

Peer-to-Peer audio conferencing using SIP: P2P Architecture

Max Weltz

May 11, 2007

Abstract

This report is part of a collection of three reports describing several aspects of Peer-to-Peer audio conferencing using SIP: security, Quality of Service and P2P architecture. This particular report will detail the P2P architecture that can be used for an audio conferencing solution. Here will be described the differences between a pure SIP solution and a P2P/SIP solution, in terms of architecture and messages. We will see what issues are solved by that system and what other issues are raised and how they can be circumvented.

Contents

1	Introduction	1
2	Peer-to-Peer Architecture	1
2.1	Introduction	1
2.2	Preliminaries on P2P	2
2.3	Audioconferencing basis	2
2.4	Existing works on P2P/SIP solutions	3
2.4.1	Objectives	3
2.4.2	Characteristics of a P2P/SIP solution	4
2.4.3	SIP using DHT	4
2.4.4	Performances	8
2.5	Alternatives	9
2.6	Other aspects of a P2P solution	10
3	Conclusion	10
	References	11

1 Introduction

Peer-to-Peer (P2P) is one of the current trends on the Internet with up to 90% [San05] of the traffic share, depending on countries and sources. P2P is praised for its scalability and as the size of the communication networks increases - for instance Skype has 80 million users [SJ06] and more than 8 million users online at peak hours [Sky07], clearly demonstrating the capacities of P2P infrastructures - it is with no surprise that SIP developers have turned towards this new solution, whether the P2PSIP working group¹ or the Damaka² company. In section 2 we will see what attempts were made to adapt the SIP protocol to be fully functional in a P2P audio conferencing context and what is left to be done in that area.

2 Peer-to-Peer Architecture

2.1 Introduction

This whole section is devoted to the integration of P2P in SIP solutions. We will first discuss briefly P2P principles and audioconferencing basics. We will then have a closer look at existing works on P2P/SIP looking at the solutions they offer and the choices they made. We will wrap up with a quick look at alternatives solutions and name a few other aspects that deserve to be mentioned when it comes to P2P/SIP solutions. We will not give exhaustive definitions of SIP or P2P in that part as it is assumed they are already in the mind of the readers. In case the reader needs or wants to go deeper into the subject, RFC 2543 and RFC 3261 relates to SIP and Stoica et al. in [MKL+04] provide a comprehensive guide to P2P.

¹<http://www.p2psip.org/>

²<http://damaka.com/>

2.2 Preliminaries on P2P

Peer-to-Peer has proven over the past years its qualities in terms of scalability, reliability and flexibility as it can be used to address various resources such as files or audio communication and be used on multiple OS and platforms. The clear success of Skype, which is partially based on the P2P technology, is particularly interesting in the context of this paper. However we cannot stop there and be satisfied of the success of a P2P VoIP solution as Skype does not use SIP and therefore will be of no further interest us. P2P provides an alternative to the usual client-server model on the edge of Internet. Therefore P2P fits well into the SIP model as SIP itself delegates a great deal to end points even though SIP communications often go through proxies and servers. In fact, except for resource location, SIP is end-to-end and the communication that results afterwards is also end-to-end [BL07], so why not make it all, end-to-end, P2P? Numerous benefits would result including the possibility to allow SIP-communications on ad-hoc wireless networks, and more generally in the cases where accessing to distant servers is not possible or not preferable [BLJ05].

