A Peer-to-Peer Overlay for the IP Multimedia Subsystem

Ling Lin  
*Vodafone Group Research & Development Department, United Kingdom, United Kingdom*

Antonio Liotta  
*Department of Electrical Engineering, Eindhoven University of Technology, Netherlands*

**ABSTRACT**

The growth of the Internet and its popular services are forcing telecom operators to provide advanced services to their subscribers, as traditional voice services are no longer enough to attract more customers. To enable more innovative and value-added IP services and take advantage of the services that the Internet provides, the IP Multimedia Subsystem (IMS) is introduced. The IMS provides a complete access-agnostic architecture and framework that facilitates the convergence of the mobile network, removing the gap between the two most successful communication networks that are the cellular and the Internet network. The harmonized All-IP platform has the potential to provide all Internet services with a more cost-effective and more efficient architecture than the circuit-switched networks do. However, by merging two of the most successful networks, the integration of two network models with different concerns and motivations is not without its problems, among which the scalability issue is the most essential, when supporting content delivery services. The purpose of this chapter is to study and design a new content delivery network infrastructure, PeerMob, merging the Peer-to-Peer technology with the IMS framework, which benefits IMS with scalability, reliability, and efficiency features coming with decentralized P2P architecture. The chapter also puts this P2P IMS paradigm under realistic network conditions and strenuous simulation to evaluate the performance of the P2P IMS system.

**INTRODUCTION**

In recent years, the Internet has become the main medium to deliver contents to end users and the multimedia services in the Internet have experienced an explosive growth in many dimensions, including size, performance, and geographical span. The growth of the Internet and its popular multimedia services are forcing telecom operators to provide comparable services to their subscribers as traditional voice services are no longer enough to attract more customers. Therefore, to enable efficient and cost-effective value-added IP-based services, cellular networks are evolving from a traditional purely circuit-switched network to an All-IP packet switched network. Furthermore, to take the advantage of the multimedia services that the Internet provides, the telecom industry is as a whole is undergoing an evolutionary transformation to be an efficient content delivery platform.

With IP spreading throughout cellular networks, the challenge of integrating voice and data services in the fixed and mobile access networks becomes more formidable. Thus, a standard convergence networks platform is required to offer data, voice, and multimedia services. Then, the IP Multimedia Subsystem (IMS) has been introduced to deploy IP-based telephony and multimedia services on every access network, including both circuit-switched networks and packet-switched networks, to eventually replace the circuit-switched core network with an All-IP core network. The IMS is an attempt to provide a complete access agnostic architecture and framework removing the gap between the two most successful communication networks that are the cellular and the Internet network, while still integrating with legacy network for existing services. The IMS allows network operators to play a vital role in the network and

that’s why IMS has generated intense research and standardization efforts. (Camarillo, Angel, & Martin, 2008)

Problem statement
These emerging content delivery services have high demands for network resources and thus create new challenges in terms of network bandwidth, service management, configuration, and deployment. By merging two of the most successful networks, the cellular network and the Internet, the integration of two network models with different concerns and motivations is not without its problems. Among these, the scalability issues, inherited from the client-server architecture from the cellular network, are the most essential.

The IMS is realized through a collection of well specified logical nodes with clearly defined interfaces to achieve network functions, such as registration, subscription, session control etc. These nodes are designed according to the client-server paradigm as historically they have been located and operated in the fashion by network operators. In the IMS, both signal flows and media flows have to traverse a long list of nodes to reach their destinations. If one or more intermediate nodes in the communication link reach their capacity limit, congestion may occur and no more communication links via these nodes are possible. Thus, this will in turn limit the performance of the IMS in a large-scale telecom network, especially when supporting content delivery services.

Main Contributions
Concerning the scalability issues in the IMS, the purpose of this chapter is to study and design a new content delivery network infrastructure, PeerMob, by merging the Peer-to-Peer (P2P) technology with the IMS framework, which benefits the IMS with scalability, reliability, and efficiency features coming with a decentralized P2P architecture.

Generally speaking, P2P is a technology that fosters resources self-deployment and self-organization, while still achieving optimized resource utilization for the deployed applications and services. P2P is designed for sharing computer resources, CPU power, storage and bandwidth, by direct exchange, rather than requiring the mediation or support of a centralized server. P2P architectures are characterized by their ability to adapt to failures and accommodate transient populations of nodes while maintaining acceptable connectivity and performance. The use of the P2P paradigm to provide content delivery services is gaining increasing attention, and has become a promising alternative to other legitimate approaches as the classical client-server model or Content Delivery Network (CDN).

Another objective of the chapter is to put this P2P IMS paradigm under realistic network conditions and strenuous simulation to evaluate the performance of the P2P IMS system. For this purpose, a new P2P IMS proof-of-concept is carefully designed, implemented and deployed for demonstration. The P2P IMS proof-of-concept includes most of the IMS components, such as x-CSCF, HSS, IMS application servers and the IMS peer application. Also, a customized P2P IMS simulation has been built to demonstrate the performance of this P2P IMS approach.

