Agent based VoIP Application with Reputation Mechanisms

Grzegorz Oryńczak and Zbigniew Kotulski

***Abstract*—In this paper we introduce our new VoIP model the aim of which is to meet the challenges of modern telephony. We present project concepts, details of implementation and our testing environment which was designed for testing many aspects of VoIP based systems. Our system combines mechanisms for ensuring best possible connection quality (QoS), load balance of servers in infrastructure, providing security mechanisms and giving control over the packet routing decisions. The system is based on Peer-to-Peer (P2P) model and data between users are routed over an overlay network, consisting of all participating peers as network nodes. In the logging process, each user is assigned to a specific node (based on his/her geographic location and nodes load). Every node also has a built-in mechanism allowing to mediate between the user and the main server (e.g. in logging process). Besides that, 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 presented system is usage of agents. Each agent acts with a 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 packets etc. The information collected by the agents from all nodes allows to built a dynamic routing table. Every node uses 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 sent by other nodes. To ensure greater security and high reliability of the system, we have provided a reputation mechanism. It is used during updating of 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.**

***Index Terms*—voice over IP, IP telephony security, speech quality control, software agents**

I. INTRODUCTION

VOICE sending is an incredibly useful and worth developing feature among many possibilities given by the Internet. 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. Nowadays, it becomes a serious competitor to standard telephony. Many advantages of this form of communication like cheap (or free) calls, wide range of additional features (video calls, conference calls, etc.) made it popular among companies and ordinary homes. Taking into account the continuous increase of Voice over IP (VoIP) users, it is safe to say that internet telephony will be one of the main forms of communication. However, there are still some challenges that it has to face, like providing a mechanism to ensure proper quality of service (QoS) and good security for data transfer and signaling.

Because VoIP is a real-time application it has specific requirements from the lower layers. The most important of them are related to delay, jitter and packet loss. 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, communication starts to be annoying. ITU-T G.114 [2] recommends maximum of 150 ms one-way latency. And because it includes the entire voice path, the network transmit latency should be significantly smaller than 150 ms. Unfortunately, for real-time applications we cannot use standard internet transport protocols such as TCP and UDP because they are not designed for this specific use, so they do not give us control over delay and jitter. Because TCP is a connection oriented protocol it is slower then UDP and built-in retransmission mechanism is often useless for real-time transmission – retransmitted packets are outdated. For multimedia data, reliability is not as important as timely delivery, so UDP is a preferable choice to base on for building real-time protocols. Although UDP has its benefits when it comes to speed, protocols based on it have to deal with lack of some important mechanisms. First of them is a congestion control mechanism which is not present in UDP and if the sender exceeds transmission rate that can be handled by the network it leads to congestion problems and network overload. The protocol should also implement mechanisms for time-stamping packets to allow synchronization and minimize jitter problems. RTP/RTCP defined in RFC 1889 [3] is currently most widely used transport protocol for real-time services. It can estimate and control actual data transmission rate but QoS is still not guaranteed.

In this paper we introduce our new VoIP model the aim of which is to meet the challenges of modern telephony. As opposed to standard client/server architecture used for example in SIP [4] or H.323 [5], we chose to base our system on 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

Grzegorz Oryńczak is a PhD student at Jagellonian University, Department of Physics, Astronomy and Applied Computer Science, Cracow, Poland (corresponding author, email grzegorz.orynczak@uj.edu.pl)

Zbigniew Kotulski a professor at Institute of Fundamental Technological Research of the Polish Academy of Sciences and professor at Department of Electronics and Information Technology of Warsaw University of Technology, Poland (email: zkotulsk@ippt.gov.p)