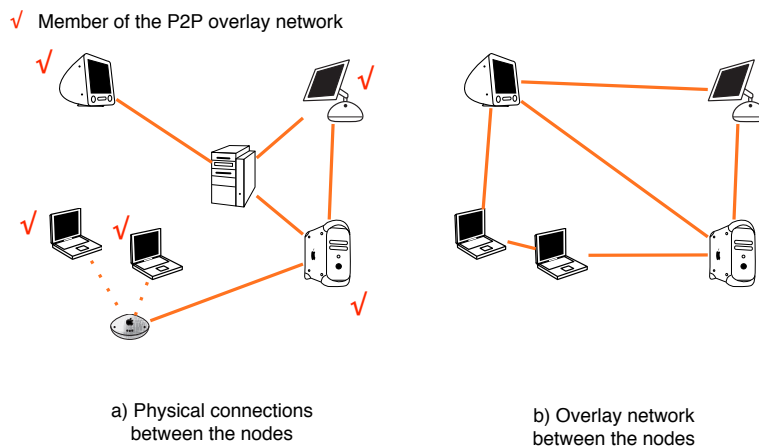


Figure 1: Difference between physical and overlay networks

Peer-to-Peer networks are based on overlay networks, that's to say virtual networks of devices not necessarily matching the lower-level connections between them (cf Figure 1). This overlay network can be of two types: unstructured or structured. Only the latter will be of interest for the rest of this study but we need to explain both to have a clear view of the subject. An unstructured network is built when new nodes are placed anarchically into the pre-existing overlay network and when there is no better way to contact a node than to flood the overlay network. On the other hand, a structured overlay network has a specific handling of new nodes joining or leaving the network and provides an easy access to resources, often via the use of a Distributed Hash Table. We will discuss the use of DHT later on.

2.3 Audioconferencing basis

Our objective is to propose a solution viable for an audioconferencing purpose. We should first describe what are the specificities of audioconferencing. SIP audioconferencing should allow users to lead a chatting session with audio support with more than two parties involved, a party being a SIP terminal.

As we are going for a P2P solution for the SIP network, it would be logical to go for a P2P solution during the actual conference, either with a completely connected network or with a ring-architecture. The problem of a ring architecture is that as we will see later it can take several hundreds milliseconds to go from one node to another when passing through a number of other nodes, which would cause too much latency during a speech conversation. A completely connected network would imply many connections established and a waste of resource. Therefore, the conferencing requires a local centralized solution.

Singh and Schulzrinne propose a good solution in [SS04]. The conferencing nodes should elect a central node who will do the mixing, resulting in a star-shaped pattern. Singh and Schulzrinne would go for the node with the best bandwidth and CPU capacity but that is not enough as it does not address the problem of codecs and that of dynamic conferences with members joining and leaving during the actual conference. The chosen node definitely has to have the largest range of codecs (or at least each other participant should have one codec in common with the central node) and to be the most likely to stay in the conference from the beginning to the end (probably the instigator of the conference). The node with the largest range of codecs is likely to be a computer (as opposed to a SIP phone or a PDA) and dispose of a larger computational power, however a SIP phone might have been designed from scratch to enable conferencing and to act as a mixer. Another issue to keep in mind in this model is the question of privacy, which limits the choice to nodes involved in the conference. The determination of the central node is thus an open problem and the result will most likely depend on assumptions made on the network usage and users.

Another solution could be to use multicast but that could lead to flooding the network, at least the part of the overlay network involved in the conference. In that case, all nodes have to do the mixing. Because of these two drawbacks this solution cannot apply to conferences with many speakers debating at the same time. The performances can be improved depending on the properties of the overlay network. For instance, some P2P overlay networks, like Pastry, provide proximity informations that can be used to smarten the multicast.

2.4 Existing works on P2P/SIP solutions

2.4.1 Objectives

From the works of Bryan, Lowekamp and Jennings in [BLJ05] and Kundan Singh and Henning Schulzrinne in [SS04], we came with a precise list of objectives to meet for a good P2P/SIP solution:³

1. **Simple lookup:** one issue with P2P systems is the problem of bootstrapping, or more clearly, how to join a P2P network without any a priori knowledge about its members, and namely, their addresses?
2. **Privacy:** often true when in the communication world, users might want their privacy, maybe by connecting only to a subnetwork only made up of trusted users.
3. **Mobility:** with a wider use of P2P/SIP solutions for VoIP, users are likely to wish for mobility, whether it is service mobility, personal mobility, session mobility or terminal mobility as defined in [SJ06].
4. **Portability:** along with mobility raises the issue of portability: users might want to keep the same solution working, whether they are using their laptop or their PDA.

