Exploring Peer-to-Peer Infrastructure for Computer Supported Collaborative Work Applications

Tianming Wei Yongjun Xu Yiyun Zhao Nishant Khanna Bing Gao Yvonne Coady
Department of Computer Science
University of Victoria, Victoria, BC, Canada

***Abstract*—Computer Supported Collaborative Work (CSCW) has been greatly enhanced by technology that provides almost real-time capture, replay, and sharing of content. "The cloud" has popularized ease of deployment for apps serviced through a centralized approach. In this work we propose a more aggressive design, decentralizing the location of shared content within a peer-to-peer structure. Ease of deployment is maintained through a novel architecture for a multilayer distributed cloud. This paper describes how this approach has shown promise in a prototype app, *ThinkTogether*, and the potential to enable a consistent user experience, even for remote peers sharing VoIP.**

***Index Terms*—Voice over IP, Peer-to-peer, Quality of service, Traffic balance, Bandwidth utilization.**

I. Introduction

Collaboration platforms for Computer Supported Collaborative Work (CSCW) enable users to leverage technology to cooperatively develop shared content. Successful platforms now support everything from education to medicine, typically supported through a centralized service or cloud. Though read-only content distribution has been successfully shared between peers [1], mutable content with complex interactions and dependencies has been more difficult to decentralize. Essentially, the problem is that decentralized approaches require coordination as nodes can come and go, and connections can fail, creating a myriad of different failure modes.

In this paper, we overview the design of a Peer-to-Peer (P2P) infrastructure for a CSCW application, ThinkTogether. ThinkTogether is an iOS prototype for creating and sharing content in almost real-time. We consider the specific problem of Voice over IP (VoIP), as it is representation of a stream of contents that are particularly sensitive to service disruptions. Our goal is to be able to share all content, including voice, live, but also allow users to replay it later based on access to a shared server or the cloud. Due to a variety of factors that could affect the quality of service (QoS) for the application, such as limited bandwidth, high volume traffic, network delay and the effects of other peers, we propose a dynamically decentralized approach, in order to optimize the network performance as the conditions change in the system. We consider this as a case study for an educational platform, which could ultimately subvert the traditional way of learning, sharing and communicating with mobile technology.

This paper is organized as follows. After an overview of related work (Section II), we present our model of a hierarchical architecture, and provide the verification of the model (Section III) as well as the design using a simulation software OMNeT++ [2]. We then describe the ways in which the simulation results inform the implementation of the system. In particular, we identify bottlenecks to anticipate in the VoIP educational platform case study. The analysis underscores the importance of augmenting the existing prototype with the means of easily distributing data throughout the peers in the system, in a way that provides adequate bandwidth for both the current users and the cloud-based content capture (Section IV). We generalize this construct, and propose that if we treat the platform as a *distributed cloud*, which may be able to address the key challenge of scaling across potentially thousands of mobile devices, all creating and modifying shared content, in a globally distributed system (Section V). We conclude with an outline of future work (Section VI).

978-1-4799-7492-4/15/$31.00 ©2015 IEEE

II. Related Work

Our case study focuses specifically on VoIP (Voice over IP), which is a technology used for voice communication over multimedia devices with the help of the IP protocols. Wook & Kang [3] focused on VoIP using SIP (Session Initiation Protocol) for smartphones in general, whereas we hope to extend some of these results to iPads. Fathi et al. [4] optimized VoIP services in wireless networks and reduced the disruptions in real-time by focusing on the network layer as opposed to the link layer. In our work we aim to use a similar concept, where we aim to reduce delay in voice streaming for live presentations for wireless devices with a focus on both the network and link layer.

