P2PSIP and the IMS:
Can they complement each other?

July 2008

IMSF-008-072008

Legal notices:

About the IMS/NGN Forum

The IMS/NGN Forum is a global, non-profit industry association devoted to interoperable IP Multimedia Subsystem services delivery architecture and solutions. The IMS/NGN Forum mission is to accelerate the interoperability of IMS revenue-generating services, enabling enterprise and residential consumers to fully benefit from the delivery of multimedia mobile and fixed services over broadband cable, wireless, wireline and fiber networks. The IMS/NGN Forum is the creator and organizer of the IMS Plugfest™, the industry's only event focused on IMS service interoperability, verification and certification through the IMS Certified™ program.

Through its organized Plugfests, working group interactions and other activities, forum members develop cost-effective technical frameworks for converged IP services over wireline, cable, 3G, WiFi and WiMAX networks. For additional information, or to join the IMS/NGN Forum and the IMS Plugfests, please visit **www.IMSForum.org**.

IMS/NGN Forum Contact Information:

IMS/NGN Forum Headquarters
211 Summit Place #292
Box 10,000
Silverthorne, Colorado 80498 USA
www.IMSForum.org
Email: Info@IMSForum.org
Telephone: +1 970-262-6100

Foreword from IMS Forum Chairman and President:

We are in the midst of the convergence of Internet and broadband over cellular, WiFi, WiMAX, cable, fiber, power lines, and increased consumer expectations of enhanced services and applications. Investors world-wide accept that “content is king,” however at the end of the day, the “consumer is king.” Consumers are forcing service providers to deliver bundled services, with the right quality of service at the right price, and with reach features tied into mobility and multimedia. These expectations are the main drivers for the implementation of the IP Multimedia Subsystem (IMS) services architecture.

The IMS/NGN Forum focus ensures that IMS architecture is tested and certified through a rigorous process. We work with service providers, vendors, regulators as well as other industry groups to inform, educate and promote interoperable IMS services working across all types of broadband networks.

The IMS Forum issues two types of documents, white papers and best practices. The white papers focus upon information dissemination, education and promotion of IMS services. The best practices focus upon clarifications and methodologies for implementation of IMS applications and services

Thanks to IMS Forum members and industry partners for their contributions to this document.

To participate in IMS Forum projects, including technical working groups, the IMS interoperability, please visit **www.IMSForum.org or www.NGNForum.org.**

Thank you,

Michael Khalilian
Chairman & President
IMS Forum
Mkhalilian@IMSForum.org

Table of Contents

Executive Summary

Is peer to peer (P2P) and IMS, or P2P versus IMS? This paper, the first in a series, addressing one of the most intensely debated issues in IP Multimedia architectures, services and applications, will be followed by other papers describing in more details solutions for load balancing, scalability and services. Many are seeing IMS and P2P as two antagonistic architectures; the first has a control layer at the core, while the other doesn't. The debate over 'centralized' versus 'decentralized' is not new and the Internet didn't really change the essence of this debate. First we need to recognize that the reality is never 'pure'. Neither fully centralized nor fully decentralized models are used in real life. The paper proposes a different approach to the IMS architecture standardized by the ETSI, ITU and Cable Labs; it follows a set of proposals made in IETF for P2P SIP services, and discusses how these could be applied to the IMS architecture at the core of the IMS network.

As this paper states, "It is important to realize that P2P is essentially a different way of distributing the "load" of a system architecture, whether the load to be shared is CPU processing, resource storage, or call control. P2P systems don't eliminate the work done by a centralized device in traditional client-server architecture, but instead distribute that work differently, pushing the functionality to the edge."

This paper provides the background necessary to understand P2P and lists the advantages of the P2P methodology and demonstrates that a combination of centralized and decentralized resources can be used in creating services and applications using SIP and be extended to the IMS architecture. The paper, however, doesn't cover some of the issues related to a completely decentralized architecture, such as billing, problem segmentation for troubleshooting and help desk support.