IMS Framework
Telecom network architectures have evolved to several access networks coexisting both in the wireless and wired domains. The networks have also evolved from the original circuit-switched networks to the packet-switched networks. The intention of IMS is not to standardize applications within it but rather aid in building all services independently of access networks aiming to aid the deployment of ALL-IP networks, making most popular Internet services and application anywhere and everywhere.
3GPP release 7 states the complete solution to support IP multimedia applications consists of User Equipment (UE), IP-Connectivity Access Network (IP-CAN), and the specific functional elements of the IP Multimedia Core network (IM CN) subsystem. (Poikselkä & Mayer, 2009) (Oredope & Liotta, 2008a)

Figure 1 illustrates a simplified version of the IMS architecture. Dotted lines represent the signal flows while the full lines give means for the media flows. It is important to notice that IMS deals just with the session signal and control while it does not tackle the actual transport of data or media flows of the sessions. The IMS architecture can be divided in three logical layers, the access layer, the session control layer and the application layer. (Poikselkä & Mayer, 2009) (Oredope & Liotta, 2008a)

Figure 1: The IMS Architecture

The access layer provides a common access interface for UEs shielding the details of various access networks. UEs might connect to the IMS infrastructure through either a packet-switched network or a circuit-switched network that interfaces the IMS through the Media Gateway Control Function (MGCF).

The session control layer handles session setup and teardown, and manages the handover of sessions between service providers. The most important elements in this layer are the HSS and the three types of Call Session Control Functions (CSCF). All of these nodes can be distributed over multiple nodes to increase redundancy and scalability.

The Home Subscriber Server (HSS), technically an evolution of the GSM Home Location Register (HLR), provides a central repository for user-related information to handle multimedia sessions. The user-related information includes location information, security information, user profile information and the S-CSCF allocated to users. The HSS is always located in the home network. If there is more than one HSS in an IMS core, a Subscription Locator Function (SLF) is used to determine the HSS that a user’s record is located.

The Proxy Call Session Control Function (P-CSCF) is a SIP proxy server, providing the first contact point in the signal plane between IMS UEs and the IMS core network. This means that all signal traffic from or to an IMS UE must traverse a P-CSCF, which will then forward SIP requests and responses in the appropriate direction. Here is how the function works. First, a P-CSCF terminates the IPSec security connection between an IMS UE and the P-CSCF. Then, the P-CSCF verifies the validity of SIP requests to or from the IMS UE. If the validation is successful, the P-CSCF forwards the SIP message to a suitable S-CSCF. In the early stage of IMS deployment, the P-CSCF needs to be placed in the home network as media traffic traverses always through GGSN and the P-CSCF has to be deployed close to GGSN. Currently, the P-CSCF may be located either in the home network or the visiting network.

The Serving Call Session Control Function (S-CSCF) is a SIP server with two key roles. One is a SIP proxy server routing SIP messages to appropriate SIP servers or application servers. The other is a SIP registrar server maintaining the binding between the current address of IMS UEs and user identities. S-CSCF has an interface to the HSS and the SLF to download authentication vectors from the HSS. The authentication vectors are required to authenticate IMS UEs to grant access. The S-CSCF also needs to inform the HSS which S-CSCF is allocated to handle a UE; subsequently the UE will send registration requests to the same S-CSCF and the P-CSCF. The S-CSCF is always located in the home network.

The Interrogating Call Session Control Function (I-CSCF) is also a SIP proxy server, placed at the edge of an IMS domain to communicate with other IMS domains, hiding the network capacity and topology from the outside. Because the I-CSCF is the first contact point to other IMS domains, the I-CSCF implements a functionality called Topology Hiding Inter-network Gateway (THIG) to encrypt sensitive information in SIP messages. The address of I-CSCF is listed in well-known domain DNS servers. When
another IMS domain wants to find an entity within a destination domain associated with an I-CSCF, it will query the address of the I-CSCF from the DNS servers and forward the SIP message to the I-CSCF. Like S-CSCF, I-CSCF also has an interface to the HSS and the SLF. I-CSCF queries the HSS to obtain the address of an appropriate S-CSCF where a request should be forwarded, then assigns the S-CSCF to handle the SIP request. The I-CSCF is usually located in the home network.

The Application Layer contains Application Servers, which are expansion slots for IMS networks where third party products and services are located. It consists of different application servers for the execution of various IMS services and the provision of end user service logic. The Application Layer offers a platform for value-added services which goes well beyond the integration of network and devices. (Ilyas & Ahson, 2008)

Each function mentioned above is not necessarily implemented on a single node. The IMS architecture is a collection of functions exposed as standardized interfaces, while implementations are free to combine two or more functions into a single physical node. Similarly, a single function can be spread over two or more nodes for load balancing or availability purposes.

**A P2P Architecture for the IMS**

P2P is a distributed application architecture in which participants can act as clients and servers. These participants are able to work without a central server but are also ready to make use of a central server, if necessary, for the efficiency of the system. In P2P networks, the information is no longer concentrated on central servers but is provided by each participant, called a peer. The source of content is not the only one to upload the content and to share its resources, CPU power and bandwidth. Most participants download content but also upload it to other peers, sharing their resources to help other peers to get the content. As most participating peers compensate the additional workload on the system by providing their own resources, a P2P system typically scales well up to a large number of users.