There are many studies on VoIP applications and how to improve the capacity of such systems in Wireless networks. For example, Rao et al. [5] provided VoIP services for GSM (Global System for Mobile) based devices without making any modifications in the GSM network. Similar work by Fong et al. [6] focused on Windows. Approaches like these are great examples of how to handle traffic dynamically, to avoid any disruption in voice streaming in real-time. Komolafe & Gardner [7] studied aggregation of VoIP calls, whereas we focus on aggregation of voice streams. Agarwal et al. [8] consider energy management of VoIP over Wi-Fi for Smartphones, which should be directly applicable to our case study using iPads. An increase in voice streaming will most definitely require the energy of the device to be managed in an efficient and environment-friendly manner. Chen et al. [9] provided a methodology to measure the Quality of VoIP over mobile devices, and Kim et al. [10] provided metrics for performance of VoIP over the mobile networks, and we intend to leverage these same methodologies in our work. Related work studying VoIP using P2P design includes Wang et al. [13], where they analyzed the delay in the Chinese internet system, which also factors in as an important concern in our case study.

An experimental study on Skype P2P VoIP Systems by Guha & Daswani [14] showed a variety of characteristics of network traffic from the data derived of online users in the world. We hope to leverage these insights in our work, along with the work done by Hobfeld & Binzenhofer [15] focusing on the Quality of service (QoS) of the end-to-end connections and the Quality of Experience (QoE) of the end user. A similar analysis of the Skype P2P Internet telephony protocol by Baset and Schulzrinne [16] establishes the superior voice quality in Skype relative to other state-of-the-art applications.

Sundar et al [11] put forward a novel algorithm along with suggested architecture to support P2P voice transmission among mobile phones. Short-distance mobile communication is set up through Bluetooth, while long-distance calls are connected through Wi-Fi. The mechanism developed uses GSM technology if no costfree networking is accessible. Our primary focus is managing data while connected to the network. Ghassan Kbar et al [12] also concentrate on eliminating costly GSM telecommunication by exploring other channels such as Bluetooth and Wi-Fi. Our goal is to extend the voice feature of the ThinkTogether app to be adaptive of the current network situation.

Fobert et al [17] provides a general look into IP telephony in Server/Client Model managing voice data packets. The paper was published in the early years when IP telephony was immature, therefore it only provides a fundamental principle of transferring voice data over the Internet. We use this as a comparison to the proposed P2P architecture in terms of the performance in voice data transmission within the ThinkTogether app.

III. Network Scenario and Modeling

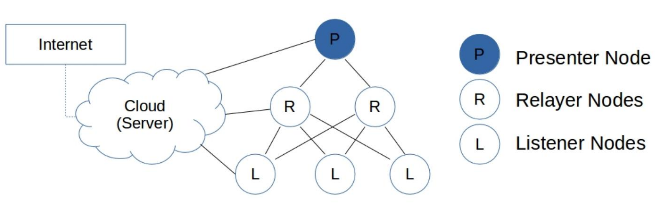

Fig. 1. P2P-based Model for VoIP

There are four main types of components/nodes in the designed scenario, a presenter, multiple relayers and listeners, and a remote cloud server. In our application all the nodes are connected with the cloud server and they do not need any kind of internet connection for transmitting voice through the network. Each component can coordinate with one another in order to stream content such as the voice or camera videos of the speaker, to the listeners in real-time. As shown in Fig. 1, the network is organized in a hierarchical architecture. The presenter will produce the streaming content from the root node and broadcast it to the relayers and listeners, as well as to the cloud servers for backup.

For example, one can think of the model as a lecture in real life. A presenter is an instructor who gives the lecture in front of the room, and the listeners are students who are listening the lecture (can also be remote listening). When a presenter creates a presentation, the voice or videos being recorded will be sent to both the child nodes and the remote cloud server so that the servers can maintain a full copy for the streaming content in the presentation. All the other nodes can passively receive the packets from the presenter. There are different routes a voice packet can travel through. The listeners receive voice packets through the relayers while the cloud server can provide backup services. If any of the listeners or relayers encounter a data loss due to a bad network condition, the node can send requests to the cloud server and retrieve the missing content. Once a relayer goes down, the listeners will request the voice stream from the cloud. Presumably the relayer will rejoin, and the system will reorganize to optimize in that case. It is presumed that relayers are closer to the listeners, affording better service due to locality and load distribution. The cloud will take over and serve as the main approach to deliver packets. For a more complex scenario, where there are multiple roles for the listeners, relayers act as filter and only forward packets to the designated attendees.

