Mobility Management Alternatives for Migration to Mobile Internet Session-Based Services

Kazutaka Murakami[1], Oliver Haase[1], JaeSheung Shin[2], and Thomas F. La Porta[2],
1 – Bell Labs Research, Lucent Technologies
2 - Penn State University

***Abstract*— Session-based IP applications, such as Internet telephony, are an important component of the emerging mobile Internet. The ubiquitous availability of these services is critical to the success of the mobile Internet. Because all-IP networks will be deployed in phases and current mobile telecommunication systems will be in operation for decades to come, the interworking and migration between current network services and all-IP services is a key problem. In this paper we address seamless roaming for SIP-based services across current cellular telecommunication networks and emerging all-IP wireless networks, such as those using 3G and WiFi networks. We present an abstract mobility model, and map this model to three basic approaches for supporting seamless mobility: a master-slave approach, a federated system, and a unified approach. We discuss the challenges and implementation of an instance of the unified mobility management approach, called the Unified Mobility Manager, and then compare the trade-offs of the three systems using a comparative performance analysis. We conclude that unified mobility management is most efficient if a great deal of interworking is required, and as more users invoke IP-based services; the federated approach is efficient when a single network technology is dominant and data access is limited, but requires sharing of data across networks; the master-slave approach is the least efficient, but is easy to introduce if the number of network types is small.**

Index Terms—**Home Location Register, Interworking, Mobility Management, SIP server**

Appeared in IEEE Journal on Selected Areas of Communications, Vol. 22, No. 5, pp. 818-833.

I. INTRODUCTION

The wireless communications industry is headed towards all-IP mobile networks [1] [2], both through the advance of standards efforts such as the 3rd Generation Partnership Project (3GPP) [3], and with the proliferation of competing technologies such as WiFi. Session-based services, such as voice-over-IP and multimedia communication, are critical for the success of these networks. The fact that it will take several years to finalize standards for all-IP networks, and many more for these networks to be deployed with great density, dictates that there will be a long migration from current circuit-based networks and services to the all-IP environment. In addition, access networks and core networks may evolve at a different pace, requiring further interworking. Figure 1 shows an overview of a likely scenario in which multiple network types must interwork to provide seamless session-based services.

Current cellular telecommunication networks, such as GSM [4], ANSI41 [5], and the emerging UMTS [6], all have similar structures. Their network architecture is precisely defined and controlled. To route calls and deliver services, mobility-supporting switches and databases, called Mobile Switching Centers (MSCs) and Visitor Location Registers (VLRs), respectively, access central databases, called Home Location Registers (HLRs) that maintain service profiles and track mobility. The serving MSCs and VLRs are dynamically assigned based on user location through location update procedures. The HLR and home MSC are assigned based on the device address. These networks use various parts of the Signaling System No. 7 protocol suite [7], such as variants of the Mobile Application Part (MAP) for mobility management, and the ISDN User Part (ISUP) for call control. Calls are delivered based on a device address such as a phone number.

Session-based services in an IP network are increasingly being provided using the Session Initiation Protocol (SIP) [8]. In SIP networks, there is a clear separation of signaling and media flows. While media is exchanged peer-to-peer, call set-up signaling typically traverses a series of SIP proxy servers that help route the call to the far end and that can execute various services. In SIP, calls are addressed to logical SIP URLs; SIP location servers, to which SIP users periodically register their current location, map these addresses to routable IP end points at the time of a call set-up request.

From a cellular telecommunications standpoint, the *IP Multimedia Subsystem* (IMS) [3] is the target network for future IP-based wireless services. The IMS can be considered a special instantiation of the SIP framework, in that the roles of various call control and service execution servers are clearly defined, and some interfaces that are unspecified in SIP have been specified for the IMS architecture. WiFi is an alternative wireless IP technology on which SIP will also likely be used for session control. The number of wireless LAN systems in so-called hot spots is rising sharply; IP voice services over WiFi are expected to be offered in the near future. At the same time, IP voice service is becoming more popular in wireline

networks, and IP enabled mobile backbone networks (core IP networks) are becoming a reality.

One approach to supporting seamless roaming between a mobile telephony network and emerging all-IP mobile networks is to use multi-mode terminals. Multi-mode terminals must satisfy two requirements. First, they must operate over multiple physical levels and low-layer protocols in order to access multiple systems. Second, they must support the application level signaling capabilities to request services from these networks. This is not always practical and may not be possible when considering a mix of mobile and fixed terminals.

Another approach is to allow subscribers to use multiple devices that are accessible via a single *user address*. This may be accomplished by introducing a *user mobility infrastructure* into the network. The user mobility infrastructure enables users to switch between end devices and still get the same, personal services. The user mobility concept requires changing the traditional *m*-to-1 address-to-terminal relationship in cellular networks to a more general *m*-to-*n* relationship between addresses and terminals. The infrastructure must be able to intercept a call to a user address and dynamically map the user address to one or several devices belonging to the addressed user.

Some existing mobile network protocols support user mobility to a limited degree. The ANSI-41 flexible alerting and multiple access hunting features [5] allow 1-to-*n* address-to-terminal relationships; however, the destination terminals are confined to ANSI-41 or PSTN terminals. SIP allows forking to multiple destinations, but requires all signaling messages to be converted to SIP even if only a circuit switched network needs to be involved in the call set-up. In order to efficiently support user mobility in a heterogeneous network environment, a user mobility infrastructure is needed that can accept calls from any packet or circuit core network and allow users to roam into the entire range of mobile access networks.

In this paper we examine three approaches for supporting seamless roaming in a wireless, mobile environment. We emphasize support for SIP terminals as our focus is on enabling the mobile Internet.

The first approach, called *master-slave mobility management* is based on using gateways to provide interworking between disparate networks. This approach requires denoting a *master* protocol (network) and *slave* protocols (networks). An example of this approach is the *GSM/ANSI-136 Interoperability Team* (GAIT) architecture [9]. While this approach may be simple to deploy if the master protocol is based on an existing system, and works efficiently if call requests are predominantly on a single network type, its performance degrades as interworking is required.

The second approach uses a *federation* of servers for providing interworking. In particular a separate user mobility component that contains mappings from personal to terminal addresses, and that interrogates a federation of terminal level location services (HLRs, SIP location servers) for availability information, is required in order to redirect incoming calls. This approach is one of the key rationales behind the current

definition of standard open interfaces for presence and availability information, such as the PAM (presence and awareness management) interface in Parlay [10]. This approach allows a fair amount of isolation between different networks, but may require multiple data accesses over a wide area that might adversely affect performance.

The third approach, called *unified mobility management* integrates mobility management functions of multiple network protocols in one logical entity, and adds user mobility management capabilities on top of this multi-protocol management system. In [11] and [12] we proposed a specific instance of a unified mobility management system through the introduction of a new network element called the Unified Mobility Manager (UMM). In [11] and [12] we discussed the internal structure of the processing components and database of the UMM, respectively. The UMM provides traditional cellular networks with standard HLR functionality, maintaining the registration and location information of the mobile terminal and the value added service settings such as call forwarding activation, barring of incoming calls, and so on. For SIP networks, it works as a SIP location server, maintaining the user's contact addresses together with related service subscriptions. The protocol interworking capability of the UMM facilitates efficient realization of user mobility in a multi-protocol mobile network infrastructure. The UMM has a novel architecture, called Common Operations (COPS), that abstracts mobility management functions so they can be accessed by multiple network interfaces. This facilitates efficient interworking, but incurs moderate overhead if only a single protocol is being used in a session. COPS is described in more detail in Section III.

In [13], we compared the performance of the UMM approach with a type of gateway approach for delivering SIP-originated calls to UMTS devices. In [14], we analyzed the signaling overhead for these scenarios. In this paper we consider the much more general case of multiple combinations of call originating and terminating devices. To provide a strong foundation for this work, we start by presenting an abstract mobility model in Section II. We map this model to the master-slave, federated, and unified mobility management systems. In Section III, we discuss the UMM COPS architecture and implementation in detail. This provides insight as to why this approach is very efficient for scenarios in which a great deal of interworking is required, and shows where overhead is incurred. In Section IV we evaluate the performance of these three systems through a comparative analysis using a technique similar to that in [15] and [13]. The processing values used in the analysis are based on experimental measures taken from an extensive system prototype. We present results showing trade-offs between the systems and the impact of call mixes, types of devices being used (single-mode or multi-mode), and mobility characteristics. In Section V, we conclude.