³We do not list here previously acknowledged advantages of P2P systems.

5. **Compatibility:** SIP products and solutions are already widely spread and it would be confusing and damageable to hinder the communications between P2P and non-P2P SIP products. Moreover compatibility with former solutions is likely to lead to a larger reuse of existing code-snippets
6. **Easy configuration:** which is not dependent of the P2P/SIP choice but must be kept in mind as nowadays people are used to plug and play solutions when it comes to P2P and communications, which means the developer is in charge of the inherent issues of NAT transversal, among others.

2.4.2 Characteristics of a P2P/SIP solution

Compatible characteristics

SIP in itself is not causing many problems for integrating it into a P2P solution. Namely, intermediate servers are optional and the SIP protocol allows user agents (UA) to communicate with each other with no other party involved. This means that the messages inherent at a P2P protocol can be transported in SIP messages (cf. 2.4.3).

Modification needs

Presence Not only should we be aware of what does or does not change inside SIP's core aspects, but also on services often associated with it, such as presence. The classical presence management over SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions) requires all UA to update their presence information on a remote centralized server. A system based on event packages has been proposed in RFC 3265 to deal with presence between nodes, using no external server but this solution might not be implemented in all existing SIP products and so might not be suitable for all SIP devices and applications already on the market.

Voicemail Another problematic aspect is that of voicemail when the callee's machine is not connected (if the callee is connected but not answering her machine can store the resulting file). While voicemail can be stored, and is actually stored, on the centralized server in most SIP implementations, there is no such server in our P2P/SIP model and adding one would break our efforts to bring SIP to a fully P2P modus operandi. Instead, we have to designate a node, but more likely a set of nodes to improve chances that the voicemail message will be available when the node reconnect⁴, on the overlay network to store the file containing the voicemail message. Encrypting the message could guarantee privacy using the callee's public key and integrity using the caller's private key. At this point, we have to further explain DHT to determine what node should stock the voicemail file.

⁴This problem is close to that of data replication on MANETs, except that all messages eventually need to be stored only by their legitimate recipient. Hayashi, Hara and Nishio in [HHN05] give good ideas on the subject.

2.4.3 SIP using DHT

Principles of a DHT architecture

Many implementations of DHT are available, Chord⁵ and Bamboo⁶ being the main ones, but they all rely on the same model that we are to explain here. As the name indicates, DHT makes use of a hash function, associating a variable-length string (a SIP AoR in our case) representing one resource of the network (a SIP user) to a unique fixed-length identifier. To be more precise, a good hash function is one that minimizes risks of collision, therefore the identifier is not unique but has great probabilities to be so. Some P2P networks are hierarchical in that they distinguish super-nodes from regular nodes. Those super-nodes exchange messages to manage the P2P network and act often like an entry point in the network. Those super-nodes make up a super-overlay network. They are not fixed, nor do they have to belong to the institution responsible for the P2P network and technology used.

Every UA being associated with an identifier, all UAs are organized as a ring network, ordered by ascending identifiers. This addressed the problem of identification so we now have to face the problem of destination lookup. Each node knows the location to its predecessor and successor in the ring. Furthermore each node stores the location of a small amount of other nodes, along with their identifiers. That amount is generally $O(\log(n))$, which tops the lookup time at $O(\log(n))$ [SS04] (see Figure 2 a). The case of audioconferencing narrows the assumptions on the network so we might decide to optimize the choice of the nodes, whose locations are stored. Namely, a node can store the locations of a few nodes from the overlay network and fetch preferentially the location of nodes present in the user's buddy list.