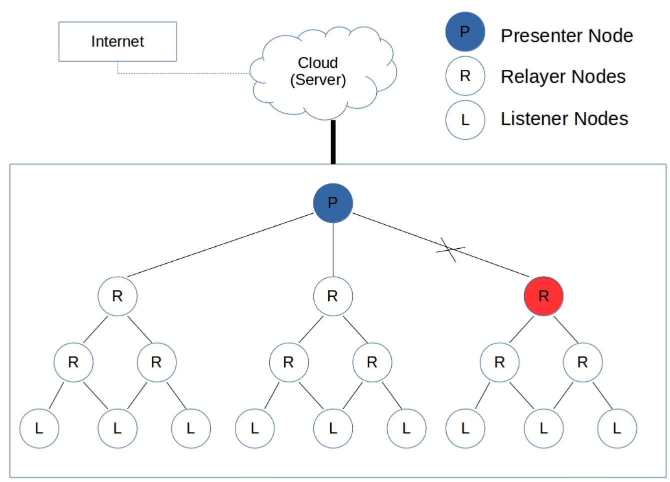

Fig. 2. Relayers Become Presenters

However, this simple hierarchical architecture cannot be adaptable to dynamic network changes and could experience an unbalanced network traffic problem which could seriously reduce the quality of service and experience. For example, Fig. 2 shows that if a high-level (near to the root) relayer suddenly leaves the network, all the nodes lower than the relayer will not be able to receive the streaming content any more, and thus the server will receive a large quantity of requests from the nodes which results in lossing the connection from the presenter. This will cause the server bandwidth to be occupied by these nodes which will make the network flow unbalanced. Also, the relayers among these nodes could lose their advantage in passing the content to the listeners, as the listeners can already obtain the requested packets from the cloud server and the bandwidth between the relayers and the listeners is wasted.

Therefore, our proposed solution is to design a P2P-based architecture among the relayers and listeners. The server could act as the tracker in this scenario. Every node will have to register with the server first when it joins in a presentation so that the server will be aware of the network situation for each node. According to the information the server receives from each node (including bandwith, account information based on the future business model, etc.), some nodes will be selected as relayers at different levels, while the other nodes will act as the listeners (leaves) and establish connections with the relayers. The nodes at the same level will be connected as well, to establish the P2P-based architecture, as shown in Fig. 3.

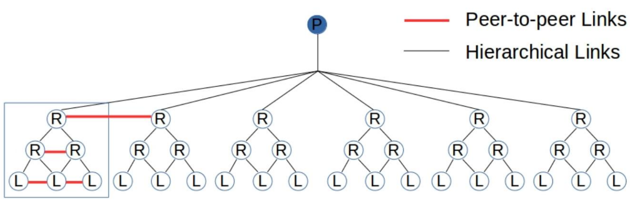

Fig. 3. Relayers Become Presenters

IV. Analysis and Case Study

We used OMNeT++ as a simulation tool to explore the most suitable model and algorithm for ThinkTogether. The study done by Varga [2] provides a brief introduction of OMNeT++, including Model Libraries, NED language, model structure. The paper also describes how to execute the simulation under a powerful graphical user interface. Work done by Anggadjaja et al. [18]. features the use of OMNeT++ based simulation for a reliable point-to-point wireless transmission. The outcome could provide inspiration for employing the P2P model in a wireless network. There are several other papers [19], [20] explaining the main infrastructure and internal design of OMNeT++.