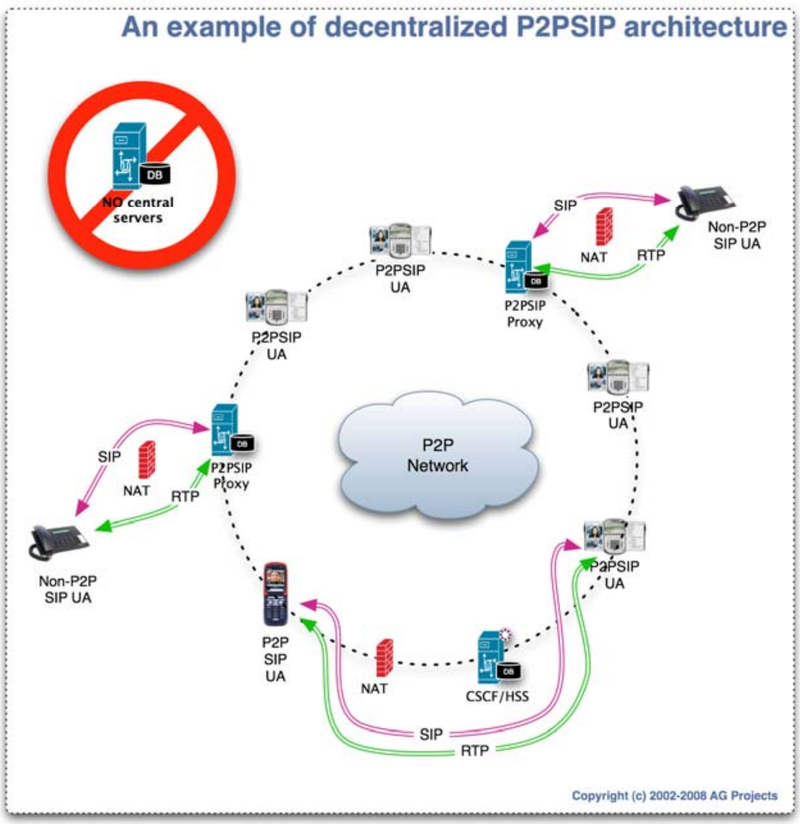
Future Work

More analysis of the relationship between P2P and IMS is necessary. The IMS/NGN Forum members are invited to contribute by discussing how the two technologies may be used by Service Providers to deliver IMS Services to a broad community of consumers and enterprise users.

Background

What is P2PSIP?

In a traditional SIP or IMS deployment, a single, or small number of tightly coupled, servers provide various services to the endpoints, including storing end-user data, performing registrations, routing calls, simple voice mail, and complex next generation multimedia services like presence or multi-party conferencing. At the heart of P2PSIP is the provision of a distributed "fabric" of devices that are able to provide these same services and functionality with little or no centralized infrastructure.



There are many types of Peer-to-Peer (P2P) Technology. P2PSIP is unique in that it provides a conceptually simple, ***standards-based*** mechanism to distribute functionality, leveraging the power of clusters of devices instead of a single centralized box. These devices work together, collectively, to replace the resources and services that would be provided by the centralized server.

As much as possible, existing technologies, protocols, and architectures are leveraged to harness the advantages of P2P, while ensuring that full advantage is taken of the development to date of standards such as SIP, DIAMETER, and ICE. It is an effort to produce a P2P framework for communications (including voice, IM, video, and others) that is compatible with today's equipment and standards. By taking this approach, existing product lines can be augmented to include P2P functionality. Existing carriers can incrementally incorporate P2P approaches, and users and enterprise administrators can be more comfortable knowing that the solution works with other equipment and is implemented in an open and transparent manner.

To date, most of the work on P2PSIP has focused on pushing as much functionality as possible all the way to the end-user devices, enabling them to collectively provide the capabilities that traditionally resides in centralized servers such as proxies, registrars, feature servers or soft-switches. The same principles can also be applied, however to the large number of servers used in an operator-controlled network. Rather than pushing functionality completely to the edge, P2P technology can be used to allow a number of servers to share basic information, therefore providing high availability, replication of information, and improved reliability.

P2P (and P2PSIP) fundamentals

It is important to realize that P2P is essentially a different way of distributing the "load" of a system architecture, whether the load to be shared is CPU processing, resource storage, or call control. P2P systems don't eliminate the work done by a centralized device in a traditional client-server architecture, but instead distribute that work differently, pushing the functionality to the edge. As a result, there are some advantages to a P2P architecture (for example, improved scalability and reliability) and some areas where a conventional CS system may be better (for example, number of hops allowed to route a message).

P2P systems exist in two basic categories: *structured and unstructured*. Unstructured systems are organized in an ad-hoc manner, typically using flood-based mechanisms to locate and store resources, and frequently with non-deterministic guarantees of search completion. In general, unstructured systems are not being discussed for anything by the smallest of systems. Structured systems use mathematical mechanisms to organize the system, and provide stronger guarantees of storage and location mechanism. In the remainder of the document, the term “P2P systems” will refer to structured P2P systems. In particular, the properties below apply best to a Distributed Hash Table, or DHT P2P approach. Chord, one of the most popular DHTs, is described in [i], and provides an excellent introduction of how a DHT works.

Every P2P system provides the following basic features and mechanisms:

Self-organization: P2PSIP systems provide a way to take a collection of peers and allow them to connect to one another in a mathematically structured way without coordination from outside. In other words, no manual provisioning or central control is required to organize the peers and determine where in the fabric they should be placed. The resulting collection of connected peers is referred to as an *overlay*. As new peers join, they query the existing peers to determine their location within the fabric. As peers are added or removed from the fabric (processes known respectively as *joining or leaving* the system), the fabric can determine how the structure changes – both in terms of how resources are stored and how the peers communicate with each other. This behavior is referred to as "self-organization".