II. MOBILITY MANAGEMENT PARADIGMS

Most new telecommunications systems will include mobility management mechanisms as a core capability. This holds true not only for emerging cellular networks, but also for IP-based wireline and wireless networks in which a user address may be dynamically bound to any IP address so that a user can receive a service from any IP terminal. Although the protocols of these systems are different in their details, the underlying principles of their mobility management are essentially the same. In this section, we present an abstract mobility management model, shown in Figure 2, which captures the common structure of the mobility management activities in cellular and voice-over-IP networks. This model will be the basis of our development of the COPS architecture discussed in the next section.

A. Abstract Mobility Management Model

The mobility model has three major elements: the *mobility manager*, and the *originating* and *destination Call Control Points* (CCPs). The mobility manager manages the user profile database in which subscribed *service information* is stored. The mobility manager database further maintains the *presence and location* information of a user. This information can be virtually accessed from any serving system to provide ubiquitous service support. The mobility manager also furnishes mobility management control logic. This logic is normally triggered by a request message from an originating CCP. Certain requests from an originating CCP, for example *User Authentication*, can be fulfilled by the mobility manager alone via a database lookup. In many other cases, such as *Registration* and *Call Delivery*, the mobility manager satisfies requests by contacting a destination CCP. These are the two fundamental mobility management operations, and thus we will provide some detailed explanation below.

Table I summarizes the mapping of the elements of the mobility model to elements of specific network types. Note that the mobility manager corresponds to a HLR and Authentication Center (AC) in cellular networks. In IP networks, it is composed of an AAA entity and voice-over-IP servers such as a SIP location server, a SIP registrar and a SIP proxy. The corresponding entities of originating CCP and destination CCP for each protocol depend on the specific call scenario.

The *registration* operation informs the mobility manager of the new location of a mobile user. This information is stored in the mobility manager database. In UMTS location registration, for example, the originating CCP is the new serving VLR. When a mobile moves to a new serving area, the VLR sends a *UMTS Update Location* request message to UMTS HLR (acting as the mobility manager) informing it of its address (Step 1 in Figure 2). In this scenario the destination CCP is the previous VLR. Upon the receipt of Update Location request, the UMTS HLR sends a *UMTS Cancel Location* request message to the previous VLR (destination CCP) to inform it that the mobile is no longer in its serving area (Step 2). After

returning the control back to the HLR (mobility manager) (Step 3), the HLR sends the user's service subscription profile information back to the new VLR (Step 4).

Call delivery begins when a setup request arrives at the originating CCP. In the example of ANSI-41 networks, an *ISUP Initial Address Message* (IAM) is delivered to the home MSC (acting as the originating CCP) of the mobile terminal. The telephone number of the mobile does not contain the information of the current location of the mobile user. Therefore, the MSC first queries the HLR (mobility manager) to determine (1) the location of the mobile user, and (2) a routing number to which it may deliver the call (Step 1 in Figure 2). In case of an ANSI-41 network, the home MSC sends an *ANSI-41 Location Request* message to the ANSI-41 HLR. The HLR can answer the first query through a local database lookup since it stores the user location information in its database. In order to answer the second query, however, it must contact the serving system to obtain routing instructions (Step 2). The HLR sends an *ANSI-41 Routing Request* message to the serving MSC (destination CCP) for this purpose. This routing information is returned back to the home MSC (originating CCP) (Steps 3 and 4) so that the call may be extended to the serving MSC (destination CCP) (Step 5).

Table II summarizes the message exchanges for the key mobility management features. The second column lists the main mobility management functions common to the diverse protocols. These functions are accomplished through message exchanges between the mobility manager and the CCPs, which correspond to Step 1 or Step 2 in Figure 2, and are shown in the third column of the table. The fourth through sixth columns list the messages corresponding to each mobility manager function for ANSI-41, GSM/UMTS, and SIP, respectively. The seventh column will be discussed in Section III.

B. Interworking of Mobility Management Protocols

As mentioned in Section I, seamless roaming needs to be solved on the level of physical access to the network, as well as on the level of *network mobility support.* As far as physical access is concerned, we envisage the use of multiple portable end devices such as cell phones, PDAs, and laptops; some of these devices may have multi-mode capabilities such as a UMTS/ANSI-41 dual-mode phone, which enables global roaming between Europe and North America.

Interworking of mobility management protocols is a key requirement to accomplish seamless inter-technology roaming. Interworking is an activity in which a request from one network (e.g. Step 1 request in the abstract mobility management model) triggers a subsequent request in a second network employing a different mobility management protocol (Step 2 request). Two types of interworking capabilities must be operational to support roaming with a multi-mode phone, one for registration and another for call delivery. When a mobile node moves from one network type, say UMTS, to another type of network, say ANSI-41,

a registration request from the originating CCP of the new network (Step 1 request: *ANSI-41 Registration Notification*) must invoke a registration cancellation request to the previous network using a different protocol (Step 2 request: *UMTS location cancellation*) via the mobility manager. Suppose a call setup request arrives at an originating CCP of one network type, say UMTS. If a dual-mode phone is currently registered in another type of network, for example an ANSI-41 network, a location request from the UMTS originating CCP (Step 1 request: *UMTS Send Routing Information*) must trigger a routing number request to the ANSI-41 destination CCP (Step 2 request: *ANSI-41 Routing Request*).

The introduction of user mobility capabilities further broadens the opportunity of seamless roaming without relying on the availability of multi-mode devices. This type of roaming will be enabled by employing "user addresses" which may be mapped to multiple physical devices, thus enabling personal mobility across devices. When a user address is used for call delivery, protocol-interworking might be required. Suppose the user address belongs to a SIP network. A location request is sent from an originating CCP in the SIP network to the mobility manager, which must recognize the network in which the user is currently roaming. If the user is roaming in a non-SIP network, protocol interworking must be triggered as in the case of a multi-mode phone.

There are several approaches for providing the interworking capabilities described above in a heterogeneous mobile network environment. In order to deploy the interworking capability without modifying other existing network components, multiple protocol-specific mobility manager components or their equivalent are required, one per network type. Their activities must be somehow coordinated to achieve efficient protocol interworking and roaming of end-users across networks. The following sections introduce three representative approaches: *master-slave*, *federated*, and *unified mobility management*.

C. Master-slave Mobility Management

Figure 3 provides a high-level view of the *master-slave mobility management approach*. This model is a generalization of the GAIT approach. GAIT [9] has been proposed for the interoperability of dual-mode GSM/ANSI phones, but can be generalized to operate with more than two protocols. One of the protocols acts as a master protocol (Protocol A in the example of Figure 3). A master mobility manager is introduced in the master network (Network A). This master mobility manager is fundamentally the same as the mobility manager native to the master protocol, potentially with minor extensions to support user mobility. It stores the terminal service and location information, as well as user mobility information such as mappings between user addresses and terminal addresses. The mobility management functions confined to the master network are solely handled by this entity.

For all other networks, which we call slave networks, a slave mobility manager relative to a master

protocol is introduced. For example, Network B has a slave mobility manager of B relative to A (*Slave MM BtoA*). The *Slave MM BtoA* is also viewed as a native mobility manager from the entities in Network B. Although the slave mobility manger will internally process some protocol specific requests, common mobility manager functions (e.g. registration and call delivery) are achieved by interactions with the master mobility manager. The slave mobility manager is viewed as a CCP from the point of view of the master network.

Suppose that a certain mobility management request arises from an originating CCP in the slave network (e.g. Network B). The request is first sent to the slave mobility manager that appears as the native mobility manager from the originating CCP. The request message of Protocol B is translated into the corresponding request message of the master protocol and eventually sent to the master mobility manager. If the request further triggers a subsequent request (e.g. registration cancellation in registration) and if the destination CCP is in a third network (e.g. Network C), the request again goes through a slave mobility manager of the target network (e.g. *Slave MM CtoA*), which translates the request message to the protocol of Network C. Protocol interworking is achieved through this series of translations. Note that the slave mobility manager corresponds to the *Interworking and Interoperability Function* entity in GAIT standard.