We evaluate several wireless and P2P frameworks [21], such as MiXiM [22], OverSim [23]. MiXiM is an OMNeT++ modeling framework created for mobile and fixed wireless networks. MiXiM concentrates on the lower layers of the protocol stack, and offers detailed models of radio wave propagation, interference estimation, radio transceiver power consumption and wireless MAC protocols. OverSim is a flexible overlay network simulation framework based on OMNeT++. OverSim includes several structured and unstructured peer-to-peer protocols like Chord, Kademlia and Gia. Researchers can use these protocols to run simulation for academic purpose as well as real world networks.

In our work, two network models will be built in order to observe the behavior of data packets transferred in between the particular networks. One of the networking protocols being used is Telnet, for which J. Postel et al [24] provides the specification and standards for using the protocol. This is a relatively old paper that was published in the early 1980s. The paper draws a whole picture of the Telnet protocol from general consideration to signal processing. Another protocol adopted is HTTP. There are wider range of papers dedicated to HTTP. One of them is given by R Fielding [25] , who provides a full-scaled overview, which can be used as a reference for modelling a network simulation.

ThinkTogether is a communication application that contains multiple layers for packet transmission. Data sent by each individual is handled differently. G. Sekin et al [26] provides an insight of how to manage multi priority traffic in ATM network. The study is content-based video objects. As the common characteristics shared between video stream and voice stream, such as low tolerance in delay, jitter and packet loss. Unfortunately, there is no evaluation involved in the paper, therefore the feasibility of the proposed algorithm remains unknown.

After designing the model, we plan to create a working network by creating a simulation using OMNeT++, as shown in Fig. 4. By creating this simulation we can have an idea of how the voice stream packets are being transferred between the different nodes present in the network. Once the simulation starts working for a basic scenario of just 2 leaf nodes, a relayer and a cloud server, we will add more number of nodes to the simulation and then analyse what changes are to be made to certain parameters of the network so as to avoid any kind of latency and disturbance while transferring the voice streams.

Presenters initialize services by sending voice packets to both relayers and the cloud. Upon receiving the packets, the server stores them locally and echoes back an acknowledgement. The services between presenter and server is now finished. On the other side, voice packets reach the relayer and will be directed to all attendees relayer connects to, which is illustrated by #6 and #7 event in Fig. 5. Each attendees request the exact same voice data from the cloud. Cloud handles the request by fetching corresponding data in server and sending back the requested content. One can refer to events from #11 to #16 for execution sequence details.

V. Implementation: Data Distribution and Layers

The simulation tests the efficiency in transferring voice data packets with a fundamental model. This model lays the groundwork for future expansion, and illustrates a simple way to consider a classroom scenario. The purpose of the simulation is to investigate the feasibility, and more importantly, the efficiency of the model. In the simulation, parameters such as packet drops, delay time, etc. are collected and extracted to generate an output file for further analysis. The output file is then used as a foundation for performance analysis.

In our simulation of the model architecture, we use multiple nodes as the leaf nodes in a network. Then we also have some nodes that act as relayers in the network that will transfer the voice stream from the presenter in the network to the corresponding leaf nodes in the network. Along with all these nodes we also have a cloud server which will act as a primary storage device for all the voice streams that will be transmitted throughout the network. All the nodes will have a direct link to the server so that if any of the leaf nodes (student nodes in our scenario) want to retrieve any voice stream they can do so by making a request to the cloud server for that voice stream directly through the relayer (big node).

To simplify, we assume all the attendees are physically located in the same classroom and they are subscribing to voice for the purpose of capturing the information. This will eliminate potential influence caused by the difference of location, such as propagation delay. There are five components in the model: presenter, relayer, cloud, server and attendee. Each component coordinates with one another in order to offer recording services to attendees. There are two test cases, one as a small-scaled network with two attendees, the other as a slightly larger-scaled network with ten attendees. The simulation runs in HTTP net, masking the lower level network layers such as TCP and IP.

