Using SIP as P2P Technology

Luca Caviglione
University of Genoa – Department of Communications, Computer and Systems Science (DIST)
Via Opera Pia 13, 16145 Genova (Italy)
Phone: +39-010-3532202, Fax: +39-010-3532154
e-mail: barone@dist.unige.it

Luca Veltri
University of Parma - Dipartimento di Ingegneria dell'Informazione
Parco Area delle Scienze 181/A, 43100 Parma (Italy)
Phone: +39-0521-905768, Fax: +39-0521-905758
e-mail: luca.veltri@unipr.it

Keywords: p2p, SIP, overlay networks, Internet telephony.

Abstract - ***Nowadays peer-to-peer (p2p) technologies are widely adopted and used for building even more sophisticated services: from ubiquitous file-sharing system to the even more popular Internet telephony. In addition, the Session Initiation Protocol (SIP) has been used for different purposes. Due to its intrinsic generality and flexibility, it could be adopted to build and manage also p2p applications. Moreover, the p2p philosophy could be applied to the existing SIP architecture, to cope with issues such as Denial of Service (DoS).***

In this paper, we survey the state of the art of the joint use of p2p and SIP. Also, some hints and examples in order to use SIP as a core technological component of the p2p world, are presented.

I. Introduction

Recently, p2p applications have been massively deployed in the modern Internet. According to [1], an important fraction of the overall Internet's traffic is generated by file sharing applications. Recent studies have also demonstrated that the global p2p traffic is not lowering as it appears. In [2] authors underline, after a thorough measurement campaign, that p2p traffic is not decreasing as a result of law threats but instead it is hiding. In fact, modern file sharing systems deploy sophisticated mechanisms such as random port hopping or HTTP encapsulation in order to produce traffic patterns hard to prevent and locate. Besides to the ubiquitous file-sharing software, p2p has becoming also adopted for different purposes: from entertainment purposes to enterprise oriented services. For instance, Skype [3] offers a p2p based system for voice communication across the Internet. Even if with some interesting investigations in the IETF, the p2p phenomena, at least, the one associated with the file sharing explosion, has been largely investigated and exploited by enthusiasts, developers and normal people, rather than standardization bodies. But a deeper investigation will reveal that IETF already has about the totality of the required technology. We identify in the Session Initiation Protocol (SIP) [4] one of the most important and promising p2p enabler within the IETF. Due to its intrinsic generality and flexibility, it could be adopted to build and manage also p2p applications. Moreover, its preexistent centralized architecture could be empowered by exploiting some mechanisms adopted in the p2p world.

The paper is structured as follows: Section II describes an infrastructure to use the classical client-server SIP model in order to establish an Internet p2p-based telephony service. Section III, investigates the state of the art within IETF for using SIP as the signaling technology for building and maintaining p2p overlays. Section IV describes one proposal about how to replace some of the core components of JXTA with SIP in order to provide a "lightweight JXTA" and Section V depicts a simple example of a p2p-SIP application. Finally, Section VI concludes.

II. SIP as a p2p system

In p2p networking all the hosts have the same capabilities and the same responsibilities. To emphasize the aspect, all the entities involved in this kind of network are named peers.

However, the p2p paradigm introduces some problems: the topology is quite trivial, and the lack of a hierarchical organization brings to major difficulties in developing any kind of algorithms. There are no "well-known" nodes (in a client-server scenario, the server is the known service provider node) and the functionalities are shared across the network. This description perfectly fits for the unstructured p2p networks, such as Gnutella[1] [5]. In this kind of network all peers interacts by self-organizing themselves in order to form a network. Studies have shown that this kind of systems follow a "power-law" rule, where few nodes have an high number of connections and many nodes have a small number of connections. The main problem with this organization is due to the complete delocalization of the information and in order to find content a controlled broadcast mechanism is mandatory.

To cope with this, many proposals have been made in order to achieve better performance in the look-up phase

[1] In Gnutella, peers are also called "*servents*" as to emphasize the duplex role of servers and clients.

both for peers and resources. The root technology is the one based on Distributed Hash Tables (DHTs), allowing to organize the overlay by mapping the peers via a hashing function. This brings to structured p2p systems such as, CHORD [6] and Kademlia [7].

