

VoPSI: Voice over a Publish-Subscribe Internetwork

Charilaos STAIS, Dimitris DIAMANTIS, Christina ARETHA, George XYLOMENOS

Department of Informatics

Athens University of Economics and Business

76, Patisson Str., 10434, Athens, Greece

Tel: +30 210 8203693, Fax: +30 210 8203686

{stais, diamantis, aretha, xgeorge}@aueb.gr

Abstract—*Information-centric networking constitutes an alternative to the conventional, IP-based internetworking, with information being identified rather than the host where it resides (which is the case for IP networking). This approach appears to be very promising for the next generation Internet. However, many challenges and critical issues remain to be addressed, associated with the range of applications that can be supported by the new architectures. Specifically, it is unclear whether information-centric networking abstractions can support conversational applications which are very important in the Internet, and even more important in other telecommunication networks. In light of those reflections, we present the design, prototype implementation and performance evaluation of a simple voice application for the Publish-Subscribe Internet architecture developed in PSIRP and now being further refined in PURSUIT, two EU FP7 research projects on clean slate Future Internet design.*

Index Terms—publish/subscribe, future internet, voice telephony

I. INTRODUCTION

The organic growth of the Internet is largely due to the ubiquitous adoption of the TCP/IP protocol stack. TCP/IP based networks were designed to simply forward traffic between pairs of communicating end hosts, following the prevailing communication pattern of previous networks, such as the telephone network. However, communication patterns have evolved, and the use of the Internet has shifted towards information-centric services and applications, such as content delivery networks (CDNs), cloud computing services and peer-to-peer (P2P) file sharing applications like BitTorrent. In these services, in sharp contrast to the underlying Internet model, the focus is on the information itself rather than on the specific end hosts producing or consuming it. Hence, these services are implemented as overlay solutions on top of an information-agnostic network substrate [1].

The current Internet design, as well as the design of computer networks in general, is based on the concept of a pair of users wanting to communicate. The design thus focuses on the location of the information/data. It has long been realized that this has led to high network resource consumption for content distribution applications, since large numbers of information consumers are served via individual connections to information producers. As Internet traffic grows, the community needs to find a solution to serve its needs with the available resources. The best approach is to design the entire network protocol stack from scratch, taking into consideration the facts of current and future Internet needs.

Many approaches have been suggested for redesigning the Internet [2], [3], [4], [5], all revolving around the concept of having information as the central entity in the architecture. We have not yet seen sufficient proofs of concept indicating that such innovative architectures will work in practice. In addition, it is unclear how the well-known applications of the current Internet will adapt to the new functionality. For example, PURSUIT [5] is an EU FP7 research project developing a clean-slate publish/subscribe internetworking architecture (the PSI architecture, or Ψ in short). The conversion of existing Internet applications for this architecture is a challenge, as traditional applications have been designed with an entirely different model in mind. In the Ψ architecture we are only interested in the information itself; we do not care about the information location. This fact means that applications must be redesigned in an information-centric way, instead of in a location-based way. This is not simple, because adapting to the publish/subscribe model requires applications to be redesigned over principles completely different to those of the traditional client-server model reflected in the Berkeley sockets API [6]. In this paper, we present a detailed description of our implementation of a voice telephony application, VoPSI (Voice over Publish/Subscribe Internetworking), paying particular attention to the ways in which such services can adapt to the publish/subscribe model of communication.

The remainder of this paper is organized as follows. In Section II we present the Ψ architecture developed in PURSUIT, while in Section III we discuss the VoPSI architecture and implementation. Section IV first provides a description of the experimentation environment and then presents and discusses the performance results obtained. We finally conclude and discuss our plans for future work in Section V.

II. Ψ ARCHITECTURE OVERVIEW

A publish/subscribe architecture involves three major entities: the publishers, the subscribers and an event notification service [7]. Publishers hold the role of information providers. They advertise the availability of specific pieces of information and they provide these pieces of information to the network. They perform these operations by issuing *publication messages*. Subscribers are information consumers, who express their interest for specific information items by issuing *subscription messages*.

The event notification service, initially locates the publishers who provide information items that satisfy the consumers'

subscriptions. Afterwards, it initializes a forwarding path from the information providers towards the information consumers. The publication and subscription operations described above, *do not have to be in sync*. Moreover, publishers and subscribers - the principal actors of the architecture - do not have to be fully aware of each other: they only need to be aware of the information that they want to exchange.