Generally, P2P networks have the following characteristics: distribution, self-organization and robustness. Distribution means that the content is distributed to different peers, and the traffic is brought to the edges of the network. There is usually no central entity that is required for the functioning of the network. Self-organization means that the network coordinates itself, during which joins and leaves of peers and other forms of interruptions are handled automatically. Robustness means that the P2P system is already designed for the imperfect behavior of the network. The more content and resources are replicated among peers, the higher overall fault tolerance may grow. However, these advantages offered by P2P are coming with a certain cost, because in a P2P network peers become responsible for managing peers and network resources, including content management and peer management. Furthermore, the dynamic nature of peers poses presents challenges in the communication overlay. (Oram, 2001) (Oredope et al., 2008b)

**Topology Overview**

The P2P topology of an overlay network is crucial to the operation of the system. The topological characteristics of P2P overlays need to be taken into account when designing P2P networks, as it is very cumbersome to change a topology module after deployment.

During its short history, P2P has already passed through several generations, which can be classified into two main groups: structured overlay network and unstructured overlay network, as shown in Figure 2. These depend upon the logical organization of the peers’ topology. In structured architectures, peers are well organized and according to defined algorithms. By contrast, peers in unstructured overlay network have the flexibility to choose the number and destinations of their connections, and adapt them to network heterogeneity to improve network performance. The unstructured overlay topology can also be
further divided into three different overlay topologies, according to the degree of centralization: centralized P2P architecture, pure P2P architecture and hybrid P2P architecture.

Figure 2: P2P Overlay architectures

The fundamental problem of P2P is resource discovery, e.g. finding a particular peer, service or content. This is achieved differently in unstructured and structured networks. Resource discovery can be classified into three categories: mechanisms with specialized index nodes (centralized search), mechanisms without any index (flooding search), mechanism with indices at super peers (distributed search), and mechanism with a Distributed Hash Table (DHT). (Schollmeier & Schollmeier, 2002)

Centralized P2P Architecture

The centralized P2P architecture, often known as the Napster architecture, is an architecture where a centralized server holds all of the meta-data of peers, such as their address and content portfolio. When a peer searches for content, it first queries the central server. Then the central server replies with the address of the peers containing the requested content. Thus, the querying peer is able to retrieve content directly from the peers suggested by the central server. All further data exchanging is directly between these peers.

In this centralized P2P architecture, queries have to go through the centralized server to identify peers containing the content of interest, which is called a centralized search. Comparing to other P2P systems, a centralized P2P system is much easier to maintain and keep consistent. Moreover, the discovery is efficient and deterministic because only a single message is required to resolve a query. However, the central server represents a single point of failure. Relying on central servers leads to less scalable and, in particular, vulnerability to Denial-of-Service attacks.

Pure P2P Architecture

The pure P2P architecture has no centralized node and all nodes in this P2P network are truly equal. As there is no centralized node for resource discovery, searching is performed by forwarding queries from peers to peers, using a flooding based lookup algorithm (e.g. in Gnutella 0.4), which effectively broadcasts the search with limited scope. Another possibility is to use more intelligent routing methods (e.g. Freenet), whereby the query is iteratively routed towards the peer that is most likely to have the requested content. (Jovanovic, Annexstein, & Berman, 2001)

Flooding propagates the query messages to every node within a specified radius, based on the query’s Time-To-Live (TTL) value. If some peers have the requested content, they reply to the original querying peer. Otherwise, queries are propagated from peer to peer, until matching resources are found or TTL has expired. In this way we avoid congesting the overlay network with search requests. Therefore, flooding can quickly retrieve any related documents within TTL hops of the initiator. However, it is possible that some peers, which are outside of the TTL limit, cannot be reached. The advantage of pure P2P systems is the fact that there is no single point of failure, which results in an enhanced reliability.

Although this mechanism is simple and robust, the resource discovery process is not very efficient, as the overlay structure is not deterministic. A node in a pure P2P network is in principle unaware of the resources its neighbor peers maintain. The flooding based discovery may lead to significant network usage by generating a large number of query messages. More importantly, the amount of traffic scales poorly as the number of peers in the network increases. (Ripeanu, Foster, & Iamnitchi, 2002)

Hybrid P2P Architecture
The hybrid P2P architecture is a combination of the centralized and pure P2P architectures. The hybrid architecture is used by many popular peer-to-peer applications, such as Gnutella 0.6, Kazaa, and JXTA.

Super peers hold meta-data about the content that single peers contains. Indices of single peers are maintained at super peers. Queries are forwarded form single peers to super peers. The super peers query each other if any of their single peers have the required content. If there is a match, peers exchange all further information directly, without super peer involvement. It is important to note that such super peers do not constitute single points of failure, since they are dynamically assigned and, if they fail, the network will automatically take action to replace them with others. Hybrid P2P networks allow a peer to function as either a single peer or a super peer, depending on the conditions, such as the available bandwidth and processing power.