A slave mobility manager is required to store user profile information specific to its network type. Furthermore, it maintains a dynamic data mapping for multi-protocol operations. For example, suppose that a dual mode phone that supports Protocols A and B currently roams into Network B (slave network). From the master network (Network A), the *slave MM BtoA* is viewed as the originating CCP of Network A, and the master mobility manager stores the address of this slave mobility manager as the current serving system of the mobile. The slave mobility manager, in turn, stores the address of the actual originating CCP in Network B together with its association to the mobile's identifications in both Network A and B. In this way, future requests from the master mobility manager with the mobile's ID in Network A can be properly translated into the request to the destination CCP in Network B.

The master-slave mobility manager approach is suitable for expanding existing mobile networks to accommodate new protocols with interworking capabilities without major system modifications. By adopting an existing network as a master, existing mobility managers can be readily converted to master mobility managers with the addition of extra call control points that correspond to the slave mobility managers in the newly deployed networks. This approach may compromise long-term system evolution if the master protocol is not carefully selected because once a master protocol is chosen, it is virtually impossible to undo this decision without complete modification of the mobility management infrastructure.

There is a tension between choosing a mature protocol as a master mobility manager, versus choosing a newer protocol that is more expressive. The choice of a mature protocol requires fewer changes to the

network, but because networks are migrating to new protocols, may result in the maintenance of the legacy protocol even after it is no longer used in the core network. The choice of a more expressive protocol enables more services, and may lead to a more efficient network in the long term, but is harder and less efficient to introduce into an existing network. Finally, this approach scatters user profile information across master and slave databases that could be running on different platforms. The consistency of these entries must be ensured through additional integrity measures.

D. Federated Mobility Management

The *federated approach* utilizes the native mobility manager in each network, as shown in Figure 4. Each mobility manager acts independently, and mobility management operations between mobility managers and CCPs are handled separately in each network. In order to realize seamless roaming across heterogeneous networks, this approach exploits a new entity called a *user mobility component* (UMC). The UMC contacts the mobility managers of the participating networks through a *database interface* instead of the communication protocols such as UMTS or SIP. In other words, the mobility manager databases act as federated data servers [16]. The UMC also belongs to one of the participating networks, which we call a *main network*. The UMC participates in communication activities through the protocol of the main network (e.g. Network A in the example of Figure 4).

The UMC must intercept control if an incoming call *may be* destined to more than one network, as is the case when a call is delivered to a multi-mode phone or to a user address rather than a terminal address. Otherwise, if the call is destined to a particular network, the call setup request can be directly sent to this network, and mobility management is handled solely inside this network with the interrogation of its own native mobility manager. In cases in which call interception by the UMC is required, an *Intelligent Network* (IN) mechanism [17] may be used in which the UMC acts as a service control point of the IN architecture. Once the UMC receives the request, it consults with its local user database to determine to which networks the call could be delivered. Then, it interrogates all related mobility manager databases for the availability of the user in their respective networks. Based on this information and the user policies, the UMC makes a destination selection decision and forwards the call to the desired network.

The federated approach solves the system evolution issue with the master-slave mobility manager approach. The mobility manager can be independently introduced per network; with this approach, only an external “gluing” entity (the UMC) is needed to deal with interworking during the evolutionary phases. On the other hand, the federated approach demands an open federated data access interface in the mobility manager. Although this may not be an issue once open interfaces for the interrogation of presence and availability information have been implemented, most current mobility managers such as cellular HLRs do

not allow this kind of access to their subscriber data.

Further obstacles include the realization of registration interworking with dual-mode phones. Registration cancellation across protocols must be triggered through the UMC. Specifically, when a mobility manager detects the registration of a dual mode phone that was not previously registered in its network, the event must be reported to the mobility manager of the previous network through the UMC. This special event notification interface must be built on top of the federated data access interface if a dual-mode phone is handled. Finally, this approach typically requires more message exchanges between entities to achieve protocol interworking because it requires two interrogations to mobility management entities, one to the UMC and another to the native mobility manager.

E. Unified Mobility Management

Figure 5 illustrates the *unified mobility management* approach. A dual-stack UMTS/ANSI-41 HLR [18] falls into this category, although it is confined to two cellular protocols and does not provide the SIP location server functions. In this approach, the mobility manager functions for all protocols are embodied in a single entity. It houses a unified user database where subscriber information pertaining to all protocols is maintained. From each network, this unified mobility manager, or *UMM*, appears to be just a native mobility manager of each protocol.

The UMM performs protocol translation internally. Therefore, it can directly send a Step 2 request of the abstract mobility management model to a destination CCP, even if the Step 1 request is received from a different network than the destination network. For example, when a dual mode phone registers from Network A, the UMM checks its own unified database to determine in which network the mobile phone was previously registered. If it was registered in a different network, for example Network B, the UMM sends a cancellation request directly to Network B. This approach requires fewer message exchanges across network entities. The main challenge to this approach is that the UMM must be structured in a way to effectively support multiple protocols and efficiently achieve protocol interworking. In our instantiation of the UMM, this is accomplished with the introduction of our innovative COPS concept, which is discussed in the next section.

III. COMMON OPERATIONS (COPS) CONCEPT

A. COPS Architecture

There are two main challenges to the UMM approach. First, the system must support multiple protocols and be easily extendable to accommodate new protocols. This meets the requirements of the anticipated heterogeneous network environment with an evolution towards all-IP based networks. Second, the UMM

must efficiently implement a protocol interworking mechanism with any combination of the protocols. A pair-wise interworking mechanism is not acceptable because the complexity of the system will grow with the square of the number of supported protocols. The *Common OPerationS* (*COPS*) architecture [11] addresses these issues.

Figure 6 depicts the structure of the COPS architecture and its application to the UMM. The heart of this architecture is the *COPS interface* that embodies the generalized mobility management methods that are common to many mobile networks. The interface is defined so that it can be applied independently of the underlying mobile network protocol. As discussed in Section II, Table II summarizes the fundamental mobility manager operations applicable to any mobile network. The last column lists the COPS method corresponding to each mobility manager operation. Note that this table only summarizes the COPS operations for registration and basic call set-up; other COPS operations for services such as messaging and call forwarding are also defined but not shown in the table.

The COPS interface is defined between two key components in the COPS architecture, a *protocol-dependent logic server* (*PDLS*) and a *core logic server* (*CLS*). The architecture can support multiple types of networks with different protocols. For each network type, a specific PDLS is defined which terminates the respective protocol interface and implements protocol-specific service logic. When a PDLS receives a request for a service that does not require interworking, it provides the service directly without contacting the CLS. If the received request may *potentially* require interworking with other networks, the PDLS invokes the COPS interface to pass the control to the CLS.

The CLS provides protocol-independent services and determines if interworking is necessary. If required, it communicates with the appropriate PDLS of a target protocol through the COPS interface to accomplish protocol interworking. In the event that there is no interworking, CLS uses the same COPS message to achieve the service, but sends the message to a PDLS of the same protocol type as the originating network.

It is worthwhile to note that the COPS architecture necessitates a PDLS of any protocol to know only its own protocol and the protocol-independent COPS interface. Knowledge of other protocols in the system is not required. Multi-protocol handling and decisions on interworking are solely performed by the CLS. This ensures the ease of introduction of new protocols since it does not require any upgrades to existing PDLS software.

One drawback of the COPS architecture is the fact that a request must be always sent from a PDLS to a CLS if there is a *potential* of protocol interworking, even if the communication ultimately takes place in a single network. This is because the CLS determines if interworking is required. This design choice affords the UMM the benefit of easily adding new protocols to the system. If the interworking decision were made in the PDLS, it would require the PDLS to know the existence of other protocols, thus requiring updates on

all existing PDLS modules. The COPS architecture avoids such major software modifications when a new protocol is introduced. The COPS message overhead is the cost incurred to effectively support multiple mobile protocols with interworking capabilities. In Section IV, we investigate the significance of the COPS penalty cost in various scenarios.