Figure 4 shows the overall picture of the simulation model in a small-scaled case. It shows the four main parts in the scenario: the sender (presenter), the receivers (attendees), the relayers (mediators), and the cloud server (backup).

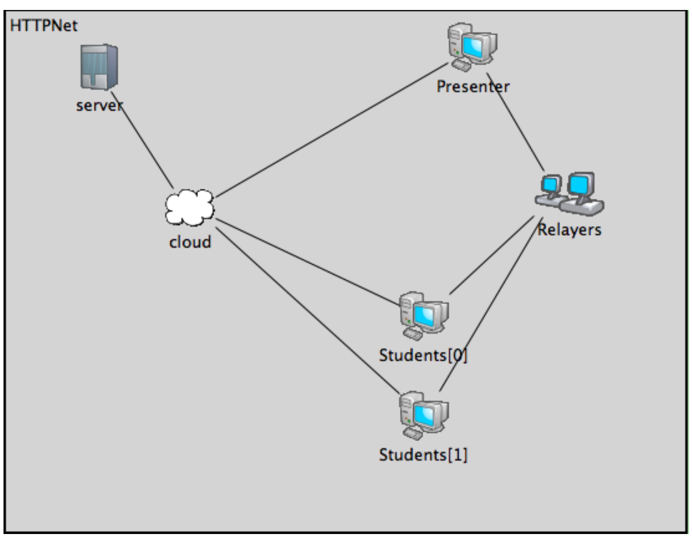

Fig. 4. Simulation in Omnet++

Presenter is the starting point of the service. It does not rely on any outer resource to trigger the sending process. Currently no packets other than acknowledgement are sent to the presenters. In other words, one can view presenters as a resource provider and there is no prerequisite for presenters to function. There are two outgoing links connected to presenters. The one connected to relayers is the main channel for delivering data. The one connected to cloud transfers the same content, even though it is less demanding in terms of propagation speed and processing time. A presenter sends voice packets to both links simultaneously, and sets a timer for the next cycle.

The cloud server can be considered as a group in terms of services purpose. They are both built to provide backup for attendees. There are two scenarios in which attendees would need backup services. The first is when packets are lost by the relayers and attendees will ask for complementary data from the cloud to ensure a smooth stream. The second is when attendees are not able to attend the class, but request a later review of that class. The server stores the lectures in the format of voice stream and delivers it to attendees via cloud. In this regard, the cloud is an interface between service providers and receivers. In addition, once the network is scaled up to include multiple servers, cloud acts as a coordinator that organizes the servers' actions.

The relayer acts as a router in this simple case simulation. Unlike servers, the relayer does not store voice data, therefore all the attendees can not request data from the relayer later on. It should be noted that this primary function is not yet implemented, and will be addressed later in the future model.

Attendees are students in the diagram. They receive voice packets from presenters, regularly through relayers. It should be noted that attendees will not send acknowledgement back to presenters. When the network between relayers and attendees are down, or for some reason, packets are lost in the regular transmission routes, attendees will request the data from the server. The target users of the ThinkTogether application are attendees and presenters.

To collect data from multiple cycles, we set a timer for both presenter and attendees based on an exponential distribution. After sending the voice packets, the presenter will create a timer that controls how long before the next cycle, namely, the next time it starts sending the packets. The duration of the timer is based on a exponential function, which allows random distribution in sending packets. The same principle applies to the rate of sending request from attendees. There is also a timer created by attendee that controls the rate of sending requests to the cloud. In real life, an attendee will need the complementary data from cloud only when the relayers fail to deliver it. However, in order to fully evaluate the function, in the simulation we configure an attendee to request voice data from the cloud whether the relayers go down or not.

The execution sequence is illustrated in Fig. 5. It is a log file generated by running small-scaled cases.

VI. Conclusions and Future Work

In this paper, we have initiated a model which decentralizes the location of shared content within a P2P structure for a CSCW application (VoIP). As a part of future work, we aim to extend this application for people who can download the voice stream from the presentation remotely in real-time.