In terms of performance, the hybrid architecture is located between the pure and the centralized P2P architecture. It scales quite well and has good resiliency. The hybrid architecture is also situated somewhere between the other two architectures when considering the search coverage in the network. When a peer sends a query, the request is forwarded to other super peers who then forward the request onwards. The search coverage is limited by the TTL of the request message, but the coverage is a lot larger in the hybrid architecture than in the decentralized P2P architecture, thanks to the backbone effect of super peers. (Beverly & Garcia-Molina, 2003)

**Structured P2P Architecture**

The structured P2P architecture is also known as Distributed Hash Table (DHT) architecture. Each peer acquires an identifier based on a cryptographic hash of some unique attribute such as a peer’s IP address or a peer’s public key. The identifier for a content item is also obtained through the same hashing function. The hash table actually stores content items as values indexed by their corresponding keys. That is, node identifiers and key value pairs are both hashed to one identifier space. Then, peers are connected to each other in a certain predefined topology, for example, a circular space in Chord (Lua, Crowcroft, Pias, Sharma, & Lim, 2005). Thanks to the structured topology, data lookup becomes a routing process with small routing table size. DHTs guarantee that the nodes distance and the number of connections per node is $O(\log(n))$, where $n$ is the number of nodes in the network. A structured P2P topology overcomes the limitations of an unstructured P2P topology by organizing the overlay with some structured content location mechanism such as DHT.

As a consequence of the deterministic organization of the structured P2P topology, maintenance efforts to keep the correct overlay are increased due to the churn. Churn is the rate by which the peers join and leave a network. If the churn handling of the algorithm is not efficient enough, the maintenance signaling would load the overlay unnecessarily and it may cause the entire algorithm to be inefficient.

**Topology Summary**

Different P2P architectures lead to different degrees of scalability, reliability, efficiency and the ability to control the network. The pure P2P architecture is the most resilient architecture against node failure. On the other hand, with the centralized P2P approach, the server may become a capacity bottleneck, causing the P2P system to stall. This type of problem has been evidenced with the Gnutella network, which was originally designed for relatively small growth rates. With the huge influx of users, Gnutella’s performance degrades significantly and becomes poorly usable.

The hybrid architecture is utilized to improve the discovery efficiency and advance the network scalability in P2P systems. The structured P2P architecture is developed in another direction to solve the weak discovery efficiency problem. The former shows its benefit in scalability and fuzzy search in a dynamic network environment, while the later has the advantage of a more search facility. Typically,
structured P2P networks obtain better performance at steady state, but as the node churn increases they
tend to show instability or congestion issues.

Neither the pure nor the centralized P2P networks can simultaneously offer both reliability and scalability
in a single algorithm. In a highly dynamic, mobile environment, the hybrid P2P architecture is preferable.
We have in fact chosen this approach to develop our P2P IMS system.

**PeerMob P2P IMS**

The aim of this study is to design a new content delivery network infrastructure for the IMS (the PeerMob
P2P IMS platform), incorporating a P2P framework into the IMS. Our system is realized through a
collection of well-defined PeerMob P2P IMS components and protocols. Since the IMS components and
interfaces have already been standardized by 3GPP, our strategy is to extend the IMS using the existing
recommendations for the incorporation of new mechanisms and applications. Thus, we don’t require any
modifications to the existing IMS platform and suggest an approach to implement P2P networking in the
form of a software upgrade.

The PeerMob P2P also adopts a group-based strategy within a hybrid architecture, to support and
maintain a low-diameter discovery network. Each peer must be assigned to one of two mutually-exclusive
roles (a single peer or a super peer), based on a super-peer election algorithm. Single peers do not join any
peer groups and are not responsible for peer group management. Super peers must join a peer group,
based on a cryptographic hash of its identity, to provide peer group services. Super peers trace the single
peers that contain the content which might be relevant to the group, and maintain content chunk bitmaps
for the peer group.

Super peers know all other super peers within the same peer group. The assignment of roles is not
permanent: a peer may start as a single peer, and later become a super peer if more super peers are
required for a particular peer groups. Alternatively, a super peer may decide to move all of its single peers
to the other super peers and become a single peer by itself, to reduce the number of super peers and thus
reduce the traffic generated by communication between super peers.

Both single peers and super peers are able to not only download content but also upload content to other
peers, sharing material to help other peers to retrieve relevant information. Each set of content belongs to
a particular peer group based on a cryptographic hash of its content ID. Super peers and content are both
hashed to one identifier space. Each peer group is responsible for managing content within the peer group
only.

We describe below the key entities of P2P IMS:

*Peer*: a peer refers to a participant in a P2P system, which not only consumes content, but also shares
content to other participants. Peers can be User Entities (UEs) or cached servers. A UE must be either a
super peer or a single peer. A cached server is always a super peer located in the IMS network, which
aims to improve the performance of the PeerMob P2P IMS.

*Peer Group*: peers self-organized into peer groups, which are responsible for distributing specific
category of content based on a cryptographic hash of content IDs. Each super peer must join one but only
one peer group based on cryptographic hash of its identity. Single peers do not belong to any peer groups.

*Cached Server*: the cached servers refer to network entities that are usually deployed at the network edge
and act as initial super peers. A cached server also belongs to a peer group based on cryptographic hash of
its identity. A cache server keeps all copies of the content belonging to its peer group.
**Tracker Server:** the tracker server is a logical entity being a super peer management server and a key store server.

**Source Server:** the source server is an application server providing user interface for content provision.

**Chunk:** a chunk is a basic unit of partitioned streaming content, which is used by a peer for the purpose of storage, and exchange among peers.