On the other hand, the current IP Telephony services has been built as phone-to-phone (peer-to-peer) system with very centric characteristics based on server/proxy nodes (in both H.323 and SIP architectures). Of course this approach may suffer of scalability problems, and some proposals to overcome this issue and to develop p2p Internet Telephony are arising. Particularly in [8] the authors present an architecture based on SIP.

The basic idea is to view the overall Internet telephony infrastructure as p2p architecture, where all the participants are organized in a p2p overlay network. Consequently, the overlay could be used to locate users for initiating a telephone call. This architectural blueprint resembles Skype but tries to overcome some of its limitations.

In fact, Skype has been developed in a closed way, it does not appear as extensible for future services and still it relies on a central server for authentication procedures. Besides, IETF has been investigated for years protocol and techniques for deploying Internet telephony. The most appealing candidate is based on the SIP architecture and with some adjustments it could be organized as a p2p system.

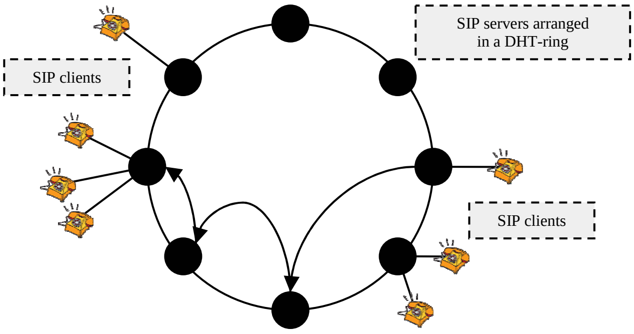

Figure 1 - SIP servers arranged in a DHT overlay.

With another respect, there are also some security issues that could be solved by using SIP in a p2p fashion. For instance, the architecture depicted in Figure 1, reduces the potential of failure due to a single node, resulting in increased robustness and allows providing a preliminary protection against Denial-of-Service (DOS) attacks. This is possible, owing to the intrinsic resistance of DHT-based system against churn, which is the continuous process of node arrival and departure.

III. SIP to build overlays

During the years, many different architectural blueprints and protocols have been developed for p2p systems, especially for file-sharing. Each developer promoted his brainchild and implemented the required protocol functionalities: from the Napster protocol to sophisticated mechanisms such as eMule [9], they are basically built from scratch. This fact also has produced a waste of resource, since some protocols are overlapped in functionalities. Moreover, the lack of a coordinated and standardized approach results in a variety of p2p systems that cannot interact each others.

In this perspective, even if not explicitly written for p2p, SIP could be used for different signaling purposes and, of course, also for building and managing p2p overlay systems. In addition, SIP can straightforwardly handle p2p traffic. This to demonstrate that albeit a big standardization campaign in p2p has not been performed, it must not be said that p2p architectures or supporting technologies are not available in existing technological pool.

For instance "*all servers in SIP are optional, allowing User Agents (UAs) to directly communicate*" [10]. Its server-less capabilities allow SIP to be one of the fundamental building block for p2p technologies.

To prove the maturity of the SIP technology to carry out all the signal procedures mandatory for exploiting a full featured p2p application, [11] tries to find out the required SIP messages in order to build an full featured Chord-based architecture. Particularly, the draft provides examples for the following operations: node registration, user registration, session establishment, node leaving the system, successful user search and unsuccessful user search.

This work looks promising allowing SIP to become a killer application, raising its popularity and, at the same time, exploiting the p2p paradigm to enhance the SIP technology itself.

IV. Using SIP to lighten the JXTA technology

In the last years several implementations of p2p systems arose, and each one came with its own nodes, protocols, and network architecture. Some of these implementations were very well developed and are today widely used.

However, although some of these implementations are open-source, none of them has still become a *de-facto* standard.

Currently there is an important open project named JXTA [13], originally sponsored by SUN Microsystems, that aims to define a standard architecture and its protocols for building up p2p applications and services. JXTA comes also with a reference open-source Java implementation of its complete suite of protocols in order to shorten the time required to develop new p2p applications.

JXTA defines a series of XML-based protocols for specific p2p basic functions, such as those for self-organizing peers to into peergroups, publishing and discovering peer resources, communicating, and monitoring other peers [14].

One of the main characteristic of JXTA is that it is independent of any particular hardware platform, operating system, or programming languages, and does not require the use of any particular network transport or topology; the JXTA protocols can be implemented on top of TCP/IP, HTTP, Bluetooth, or other protocols and technologies.