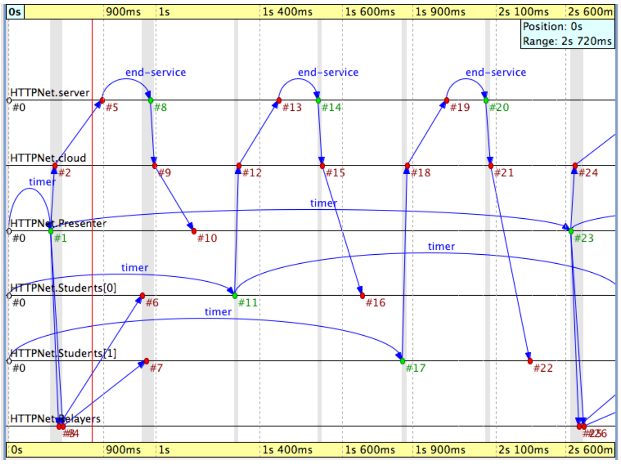

Fig. 5. Events in Time Sequence

We also aim to use a hierarchical structure for the network which will help the different nodes present in the network to change their roles during the presentation at any time. Once the application is in operation for a limited number of users, we aim to extend this application in a way so that it can be used for any number of users without having any storage issues, any disruption in the voice streaming or any bandwidth issues in the network by adding large number of users to the presentation. After this we aim to develop different use cases for scenarios where the system might have some complications and we will try to correct them in a way that is best suited for our application. Another thing that we hope to address in the future is that if any of the relayer nodes fails then all the leaf nodes linked to that relayer will be dynamically transferred to another relayer. New schemes will be developed to guarantee that these dynamic changes will not significantly reduce the network performance, as well as to keep a balanced traffic flow with the optimal bandwidth utilization. Once we have developed new schemes for our application, using OMNeT++ we will generate detailed log files to analyze the delay in routers during transmission and other parameters that might some kind of latency in the application. With the help of these log files we will be able to analyze any problems that the application might have in the real-life situation and we will try to alter our schemes accordingly to avoid any such problems from occurring in the future.

REFERENCES

[1] X. Yang, and G. Veciana, "Service capacity of peer to peer networks," In *IEEE Proc. of the Twenty-third Annual Joint Conference on Computer and Communications (INFOCOM)*, pp. 2242–2252, 2004.

[2] A. Varga, "The OMNeT++ discrete event simulation system," In *Proc. of the European simulation multiconference (ESM)*, 9(185), 2001.

[3] H. Wook and S. Kang, "Design and implementation of SIP-based mobile VoIP application for multiple smartphone OS," In *Proc. Information and Communication Technology Convergence (ICTC)*, pp. 565–568, 2010.

[4] H. Fathi, S. S. Chakraborty, and R. Prasad, "Optimization of Mobile IPv6-Based Handovers to Support VoIP Services in Wireless Heterogeneous Networks," In *IEEE Transactions on Vehicular Technology*, 56(1):260–270, 2007. doi: 10.1109/TVT.2006.883806.

[5] H. C. Rao, Y. Lin, S. Cho, "iGSM: VoIP service for mobile networks," In *IEEE Communications Magazine*, 38(4):62–69, 2000. doi: 10.1109/35.833558.

[6] M. Fong, R. Novak, S. Mcbeath, and R. Srinivasan, "Improved VoIP capacity in mobile WiMAX systems using persistent resource allocation," In *IEEE Communications Magazine*, 46(10):50–57, 2008. doi: 10.1109/MCOM.2008.4644119.

[7] O. Komolafe and R. Gardner, "Aggregation of VoIP streams in a 3G mobile network: a teletraffic perspective," In *Proc. Personal Mobile Communications Conference*, pp. 545–549, 2003. doi: 10.1049/cp:20030314.