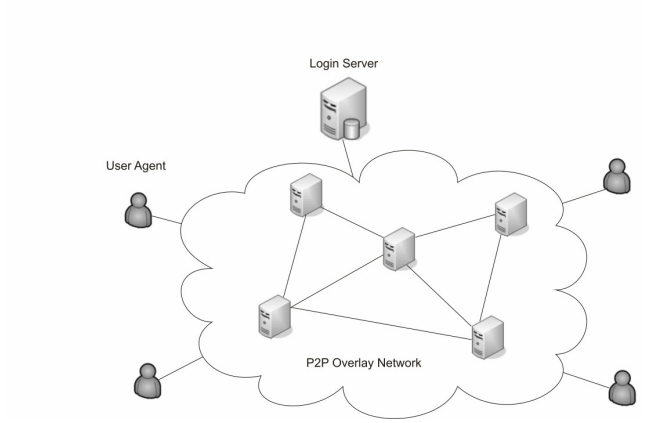

Fig. 1. Overlay network model.

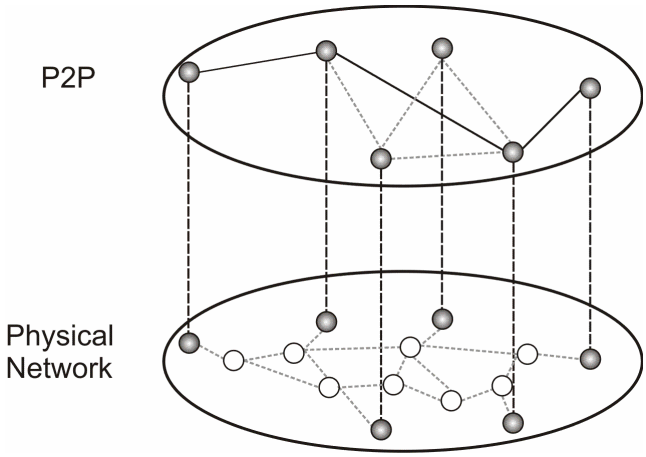

Fig. 2. Infrastructure components.

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 VoIP 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 the Section 2 the system architecture is described with node model and general communication flow. The Section 3 is devoted to security and QoS mechanisms. Finally, the Section 4 concludes our work.

II. System Architecture

VoIP application presented in this paper is based on a P2P network. As opposed to 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. When using a P2P model it is possible to implement an abstract overlay network that is build at Application Layer and consists of all participating peers as network nodes. This abstract network allows to build system independent from the physical network topology, because data transport service is provided by Transport Layer considered as part of underlying network.

We chose the P2P model for our application because of several important reasons. First of all, owing to overlay network it is possible to make our own routing decisions and be more independent from external routing protocols. It is important because widely used routing protocols are often not real-time traffic friendly [7], 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 described VoIP application, 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, system can be easily extended and is more reliable and less vulnerable to failures and attacks. Additionally, nodes load-balancing mechanisms implemented in this application 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) [8] 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, they are often ineffective [9]. 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 Section III.B.

A. Application components

Our VoIP 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 who wants to connect to the VoIP system must be previously logged into this server. Registration of new users and account management is also supported by the server. 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. The server contains up-to-date information about every user state, its

current IP address, port number and assigned node ID, so it is used to determine current user location by other VoIP users. Because the server has full knowledge about current state of nodes and path qualities it plays the most important function in informing other nodes about any changes, so nodes can quickly recalculate their routing tables. As one may notice, the presence of the server is critical for this VoIP 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 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 VoIP signaling 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 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 status of the path between neighbors and receive information about the 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 the code, so it becomes more transparent. The 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, number of lost packets, time interval between packets etc. Agents also test state of the other temporarily not being used links for detecting any changes. More about agents and routing mechanism is written in the next section.

From the 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. Apart from that, 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 behavior. Those reputation indexes are used for supporting mechanisms for building routing tables.

End terminals

End terminals are applications installed on computers (or smartphones) that are used for making and receiving calls. Because application uses only two ports: TCP for signaling and UDP for real-time data transfer, it is easy to configure firewall to work with it. 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) application connects to a node assigned by Server and it is ready for use.