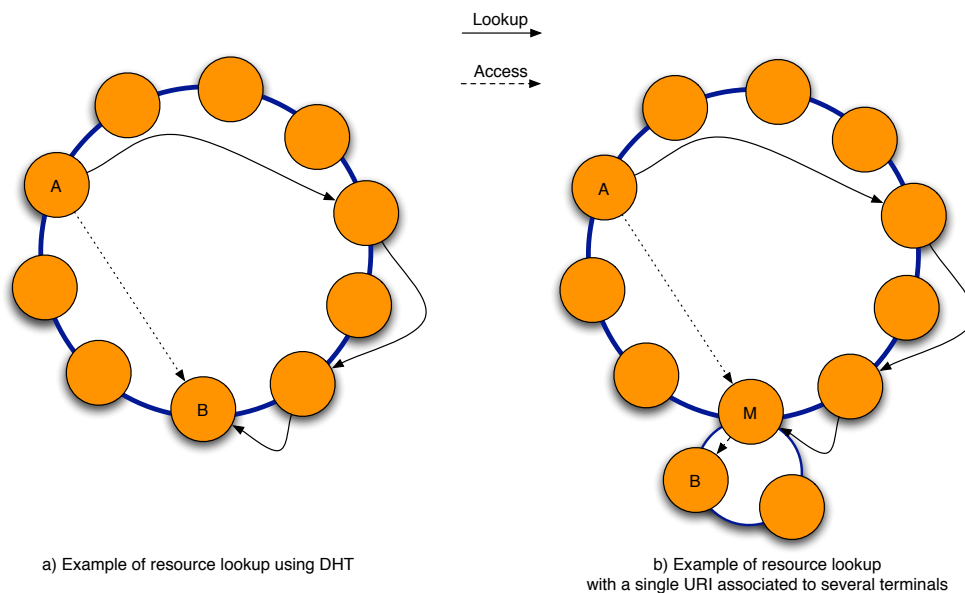


Figure 2: Example of resource lookups using DHT

⁵<http://pdos.csail.mit.edu/chord/>

⁶<http://bamboo-dht.org/>

This leaves the problem of several devices associated to a single user. This issue is also logically noticed in [SS04], as the choices we made above are similar to those of Singh and Schulzrinne. Here is a suggestion we make to deal with that issue.

To locate a node on the network, one needs to know to what value apply the hash function. On one hand, if we consider applying the hash value to the SIP AoR, then how do we know all URIs associated to that AoR have been alerted? On the other hand, if we decide to apply the hash value to the IP or phone number of a particular device, we then need to know it beforehand, which would then make any sort of lookup useless. A solution would be to have all devices associated to the same AoR elect a representing node and let the other nodes pick different hashes known by the *master* node M. This is no longer strict, flatly structured P2P (see Figure 2 b). Bryan, Lowekamp and Jennings in [BLJ05] propose to use the IP address as input value for the hash function but in that case we need to maintain a parallel mapping between IP addresses and AoRs.

SIP exchanges in the DHT architecture Singh and Schulzrinne in [SS04] develop explanations on appropriate SIP exchanges using their P2P model that we will detail and comment here. We also would like to remind that the use of a P2P overlay network does not mean that the same SIP client cannot use a SIP only network based on servers and proxies, especially since it was decided in this model to carry the P2P messages using SIP.

Node Joining For a node to join the network, two phases are to be observed:

1. *Network discovery*: to be able to join the network, the node has to find an entry point, quite often this means a super-node⁷. This can be achieved using the Service Location Protocol (SLP) or by flooding the local network for super-nodes or nodes that can provide useful information on their location. Otherwise, the cache of the joining node can be used for super-nodes lookup. In the worst case scenario, there can be a few fixed super-nodes, for instance super-nodes belonging to the protocol's creators that might be used for statistical purposes.
2. *Node registration*:⁸ suppose Alice wants to join the network. She will send:

```
REGISTER sip:123.234.1.1:5060 SIP/2.0
To:  alice@atlanta.com
From:  alice@atlanta.com
```

where `sip:123.234.1.1:5060` is the address of the super-node Alice is contacting. The super-node can now calculate the hash of Alice's AoR and register the new node in the DHT table. The P2P network will handle the registration following its own fashion. Namely, the P2P network still has to inform Alice she has been accepted and who and where are her neighbors. Alice is also still lacking the addresses of the above-mentioned nodes used to reduce the lookup time. All this depends on the P2P network solution chosen. In the case of Chord, informations on successors and predecessors are obtained by looking-up for yourself on the network. This message is also used in [SS04] to make sure that the connection is alive between the node and its supernode.