The approach followed by JXTA designers was to completely re-define a new network architecture and protocols, with poor reuse of already existing protocols, and this leaded on one hand to some duplication of network protocols and functionality, and on the other hand to a scarce interoperability with already existing p2p-like services such as real-time multimedia communications (also known as "IP Telephony"), building up with SIP.

As described in Section II and III, a completely different and challenging approach for building a p2p service could be based on existing p2p-like protocol and platform such SIP, adding up new specific p2p function and services.

In the rest of this section we describe an architecture JXTA/SIP that try to integrate the JXTA architecture with the SIP platform and functionality. The basic idea is to place the JXTA architecture on top of the SIP, replacing all JXTA protocols and services that are already provided by the SIP architecture.

JXTA originally defined six main protocols, that are:

- Peer Resolver Protocol (PRP) is the mechanism by which a peer can send a query to one or more peers, and receive responses to the query.
- Peer Discovery Protocol (PDP) is the mechanism by which a peer can advertise its own resources, and discover the resources from other peers. Every peer resource is described and published using an advertisement.
- Peer Information Protocol (PIP) is the mechanism by a which a peer may obtain status information about other peers, (such as state, uptime, traffic load, capabilities, and other information).
- Pipe Binding Protocol (PBP) is the mechanism by which a peer can establish a virtual communication channel or “pipe” between one or more peers. The PBP is used by a peer to bind two or more ends of the connection.
- Endpoint Routing Protocol (ERP) is the mechanism by which a peer can discover a route (sequence of hops) used to send a message to another peer.
- Rendezvous Protocol (RVP) is the mechanism by which peers can subscribe or be a subscriber to a propagation service. The RVP is used by the Peer Resolver Protocol and by the Pipe Binding Protocol in order to propagate messages.

These protocols lay on a common transport layer named “Endpoint Service” that is responsible for performing end-to-end messaging between two JXTA peers, using one of the underlying transport protocols, such as the JXTA TCP, TLS, or HTTP.

The Endpoint Service, together with the PRP, ERP, and PBP protocols are the building blocks that a JXTA application uses to communicate with other peers. They act as signaling and messaging infrastructure allowing communications between remote peers based on peer IDs. However SIP protocol also provides a signaling platform for communicating and establishing sessions between end–systems/peers (referred as UAs).

For this reason, the integration of the two technologies in the JXTA/SIP architecture seems to be very promising, providing twofold benefit: i) reuse of well-known and tested signaling platform without replicating functionality, and ii) integration with existing multimedia/IPTel infrastructure and devices.

Particularly, in the proposed JXTA/SIP architecture the upper JXTA protocols PDP and PIP, and resolver functionality of PRP and RVP are mapped directly upon the SIP protocol and its signaling platform. In such way, all transport and messaging infrastructure offered by the Endpoint Service, by PRP, ERP, and PBP is no more required and is replaced by SIP.

The JXTA/SIP protocol architecture is depicted in Figure 2.

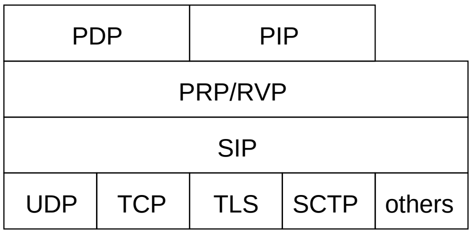

Figure 2 - JXTA/SIP protocol architecture.

The SIP protocol provides all addressing, messaging and routing functionality, reusing different types of underling transport protocols (e.g. UDP, TCP, TLS, SCTP).

The addressing scheme of SIP allow the unique identification of UAs and SIP users. However in a p2p environments can be also useful to be able to generically address single resources, or services. For this reason JXTA/SIP extends the traditional SIP URI scheme, by providing a simple and flexible mechanism to address any kind of resource owned by or associated to a generic SIP user. The SIP URI scheme

```
sip:[user@]nameaddress[:port][;uri-params]
```

is still used, and the new parameter `resource` is introduced, obtaining the following resource address

```
sip:user@nameaddress;resource=resource-name
```

where the value of the parameter `resource` (i.e. the *resouce-name*) is the peer URN (Uniform Resource Names) used in JXTA for identifying peer entities.