Routing: P2P systems may contain up to several million peers. In such large systems, it isn't possible for every peer to be connected directly to every other peer. The P2P algorithm helps determine a very small subset of other peers that each peer connects to, and provides mechanisms to ensure that through this subset, every other peer can be reached. This is achieved collectively, with a number of intermediate peers being used to route messages (or provide "next hop" information) between source and destination peers. Most P2P algorithms provide a mathematical guarantee of this routing's performance. Many common systems ensure that no more than $\log_2(n)$ hops will be required to route a message, where n is the number of peers deployed in the system.

Resource Distribution: P2P systems provide a mechanism for mapping resources onto the fabric, determining particular peers (typically more than one for redundancy) that will be responsible for providing a resource (storage, service, etc.). This ensures that all peers understand and share an underlying set of assumptions of where resources will be located. Most approaches strive to make this distribution "fair", where fair may be evenly distributed, or distributed on the device most capable of providing that service, or another equitable solution.

Join and leave: The peers together perform services that are normally provided by a central device. As an example, they might store registrations for a phone system that would normally be stored by a SIP registrar. As new nodes join or leave, the P2P algorithm is responsible for ensuring that a fraction of the distributed resources stored by the existing peers (in this example, registrations) is passed to the new peer. Similarly, as a peer leaves, the algorithm will ensure that the resources that the peer is responsible for are passed to another in the system. The join and leave operations are handled automatically by the network without disruption of other resources and do not require provisioning or central coordination as in a client-server environment.

Lookup, Get and Put: When a peer wishes to locate a particular resource, the resource distribution and routing mechanisms defined by the P2P algorithm help determine which other peer is responsible for storing that resource (or providing the desired service), and to locate and communicate with that peer, even if it is one of millions in the system. Again, in most mathematical algorithms used in P2P, performance of at least $\log_2(n)$ can be guaranteed in looking up resources as well. Once the responsible peer is located, a resource can be stored or retrieved from that location, providing an efficient, scalable, and fully distributed lookup system. Note that a traditional client server lookup can usually be completed in constant time, simply by sending a message to the central server, rather than through several hops. However, note that P2P's advantages come at the cost of slightly higher lookup hop counts.

The major advantages offered by a P2P approach include:

Redundancy/Reliability: Resources or services can be located on more than one device on the fabric. Again, this is mathematically structured, so that finding these replicas can be done efficiently and deterministically. By combining this with the load balancing aspects of the system, this allows for nodes to fail, using replicas in their place, and to then propagate the replicas back to the original locations later, resulting in a high reliability system.

No Single Point of Failure: Because no single peer is storing all of the data, the failure of any single peer cannot bring the network down. There are typically several ways to route messages through the fabric, and resources are replicated on multiple peers. In fact, in many systems, large fractions of the peers can fail and the system will still function.

Ease of Configuration: For certain applications, the self-organizing aspect of P2P can enable systems to be configured more easily. For example, an ad-hoc system can be quickly set up, with no need to provision a central box in advance, for the devices joining the network.

Scalability: Since P2P systems use a fixed algorithm to determine what information each peer is responsible for, each added peer becomes responsible for storing information or providing services as well. Each new peer brings new demands on the system, but also provides additional services or storage. In this way, P2P systems have scaling properties that make them very attractive for removing servers, and as a technique for building high availability server clusters.

P2PSIP is an approach to P2P that offers the following advantages over earlier proposed or deployed mechanisms:

Standards-based: P2PSIP systems are being developed within the IETF, with an eye towards standardizing the protocol used by the peers to connect. This enables administrators, developers, and end users to interchange elements, ensure they work together, and most importantly, understand and control the traffic on their networks, which is not possible with non-standard P2P systems.

NAT traversal: P2PSIP systems are using the IETF NAT traversal suite (STUN/TURN/ICE) to provide NAT traversal functionality, which is critical for assuring high success rate of communications over any network topology connected today to the Internet.

SIP Compatibility: Because P2PSIP has been developed with SIP as the primary application, making deployment of SIP systems simple has been emphasized from the beginning. This means that P2PSIP offers a number of advantages to developers of SIP (and IMS) applications. While extensions may be added in the future, P2PSIP currently specifies no changes to the SIP messages used for call signaling - mechanisms using only SIP primitives exist for conventional SIP devices to access P2PSIP systems, including locating other devices and routing calls. Because IMS is based on SIP, these design decisions mean that integrating P2PSIP into an IMS deployment should be more straightforward and require less new code than any other currently available or proposed P2P approach. It should allow for a smoother (and more gradual) transition if an IMS vendor or operator decides to incorporate P2PSIP technology into their architecture.

Security, Authentication and Charging: P2PSIP systems have been designed to be highly secure, including mechanisms to verify the identity of users, verify that peers are authorized to join, and provide centralized authentication, if needed - all while still allowing other aspects of the system to be fully distributed. While not defining new mechanisms, the design of P2PSIP uses conventional SIP to set up and manage calls, meaning that ongoing work being done by the SIP and IMS community to allow for security, authentication and charging, can be leveraged in a P2PSIP architecture.