A Rendezvous Network (RENE) implements the event notification service and is therefore responsible for matching subscriptions with the appropriate publications, as well as for orchestrating the information forwarding process. A RENE is composed of several rendezvous nodes (RNs), each of which is responsible for a set of publications. We refer to the RN that is responsible for a publication as the Rendezvous Point (RP) of this specific publication. For the purpose of identifying objects, we use (statistically) unique labels for each discrete piece of information. Publishers wishing to issue a new publication must use two identifiers: RId and SId. A publication's RId (rendezvous identifier) can be derived by an application specific function. A publication's SId (scope Identifier) denotes to which extent the publisher wishes the publication to be made available, by denoting a scope with appropriate access controls. Both the RId and SId are independent of the endpoints producing and consuming the associated information items.

Scoping mechanisms are used to limit the reachability of information only to the parties having access to a particular scope. Within a scope, the architecture is neutral with respect to the semantics and structure of the data, although governance rules regarding the available information may be defined. Scopes employ a hierarchical structure where parent-children and sibling relationships exist [8]. In the PURSUIT architecture, there can be physical scopes such as a corporate network, and logical scopes such as a social network.

Prior to publishing an information element, publishers have to locate the nodes that are responsible for managing the desired scope. One of these nodes will be the RP for the publication. What is actually published to the RP is the publication's metadata, which contain information about the data of the publication and not the data itself; this metadata can be, for instance, the author of the publication, its size and, perhaps, a small description of the publication.

In order for a subscriber to access a publication, she must be aware of its RId and SId. She expresses her interest in a specific publication by issuing a subscription message towards the publication's RP. Upon receiving a subscription message, and provided that an appropriate publication exists, the RP initiates a process that creates a forwarding path from the publisher towards the subscriber. This forwarding path is identified by a forwarding identifier (FId). The FId is a Bloom filter based structure that includes all the link identifiers which the publication needs to traverse in order to reach the subscriber(s) [9]. Multicast is the preferred delivery method in PURSUIT, therefore in cases where multiple subscribers exist for a particular RId, a multicast delivery tree is created. Moreover caches may act as publishers and the RP can choose a cache-publisher that is closer to the subscriber than the original publisher.

In Figure 1, we present a simple example of data delivery in a Ψ network, using a random topology. All the aforementioned components are depicted in the figure. The red arrows indicate the multicast tree constructed to transport data from the publisher to the three subscribers.

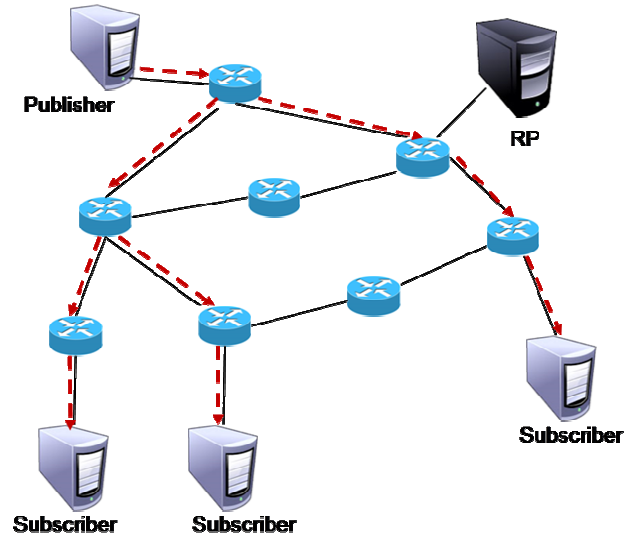


Fig. 1. PSI example.

III. THE VOPSI APPLICATION

The Session Initiation Protocol (SIP) [10] is a signalling protocol used for establishing sessions in an IP network. A session can be a simple two-way telephone call or it can be a collaborative multi-media conference session, based on the request-response paradigm. The protocol can be used for creating, modifying and terminating two-party (unicast) or multiparty (multicast) sessions consisting of one or several media streams.

Skype [11], [12] is currently the most popular VoIP application. The infrastructure of Skype depends on servers, but not only on them. The term Super Users indicates that any Skype user may be promoted to a special user of the system, if a few prerequisites exist, such as uptime and bandwidth availability. That is the reason why Skype belongs to the peer-to-peer system category.

In SIP, each user receives an identity from the SIP Gateway she connects to, e.g. if user A registered herself to sipgate.com, she will receive the identity $y@sipgate.com$, where y is the registration number. If user B wants to communicate with user A, user B has to look for registration y which is accessible from sipgate.com or other SIP gateways that interconnect with sipgate.com. On the other hand, in Skype there is a global discovery mechanism, where each user chooses a unique username and everyone is able to find her through the discovery mechanism.