By introducing the concept of group-based super peers, the topology is now organized through a two-level peer hierarchy: the super peers, included in different peer groups, and the single peers. Super peers are nodes that usually have more bandwidth, CPU power and reliable links than single peers. Because of that they can take server-like responsibilities, providing indexing services to a set of single peers. The design based on super peers has two main advantages. Firstly, certain system tasks, such as the indexing service, can be assigned to super peers, to improve the overall system reliability and performance. Secondly, the use of super peers allows the P2P system to limit the number of participants in certain distributed algorithms, such as search, which do not scale well and become too expensive when the system size is large. In the evaluation section below, we show how the super peer design can improve the system scalability.

However, the use of super peers introduces a number of new challenges to be addressed. To manage super peers, the P2P system needs to maintain a super peer list and the peer group structure. Furthermore, the P2P system has to continuously adjust the super peer list in response to peers’ arrivals and departures. Also, the load between super peers must be kept balanced to ensure the system’s scalability and fault-tolerance.

Figure 3 depicts a typical content delivery user case for PeerMob P2P IMS, illustrating the interaction between the PeerMob P2P IMS and a user, Alice, who wants to view content.

**Figure 3: P2P IMS Use Case**

The sequence of operations can be described as follows:

1. **User Alice** completes the IMS registration and authentication processes. Then, Alice updates her content list from a tracker server.
2. **Alice** wants to watch content and sends a watch query to the tracker server (if no super peer cached for the peer group that the content belongs to). The tracker server computes the peer group for the content, depending on a pre-defined hash function, and responds to Alice with both a cached server list and a super peer list of the peer group. The tracker server returns also a license key to Alice for content decryption. Alice caches the super peer list for future discovery.
3. **Alice** is now able to query either a cache server in the cache server list or a super peer (Bob in this case) in the super peer list, in order to find peers who actually contain the required content.
4. **Bob** responds to Alice that both Carol and Daniel contain the requested content.
5. **To speed up downloading,** user Alice downloads some chunks of the content from cached server A (cache servers always keep the content in the peer group) and others from Carol and Daniel. Alice decrypts these chunks in real time with the key provided in step 1.

**System Architecture**

Figure 4 depicts the high-level architecture of the proposed PeerMob P2P IMS. The IMS is mostly used for signal control while P2P is mainly to optimize the media traffic in the media plane. Apart from the
standard IMS network nodes, such as core IMS network elements (including P-CSCF, I-CSCF and S-CSCF) and HSS, three new application servers are introduced: the source server, the cache server, and the tracker server.

In IMS terms, the source server is an application server providing a new user interface for content provision. The content contained in the source server will be automatically distributed to a cached server belonging to the same group as the content itself. The cached server caches content from the source server, acting as the initial super peer. To save the bandwidth in the core network, the cached server should be deployed in the access network, sitting as close to the UEs as possible.

The tracker server plays an essential role in PeerMob P2P IMS. It has two main roles: a super peer management server and a key store server. To be a super peer management server, the main focus is to construct super peer lists for different peer groups and to respond with relevant information to the UE’s queries. To be a key store server, the tracker server is required to maintain the pairs of content and license keys and to issue proper keys to a user, based on the user’s subscription.

Figure 4: The P2P IMS Architecture

**Tracker Protocol**

To deploy a content delivery service on our group-based, hybrid P2P IMS framework, two key protocols are required. One is the signaling protocol between tracker servers and peers, namely the tracker protocol. The tracker protocol is responsible for the discovery of the correct cache servers, super peers and peer groups. The latter manage the single peers that contain the actual content.

The second protocol, namely the peer protocol, handles the communication signals among the peers. The peer protocol is responsible for the efficient transmission of data within peer groups. For instance, it locates the relevant peers (both single peers and super peers), which actually contain the content.

1. The tracker protocol handles the initial and periodic exchange of metadata between trackers and peers, such as peer lists and content information, as showed in

Figure 5. The peer protocol controls the advertising and exchange of media data availability between peers. Both tracker protocols and peer protocols can be carried over TCP or UDP, when delivery requirements cannot be met by TCP.

The sequence of operations of the Tracker Protocol is depicted in Figure 10:

2. A cache server registers at the tracker server.
3. The tracker server computes the group ID which the cache server belongs to, based on the cache server ID.
4. The tracker server records the cache server’s address together with its group ID.
5. The tracker server responds to the cache server with the group ID and cache server list in the peer group. From the returned cache server list, the cache server knows the entire current cache servers of the peer group.
6. The cache server downloads content in the peer group.
7. A peer is selected to be a super peer, based either on its capacity or randomly.
8. The super peer registers to the IMS core.
9. The super peer registers to the tracker server.
10. The tracker server computes the group ID which the super peer belongs to, based on the super peer ID.
11. The tracker server records the super peer’s address together with its group ID.
12. The tracker server selects the super peers of the peer group.
13. The tracker server responds to the super peer with group ID, cache server list and super peer list of the peer group.
14. The tracker server responds to the super peer with group ID, cache server list and super peer list of the peer group.