P2PSIP Standards-body and Community Activities

P2PSIP has been a topic of interest in the IETF for several years. The topic was first proposed to the SIPPING Working Group (WG) by David Bryan in January of 2005 [ii], and in the ensuing years, interest has snowballed. P2PSIP is now embraced by the IETF as an official WG [iii], and has met three times by the end of 2007. The primary work items the group has been chartered to address include:

Development of a "Concepts and Terminology" document [iv], which will explain what P2PSIP is, and the terms used by the WG to describe concepts. Because the topic is quite new, many new terms have been invented for concepts, and some cohesion is required among WG items. Additionally, the structure of the components of P2PSIP needs to be outlined. This document addresses these topics.
Development of a "Peer Protocol Specification", which will document the protocol spoken between the peers that make up the fabric. At the date this article was published, the form this document will take was still a topic of debate.

The P2PSIP group is very active. In several of the recent IETF meetings, it has been the most popular working group, with approximately 250 attendees. Many other topics of interest to the community, if not yet formally adopted as items the group will work on, include a client protocol specifying how non-peer devices could query the network using a method other than SIP, applications scenarios descriptions, and usage documents.

Additionally, there is a great deal of work on how to incorporate P2P into SIP and IMS networks, including several products that have been brought to market [v,vi].

The community site P2Psip.org [vii] has been active for several years. In addition to providing an easy index of the activities of the IETF group, the site hosts background material on P2PSIP and P2P in general, and links of interest to the community. In addition, a large number of academic papers related to P2PSIP exist, including two of the earliest ones [[viii],[ix]].

P2PSIP Architectures

The following are all examples of proposals for structuring P2PSIP systems:

P2P software is installed on a collection of endpoints (ranging from a few dozen to millions). Each endpoint stores a fraction of the registrations in the telephone system. A user wishing to call another user queries the collection of endpoints, eventually locating the particular phone that stores the location of the destination party. The call is then placed directly between the devices. The collection of devices in this case can completely replace the functionality of the registrar and proxy, and potentially, any feature servers in the system. In such an approach, the only "centralized" component that is required is a gateway or SIP trunking infrastructure.
A system of tens to hundreds of SIP Proxy/Registrar uses a P2P mechanism to share registration information among them. When a new registration is received, the proxy mathematically selected by the P2P algorithm stores that value, regardless of which proxy received the message. Similarly, when a query to place a call is received, the collection of proxies is queried to locate the information. A user can contact any proxy to get the answer, although any other proxy may store the information and a minimum number of proxies is involved in each call setup. Similarly, other services like Voicemail and Presence can be scaled up in a similar and cost-effective manner.

A hybrid solution, where some call control is distributed, but other aspects (perhaps billing or authorization) are centralized as in a traditional deployment.

Note that other concepts are occasionally described as "P2P" by other groups or vendors in the SIP community, but are not the same as the definition of P2P discussed by the P2PSIP WG at the IETF. Some examples include:

The basic functionality of SIP. A configuration of two SIP devices connecting directly to each other is sometimes referred to as "P2P" mode, in the sense that no central devices or servers are being used to route the call. However, this isn't a true P2P deployment because the phones are not working together to provide the functionality the central server would have provided – such a configuration does not scale beyond two devices (or as many IP addresses that can be remembered). Similarly, a system where "the media flows directly between the peers", while a central box provides registrar and proxy functionality – the standard functionality of SIP – is sometimes called P2PSIP. In these cases, this functionality has always been available in SIP, and is not what we refer to as P2PSIP.

Provision by each user of a phone line for collective use of all callers. When a user places a call, the call can be placed either for free or at very low cost, using the line of another user on the network. [x,xi] This type of resource sharing is a different kind of P2P, in that the phone lines, rather than the call control, are being distributed. A central server is still used to complete calls.

Building a Call Control Platform on a P2P Fabric

Once a P2P basic fabric exists for distributing services, it is straightforward to use it to develop call-control services used by the IMS. Later, we'll discuss using this in an IMS deployment, but first, let's take a look at how this fabric can be used to replace the registrar and proxy in a traditional SIP deployment of a proxy/registrar server and a few phones.

Each phone, in addition to supporting traditional SIP, is enhanced by the addition of the P2PSIP stack. As such, each phone, once connected to the fabric, can mathematically determine where it belongs, relative to the other devices, and which resources (in this case, only registrations) it should store. It also can use this same mathematical algorithm to easily locate the peers that will be storing a particular resource. In this case, the collectively stored resource is the registrations, and each device will store a number of registrations. The most common mechanism to distribute this service is a *hash table*. Each user name or phone number is hashed to produce a unique signature within a known, finite hash space. The various devices making up the fabric split and share the hash space; each device stores the registrations that hash to a particular interval that device is responsible for.

When the first phone joins, it realizes there are no other peers, and starts a new P2P node. At this point, it is responsible for storing all resources in the fabric. A registration for the user of that phone, mapping from the extension to the IP address of the device, is generated, hashed to produce a location to store it in, and stored in the fabric. At this point, this is the first and only phone in the fabric.