[8] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta, "Wireless wakeups revisited: energy management for voip over WiFi smartphones," In *ACM Proc. of the 5th international conference on Mobile systems, applications and services (MobiSys)*, pp. 179–191, 2007. doi=10.1145/1247660.1247682.

[9] W. Chen, P. Lin, and Y. Lin, "Real-Time VoIP Quality Measurement for Mobile Devices," In *IEEE Systems Journal*, 5(4):538–544, 2011. doi: 10.1109/JSYST.2011.2165604.

[10] D. Kim, H. Cai, M. Na, S. Choi, "Performance measurement over Mobile WiMAX/IEEE 802.16e network," In *International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 1(8):23–26, June 2008. doi: 10.1109/WOWMOM.2008.4594817.

[11] S. Sundar, M. K. Kumar, P. Selvinpremkumar, and M. Chinnadurai, "Voice over IP via Bluetooth/Wi-Fi Peer to Peer," In *International Conference on Advances in Engineering, Science and Management (ICAESM)*, pp. 828–837, March 2012.

[12] G. Kbar, W. Mansoor, A. Naim, "Voice over IP Mobile Telephony Using WiFi P2P," In *the 6th International Conference on Wireless and Mobile Communications (ICWMC)*, pp. 268–273, September 2010. doi: 10.1109/ICWMC.2010.97.

[13] G. Wang, C. Zhang, X. Qiu, and Z. Zeng, "An efficient relay node selection scheme to improve the performance of P2P-based VoIP applications in Chinese internet," In *Multimedia Tools and Applications*, 64(3):599–625, 2013. doi: 10.1007/s11042-011-0952-5.

[14] S. Guha and N. Daswani, "An Experimental Study of the Skype Peer-to-Peer VoIP System," In *Computing and Information Science Technical Reports*, 2005.

[15] T. Hoßfeld and A. Binzenhöfer, "Analysis of Skype VoIP traffic in UMTS: End-to-end QoS and QoE measurements," In *Computer Networks*, 52(3):650–666, 2008. doi: 10.1016/j.comnet.2007.10.008.

[16] S. A. Baset, and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," In *IEEE Proc. of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, pp.1–11, April 2006. doi: 10.1109/INFOCOM.2006.312.

[17] J. Fobert, S. Navaratnam, P. J. Dagert, and S. J. McKinnon, "Client-server network for managing Internet protocol voice packets" U.S. Patent No. 6,853,713, February 2005.

[18] E. Anggadjaja, and I. McLoughlin, "Point-to-point OMNeT++ based simulation of reliable transmission using realistic segmentation and reassembly with error control," In *IEEE second International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT)*, pp. 125–128, 2010.

[19] M. Bohge, and M. Renwanz, "A realistic VoIP traffic generation and evaluation tool for OMNeT++," In *Proc. of the 1st International Workshop on OMNeT++*, March 2008.

[20] X. Qing, and X. Ren, "Research of routing protocols simulation for wireless sensor networks based on OMNeT++," In *IEEE International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE)*, pp. 79–82, June 2012.

[21] F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz, and I. Stoica, "Towards a common API for structured peer-to-peer overlays," In *Proceedings of the 2nd International Workshop on Peer-toPeer Systems (IPTPS)*, pp. 33–44, 2003.

[22] X. Li, L. Wang, L Liu, X. Hu, F. Xu and J. Chen, "OMNeT++ and Mixim-based protocol simulator for satellite network," In *IEEE Aerospace Conference*, pp. 1–9, 2011.

[23] I. Baumgart, B. Heep, S. Krause, "OverSim: A Flexible Overlay Network Simulation Framework," In *IEEE Global Internet Symposium*, pp. 79–84, 2007.

[24] J. Postel, and J. K. Reynolds, "Telnet protocol specification," 1983.

[25] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol–HTTP/1.1," 1999.

[26] G. Seckin, and F. Golshani, "Real-time transmission of multilayer video over ATM networks," In *Computer Communications*, 23(10):962–974, 2000.