⁷Singh and Schulzrinne consider a Chord network and the use of super-nodes. A regular Chord network is flat and does not include super-nodes. This means we have to consider adding a super-Chord network made of super-nodes only as in [JW06] or replace Chord by another system.

⁸We will only discuss the registration of the node on the P2P network here.

More evolved scenarios can be considered when a node is joining the network. For instance, super-nodes may have been designed to split to the task of incoming nodes' management depending on areas, functionalities or client versions. In that case a REGISTER message might be forwarded between nodes. Similarly, for one reason or another, a REGISTER message might be received by a node that is not a super-node (because it has lost this status recently for instance) and in that case again, the REGISTER message might be forwarded. The forwarding may be done by a REFER message. Alan B. Johnston and Henry Sinnreich claims in [JS06] that this solution is not suitable as some existing solutions might not tolerate the forwarding of REGISTER messages. More generally they are opposed to the transport of P2P message on SIP for reasons of performances (cf. paragraph 2.4.4).

Node quitting A node quitting abnormally will be detected by the process above, but a node quitting normally has to unregister from the P2P overlay network. This can be achieved by sending a REGISTER message with an expiry time set at zero. However due to their specific roles, super-nodes cannot just leave on an unregistering message.

In the specific case of a super-node leaving the network, it has to communicate with its fellow super-nodes on the super-network the information concerning the nodes it was in charge of. This can be achieved by the mean of REGISTER messages exchanged between the two super-nodes. The authors of [SS04] consider that nodes will notice the departure of their super-node during the next refresh and then look for the new super-node responsible for them. The resulting delay can be avoided if the leaving super-node warns the nodes it is in charge of with a REFER message for instance.

This question of warning other nodes leads us to discuss the case of node failure. The failure of a super-node matters especially and in such a case the remaining super-nodes will notice it and redistribute among themselves the dropped nodes. Once again, the dropped nodes might be notified of the change by the mean of a REFER message or wait until they realize the change, like in [SS04].

As we will see later, if some information is to be stored on the network, the failure of any node can become a serious issue. When online storage of information is considered, a disconnecting node has to pass on its data to other nodes before the actual disconnection to allow the information to remain available afterwards. This can be a problem and cause a long time between the request for disconnection and the actual disconnection.

Interaction between two nodes Here comes the aspect that interest the users the most: opening sessions with another node. Just like in non-P2P SIP, it consists of two parts: 1) finding the requested node, 2) establishing a session with that node. Once again, using a P2P/SIP solution does not mean that this solution is isolated from the pure SIP world, for instance, instead of the P2P lookup phase we are going to describe further down, the node might successfully use a traditional SIP lookup.

1. **Node lookup:** The initiating node sends an INVITE message to its super-node that will forward it to the called party. The super-node might have to act as a proxy in specific cases, for NAT-transversal for instance. The lookup is done by the super-node using the DHT mechanism. Note that a node might also be a super-node and in that case it can issue directly the INVITE message to the called node.
2. **Session:** After the lookup is done, the SIP exchange can go on like a normal SIP exchange, either including the super-node as a proxy or as an end-to-end exchange.

Voicemail storage As we discussed before, the voicemail message has to be stored as long as the node concerned by the message has not fetched it. The most intuitive way is probably to assign that message to the super-node that should be in charge of the recipient node. "Be in charge" can either mean store it and send it to the recipient node when it connects or manage the replication of the message on the overlay network and alert the recipient node it has a message once it is connected. It is possible that at some point the super-node disconnects, in that case it has to pass its duty to another super node. Once that the message has been delivered, the question of the confirmation creates another issue. If the sender is online, an acknowledging message can be sent and if it is not online, a system similar to the one described above can be used.