When a second phone joins, it sees there is an existing node. The new phone now controls a fraction of the resource space. A new registration is generated for the new phone's user, hashed, and stored in the fabric. Statistically, it is likely that one registration is stored on each phone, although in theory, both registrations could reside on the same device. Also, redundancy ensures that multiple copies of the hashed resource (using different hash techniques to provide different storage locations) are stored in the fabric.

Each new phone repeats this process. As it joins, it takes over a small fraction of the hash space, and stores the registrations that map to that region. The added phone also adds a new registration for its user. As additional peers come, the cluster gets larger, and registrations are moved to the new peers. As they leave, the peers hand off their resources along with the slice of hash space they control. In the event of a failure, replicas serve until the failure is detected and resources propagated from the replicas are used to replace the missing information.

When a call is placed, a similar mechanism is used. The address of the called party is hashed, and the routing algorithm of the cluster is used to locate the responsible peer, which stores the registration. Since each peer only knows about some of the peers in the system, intermediate peers will either route the message on the sender's behalf, or will provide a "next hop" for the location of the peer. If the search fails, replicas can be consulted. The peer provides the registration, and the caller now has the IP address of the remote party's phone. At this point, a normal SIP call can be placed between the devices.

Cost Model of P2P Primitives

Having described the basic concepts of P2PSIP, we can examine how these concepts can be used for the benefit of IMS. In particular, P2PSIP can help ensure high availability of resources, and help a system scale in a simple way, while being designed from the outset to work with SIP. Note that while we will talk primarily about its utility in an IMS deployment, there is nothing unique in IMS about how P2P concepts can benefit the solution. Any large-scale system consisting of a large number of devices, including a traditional non-IMS SIP network, would benefit from a similar approach. SIP deployments and the IMS are unique in being able to leverage the "SIP friendly" aspects of P2PSIP directly, based on their SIP lineage.

High Availability

Traditional telecom infrastructures, including SIP and IMS, aim to provide high availability of resources and services for the end-users. The infrastructure is critical for reliably performing sensitive services like emergency calling, and possible malfunctions in a high availability system have to be carefully taken into consideration both by the designer and the operator of the systems.

The high availability aspect is a well-understood field of engineering in the telecom industry. The generic way to solve this problem was to design a resilient system for each separate function that has reserve resources to fall back on in case of a certain component failure. An example of a resilient system is *cluster technology*, in which a clustered database employs several mirrored servers; if one server fails, the other takes over the load originally associated with the failed server.

Cluster technologies pose several disadvantages. Among the most important are:

Need for monitoring. It is common for the operations department of a provider to closely monitor the health of its clustered systems 24/7. While a cluster system has a degree of high availability by design, it can usually sustain only a single or few component failure(s) at a time. A secondary failure on a reserve server can bring down the cluster service completely. It is therefore essential to detect failures quickly and to eliminate the reason(s) for failure before a secondary failure occurs. Hence, specialized systems and costly trained personnel are dedicated for monitoring activities.

Inefficient use of resources. Typical cluster designs employ a master/slave schema, by which the slave machine sits idle waiting to take over in case of failure of the master. This means a 50% waste of available resources, as a slave system may not be used until a disaster occurs.

Non-deterministic fail-over. While the design of a cluster assumes the members will be able to automatically take over each other’s function, this seldom occurs in reality. Many problems can prevent a correct fail-over. These problems can include different or out-of-date software versions among the cluster members, unsynchronized data at the moment of fail-over, untested procedures, and inexperienced personnel who have seldom had the chance to rehearse a fail-over and its recovery processes. Additionally, after a failure and the associated repairs, the cluster has to revert to its original state, generating additional service downtime.

Operational costs. Clusters usually have high costs. They typically make use of redundant instances of expensive hardware with redundant components. The price tag of personnel specializing in troubleshooting and recovering these clusters can also be very large. For example, the cost of maintenance for a simple hard-disk repair in a high-availability server using RAID controller and multiple disks, is usually higher over its lifetime than the cost of the hardware itself. This leads to high capital expenses, and even higher operational expenses.

Scalability

Services related to telephony and real-time communications must be designed to scale for millions of end-users. It is likely that IMS deployments will need to scale to at least the size of a [small? medium? large?] country, with millions of active subscribers connected to a centralized system. IMS is a complex architecture with many discrete components and interfaces.

Most IMS call flows involve many components and use several interfaces and databases to fulfill the accessed service. For many portions of the IMS architecture, the implementers are responsible for finding the best way to implement scalability. Each call flow must identify the bottlenecks associated with heavy usage and find a solution for each. Many interfaces mean many ways to solve scalability for each of them.

The scalability issue has been addressed by the telecom industry in a number of ways. For example, high-density hardware combined with the use of clusters and load balancers have been successfully deployed in TDM/ATM networks for decades. Another technique to achieve scalability is through static provisioning, in which each device is preconfigured to use a different set of servers. Dedicated servers are allocated to serve different cities, parts of a country, or regions. The data containing the subscriber information or the routing logic of the switching equipment (such as dial plans) is replicated across multiple sites using synchronization mechanisms, which are usually proprietary to a particular vendor. The provisioning of the servers and various customer devices form an integral part of the operator operational activities.