There currently exist two different approaches for creating a telephony application over the Internet. The first is to build an application based on SIP for signalling, and the second is

to implement your own connection control, as in Skype. If we want to describe the VoPSI application in a few words, we can say that VoPSI is a SIP-like initiated telephony application with some influences from Skype.

A. Architecture Overview

The design and implementation of VoPSI is based on the fundamental principles and methods provided by the PUR-SUIT architecture, which enable us to perform the essential functions of an Internet telephony service, that is, establishing a connection between the communicating nodes (signalling) and *two-way voice communication* between the nodes (voice data transfer). The establishment of the connection between two nodes requires a rendezvous point. The Ψ architecture allows subscribers to ask for content that has not yet been published in the network. In this case, the subscribers are the ones who specify the rendezvous point where the publishers are going to publish their content. In our case, the callee is the subscriber and the caller is the publisher, therefore the callee subscribes in order to indicate its willingness to accept calls, and the caller publishes its desire to establish a call.

After the connection is established, the participants can commence a two-way voice data message exchange. This raises the need for the unambiguous identification of successive voice messages exchanged between the two parties. The information exchanged during the establishment of the connection along with the use of sequence numbers provides some degree of flow control for the conversation. If a packet arrives with a sequence number less or equal to the previous one, we can drop it rather than push it to the audio component of the application.

Every node of the network using VoPSI has a unique name, which is disclosed only to nodes wishing to communicate with it. We assume that a third-party authority, similar to the Skype user search service [11], is responsible for the names. So, if a node desires to find a name, it will request it from this authority. In VoPSI, the SId represents the space where all the incoming calls for a host will arrive, therefore the name of the node is used to produce the SId of the information network that will carry out the conversation and the RId of the messages sent to that node (RId₁). Similarly, the RId of the messages sent by the node is produced by the name of the node located on the other end of the conversation (RId₂), using however the same SId. RId₁ and RId₂ are produced by applying a hash function (SHA-1) on the name of the callee and the caller, respectively, while the SId is obtained by applying the same hash function to RId₁. This does not mean that the entire scope indicated by that SId is bound to VoPSI; the user is able to reuse the same SId for another purpose, e.g. for participation in a social network.

The communication model is illustrated in Figure 2. The figure depicts the conversation between the caller and callee inside the information network denoted by the SId. Each endpoint uses two RIds, one for publication messages and one for subscription messages.

In a VoPSI conversation, the callee states in advance that she wishes to accept an INVITE message, through a subscribe

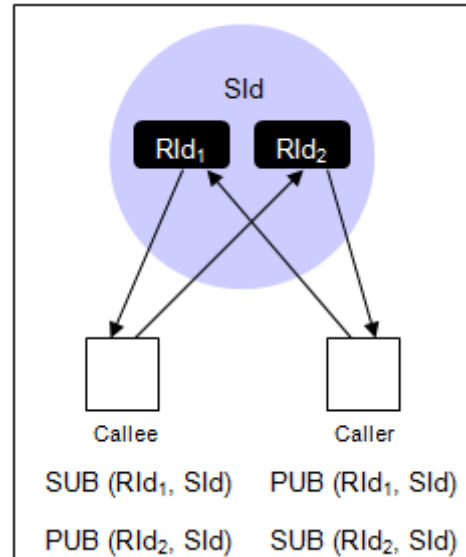


Fig. 2. Management and use of SId and RIds.

message identified by RId₁ in the SId information network. The caller publishes an INVITE message on the same SId information network. The message's identifier is RId₁ and it includes RId₂. This is the identifier of the point where the caller will receive messages should the callee accept the invitation.

If the callee accepts the invitation, she notifies the other end through a publication (RESPONSE message) in the SId information network, identified by RId₂. By sending this response, the caller is sure that the callee got her INVITE message. Otherwise, after a timeout, the caller resends the INVITE message, because she assumes that the first one was never delivered to the callee. When the caller gets the RESPONSE message, the connection between the two parties has been established and they are now able to exchange voice data.

Figure 3 illustrates the communication protocol between the two ends of the conversation. Initially, the connection is established and then the two parties exchange voice data messages.