B. Unified Mobility Manager (UMM) with COPS

As shown in Figure 6, the UMM is built on top of the COPS architecture. The UMM also contains a database which maintains the subscriber profile information for the services of all network types to which the user subscribes. Unified profile management in the UMM eliminates redundant data replication for a single user across mobility managers, thus reducing data management and provisioning costs. The UMM subscriber-base also contains the subscriber information supporting user mobility across network technologies, including terminal independent user addresses. Since the UMM acts as a protocol specific mobility management component from the perspective of each individual network, it receives all registration and location resolution requests of the networks for which it is responsible. Consequently, the UMM multi-protocol subscriber base stores the location information for all terminals a user owns. Taking advantage of this combined information about multiple networks, the UMM can transparently provide *user addressing and mobility*, without requiring any changes in other network elements, such as MSCs or SIP servers and user agents. Refer to [12] for more details on UMM database support of user mobility infrastructure.

Figure 7 shows an example of a call flow with protocol interworking and user mobility. In this example, a user owns two types of terminals, one for ANSI-41 and another for SIP, and the latter is assumed to be active at the time when a call set-up request arises. It is further assumed that the telephone number of the user, namely a user address, belongs to the ANSI-41 network. Thus, an *ISUP IAM* is first delivered to the home MSC of the ANSI-41 network (originating CCP) to establish the call to a mobile user. The home MSC then interrogates the ANSI-41 HLR (mobility manager) for the user location with an *ANSI-41 Location Request* message. Since the UMM is acting as the *ANSI-41* HLR, it receives this message. Internally, the location request message is received at the *ANSI-41* PDLS, translated to a *COPS Request Location* (RL) message, and sent to the CLS. The CLS then contacts the UMM database to determine the user associated with the called address, and selects one, or several, of the user's devices to terminate the call. In this example, the user's SIP terminal is selected and the current SIP contact universal resource identifier (URI) is retrieved from the UMM database.

Because the originating CCP belongs to a cellular network, which does not understand a SIP contact address, we must emulate the temporary number allocation process at the destination CCP. The temporary

number allocation process allocates a short-lived routable telephone number to extend a call leg towards the destination CCP, e.g. serving MSC, in traditional circuit-switched cellular networks. The destination CCP must also maintain a mapping from the temporary number to the identity of the final destination, namely a SIP contact address in this cellular-to-IP interworking scenario. The *SIP ALLOCATE* method [19] is proposed as an extension to SIP to accomplish this goal.

The CLS first sends a *COPS Request Route Information* (RRI) message to the PDLS of the destination network, namely the SIP PDLS. Then, the SIP PDLS selects an appropriate PSTN/IP-SIP gateway and sends a *SIP ALLOCATE* message to a *gateway broker* (destination CCP in this scenario) to retrieve a temporary routing number. A mapping from the routing number to the SIP contact address is also stored. The routing number is sent back to the SIP PDLS in a 200 OK response and is eventually delivered back to ANSI-41 home MSC via *COPS RRI*, *RL* responses and *ANSI-41 LOCREQ* response. The home MSC then sends an *ISUP IAM* using the temporary number as its destination to the PSTN/IP-SIP gateway. The gateway initiates a *SIP INVITE* message to the corresponding gateway broker, which completes the call delivery by sending a *SIP INVITE* message with an appropriate SIP contact address obtained from the mapping table.

The gateway selection process in the UMM above can take advantage of the destination device's current location information, and thus select a gateway close to the destination. This is in contrast to both the master-slave and the federated mobility approaches where the originating call is first routed to a statically provisioned gateway, and then the location information in the target SIP network is looked up. Therefore, the integrated approach of the UMM allows for more efficient routing of transport information.

However, the situation is more complicated if the SIP user agent runs on a mobile IP client. In this case, the IP address loses its topological significance. With mobile IP [20], packets to the mobile client are routed to its home network via its home address. In the home network, packets are intercepted by a home agent which then forwards the packets to the mobile node's current Care-of-Address in the foreign network. To make this work, the mobile node (or a so-called foreign agent acting on behalf of it) registers its Care-of-Address with its home agent, when it is roaming.

In order to optimize the gateway selection process when using mobile IP, the UMM needs to know the current Care-of-Address of the mobile client. If Mobile IP route optimization is supported - which is optional in IPv4 and mandatory in IPv6 – the UMM can send a mobile IP binding request packet to the mobile IP client in order to learn the mobile client's current location[1].

In any case, it is important to stress that Mobile IP eliminates the opportunity to perform better gateway

[1] In case of IPv4, the home agent sends the binding update; while in IPv6, the mobile host sends it.

selection than the alternative approaches if no special care is taken, but it will not break the logic of the presented UMM signaling flows.

IV. EVALUATION

In this section, we compare the performance of the three approaches: master-slave, federated, and unified mobility management. As a metric, we choose the call setup signaling latency because this is the most important factor in the telecommunications signaling systems. We first describe the methodology used to compare the performance of the different systems, and then discuss the analytical results. In this discussion we refer to each approach as a *configuration*, and each mix of terminal types, e.g., call originated from an ANSI terminal to be delivered to a SIP device, as a *scenario*. Because the focus of this paper is on providing mobile Internet services, we concentrate on the performance of calls delivered to SIP devices. We varied several parameters, shown in Table III, including the offered call arrival rate (λ_{call}), offered mobility rate (λ_{mm} – the rate at which update location procedures are invoked), call mix (*o_call_mix* - the percentage of ANSI originated calls, and *d_call_mix* – the percentage of calls to ANSI destinations), and addressing type mix (user address or terminal address – *P(um)*).

For this evaluation we assume that an ANSI-41 network serves as the master network in the master-slave configuration, and as the main network in the federated configuration. We make this choice because cellular telecommunication networks are the dominant mobile networking technology today and will be the starting point for any interworking solutions. Likewise, we assume that only single-mode phones are used.

A. Model

We model each component in the network as an M/M/1 queuing system. Each message that arrives at a component, e.g., a MSC, will experience a certain queuing delay and service time. Combined, this is the sojourn time of a message through the network element. Summing the sojourn times of the messages required to complete a task at all elements they pass through yields the time taken to carry out the task. This method is based on that described in [15] with some simplifying assumptions. While we make simplifying assumptions, none affect the main goal of our analysis – comparing the performance of the various configurations under like conditions. By relaxing the assumptions stated below, the absolute values of the figures may change, but the relative performance of the configurations, and hence main conclusions, will not.

To determine the sojourn time of each message we first calculate the load on each element of the system. We assume that service requests (call establishment/release requests) arrive with a Poisson distribution. Because there is significant aggregation in the network between elements, we can assume that the arrival process at each element is also Poisson [15]. For simplicity we assume processing times are exponential.

To determine the load on each element we developed extensive message flows for all possible call scenarios and configurations. These include procedures for call establishment for network originated calls, call release, and location updates. Note that when determining our results, we considered different mixes of call scenarios. Due to space limitations, we do not include all the message flows here; please refer to Figures 7 – 9 for representative flows of SIP-terminated calls for the unified, master-slave, and federated mobility management configurations, respectively.

Figure 7 is explained in Section III; we provide a brief explanation of Figures 8 and 9 here. In Figure 8, a call is placed from a PSTN device through an ANSI-41 core network, to a mobile SIP device using the master-slave network configuration. The ANSI-41 home MSC acts as the originating CCP and terminates the standard ANSI-41 and ISUP protocols for location management and call control, respectively. The ANSI-41 HLR is the master mobility manager. The slave mobility manager terminates ANSI-41 from the master mobility manager, and translates the mobility management messages into SIP for the slave network. The PSTN/IP SIP gateway performs the dual function for the call control messages. The gateway broker and SIP terminal use the standard SIP protocols.

In Figure 9, the same call scenario is shown using a federated mobility management configuration. The User Mobility Component accesses mobility management and service profile information directly from the ANSI-41 and SIP location server databases. The ANSI-41 home MSC uses standard ANSI-41 and ISUP protocols; likewise, the SIP location server, SIP Proxy and SIP terminal use standard SIP.