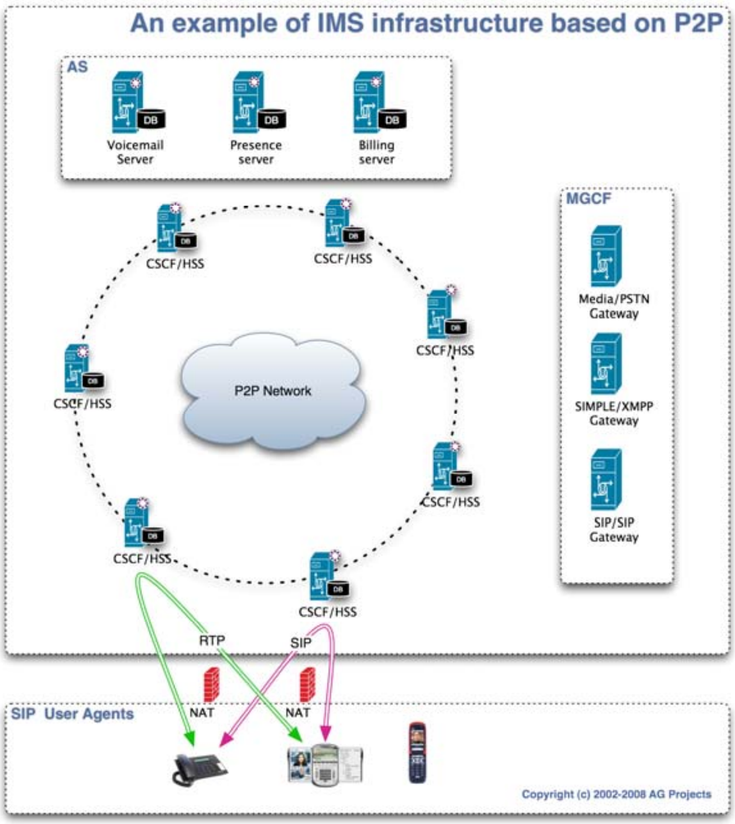
In addition to the reliability issue, operating expenditures, and capital expenditures discussed above, there are several other disadvantages of using classic technologies for achieving scalability of IMS.

Load balancers are points of failure. Load balancers are expensive and constitute major points of failure. The load balancers are usually able to increase the capacity of a certain function by employing light-weight service logic to distribute requests to a server farm behind them. The capacity increase by the use of a load balancer usually ranges from a simple multiple to one order of magnitude. For example, a stateless SIP load balancer might be able to load balance requests to ten transaction-stateful SIP Proxies behind it. After a certain load, the load balancer itself becomes subject to the same issues that initially were supposed to be solved by its addition.

Static provisioning is unreliable and error-prone. Static provisioning implies major changes whenever customers or their devices must be assigned to specific servers or locations. Data has to be prepared and uploaded to carefully identified devices. This requires time-consuming planning as well as coordination of changes with many third parties at a later date. Preserving data integrity across multiple locations is an unreliable and costly technique. Specialized database backend using replication techniques can solve this problem partially, but only at a high operational cost. This mechanism does not allow for improving the speed of changes under the increased load caused by the addition of new customers. Static provisioning can also cause problems if, for whatever reason, a different server needs to be consulted instead.

Can P2PSIP and the IMS Compliment Each Other?

While it is clear that some proposed uses of P2PSIP are at odds with the IMS (such as deployment of free, completely open communications systems on the WWW), there are many ways that P2PSIP and IMS can be combined to produce a product that is "greater than the sum of the parts." P2PSIP can help by eliminating some of IMS's complexity aspects while simplifying IMS's high-availability and scalability aspects.



As mentioned in the previous sections, P2PSIP has a primitive called **Leave**. This is equivalent to a "failure" in telecom terminology. P2P deals with failure in a completely different and (we believe) superior way than other reliability models that could be used in the IMS. Because failure handling has been moved in the network protocol itself, the need for monitoring and human intervention is significantly reduced. The network is able to detect and "heal" itself faster than any human operator can. It also deals with failover in a deterministic way.

Many IMS functions that require high availability and must quickly recover from failure can benefit from P2P. Such functions include the subscriber database, the SIP Proxy and SIP Registrar and the application servers that provide value-added services. Any function in which information is retrieved based on a known key (such as the subscriber SIP addresses) can be easily ported to a P2P environment.

By applying P2P concepts to solve the high availability of IMS functions, the operator can significantly reduce operating costs for the IMS.

Another P2PSIP primitive is **Join**. Join is the equivalent of increasing the capacity of a service provided in a telecom environment. In a traditional telecom deployment, even an IMS-based one, capacity increase is realized by the careful planning, reengineering, configuring, testing and deploying equipment and software by expert technicians, often at great cost. Such activities must usually be coordinated among multiple vendors and an integrator, adding to the costs of the upgrade. By using P2P in the core of the architecture, the costs related to capacity increase can be significantly reduced. P2P Join allows additional servers, and thus additional capacity, to be added to the network transparently, with an automatic operation handled by the protocol. Adding new services may require additional development and engineering, but adding additional processing power to a P2P cluster requires no new engineering and no downtime.