Figure 5: The Tracker Protocol

**Peer Protocol**

To prevent isolation between super peers in the same peer group, a mechanism based on the gossip protocol (Allavena, Demers, & Hopcroft, 2005) is adopted in the peer protocol. The peer protocol behavior periodically performs the following actions: first it selects a super peer from the super peer list of its peer group and exchanges information. Then, both participants update their actual information according to the received one. The peers involved in the exchange send to each other information about their current super peer lists and content chunk availability. Based on the received information, super peers update their super peer lists in order to obtain a better approximation of the super peer topology.

The continuous gossiping of topology information captures the dynamic nature of P2P systems. In this way super peers learn about new super peers in the peer group by receiving their identifier in an exchange, while crashed super peers are progressively forgotten and then removed from the super peer overlay. Thus, super peers are provided with continuously fresh samples of the entire peer group. The gossip protocol has been used successfully to implement several P2P protocols, including newscast, broadcast and aggregation protocol.

1. The sequence of operations of the Peer Protocol is depicted in

Figure 6:
2. Super peer 1 is elected to be a super peer of its peer group.
3. Super peer 1 exchanges information with other super peers and cached servers in same peer group, in this case, cached server 1.
4. The cached server updates its super peer list and cache server list.
5. The cached server response to the super peer 1 with its super peer list and cache server list.
6. Super peer 1 updates its super peer list and cache server list.

Figure 6: The Peer Protocol

**System Implementation**

In order to validate the protocols proposed in PeerMob P2P IMS, we realized a prototype which runs over a standard IMS platform. Figure 7 depicts the deployment diagram of the PeerMob P2P IMS proof-of-concept testbed.

Figure 7: Deployment Diagram

The PeerMob implementation follows the module design approach, in which functionalities are grouped into modules and each module depends on the services provided by other modules. This ensures maximum module reuse. The PeerMob P2P IMS proof-of-concept has been successfully demonstrated in the University of Essex, UK, using HP IPAQ 5500 hand-held devices.

The server side is implemented in Java; the peer implementation is coded in J2ME to make it runnable in hand-held devices. The PeerMob proof-of-concept has been tested successfully on the J2ME mobile phone simulator (running J2ME CLDC1.1 and MIDP 2.0) and HP IPAQ 5500 with familiar Linux. The
server implementation is based on Jain SIP (JainSipApiv1_1.jar) and NIST SIP (Nist-sip-1.2.jar). Java media framework has been integrated for media player.

**PeerMob Simulation**

Evaluation is vital in designing P2P systems by comparing a range of conditions which is broader than those tested on the test-bed. A customized simulator has been developed to analyze the behavior of PeerMob P2P IMS. In order to avoid the tight scalability limits imposed by packet-based simulators and model networks dynamic with acceptable accuracy level, the PeerMob simulator follows discrete-event and flow-based approach, with a simplified network layer, aimed at modeling PeerMob P2P IMS protocols accurately and efficiently in a dynamic P2P environment.

The PeerMob simulation follows the procedure for the standard discrete-event model. The simulator is initialized by a configuration entity targeted to read a configuration file, which is a plain xml file used to specify the simulation scenarios. After the initialization of the system variables and clock, the simulator calculates next clock time and processes subsequent events in the scheduling queue. The ending condition of the simulator happens once the maximum simulation time has been reached or when the scheduling queue is empty. Instead of advancing the simulation time at fixed increments and processing events synchronously at each clock tick, in the PeerMob simulator the processing and time advancement is triggered by the occurrence of events.

Inspired by the modular design of the P2P simulator OMNet++, everything in the PeerMob simulator is a module. These modules can be further divided into simple modules and compound modules. Simple modules include node, event and resource class; compound modules contain process and engine classes. Compound modules may consist of one or more other modules. Each module is defined by a time base, input, states, output and functions to compute the next states and outputs.

For the purpose of portability, ease of development and extensibility, the PeerMob simulator is implemented in Java. In the PeerMob experiments, the PeerMob simulator consumes around 1 Gigabyte of memory when simulating 100,000 peers. We could have simulated a larger system if we adopted a more powerful computer.

**Figure 8: The simulator core class diagram**

The simulator’s core class diagram is depicted in

Figure 8. The engine class represents the simulation engine. First, the engine parses the configuration file and loads nodes, events and processes from the configuration file into memory. Each event is assigned a time-stamp which indicates the logical time at which the event should be processed. Then, the engine initializes the scheduling queue, which is in control of the execution driven by events. After initialization, the engine is ready to run. The engine starts by inserting events into the scheduling queue and processes these events individually one after the other. The events are stored in the scheduling queue in increasing time-stamp order. The engine processes each event in the scheduling queue until a maximum virtual time is reached or the queue is empty. After being executed, each event is removed from the top of the scheduling queue and the virtual time of the simulation is updated.

**Simulation Parameter**

The parameters of the PeerMob simulator can be divided into four categories: network parameters, cached server parameters, peer parameters and super peer parameters. Table 1 lists the network parameters and their default values in the PeerMob simulator. These default values are used as the initial setting for the experiments, as further detailed in the next section. In each experiment, only one parameter at a time is
varied, while other ones are kept fixed. The total default number of peers, including single peers and super peers, in the simulation network is 100,000; the default number for peer groups is 10; and 10,000 data files are available in the simulation network.