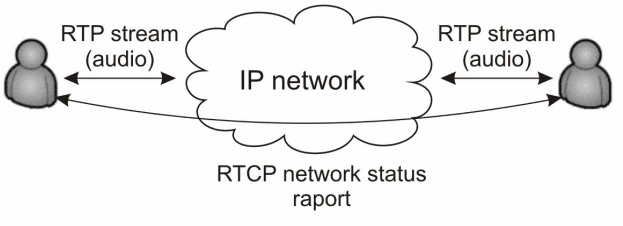

Fig. 3. RTP/RTCP transmission schema.

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

Other elements

Apart from mentioned elements, the presented application can be easily extended with additional specialized nodes, like public switched telephone network (PSTN) gateway or other IP telephony standards (e.g. SIP or Skype) gateways.

III. Design and implementation details

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

A. Quality of Service

As it was said before, the standard best effort Internet is not real time traffic friendly. Although there are techniques for providing QoS like Intserv [10] or Diffserv [11] 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. Method proposed in this paper can be used to complement those existing mechanisms. Our application combines traffic flow adjustment method and path switching technique to ensure best possible connection quality. 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(e.g. using watermarking techniques) or into packet header [12].

To make is simple, 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 neighbor about the quality of links between them. It is done by inserting additional information, like number of sent and received packets, average delay and jitter, into header. By analyzing that data, the 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 another 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 on this link reputation by decreasing it. Temporary not being used links are also regularly tested by agents, they are sending (with desired time interval) series of test packets to simulate real-time traffic and analyze the responses. For performing routing decisions every node is building graph that represent current network state, then Dijkstra's shortest path algorithm [13] is applied, but instead of shortest path counting, paths with 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.

B. Security

In case of designing security mechanism for real-time traffic, it is very important 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 affects on QoS, so call quality may be degraded. It is also possible that VoIP users may choose to disable these mechanisms to get better call quality. In this application we used following security schema:

User logging – for logging process TLS connection is established. The user verifies the authenticity of the Logging Server using his CA certificate (with was previously delivered with 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.

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

Signaling and real-time traffic. Nodes are used to assist in signaling between end-users. Main reason of choosing this signaling method is willingness to provide mechanism for maintaining secrecy of end-users location, so IP address needs to be hidden from other users. In order to establish phone-to-phone call, only user names and indexes of nodes to with they are attached are needed (indexes 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 calling user is connected establishes TLS connection with destination user node, then they forward signaling data between users. Diffie-Hellman key exchange

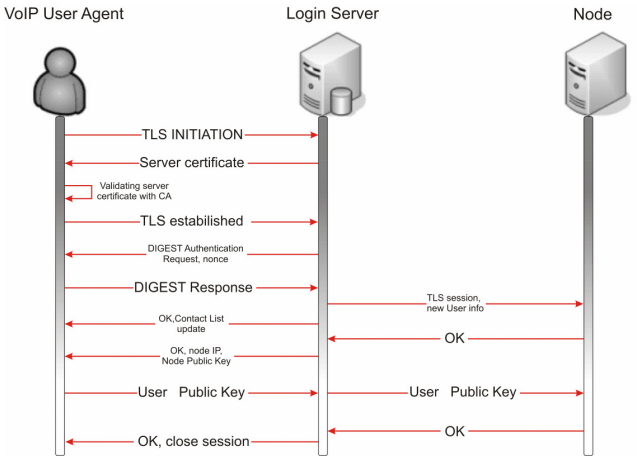

Fig. 4. User logging schema.

protocol is used to establish 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 case additional security is needed, we provided 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 integer counter mode [14] (with the key agreed within signaling process) as a stream cipher. Bits from cipher are XORed with sound data, and SHA-1 hash function is used to ensure packet integrity.

Apart from that, because nodes are participating in data transmission, we have greater 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.

C. Implementation and testing

Our system 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 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 behaviors 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.