The IMS operator can also achieve a near-perfect distribution of resources among different racks, data-centers, cities or countries, since P2P systems use mathematical mechanisms that statistically guarantee that resources are fairly distributed. Assuming that sufficient connectivity with low latency is available, systems can even be balanced across WAN links, rather than just physical LAN links as in the case of clusters.

The **Self-organization** primitive of P2P provides an innovative way to deal with join and leave events. Because the network has its own organization and clustering primitives at the application layer, a truly accessible independent IMS can be implemented, regardless of the access technology used in the core or last mile access. IMS can be fully distributed in multiple physical locations that automatically share the load generated by the end-users.

The **Routing** primitive provided by the P2P protocol, along with replication techniques, assures deterministic routability of service requests among the servers even during server failures, without any manual intervention. Because the routing is performed using mathematical formulas, it is not subject to the provisioning of static data and the hassle associated with manually replicating such data across multiple nodes.

Unprecedented **scalability** can be achieved by using P2P. As the total capacity of the network scales with the amount of nodes available, the incremental costs of upgrade is small compared to major reengineering. The major monitoring function required for a P2P network is the amount of nodes available to serve a specific function.

High economy-of-scale cost savings can be achieved by using off-the-shelf commodity hardware instead of highly available servers. It is cheaper to replace standard hardware rather than repair highly available hardware. A defective server can be simply replaced by booting a standard P2P software build; the server will provision itself by communicating with the other peers in the cluster, and become part of the operational network without any manual setup.

Implication for End Devices

In a combined P2PSIP/IMS approach, few (if any) of the capabilities will be pushed entirely to the endpoint. There are several reasons for this. First, mainly mobile devices are envisioned to be deployed in the IMS. The requirement that these devices consume little power means that they are poorly suited to being used as peers, since the additional transmission power needed to serve requests can compromise the life of the battery. Also, mobile devices can be in and out of range, or simply turned off for an extended period, such as an intercontinental flight. Similarly, these devices are likely to have low bandwidth capability, and are more likely than fixed devices to come and go on the network. Rapid join/leave cycles of devices in the fabric, a condition referred to as churn, is undesirable in a P2P network because it leads to a greater dependence on replicas for locating information, increases the load on all peers as resources are moved, and leads to less stable routing.

Because of this, the industry is exploring mechanisms to allow the devices to communicate with the fabric, either using conventional SIP or by using a client protocol designed to query and store information without being a peer.

In the case of IMS and P2PSIP, it is more likely that, at least initially, P2PSIP will be used among the servers, leaving the relationship between the end-devices and the servers unchanged. This provides the additional benefit of leaving the current authorization and charging mechanisms in place, while leveraging the advantages of P2P. As the technology becomes more established in the IMS, more functionality may move to the edges, while still centralizing aspects required to maintain control of the network.

Applications of the IMS to P2PSIP

P2PSIP was designed from the outset to reuse as much existing infrastructure from other protocols and architectures as possible. There has been a concerted effort not to "reinvent" the wheel. As such, basic aspects of SIP have not been revised, but this also means that certain aspects of a telecommunications system – most notably the charging functions – have not been addressed in P2PSIP.

As a result, persons applying even a purely P2PSIP deployment may find interesting aspects of IMS to use in their systems. For example, servers and mechanisms for charging function can be integrated into a P2PSIP system when placing calls to the PSTN. When peering between two isolated P2PSIP deployments, the mutual authentication mechanisms proposed by the IMS can be applied.

Conclusions

While it is tempting to dismiss P2PSIP as a competitor to IMS – a development that may lead to the downfall of some of today's carriers, some aspects of P2PSIP can be easily integrated into an IMS architecture, resulting in a system that is easier to manage, to scale, and is more resilient against faults. It is certain that the relationship between P2PSIP and IMS will be complex. It is true that P2PSIP can be used without the IMS to deliver many of the same services that are also being proposed for the IMS, and such solutions are certain to emerge. But P2PSIP and the IMS can also live together in some deployments, enabling the advantages of P2PSIP to be paired with IMS.

A hybrid network which combines these aspects of P2PSIP with the authorization, charging, QoS, and new application deployment ease offered by IMS, could be an attractive design for a next-generation network. All savings associated with operating an IMS based on P2P principles can be passed down to subscribers, turning IMS into a cost-effective way to provide reliable services to millions of users while staying competitive with other Internet players.

Authors

David A. Bryan, CEO SIPeerior Technologies

David is a recognized thought leader in the area of P2PSIP and is co-chair of the IETF P2PSIP Working Group. He has published numerous IETF drafts, industry trade articles, and academic papers on the subject and is active in the SIP community. Prior to founding SIPeerior, David was co-founder and CTO of Jasomi Networks, a pioneer in the SIP Session Border Controller (SBC) market, which was sold to Ditech Communications in 2005. David previously worked for Cisco Systems via its acquisition of Vovida Networks, where he led a team of developers creating the first open-source softswitch. David holds bachelor's degrees in Computer Science and Physics from Richard Stockton College in NJ, as well as a master's degree in Computer Science from The College of William and Mary.