SIP provides to JXTA also the method for delivering XML-based JXTA messages, encapsulated within the body filed of SIP messages, with Content-Type `application/xml-jxta`.

SIP methods used to deliver JXTA messages can vary according to the implemented upper layer service. However the most common SIP methods used with JXTA are: MESSAGE, SUBSCRIBE and NOTIFY. Here is an example of SIP message used for sending a JXTA Discovery Query:

```
MESSAGE SIP:rdzv@zoolu.org;resource=jxta://abdcf23 SIP/2.0
Via: SIP/2.0/TCP 192.168.5.2;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: SIP:peer@neverland.net;resource=jxta://678ffr
To: SIP: rdzv@ zoolu.org;resource=jxta://abdcf23
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Type: application/xml-jxta
Content-Length: 340
Resolver query
   Discovery query
```

Upon SIP, there are the PRP/RVP layer that comprises the various JXTA resolver and propagation related protocols such as PRP and RVP.

PRP (Peer Resolver Protocol) is a simple query/response protocol that provides a simple addressing resolution mechanism for upper layer protocols. It can be used in point-to-point, multicast, and propagation mode. It uses the SIP MESSAGE methods for message delivery.

Although SIP provides direct peer addressing and routing, message propagation within a peer group is also required in a p2p scenario. For this scope the RVP (Rendezvous Protocol) is used within the PRP/RVP layer. The RVP provides mechanisms which enable propagation of messages to be performed in a controlled way. To most efficiently propagate messages some peers acts as Rendezvous Peer cooperating with other Rendezvous Peers and with client peers to propagate messages amongst the peers of a peer group. The PRP/RVP layer will propagate a message unless of the following conditions are detected:

i) Loop detected: if a propagated messages has already been processed on a peer, it is discarded;

ii) Time To Live (TTL) exceeded: propagated messages are associated with a TTL that is decreased each time a propagated message is received on a peer; when the TTL of a message drops to zero, the message is discarded;

iii) Message duplication: when peer receives a copy of an already received message, it discards the message.

Upon the PRP/RVP layer there are the JXTA upper level protocols, mainly the PDP (Peer Discovery Protocol) and the PIP (Peer Information Protocol).

PDP is used to discover any published peer resource. Resources are represented as advertisements. A resource can be a peer, a peer-group, a pipe, a module, or any resource that has an advertisement. Each resource must be represented by an advertisement. The PDP enables a peer to find advertisements in its group. Custom discovery services may choose to leverage PDP.

Once a peer is located, its capabilities and status may be queried. PIP provides a set of messages to obtain a peer status information. PIP is an optional protocol. Peers are not required to respond to PIP requests.

V. An example of p2p-SIP file-sharing system

In order to provide a comprehensive idea of how SIP could be used to build a full fledged p2p system, in this section we briefly discuss how to implement a very small portion of a file sharing system directly on top of SIP. Especially, we concentrate in the procedures required for discovering a set of working nodes in order to join the overlay.

Basically, p2p file-sharing systems are composed by two main functionalities: file transfer functionalities and overlay related functionalities. The file transfer functionalities are somewhat independent of the underlying p2p layer; basically when an endpoint is found the transfer phase can be performed straightforwardly. The overlay related functionalities are related to keeping coherent the overlay. For what concerns a file sharing application, there are two critical operations:

- Join/Leave operations: mandatory in order to gain access to the file sharing network and to leave.

- Search operations: mandatory in order to locate a desired resource.

One of the key problems of p2p infrastructure is locating the first node in order to join the p2p network. This operation is known in literature as the “*first node problem*”. There are several techniques to cope with this problem: a thorough investigation is not in the scope of this work, but we briefly discuss the one adopted here.

The solution adopted here is called the “peer cache” and it used by the WinMX [12] file sharing system. Basically, there is a well-known machine that stores a certain amount of peers who recently joined the network and supposed to be active. This entity is crucial, because it must be safe and always available to allow other peers join. Besides, if the peer cache goes off-line, the p2p infrastructure could continue to work properly, but the join of new peers will be impeded.

Figure 3 depicts a simple file sharing network for retrieving a well known node in order to join the network.