Consider network element *e* (e.g. MSC, ANSI-PDLS, CLS, etc.) and network configuration *c* (i.e. master-slave, federated, or unified). Let $M^{(e,c,m)}$ denote the set of messages processed at element *e* in configuration *c* for a specific mix of scenarios *m*. From [21], the utilization, $\rho^{(e,c,m)}$, of network element *e* in configuration *c* with scenario mix *m* is given by

$$\rho^{(e,c,m)} = \sum_{i \in M^{(e,c,m)}} \frac{\lambda_i}{\mu_i^{(e)}} \quad (1)$$

where λ_i is the mean arrival rate of messages of type *i* and $\mu_i^{(e)}$ is the mean service rate of message *i* for element *e*. The messages considered include those related to call establishment, release, and location update procedures. The rates for the call procedures are equal as we assume all calls are completed and hence eventually released. The rates for the location update procedures are set independently of the call procedures.

To determine the average call setup latency for each scenario, we need to know the average sojourn time of message *i* through element *e*, $E_e(T^i)$, which is given by

$$E_e(T^i) = \frac{1}{\mu_i^{(e)}(1-\rho^{(e,c,m)})} \quad (2)$$

with $i \in M^{(e,c,m)}$.

In a typical network, there will be a different number of each type of network elements. For example, a unified mobility manager-based network may contain *X* MSCs, *Y* ANSI-PDLSs, *Z* SIP-PDLSs, etc. In a properly engineered network, the ratio between the different network elements is chosen to have all elements equally loaded. For each configuration and scenario under consideration, we calculate the optimal ratio between all involved network elements in this respect. To do this we consider the overall processing load in each network configuration, and the processing load of each element in the network.

To balance the load across each element in configuration *c* we apply more processors to the elements that experience higher load. This has the effect of adjusting the message arrival rates at the individual elements. To determine the effective call arrival and mobility rates at each element *e* for configuration *c* under scenario *m*, while holding total processing resources constant across configurations, we first determine the average utilization of each element if perfect load balancing was achieved, $\rho_{ave}^{(c,m)}$, using

$$\rho_{ave}^{(c,m)} = \frac{\sum_{e \in EL^c} \sum_{i \in M^{(e,c,m)}} \frac{\lambda_i}{\mu_i}}{|EL_c|} \quad (3)$$

We define $\lambda_{eff\_mm}^{(e,c,m)}$ and $\lambda_{eff\_call}^{(e,c,m)}$ to be the effective rate of mobility procedure arrivals and call arrivals at an element, respectively, when the load on the system is perfectly balanced. Therefore, we can also express $\rho_{ave}^{(c,m)}$ as

$$\rho_{ave}^{(c,m)} = \lambda_{eff\_call}^{(e,c,m)} \sum_{i \in M_{call}^{(e,c,m)}} \frac{1}{\mu_i} + \lambda_{eff\_mm}^{(e,c,m)} \sum_{j \in M_{mm}^{(e,c,m)}} \frac{1}{\mu_j} \quad (4)$$

where $M_{call}^{(e,c,m)}$ is the set of call related messages processed at element *e* in configuration *c* for scenario *m*, and $M_{mm}^{(e,c,m)}$ is the set of mobility related messages processed at element *e* in configuration *c* for scenario *m*. Note that the ratio of call arrivals to mobility procedure arrivals is maintained when the system is load balanced. This can be expressed as

$$\frac{\lambda_{eff\_call}^{(e,c,m)}}{\lambda_{eff\_mm}^{(e,c,m)}} = \frac{\lambda_{call}}{\lambda_{mm}} \quad (5)$$

We use equations 3-5 to determine the effective call arrival rates at each element. In order to do a fair comparison between the different network configurations, we have fixed the total processing cost

(resources) for all scenarios. For this purpose, we first determine the processing load on network element *e* for configuration *c* with a scenario call mix *m* , $L^{(e,c,m)}$ by

$$L^{(e,c,m)} = \sum_{i \in M^{(e,c,m)}} 1/\mu_i^{(e)} . \quad (6)$$

To get the total processing load for configuration *c*, $L^{(c,m)}$, we use

$$L^{c,m} = \sum_{e \in EL^c} L^{(e,c,m)} \quad (7)$$

where EL^c is the set of network elements employed in configuration *c*. We use the processing load for the UMM configuration, $L^{UMM,m}$, as our baseline, and hence normalize all other configurations to this load in order to compare networks of the same cost. To do this normalization, we calculate the processing factor, $PF^{(c,m)}$, for each configuration *c* using

$$PF^{(c,m)} = \frac{L^{UMM,m}}{L^{c,m}} . \quad (8)$$

Finally, we apply the normalization factor to $\lambda_{eff\_mm}^{(e,c,m)}$ and $\lambda_{eff\_call}^{(e,c,m)}$ to obtain the normalized effective rate of mobility procedure arrivals and call arrivals, $\overline{\lambda_{eff\_mm}^{(e,c,m)}}$ and $\overline{\lambda_{eff\_call}^{(e,c,m)}}$, respectively, as

$$\overline{\lambda_{eff\_mm}^{(e,c,m)}} = \frac{\lambda_{eff\_mm}^{(e,c,m)}}{PF^{(c,m)}} \quad (9)$$

and

$$\overline{\lambda_{eff\_call}^{(e,c,m)}} = \frac{\lambda_{eff\_call}^{(e,c,m)}}{PF^{(c,m)}} \quad (10)$$

These rates are used to determine the message arrival rates in equation 1 by considering the number of messages per call and mobility procedure arrival. Using equation 2, and the derived message flows, we determine the mean sojourn time for each element in the system. We then determine the time taken to deliver a call to a mobile user by summing the sojourn times of each message required to carry out this task for each element.

For example, consider the UMM system shown in Figure 7. The flow for an ANSI originated call to a SIP terminal is shown. The time taken to deliver a call originated from the MSC to the SIP proxy is the time taken for the following messages to be processed: *ISUP* IAM (MSC), *ANSI-41* LOCREQ (ANSI-41 PDLS), *COPS* RI (CLS), DB Query (User Location DB), DB Response (CLS), *COPS* RRI (SIP PDLS), *SIP* ALLOCATE (Gateway Broker), *SIP* 200 OK (SIP PDLS), *COPS* RRI Response (CLS), *ANSI-41* LOCREQ Response (MSC), *ISUP* IAM (Gateway), *SIP* INVITE (Gateway Broker, a.k.a. a SIP Proxy). Therefore, we sum the sojourn times of these messages to determine the call delivery time.

B. Service Rates

The service rates for each message and network element were determined through a combination of experimental measurement of an extensive system prototype and inference.

In order to eliminate the impact of specific processing platforms on our results, we use relative processing times in this analysis. To do this, we normalize the total processing times for all messages required in an MSC to originate and terminate an ANSI-ANSI call in the unified mobility management configuration to be equal to one, i.e., $L^{(MSC,UMM,ANSI\_ANSI)} = 1$. We chose this element, configuration and scenario as our baseline because it is the current standard. We then normalize all other processing times to these values. We measured protocol processing times and call processing times separately so we could accurately account for the different overheads occurred for SIP communication vs. ANSI communication vs. COPS communication, etc. To determine the processing time of a message in an element, we simply summed the corresponding protocol time and call processing time. For example, the processing time of a SIP INVITE at a SIP PDLS is SIP_TIME + SIP_PDLS_TIME time where SIP_TIME is the protocol processing time and SIP_PDLS_TIME is the call processing in the PDLS associated with SIP.

Table A.1 in the Appendix lists the processing times, and how they were determined, for each protocol and element.