SIPeerior Technologies is the leading provider of standards-based, OEM software solutions designed to enable server-less implementations of real-time communications applications. SIPeerior was founded in 2006 by a team composed of industry veterans, serial entrepreneurs, and technology experts. SIPeerior is a privately held company with headquarters in Williamsburg, Virginia, USA.

Adrian Georgescu, Founder and CEO of AG Projects

Adrian is an advocate of IETF standards, member of OpenSER management board, founder and co-chair of the Dutch ISOC SIP Special Interest Group. Prior to starting of his own company, Adrian had accumulated experience with Esprit Telecom, GTS, Ebone and KPNQwest. Adrian's expertise is the design of high-availability infrastructures, addressing and numbering systems, signaling protocols, provisioning and mediation systems, implementation of disaster recovery plans. Adrian has received in 1996 a degree in Electrical Engineering from Faculty of at the Polytechnic University of Bucharest, Romania. He personally contributed to and supported the development of projects like OpenSER, MediaProxy, CDRTool, OpenXCAP and MSRP-Relay.

Based in the Netherlands, AG Projects is a leading European company supplying cable operators and Internet service providers with turnkey SIP infrastructure systems. AG Projects focuses on the delivery of scalable SIP infrastructure that complements fiber to the home deployments, triple play and converged fixed/mobile communication services. Delivery includes the analysis of requirements, tailoring of solutions to meet the requirements, project management, training and support activities.

References

---

i Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, Proceedings of ACM SIGCOMM, I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balikrishnan, 2001

ii A P2P Approach to SIP Registration, draft-bryan-sipping-p2p-00, D. Bryan and C. Jennings, January 2005, IETF Internet-Draft (expired draft, work-in-progress)

iii IETF P2PSIP Working Group Charter, http://www.ietf.org/html.charters/p2psip-charter.html

iv Concepts and Terminology for Peer to Peer SIP, draft-ietf-p2psip-concepts-01, November 2007, D. Bryan, P. Matthews, E. Shim and D. Willis, IETF Internet-Draft (work-in-progress)

v AG Projects' SIP Thor, P2P infrastructure for operators, http://ag-projects.com/SIPThor

vi SIPeerior Technologies, http://www.SIPeerior.com

vii P2PSIP.org, http://www.P2PSIP.org

viii SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System, International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA), D. Bryan, B. Lowekamp and C. Jennings, June 2005

ix Peer-to-Peer Internet Telephony using SIP, Columbia University Technical Report CUCS-044-04, K. Singh and H. Schulzrinne, October 2004

x Free world Dialup: http://www.freeworlddialup.com

xi OOMA, http://www.ooma.com/

Glossary

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
API	Application Program Interface
AS	Application Server
ARIB	Association of Radio Industries and Businesses (ARIB)
ATIS	Alliance for Telecommunications Industry Solutions
AVP	Attribute-Value Pair
CAMEL	Customized Application Mobile Enhanced Logic
CCSA	China Communications Standards Association (CCSA)
CDF	Charging Data Feature
CN	Core Network
COPS	(Common Open Policy Service)– RFC 2748
CS	Circuit Switched
CSCF	Call Session Control Function
Cx	Diameter interface for interactions between HSS and CSCF
DIAMETER	Successor to RADIUS – RFC 3588 – Need for Mobile IP
DSL	Digital Subscriber Line
ETSI	European Telecommunications Standards Institute
FMC	Fixed/Mobile Convergence
FTTH	Fiber to the Home
GSM	Global System for Mobile Communications
HSS	Home Subscriber Server
I-CSCF	Interrogating Call Session Control Function
IPSec	IP Security Protocol

IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
MGCF	Media Gateway Control Function
MRFC	Media Resource Function Controller
MRFP	Media Resource Function Processor
MSF	MultiService Forum
NAT	Network Address Translation
OSA	Open Services Architecture
PRACK	Provision Response Acknowledgement (SIP Message)
P-CSCF	Proxy-Call Session Control Function
PSTN	Public Switched Telephone Network
PDF	Policy Description Function
QoS	Quality of Service
RADIUS	RFC 2865 – Remote Authentication Dial In User Service
SCS	Service Capability Server
S-CSCF	Serving-Call Session Control Function
SEG	Security Gateway
SGSN	Serving GPRS Support Node
SIP	Session Initiation Protocol
Sh	User profile interface between HSS and AS
TTA	Telecommunications Technology Association (TTA)
TTC	Telecommunication Technology Committee
UE	User Equipment (IMS Terminal)
VCC	Voice Call Continuity
WIFI	Wireless Fidelity (IEEE 802.11)
WI-MAX	Worldwide Interoperability for Microwave Access, Inc (IEEE 802.16)
XCAP	XML Configuration Access Protocol