate in data routing and are directly connected in the overlay network.

References

[1] Cohen D., A protocol for packet-switching voice communication, Computer Networks 2(4-5) (1976).
[2] Perey C., Feldman M., Videoconferencing over IP Networks, in Broadband Networking, J.Trulove, ed., CRC Press (2000): 193.
[3] Tsui K. M., Desai M., Yanco H. A., Uhlik C., Exploring Use Cases for Telepresence Robots, in Proceedings of the 6th ACM/IEEE.
[4] Rosenberg J. et al., SIP: Session Initiation Protocol, RFC 3261, IETF (2002).
[5] H.323, Packet-based multimedia communication systems, ITU-T (2003).
[6] Wooldridge M., Jennings N. R., Intelligent agents: theory and practice, The Knowledge Engineering Review (1995).
[7] G.114, One way transmission time, ITU-T (2003).

[8] Gross J.H., Etter D.M., Comparison of echo cancellation algorithms for the adaptive delay filter, 42nd IEEE VTC (1992).
[9] Henderson T., Katz R., Transport protocols for Internet-compatible satellite networks, IEEE JSAC 17(2) (1999): 345.
[10] Quinn B., Almeroth K., IP Multicast Applications: Challenges and Solutions, RFC 3170 (2001).
[11] Picconi F., Massoulié L., Is There a Future for Mesh-Based live Video Streaming?, p2p (2008): 289.
[12] Xiaojun H., Yong L., Ross K.W., IPTV over P2P streaming networks: the mesh-pull approach, IEEE Communications Magazine 46(2) (2008).
[13] Almeroth K.C, Ammar M.H., The use of multicast delivery to provide a scalable and interactive video-on-demand service, IEEE JSAC 3 (2004): 1467.
[14] Sengodan S., Li V. O. K. , A Shared Buffer Architecture for Interactive VOD Servers., infocom, pp.1341, INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution (1997).
[15] Henderson T., The effects of relative delay in networked games, PhD thesis, University of London (2003).
[16] Che X., Cobley L. J., VoIP Performance over Different Interior Gateway Protocols, IJCNIS (2009).
[17] Rosenberg J., Mahy R., Matthews P., Wing D., Session Traversal Utilities for NAT, RFC 5389 (2008).
[18] Hu Z., NAT Traversal Techniques and Peer-to-Peer Applications, HUT T-110.551 Seminar on Internetworkingl (2005).
[19] Baden R. et al., Integrated services in the Internet architecture: An overview, Tech. Rep. IETF RFC 1633 (1994).
[20] Blake S. et al., An architecture for differentiated services, Tech. Rep. IETF RFC. 2475 (1998).
[21] Mazurczyk W., Kotulski Z., Adaptive VoIP with Audio Watermarking for Improved Call Quality and Security, Journal of Information Assurance and Security 2(3) (2007): 226.
[22] Pioro M., Medhi D., Routing, Flow, and Capacity Design in Communication and Computer Networks, The Morgan Kaufmann Series in Networking (2004).
[23] NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation, NIST (2001).