C. Results

We examined a system with a combination of ANSI and SIP terminals so calls to and from both types of terminals are loading the system. The results presented show the performance of the SIP terminated calls. As previously discussed, we varied the parameters shown in Table III. Interworking occurs whenever a call is terminated on a different network type than on which it was originated. Informally, the percentage of calls requiring interworking is simply the percentage of ANSI-41 originated calls terminating on SIP devices plus the percentage of SIP originated calls terminating on ANSI-41 devices. More formally, the percentage of interworking, I, is:

$$I = o\_call\_mix(1 - d\_call\_mix) + (1 - o\_call\_mix)(d\_call\_mix). \quad (11)$$

Figure 10 shows the scaling properties of the three configurations for various call mixes. In addition, three sets of curves are presented for the federated configuration for different probabilities that the call is delivered via a *user address* (SIP URI) as opposed to a *device address (P(um)=1* corresponds to all calls being addressed to a user address, thus invoking user mobility, while *P(um)=0* corresponds to all calls being address to device addresses, thus invoking no user mobility). Although user addresses will be more

likely in cases when SIP terminals are used, they will also become prevalent when multi-mode devices are used, or users have more than one type of device. Both the UMM and master-slave configurations are not affected by the addressing mechanism because they have integrated solutions for address resolution. Note that the master-slave configuration is the least efficient in terms of performance in almost all cases. This is because we assume we always have an ANSI-41 master network requiring some type of protocol conversion.

In Figure 10a, we illustrate a scenario in which 80% of both call origination and termination are via ANSI-41 phones, while the remaining 20% are via SIP phones. This is typical of the time period at which migration to SIP phones is in the early stages. We see that federated mobility management is most efficient in these cases. This is because in this scenario very little interworking is required (about 30% from equation 11), and due to the network isolation properties discussed in Section II, the federate approach is acting almost as a pure ANSI-41 system. Note that the federate performance degrades rapidly as the percentage of calls being delivered to a user address increases (P(um)). This is because the federated system must now perform multiple data accesses and search multiple possible networks (see Figure 9) to deliver the call. The UMM is not as efficient in this scenario because of the extra processing incurred with ANSI-41 signaling, and the fact that its interworking capabilities are not taken advantage of.

In Figure 10b, we illustrate a scenario in which call origination and termination are evenly split between ANSI-41 and SIP terminals. In this scenario, the federate approach is only the most efficient if no user addressing is used (P(um) = 0), which is highly unlikely. It is more likely that if 50% of calls are either being originated or terminated by SIP devices, a significant amount of the calls would be addressed to a user address because the devices will be either pure SIP or multi-mode. In these cases, P(um) will be greater than 0.5, in which case it is outperformed by the UMM. This is because the UMM has capabilities for address resolution integrated into the CLS. Furthermore, since the main network in the federated configuration is an ANSI-41 network, more protocol conversion is required as the P(um) value and the number of calls in which SIP is involved grow. In Figure 10c, we illustrate a scenario in which most users have migrated to and use SIP phones (90% of call originations and terminations are via SIP phones). Under these conditions, the UMM outperforms the federated configuration unless P(um) is very low which is not realistic.

Another interesting case, not shown here, is when most calls are originated by ANSI-41 networks but terminated on mobile SIP devices. This condition may exist as mobile users move towards mobile Internet technology but interwork with the large legacy base of wireline PSTN phones. In this case, we would expect most calls to be delivered via a user address. Our results show that the UMM scales better than the federated case under these circumstances due to its integrated support of interworking and user address

resolution. In summary, from Figure 10, the federated approach is the most efficient if ANSI-41 and terminal addressing is the dominant technology used because the system effectively operates as a pure ANSI-41 system. The UMM approach is the most efficient when interworking is required and as more calls are delivered via a user address, because these functions are integrated into the CLS via the COPS interface.

Figure 11 shows the impact of mobility on the different configurations for the case of devices being equally divided between SIP and ANSI-41. There is little impact on the performance of all the scenarios, with the performance of all the systems improving as the mobility rate decreases, i.e., more calls are made per move.

Figure 12 shows the impact of the call mix on the different configurations when none of the systems are in an overloaded state. Recall from Table III that *destination call mix* refers to the percentage of ANSI-41 terminated calls, and *originating call mix* refers to the percentage of ANSI-41 originating calls.

Figure 12a shows the performance of the master-slave configuration. We see that this configuration performs poorly when most calls are delivered to SIP terminals (low values of destination call mix). That is because interworking is required with the core ANSI-41 network which results in significant messaging overhead. The master-slave approach is more efficient for ANSI-41 terminated calls (high destination call mix) because the call is being delivered via the master protocol and hence less messaging is required.

Figure 12b shows the performance of the federated configuration. In this configuration, ANSI-41 terminated calls require incur the most delay because of the extra processing associated with ANSI-41. The most overhead is incurred when all calls are originated by SIP users and delivered to ANSI-41 terminals because of the interworking required and name translation.

Figure 12c shows the performance of the UMM configuration. The UMM is least efficient when the system is pure ANSI-41 because of the high processing cost of this protocol. It is more efficient for SIP originated calls because it is effective at performing address translations.

Figure 13 shows the performance difference between the UMM and federated configurations. The surface represents the values of call delay of the UMM configuration minus the call delay of the federated configuration. Therefore, when the value on the vertical axis is below 0, the UMM configuration is more efficient. As shown in the figure, even though the performance of the federate configuration improves as more SIP terminals are involved (Figure 12b), the UMM outperforms the federated approach as more SIP terminals are involved in calls. This is because the federated system requires significant messaging via the UMC to coordinate the networks (see Figure 9) which reduces the benefit of the simplified SIP processing.

V. Discussions and Conclusions

The ultimate goal of the systems presented in Sections II and III of this paper are to enable the smooth migration to an all-IP services-based network, in particular, a network using SIP for session control. Given the current state of wireless networks, and the investment already made in current telecommunication networks, the migration to all-IP networks will occur over many years. Therefore, the systems must be evaluated on their ability to be easily introduced into the current networks, provide efficient interworking during the migration period, and eventually efficiently support services in a predominantly IP environment.

The master-slave approach is easy to introduce into existing systems because, while it requires new equipment to be deployed in new networks, most existing equipment is untouched. The disadvantage of the master-slave approach from a deployment point-of-view is that user data must be spread over multiple networks so that services may be provided in a seamless fashion. This requires the use of multiple provisioning systems to support a single user that roams between network types. Also, the performance of the master-slave configuration is not as efficient as the other approaches.

The federated approach is very efficient when there is a single dominant network protocol and little data access is required to deliver a call. This is because the federated approach affectively isolates networks, so if a call is originated and terminated in the same network type, the system behaves as a pure native mode system. This advantage dissipates as more data access is required, for example to resolve a user address or to access data to provide interworking functions, such as determining which of multiple possible terminals or networks to route a call. This is because the federated system is comprised of specialized databases and therefore multiple data accesses must be made to support these services. Unfortunately, these types of functions are critical to supporting SIP, and therefore the federated system does not fare well under the conditions likely to be present as the migration period matures. In addition, due to the nature of a federated database system, user data tends to be spread over multiple databases which require synchronization and perhaps multiple provisioning systems.

The unified mobility management approach is very efficient for cases in which interworking and user mobility support, e.g., address resolution, are required. This is because the UMM treats these as core services, and thus integrates them into its basic capabilities via the CLS and unified database. This exemplifies the benefits of identifying key services that are required for most call requests and implementing them efficiently as part of the base system. The cost of this approach is that, if the services are not required, there is some extra overhead. This is evident when examining the performance results of the pure ANSI-41 case with the UMM. The overhead of separating the ANSI-41 PDLS and CLS via the COPS interface causes the system to be less efficient in this case. The UMM has other benefits for deployment as well. For basic services supporting SIP and interworking, only a single database is used, so

user provisioning will be easier. Also, because the UMM supports native interfaces to several networks, and handles all interworking functions, it is the only new network element that needs to be introduced into a network to enable roaming; no other elements require modification.

In summary, the unified mobility management approach is highly suitable for supporting the migration to all-IP mobile networks because it implements the basic functions to support interworking in an integrated fashion. This approach may have limitations as more advanced services are introduced and multiple providers provide services on a single call, because the management of data in a single database may become impractical. In these cases, the combination of using the UMM for basic session services and interworking, and a federate approach for providing access to other service-specific data, may be an attractive solution.

ACKNOWLEDGMENT

The authors would like to acknowledge K. Sheta, K. W. Chen and P. Doshi who provided much of the implementation and great insight on interpreting the data gathered via the performance measurements.