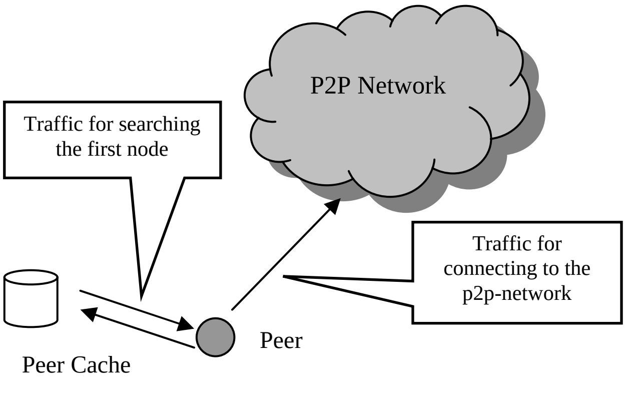
P2P Network
Traffic for searching the first node
Traffic for connecting to the p2p-network
Peer
Peer Cache

Figure 3 - Signaling traffic for solving the first node problem and joining the network.

When the new peer (e.g. sip:alice@wonderland.net) wants to attach to the p2p network and discover other peer nodes it first send a SUBSCRIBE request to a known Peer Cache with package “p2p-join”. Then the Peer Cache responds with a 200 OK and then sends a NOTIFY with the updated list of known active nodes/peers. At this point the new peer can directly contacts any of such peers to join them. Figure 4 shows the corresponding signaling flows, while above there are the details of such messages.

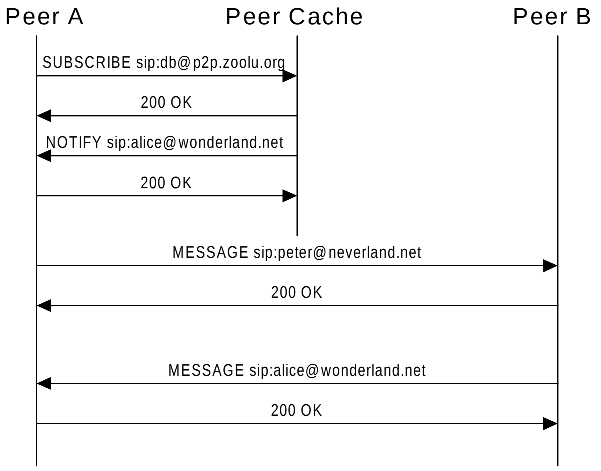
Peer A
Peer Cache
Peer B
SUBSCRIBE sip:db@p2p.zoolu.org
200 OK
NOTIFY sip:alice@wonderland.net
200 OK
MESSAGE sip:peter@neverland.net
200 OK
MESSAGE sip:alice@wonderland.net
200 OK

Figure 4 - Message flow.

```
SUBSCRIBE sip:users-db@p2p.zoolu.org SIP/2.0
Via: SIP/2.0/UDP 10.10.0.5;branch=z9hG4bK85c2f
Max-Forwards: 70
To: <users-db@p2p.zoolu.org>
From: Alice <sip:alice@wonderland.net>;tag=3758f51b
Call-ID: k3l43id034@10.10.0.5
CSeq: 22 SUBSCRIBE
Contact: <sip:alice@wonderland.net>
Event: p2p-join
Content-Length: 0
```

```
NOTIFY sip:alice@wonderland.net SIP/2.0
Via: SIP/2.0/UDP p2p.zoolu.org;branch=z9hG4bKb1454
Max-Forwards: 70
To: Alice <sip:alice@wonderland.net>;tag=3758f51b
From: <sip:users-db@p2p.zoolu.org>;tag=763287060
Call-ID: k3l43id034@10.10.0.5
CSeq: 34321 NOTIFY
Event: p2p-join
Content-Type: application/p2p-info+xml
Content-Length: ...

<p2p-info version="0" entity="sip:users-db@p2p.zoolu.org">
  <user uri="sip:alice@wonderland.net">
    <activity-status>active</activity-status>
  </user>
  <user uri="sip:peter@neverland.net">
    <activity-status>active</activity-status>
  </user>
  ...
</p2p-info>
```

```
MESSAGE sip:peter@neverland.net SIP/2.0
Via: SIP/2.0/UDP 10.10.0.5;branch=z9hG4bKdf334
Max-Forwards: 70
To: Peter <sip:peter@neverland.net>
From: Alice <sip:alice@wonderland.net>;tag=1166ccae
Call-ID: asd88asd77a@10.10.0.5
CSeq: 1 MESSAGE
Content-Type: application/p2p-info+xml
Content-Length: ...

<p2p-info version="0" entity="sip:alice@wonderland.net">
  <join-query>
  ...
  </join-query>
</p2p-info>
```

```
MESSAGE sip:peter@neverland.net SIP/2.0
Via: SIP/2.0/UDP 192.168.0.44;branch=z9hG4bKfd226
Max-Forwards: 70
To: Alice <sip:alice@wonderland.net>
From: Peter <sip:peter@neverland.net>;tag=99a52030
Call-ID: 997dfghj@192.168.0.44
CSeq: 1 MESSAGE
Content-Type: application/p2p-info+xml
Content-Length: ...

<p2p-info version="0" entity="sip:peter@neverland.net">
  <join-response>
  ...
  </join-response>
</p2p-info>
```