IV. Conclusions and Future Work

In this paper we presented new, based on Peer-to-Peer network model, IP telephony system. The system model, infrastructure elements and some implementation details were

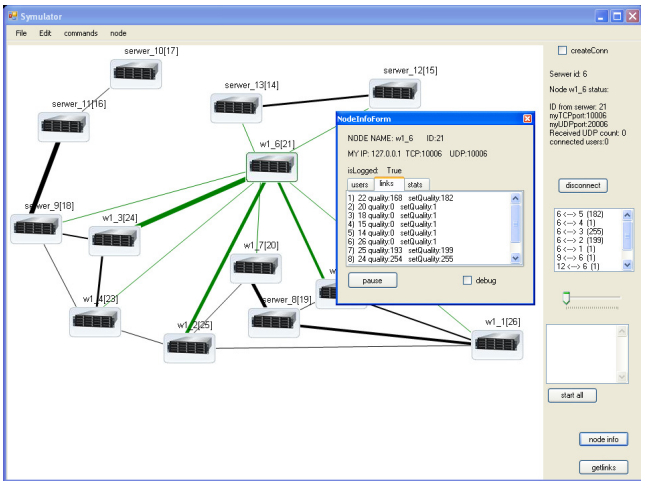

Fig. 5. Infrastructure simulator.

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 call quality, and security. In particular, a mechanism for improving QoS through continuous measurement of sound 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 login process, AES encryption of real-time data and SHA-1 hash function for packets integrality. Additionally, owing to 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 a 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 i.e. reputation mechanism behavior 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 behavior in real environment.

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] D. Cohen, "A Protocol for Packet-Switching Voice Communication," *Computer Networks*, vol. 2, no. 4–5, 1976.
[2] *One Way Transmission Time*, ITU-T, G.114, 2003.
[3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF, RFC 3550, Tech. Rep., Jul. 2003.
[4] *H.323. Packet-Based Multimedia Communication Systems*, ITU-T, Jul. 2003.
[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," IETF, RFC 3261, Tech. Rep., Jun. 2002.
[6] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice," *The Knowledge Engineering Review*, 1995.
[7] X. Che and L. J. Cobley, "VoIP Performance over Different Interior Gateway Protocols," in *IJCNIS*, Apr. 2009.
[8] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "Session Traversal Utilities for NAT," RFC 5389, Tech. Rep., Oct. 2008.
[9] Z. Hu, "NAT Traversal Techniques and Peer-to-Peer Applications," in *HUT T-110.551 Seminar on Internetworking*, Apr. 2005.
[10] R. Baden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," IETF RFC 1633, Tech. Rep., Jun. 1994.
[11] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," IETF RFC. 2475, Tech. Rep., Dec. 1998.
[12] W. Mazurczyk and Z. Kotulski, "Adaptive VoIP with Audio Watermarking for Improved Call Quality and Security," *Journal of Information Assurance and Security*, vol. 2, no. 3, pp. 226–234, 2007.
[13] M. Pioro and D. Medhi, "Routing, Flow, and Capacity Design in Communication and Computer Networks," *The Morgan Kaufmann Series in Networking*, 2004.
[14] M. Dworkin, "Recommendation for Block Cipher Modes of Operation," *NIST Special Publication 800-38A, NIST*, 2001.

Zbigniew Kotulski received his M.Sc. in applied mathematics from Warsaw University of Technology and Ph.D. and D.Sc. Degrees from Institute of Fundamental Technological Research of the Polish Academy of Sciences. He is currently professor at IFTR PAS and professor and head of Security Research Group at Department of Electronics and Information Technology of Warsaw University of Technology, Poland.

Grzegorz Oryńczak received his M.Sc in computer science from Jagiellonian University. He is currently a Ph.D. student in computer science at the Jagiellonian University and Institute of Fundamental Technological Research of the Polish Academy of Sciences. He also works as a senior specialist at the National Center for Nuclear Research, Świerk.