REFERENCES

[1] G. Patel, S. Dennett, "The 3GPP and 3GPP2 Movements Toward and All-IP Mobile Network," *IEEE Personal Communications Magazine*, Vol. 7, No. 4, pp. 62-64, August 2000.

[2] D. Plasse, "Call Control Scenarios in the "All-IP" UMTS Core Network," Proc. Of *IEEE PIMRC'2000*, London, UK, September, 2000.

[3] 3rd Generation Partnership Project, "Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 5)," (version 5.5.0), 3GPP TS 23.228, Jun. 2002.

[4] ETSI, "Digital Cellular Telecommunications System, Network Architecture," GSM 03.02 version 7.1.0, ETSI, Sophia Antipolis, France, Feb. 2000.

[5] TIA/EIA, "Cellular Radio Telecommunications Intersystem Operations," TIA/EIA ANSI-41-D, TIA, Arlington, VA, Dec. 1997.

[6] ETSI, "Universal Mobile Telecommunications System (UMTS), General UMTS Architecture, 3G TS 23.101, ETSI, Sophia, Antipolis, France, Jan. 2000.

[7] ITU Recommendations Q.700-Q.795, "Specification of Signaling System No. 7," 1989.

[8] M. Handley et al, "SIP: Session Initiation Protocol," IETF RFC 2543, March 1999.

[9] GSM/ANSI-136 Interoperability Team (GAIT), "Network Interworking between GSM MAP and ANSI-41 MAP," PN-4857, TR-46, Dec. 2000.

[10]ETSI ES 202 915, "Open Service Access (OSA); Application Programming Interface (API)," v1.2.1, June 2003.

[11]R. Isukapalli, T. Alexiou, and K. Murakami, "Global Roaming and Personal Mobility with COPS Architecture in SuperDHLR," Bell Labs Technical Journal, vol.7, no.2, pp.3-18, Nov. 2002.

[12]O. Haase, M. Xiong, and K. Murakami, "Multi-Protocol Profiles to Support User Mobility Across Network Technologies," *2004 IEEE International Conference on Mobile Data Management*, *under submission*.

[13]O. Haase, K. Murakami, and T.F. La Porta, "Unified Mobility Manager – Enabling Efficient SIP/UMTS Mobile Network Control," *IEEE Wireless Communications Magazine*, vol. 10, no. 4, pp. 66-75, Aug. 2003.

[14]J. Lennox, K. Murakami, M. Karaul, and T.F. La Porta, "Interworking Internet Telephony and Wireless Telecommunications Networks," *ACM Computer Communications Review*, vol.31, no.5, pp.25-36, Oct. 2001.

[15]G. Willman, and Paul Kuhn, "Performance Modeling of Signaling System No. 7," *IEEE Communications Magazine*, Vol. 28, No. 8, pp. 44-56, July 1990.

[16]A. P. Sheth, J. A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, Vol. 22, No. 3, pp. 183-236, Sept. 1990.

[17]T. Magedanz, and R. Popescu-Zeletin. *Intelligent Networks*. International Thomson Publishing, 1996, ISBN 1850322937.

[18]S. Kim, H.J. Cho, H.H. Hahm, S.Y. Lee, and M.S. Lee, "Interoperability between UMTS and CDMA2000 Networks," *IEEE Wireless Communications Magazine*, vol.10, no.1, pp.22-28, Feb. 2003.

[19]T. Alexiou, J. Lennox, and K. Murakami, "The SIP ALLOCATE method," IETF Internet Draft, <draft-alexiou-sipping-allocate-00.txt>, Feb. 2002.

[20]Perkins, C., "IP Mobility Support," *IETF RFC 2002*, October, 1996.

[21]L. Kleinrock, *Queuing Systems*, Vol. 2: Computer Applications, New York, Wiley Interscience, 1976.