VI. Conclusions and future work

In this paper we illustrated how SIP could be used for building p2p services and how the p2p philosophy could be used to empower the a classical SIP infrastructure. Then we proposed some hints to lighten a widely adopted p2p technology, such as JXTA. Also, a simple example of how to use SIP for implement a little part of the WinMX protocol has been shown. Future work will be aimed in continuing to analyze the joint use of p2p and SIP and to find out if interesting mechanisms like the Host Identity Protocol (HIP) [13] will be useful for building the next generation of standard-based p2p systems.

Acknowledgements

Thanks are due to Stefano Busanelli, for precious suggestions about JXTA and SIP interworking.

References

[1] Subhabrata Sen, Jia Wang “Analysing Peer-To-Peer Traffic Across Large Networks”, IEEE/ACM TRANSACTIONS ON NETWORKING, April 2004.

[2] T. Karagiannis, A. Broido, N. Brownlee, kc claffy, and M. Faloutsos, “Is p2p dying or just hiding?”, IEEE Globecom 2004, Global Internet and Next Generation Networks, Dallas, November, 2004.

[3] Skype P2P Telephony System. http://www.skype.com

[4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[5] Gnutella Protocol Specification, http://www.gnutella.com.

[6] Stoica, I., Morris, R., Karger, D., Kaashoek, M. and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", SIGCOMM '01, August 2001.

[7] Kademlia: A Peer-to-peer Information System Based on the XOR Metric, http://kademlia.scs.cs.nyu.edu/.

[8] The p2p-SIP project at Columbia University, http://www1.cs.columbia.edu/~kns10/research/p2p-sip/

[9] eMule project, http://www.emule-project.net.

[10] A. Johnston, “SIP, P2P, and Internet Communications”, draft-johnston-sipping-p2p-ipcom-00, January, 2005.

[11] D. Bryan, C. Jennings, “A P2P Approach to SIP Registration”, draft-bryan-sipping-p2p-00, January 2005.

[12] WinMX, http://www.winmx.com

[13] R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, “Host Identity Protocol“, February, 2005.

[14] ”JXTA v2.0 Protocols Specification”, http://www.jxta.org

[15] M. Duigou, “JXTA v2.0 Protocols Specification”, IETF Internet Draft <draft-duigou-jxta-protocols-04>, May 6, 2004.

Luca Caviglione holds a Laurea degree in Telecommunication Engineering from the University of Genoa, Italy. He works in Italy both for CNIT (Italian National Consortium for Telecommunications) and for (DIST) - Department of Communications, Computer and Systems Science, University of Genoa, where he is currently a Ph.D. Student. His main interests are IP-networks, p2p-networking, GRID computing and Unix Systems. Luca Caviglione participated in several research projects funded by the EU, by ESA and by the Italian Ministry of Research. He is author and co-author of several academic publications about TCP/IP networking and satellite systems, p2p-based algorithms and QoS architectures. He is a member of the Italian IPv6 Task Force.

Luca Veltri received his Laurea degree in Telecommunication Engineering from University of Rome "La Sapienza" in 1994. He received the Ph.D. in Communication and Computer Science from the same University in 1999. From 1999 to 2002 he worked for CoRiTeL, a research institute on telecommunications that joins Ericsson Lab Italy and universities. From November 2002 he is assistant professor at the University of Parma, where he teaches classes in Telecommunication Networks, Network security, and Multimedia communications. He has participated in several research projects founded by the EU, by ESA and by the Italian Ministry of Research. His current research interests include IP telephony, P2P systems, Security, and Mobility management in next generation networks.