Table 1: Network Parameters

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Peers</td>
<td>100,000</td>
</tr>
<tr>
<td>Number of Peer Groups</td>
<td>10</td>
</tr>
<tr>
<td>Number of Data Files</td>
<td>10,000</td>
</tr>
</tbody>
</table>

Table 2 illustrates the parameters required for the cached server. Cached Servers are usually deployed at the edge of the network before the content delivery service is started. The Number of Cached Servers indicates the servers deployed at service start-up time. The Cached Server Refresh Rate defines the delay that every cached server uses to randomly selects another cached server or super peer in its peer group, to exchange information (according to the gossip protocol). The Max Super Peer in Cached Server represents the length of the super peer list of the cached servers, which defines the maximum number of super peers that could be possible cached in a cached server. Cache servers will remove older cached super peers from the super peer list as soon as the length of the super peer list reaches its limit.

Table 2: Cache Server Parameters

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Cached Server</td>
<td>10</td>
</tr>
<tr>
<td>Cached Server Refresh Rate</td>
<td>100 seconds</td>
</tr>
<tr>
<td>Max Super Peer in Cached Server</td>
<td>1,000</td>
</tr>
</tbody>
</table>

In Table 3, the parameters required to simulate the peers are listed. Peer Join Rate signifies the churn of a simulation network, which indicates the average number of peer joining the simulation network at every second. Content Access Rate here refers to the content access rate of each peer (including single peer and super peer).

Table 3: Peer Parameters

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer Join Rate</td>
<td>10 peer per second</td>
</tr>
<tr>
<td>Content Access Rate</td>
<td>0 content per second</td>
</tr>
</tbody>
</table>

Table 4 includes the simulation parameters of the super peers. Super Peer Percentage refers to the probability of a joining peer is elected super peer. The Super Peer Refresh Rate represents the period after which super peers may randomly select some other cached server or super peer for the purpose of information exchange. Max Cached Super Peer in Super Peer specifies the maximum number of super peers that can be cached in a super peer. Max Cached Single Peer in Super Peer represents the number of single peer that a super peer can afford to handle.

Table 4: Super Peer Parameters
### Scalability Experiment

This section provides an evaluation of the proposed PeerMob P2P overlay network, based on simulation results and cross-validations with the prototype implementation. Our scalability experiments look at how the communication costs increase as the network grows. We select the results obtained at steady state, i.e. not during peer joining/leaving transients. Each peer selects content randomly across the network, switching content every 100 seconds.

The communication cost is measured by the average signal messages that traverse super peers, including outgoing requests and incoming responses. The super peers are the most limiting entities in the group-based hybrid P2P overlay network and that is why we focus on their signaling overheads.

Figure 9 depicts the results of the scalability experiment. The x-axis represents the total number of peers (including single peers and super peers) in the simulation network, while the y-axis refers to the average amount of signaling messages, including the outgoing requests and the incoming responses that each super peer sends and receives every 200 seconds. The average amount of signaling messages that traverses the super peer remains stable as the number of peers grows, showing that we have a scalable architecture.

Figure 9: Scalability Experiment

### Reliability Experiment

As mentioned previously, an important characteristic of P2P networks is that peers join and leave the overlay on a continuous basis (this is referred to as churn). The churn rate refers to the average fraction of peers joining and leaving the system per time unit. The reliability of a P2P network is evaluated by assessing the P2P network under different churn rates.

The first reliability experiment tries to assess the performance of the PeerMob network at fixed churn. In this experiment, initially there is no peer. When the simulation starts, new peers join the network at the rate of 10 peers per second. Also, as soon as a peer (single or super) joins the network, it starts accessing randomly selected content, switching content every 100 seconds. All other parameters are set to the default values.

Figure 10 illustrates how the number of single peer and super peer grows in this fixed churn reliability experiment. In this figure, X-axis represents the time eclipses in seconds while the y-axis refers to the number of peers in the simulation network.

Figure 10: Fixed Churn Reliability Experiment
Figure 11 depicts the load of super peers in this fixed churn reliability experiment. The x-axis represents the elapsed time slots (in seconds), while the y-axis refers to the average amount of signaling messages (including outgoing requests and incoming responses) that each super peer sends and receives in every 200 seconds. The chart shows that the average signaling increases sharply in the first 600 seconds and remains stable at around 45 signals for every 200 seconds thereafter.

Figure 11: Fixed Churn Reliability Experiment

The second reliability experiment aims at evaluating the load of super peers at different churn rates. We repeat the first reliability experiment but churn rates are varied. Only the rate of joining peers is variable, while we keep the other parameters fixed. The x-axis represents the churn rate, which is the number of peers per second joining the PeerMob simulation network. On the other hand, the y-axis refers to the average signal messages that traverse the super peers in 200 seconds at steady state. As shown in Figure 12, the average signaling messages per super peer raises slightly with churn rate. This result indicates that the average message per super peer is not determined by churn rate.

Figure 12: Variable Churn Reliability Experiment

Other Experiments

Figure 13 illustrates how the load of super peers is affected by their churn rate, which is the percentage of super peers in the PeerMob simulation network. The super peer rate experiment follows the method used in the fixed churn reliability experiment (above). The churn rate is kept constant at 1 peer per second while the super peer rate is increased. The x-axis represents the super peer rate while the y-axis refers to the average number of signaling messages that traverses the super peers in 200 seconds (at steady state). We can see that the signaling load increases gradually with the percentage of super peers.