Device independence and presence Using a P2p network implies a greater facility to connect to the P2P so it is likely that users might roam and connect from distant places, not having their main device with them. Therefore some settings, and especially the contact list, can be stored on the network, just like for a voicemail message. Once again, privacy and integrity can be achieved by the mean of encryption as nowadays most people are used to entering passwords when they connect to an online service. The main problem remains availability in that model as there is no guarantee any replicates of the data will be online at any moment, especially if a node has a failure and cannot pass on the information to another node before disconnecting.

Once the issue of contact list retrieval is addressed, fetching presence information can be done directly by requesting it from the concerned nodes, via the super-node. The super-node could store presence information for all its associated nodes but it is preferable not to add to the burden of super-nodes something that is not of primary priority.

More on SIP messages In [Bry05], David A. Bryan gives some smart hints on how to use existing headers and parameters of SIP messages to convey P2P information:

- **Supported/Require:** respectively to specify the ability of a node to handle P2P interaction and if the current message is of that kind
- **alg:**⁹ to convey information on the hash algorithm used by the DHT
- **user:**⁹ to specify if the message is relative to a user-to-user SIP exchange or to a P2P message over SIP

Obviously those are just suggestions and P2P/SIP solutions are likely to use them differently even though an overall agreement would be an important towards unification, which is the current drawback of all widespread IM and voice solutions.

2.4.4 Performances

Singh and Schulzrinne in [SS05] provide "performance predictions" using the Chord DHT system based on the work of Stoica et al. in [SMLN⁺03] and claims the following:

- with weak assumptions on the network and the nodes and strong assumptions on the use of the network¹⁰, the maximum number of nodes on the network is of 2^{300} ,

⁹Used inside a To, From or Contact header

¹⁰10 requests per second per node, one call per minute per node

- considering a network of 10,000 nodes using chord, six hops is the mean lookup path length, taking approximately 200 ms.

This latter assumption of 10,000 nodes is pretty weak compared to a network like Skype with 8 million users online [Sky07]. Based on the results found in [SMLN⁺03], such a network would require a mean lookup path length of 10 hops, or approximately 350 ms. Compared with the actual values observed on the Skype network (three to eight seconds according to Baset and Schulzrinne in [BS04]), this probably means the assumptions made are not realistic for such a wide network. With optimization those values could be lowered, especially considering the large bandwidth of current networks, one hop lookup is conceivable [GLR03] and even if not all nodes¹¹ do not have that capacity, the delay would still be greatly reduced. In any case, the latency in a P2P network is longer than that of a classic centralized network.

2.5 Alternatives

Alternatives in the domain of P2P/SIP The different papers studied above list several alternatives to P2P/SIP, each with their drawbacks and advantages. Other solutions have been designed that were not discussed above:

- Damaka¹² is offering a P2P/SIP client as a non-open source software.
- In [SBC06], Stiemerling, Brunner and Cassini discuss the use of Service-Aware Transport Overlay (SATO), a technology relative to ambient networks defining another way to handle overlay networks and DHTs.

Alternatives to P2P/SIP Other projects exists that do not use SIP, or P2P or non of them, but that can still be applied for an audio conferencing solution:

- Skype, which is a commercial, proprietary, non-disclosed solution but widely in use. The model is not strictly P2P and includes central servers for authentication and charging. Skype also adds IM support and NAT transversal functionalities and more generally enjoys a larger support thanks to its commercial nature.
- vop2p, a currently inactive project, based on JXTA, a Java-based P2P technology which is however listed as inactive on the JXTA projects' list¹³. Such a project could have been promising as JXTA provides service discovery functionalities, among others.
- Solutions using a Dynamic DNS are also considered as a solution mixing the P2P and server-based approaches. The use of a REGISTER message can then be avoided by the using of Dynamic DNS to resolve SIP URIs. Another solution is to have servers joining and leaving dynamically and still use the Dynamic DNS resolving to access them.

¹¹Namely mobile devices might not have all 384 kbps (in the case of a network with 10⁶ nodes) of bandwidth to give away.

¹²We contacted Damaka to have further information on their system but they did not reply.