B. Implementation

Within the PURSUIT project there is a need for a testbed where all the introduced ideas can be tested. For this purpose, a specific prototype implementation was developed, called Blackhawk [13], [14], [15]. Blackhawk is implemented as a kernel module in FreeBSD and makes the OS capable of using TCP/IP and pub/sub communication simultaneously. FreeBSD using Blackhawk is able to send metadata and publications and receive subscriptions over Ethernet adapters, bypassing the entire TCP/IP stack for this purpose. Additionally, in Blackhawk, apart from the publisher and subscriber, another entity is implemented, the Topology Manager (TM). The TM is an early version of a router for a Ψ network, which routes data and has a cache. When the TM receives a subscription,

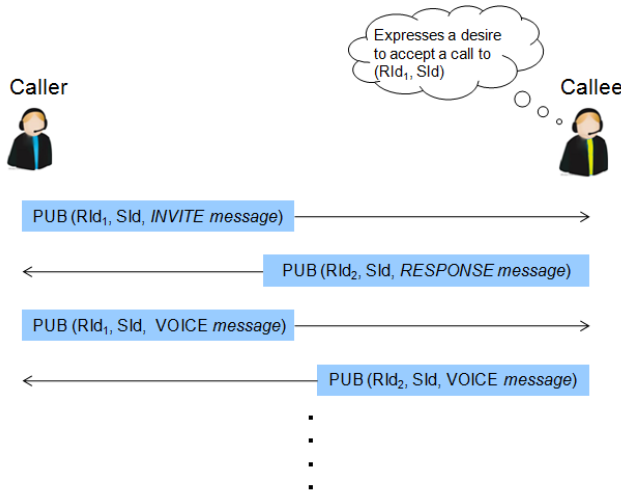


Fig. 3. VoPSI call initiation.

it checks if this message already exists in its cache. Similarly, for each publication, the TM checks if the data exists in its cache and, if not, it stores it.

On top of Blackhawk, Libpsirp was developed. Libpsirp is a shared C library that hides the system call specific details from applications and can also perform other helper functions. Its main purpose is to exploit the system calls provided by the Blackhawk kernel module to implement an API for the applications. Finally, we developed a Java Language Wrapper, in order to be able to develop pub/sub applications while still taking advantage of Java's garbage collecting environment. Moreover, developers can utilize a variety of external tools and libraries (data structures, graphical user interfaces, etc) developed by the large Java open source community.

For the development of VoPSI we used all the tools and libraries described above. In order to achieve asynchronous communication between caller and callee, the separation of application into two functions was necessary, one for data transfer over the Ψ network and one for sound management (capturing and reproducing). Concerning the sound component, we used the WAV audio format, a sample rate of 44100 Hz and 16 bits per sample. We did not use compression, since our target is not to introduce a new voice application with better performance than the existing ones, but to provide a proof of concept for the Ψ architecture.

Unfortunately, due to some limitations of the Blackhawk implementation, a SId and RId pair is rapidly consumed by the exchanged messages, since only a limited number of publications can be sent over the same SId/RId pair as consecutive versions of the same publication. In order for our application to run for a reasonable time, when all the available versions of the current SId and RId pair have been consumed, we generate a new set of RIds. For this purpose, we use a scheme called Algorithmic RIds, where a new Id is generated via the SHA-1 hash function, using as input the previous RId. In this way, VoPSI is able to run for as much time as we want it to.

IV. EXPERIMENTATION AND EVALUATION

The first step for the evaluation of an application is to test it in the lab. To this end, we decided to use two different setups. The first one consists of two nodes connected on the same switch and running VoPSI. As we expected, the results were ideal and this makes sense, because there was no overhead at all. The second step is to repeat the previous experiment with cross traffic, in order to stretch the link and increase the possibility of data loss and delay. Again, we did not notice any loss, only a few cases of delays on the order of milliseconds. In all experiments, the duration of application execution was one minute and there was no sound distortion.

We repeated the experiments using a TM as an intermediate node between caller and callee. In our case, since every data exchange is unique, the caching process at the TM (check and store) is repeated for each packet, introducing additional overhead and increasing dramatically the time for a packet to travel from one end to the other. We noticed after several experiments that a delay of 4 seconds on average is introduced that cannot be resolved with the current implementation of the TM. Note that there was no sound distortion due to the TM, only a delay.

PURSUIT also provides a testbed [16] with the purpose of serving as an early testing ground for the implementation work, but also for demonstrating application concepts. It is one of the engagement tools that works with external collaborators when it comes to experiments. There are currently eight sites established, with seven located in Europe and one in the US. The majority of the testbed machines are connected through a VPN. It is well-known that the VPN delay consists of two things: the latency of the connection between the connected machines and a small overhead caused by the encapsulation and decapsulation of each packet. The PURSUIT testbed has a mean latency of 500 milliseconds for the topology described above; therefore it was not reasonable to evaluate the performance of VoPSI through this setup.