Figure 13: Super Peer Rate Experiment

Figure 14 illustrates how the load of super peer is affected by the number of peer groups. The group experiment follows the method of the fixed churn reliability experiment. The churn rate is fixed at 1 peer per second, while the number of groups increases. The x-axis represents the number of groups, while the y-axis refers to the average number of signaling messages that traverses the super peers in 200 seconds (at steady state). We observe a sustained load decrease as more groups are introduced in the system.

Figure 14: Group Experiment

FUTURE RESEARCH DIRECTIONS

The use of the P2P paradigm to provide content delivery services in the IMS is gaining increasing attention recently. China Mobile Research Institute started IMS based P2P Streaming project in 2009. China Mobile is going to trail this IMS based P2P Streaming service in a few provinces in China in 2011. Furthermore, China Mobile extended this P2P idea and proposed a P2P media streaming standardization, Peer-to-Peer Streaming Protocol (PPSP), as an IETF draft in 2011. The key distinction between PPSP and PeerMob proposed in this thesis is that PPSP is based on Napster P2P topology while PeerMob is relying on group-based Hybrid P2P topology.

A large scale content delivery platform over telecommunication networks is a complex task. There are some challenges that remain for possible future work. First, P2P content delivery services tend to increase
the traffic of network providers, as peers first download content and then upload the content to other peers. The P2P approach approximately doubles the amount of traffic in the network compared to client-server architecture, where most of the traffic flows one direction. This motivates research in locality-aware P2P optimization solutions that exploit network proximity between peers to reduce the traffic in the core networks. Adapt PeerMob P2P IMS with locality-aware P2P protocols remains open for future work.

CONCLUSIONS
The current trend is going more and more towards IP-based delivery for all kind of digital content. At the same time, the increasing quality of the digital content is simultaneously increasing the size of the content, which will also increase the requirements for the capacity of the content delivery platforms. At the time of writing this chapter, Content Delivery Networks (CDNs), such as Akamai, have been widely deployed to provide content delivery services. However, CDNs can only cover a small portion of the open Internet, leaving a large chunk of traffic to ordinary web servers (Client-Server delivery). The same applies to IPTV solutions which employ multicast within the IPTV network only (global IP Multicast has not been deployed yet). Therefore, more scalable solutions are required for content delivery services and P2P has been seen as a potential mechanism to complement the other media delivery approaches.

P2P fosters resource self-deployment and self-organisation, reducing operational costs, while still achieving optimized resource utilization for the deployed applications and services. P2P architectures are characterized by the ability to adapt to failures and accommodate transient populations of nodes, while maintaining acceptable connectivity and performance. Although P2P technologies do not directly reduce the overall network traffic in comparison with the traditional client-server approach, the network load is distributed more evenly to the whole network. On the other hand, the IMS, as a control plane technology, addresses primarily issues of heterogeneous technologies for access, addressing schemes, Authentication, Authorization and Accounting (AAA), security and mobility management. These functions are missing in ordinary P2P systems. So our construction of a P2P IMS system brings the benefits of P2P and IMS into a single platform for network operators, service providers and content distributors.

We have prototyped PeerMob P2P IMS, assessing its performance through a purpose-made simulator. Our work highlights the benefits of extending the conventional IMS platform (which is designed according to the client-server paradigm) with the P2P paradigm.

REFERENCES


**ADDITIONAL READING SECTION**


Huang, C. M., & Li, J. W. (2007). One-pass authentication and key agreement procedure in IP multimedia subsystem for UMTS.


Rhea, S., Geels, D., Roscoe, T., & Kubiatowicz, J. (2004). Handling churn in a DHT.


**KEY TERMS & DEFINITIONS**

*Peer to Peer: P2P* is a distributed application architecture in which participants can act both as clients and servers to pursue efficient resource utilization. These participants are able to work without a central server but, in same implementations, make use of servers for purposes such as registration, authentication and indexing.

*IP Multimedia Subsystem: IMS* is an architectural framework for delivering multimedia services over converged all-IP networks. It was originally designed by the wireless standards body known as 3rd Generation Partnership Project (3GPP), as a part of the vision for evolving mobile networks beyond GSM.

*Session Initiation Protocol: SIP* is an IETF-defined signaling protocol, widely used for controlling communication sessions such as voice and video calls over the Internet Protocol (IP).

*Content Delivery Network: CDN* is a system of networked computers containing replicas of the data to be transmitted at various points in the network. When properly designed and implemented, a CDN can significantly improve network utilization and latency.

*Distributed Hash Table: DHT* is a type of decentralized distributed system that provides a lookup service similar to a hash table. Key and value pairs are stored in a DHT and any participating node can efficiently retrieve the value associated with a given key.

*Proxy-Call Session Control Function: P-CSCF* is a SIP proxy, which is the first point of contact for the IMS terminal.

*Serving-Call Session Control Function: S-CSCF* is a SIP server acting as both a SIP proxy server and a SIP registrar server.

*Interrogating-Call Session Control Function: I-CSCF* is a SIP proxy server placed at the edge of an IMS domain to communicate with other IMS domains, hiding the network capacity and topology from the outside.