¹³<http://vop2p.jxta.org/>

2.6 Other aspects of a P2P solution

The use of P2P in SIP solutions sharpens some pre-existing problems that are that of security and privacy, of NAT transversal and QoS. When non-trusted machines are involved, users get worried about their personal informations. Finally, a larger public often means more chances to encounter NAT on the path of the message and also means great constraints in term of quality of service, or at least a sufficient quality/price ratio. This latter issue is addressed in [\[Vio07\]](#) and the security issue is addressed in [\[Mac07\]](#).

3 Conclusion

Many solutions are actually studied to implement SIP communication on a P2P network. Some like the one we presented here send P2P-related messages using SIP, some do not. We also tried to discuss some issues that were not, or not sufficiently, discussed by Singh and Schulzrinne in [\[SS04\]](#) on which we based our report. The darkest spot for the future of P2P/SIP solutions seem to be the availability of some information, such as contact list and voicemail messages, inherent to pure P2P systems. One solution could be to have hybrids systems using private nodes and super-nodes but including a few servers ran by the company promoting the solution, or possibly some service providers. Nevertheless we feel confident that a P2P/SIP solution for voice and audio conferencing is not impossible to deploy at a large scale and for a common public use.

References

- [BL07] David A. Bryan and Bruce B. Lowekamp. Decentralizing sip. *ACM Queue*, March 2007.
- [BLJ05] David A. Bryan, Bruce B. Lowekamp, and Cullen Jennings. SOSIMPLE: A serverless, standards-based, P2P SIP communication system. Technical report, College of William and Mary Williamsburg, Vancouver, 2005.
- [Bry05] David A. Bryan. Peer-to-peer SIP. 2005.
- [BS04] Salman A. Baset and Henning Schulzrinne. An analysis of the Skype peer-to-peer internet telephony protocol. Technical report, Columbia University, September 2004.
- [GLR03] Anjali Gupta, Barbara Liskov, and Rodrigo Rodrigues. One hop lookups for peer-to-peer overlays. Technical report, MIT, 2003.
- [HHN05] Hideki Hayashi, Takahiro Hara, and Shojiro Nishio. A replica allocation method adapting to topology changes in ad hoc networks. Technical report, Osaka University, 2005.
- [JS06] Alan B. Johnston and Henry Sinnreich. SIP, P2P, and Internet Communications. Technical report, IETF, SIPING Working Group, 2006.
- [JW06] Yuh-Jzer Joung and Jiaw-Chang Wang. Chord²: A two-layer Chord for reducing maintenance overhead via heterogeneity. Technical report, National Taiwan University, 2006.
- [Mac07] Luca Maccari. Peer-to-peer audio conferencing using SIP: Security issues. Technical report, Kungliga Tekniska Högskolan, 2007.
- [MKL⁺04] Dejan S. Milojevic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer-to-peer computing. Technical report, HP, 2004.
- [San05] Sandvine. EDonkey - still king of P2P in France and Germany. Technical report, Sandvine, 2005.
- [SBC06] M. Stiemerling, M. Brunner, and M. Cassini. Peer-to-peer SIP Implementation Report. Technical report, IETF, P2PSIP, 2006.
- [SJ06] Henry Sinnreich and Alan B. Johnston. *Internet Communications Using SIP - Delivering VoIP and Multimedia Services with Session Initiation Protocol*. Wiley, 2006.
- [Sky07] Skype. Skype statistics. http://share.skype.com/stats_rss.xml, 2007.
- [SMLN⁺03] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup protocol for internet applications. Technical report, IEEE, February 2003.
- [SS04] Kundan Singh and Henning Schulzrinne. Peer-to-peer internet telephony using SIP. Technical report, Columbia University, 2004.
- [SS05] Kundan Singh and Henning Schulzrinne. Peer-to-peer internet telephony using SIP. Technical report, Columbia University, 2005.
- [Vio07] Gabriele Violino. Peer-to-peer audio conferencing using SIP: QoS issues. Technical report, Kungliga Tekniska Högskolan, 2007.