For a substantial evaluation of VoPSI, we decided to implement an alternative to VoPSI, where the message exchange remains the same, but UDP transport is used for communication. In this way, we are able to measure the initial setup delay in each case and the overhead due to Ψ architecture. The experiments performed over an Ethernet LAN showed that the setup delay in VoPSI takes 0.2 sec more than in the native UDP version of VoPSI. Additionally, we measured the time needed for one packet of voice to travel from mouth to ear and found that there is a small delay of 0.1 sec in VoPSI, again in comparison with the native UDP version of VoPSI. Less than 0.3 sec latency is almost never noticeable to the user, provided that there are no additional latencies, as those in our prototype kernel implementation of Ψ . If we consider the early stage of development of the protocol stack of our implementation, these latencies are absolutely reasonable, but they are not large enough to influence performance.

V. CONCLUSIONS AND FUTURE WORK

In this paper we show that established and popular real-time streaming applications, such as voice telephony, can be

mapped to an information centric publish-subscribe framework for a future Internet. The approach presented here takes advantage of the proposed Ψ architecture and the new principles and methods that come with it. This work is part of ongoing research towards a Ψ based which is expected to answer questions left open here. We plan to further investigate the correctness and effectiveness of our proposed solution more extensively, test it over the project's international testbed, which we expect to be improved soon in terms of latency, and compare it with VoIP [17]. Finally, we plan to try the same conversion mechanism with other well-known applications, such as BitTorrent [18].

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their useful and important notices and comments. The work reported in this paper was supported by the FP7 PURSUIT project under contract FP7-ICT-2010-257217.

REFERENCES

- [1] D. Clark, B. Lehr, et al. "Overlay networks and the future of the Internet," *Communication & Strategies*, vol. 63, 2006.
- [2] V. Jacobson, D.K. Smetters, et al.: "Networking named content." Proceedings of the 5th international conference on Emerging networking experiments and technologies. CoNEXT, 2009.
- [3] Named Data Networking Project, <http://www.named-data.net/>, Last visited: December 2010.
- [4] The FP7 4WARD Project, <http://www.4ward-project.eu/>, Last visited: December 2010.
- [5] The FP7 PURSUIT Project, <http://www.fp7-pursuit.eu/>, Last visited: December 2010.
- [6] G. Xylomenos, B. Cici: "Design and Evaluation of a Socket Emulator for Publish/Subscribe Networks," Proceedings of the Future Internet Symposium, Berlin, 2010.
- [7] P.T. Eugster, P.A. Felber, et al.: "The many faces of publish/subscribe." *ACM Comput. Surv.* 35, pp. 114-131, 2003.
- [8] M. Ain, S. Tarkoma, et al.: "FP7-PSIRP Deliverable 2.2: Conceptual Architecture of PSIRP Including Subcomponent Descriptions", p. 38, 2008.
- [9] P. Jokela, A. Zahemszky, et al.: "LIPSIN: line speed publish/subscribe internetworking." In: SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication, pp. 195-206. ACM, New York, NY, USA (2009).
- [10] J. Rosenberg, H. Schulzrinne, et al.: "SIP: Session Initiation Protocol." RFC 3261, IETF, June 2002.
- [11] S.A. Baset, H.G. Schulzrinne. "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol." *CoRR*, 2004.
- [12] S Guha, N. Daswani. "An experimental study of the Skype peer-to-peer VoIP system." Proceedings of IPTPS, 2006.
- [13] P. Jokela, J. Tuononen, et al.: "FP7-PSIRP Deliverable 3.3: Progress Report and Evaluation of Implemented Upper and Lower Layer", p.4, 2008.
- [14] P. Jokela, J. Tuononen, et al.: "FP7-PSIRP Deliverable 3.4: Integration and Demonstration Plan", p.20, 2009.
- [15] D. Trossen, J. Tuononen, et al.: "FP7-PSIRP Deliverable 3.5: Final Description of the Implementation", p.4, 2010.
- [16] PURSUIT Testbed, http://www.fp7-pursuit.eu/PursuitWeb/?page_id=237, Last visited: December 2010.
- [17] B. Goode: "Voice over Internet protocol (VoIP)," Proceedings of the IEEE, vol.90, no.9, pp. 1495-1517, Sep 2002.
- [18] BitTorrent Protocol Specification, <http://www.bittorrent.org/beps/bep0003.html>, Last visited: December 2010.