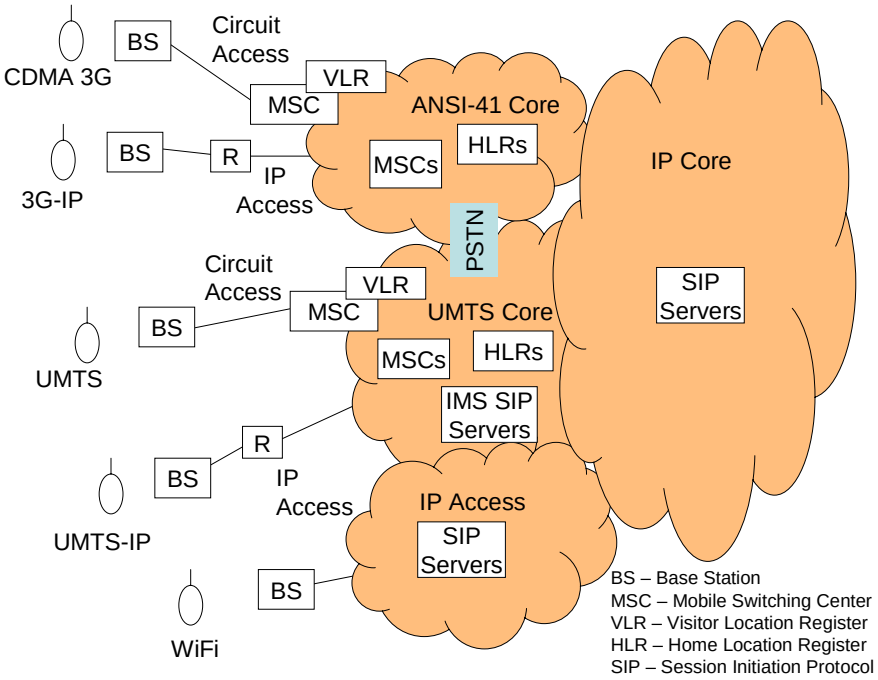


Figure 1. Heterogeneous network environment in the future

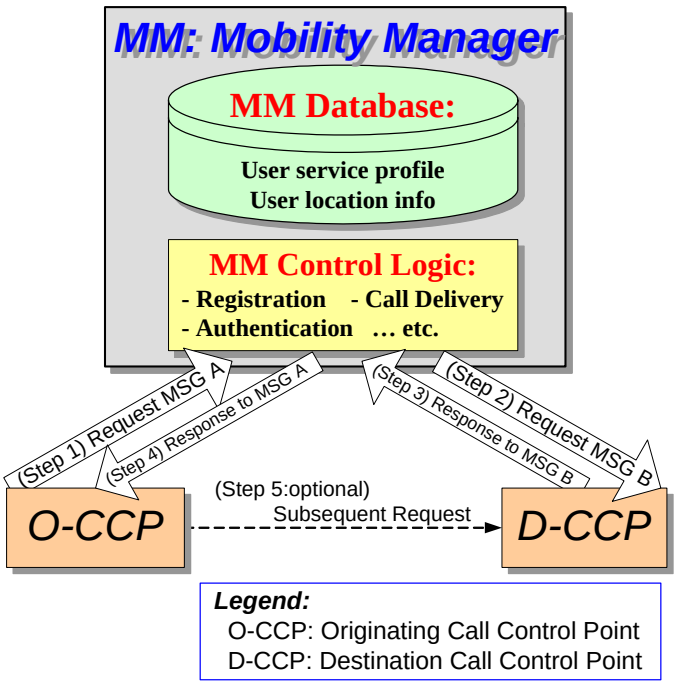


Figure 2. Abstract Mobility Management Model

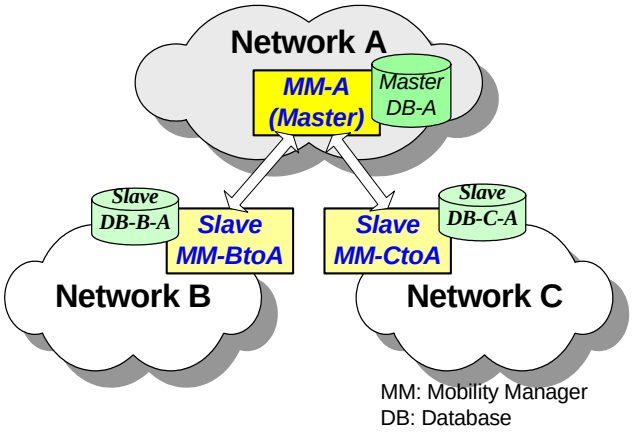


Figure 3. Master-Slave Approach

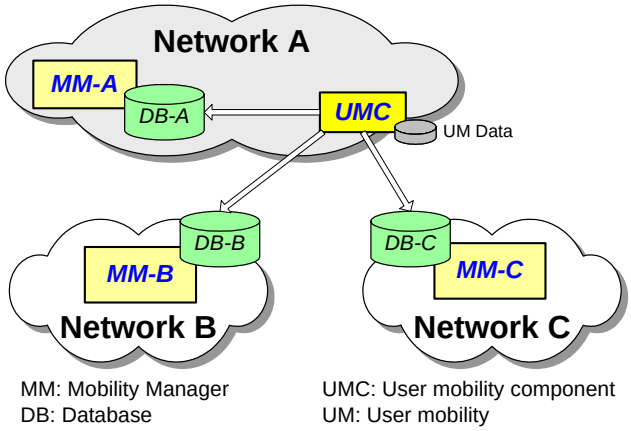


Figure 4. Federated Approach

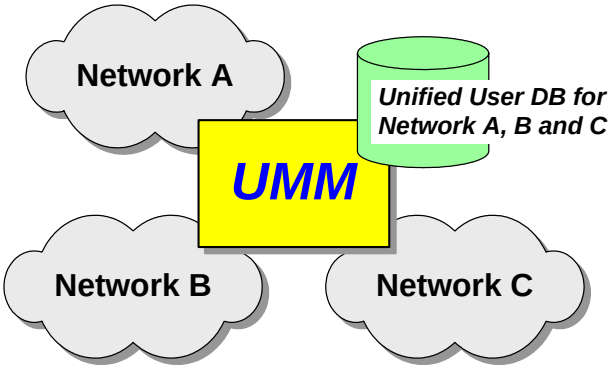
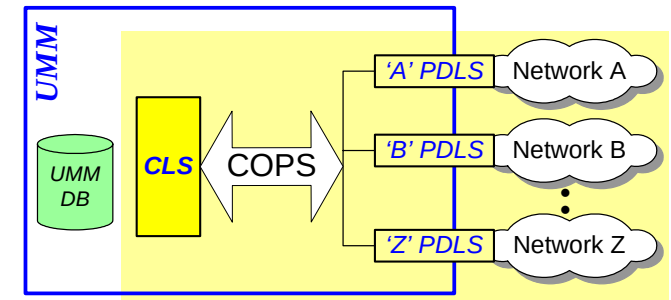


Figure 5. UMM Approach



CLS: Core Logic Server
PDLS: Protocol Dependent Logic Server
COPS: Common OPerationS

Figure 6. COPS architecture and its application to UMM

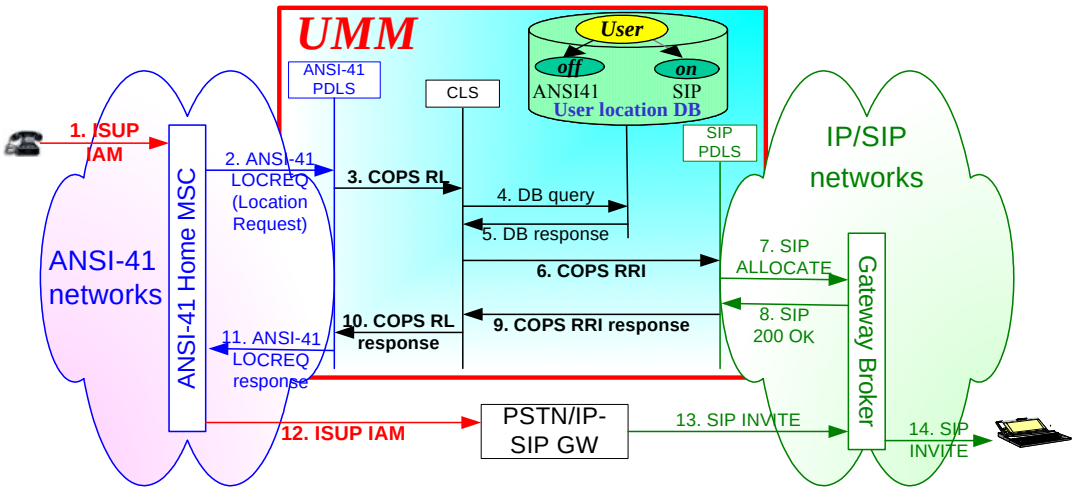


Figure 7. UMM Call Flow Example: ANSI-41 to SIP

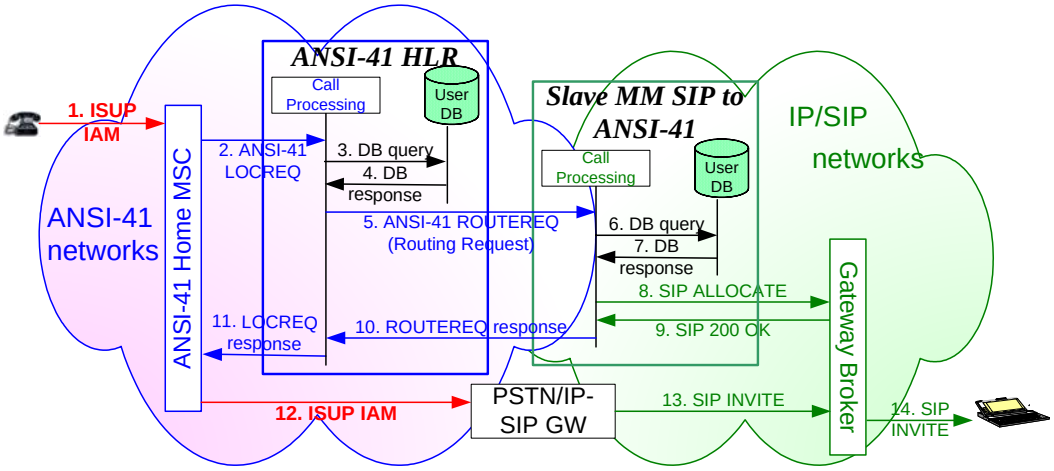


Figure 8. Master-Slave Call Flow Example: ANSI-41 to SIP

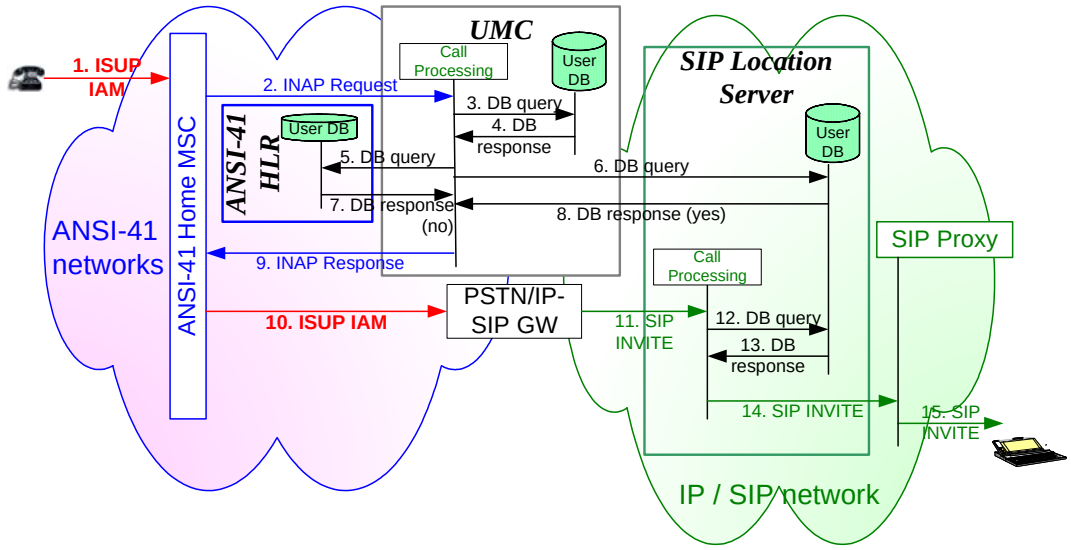
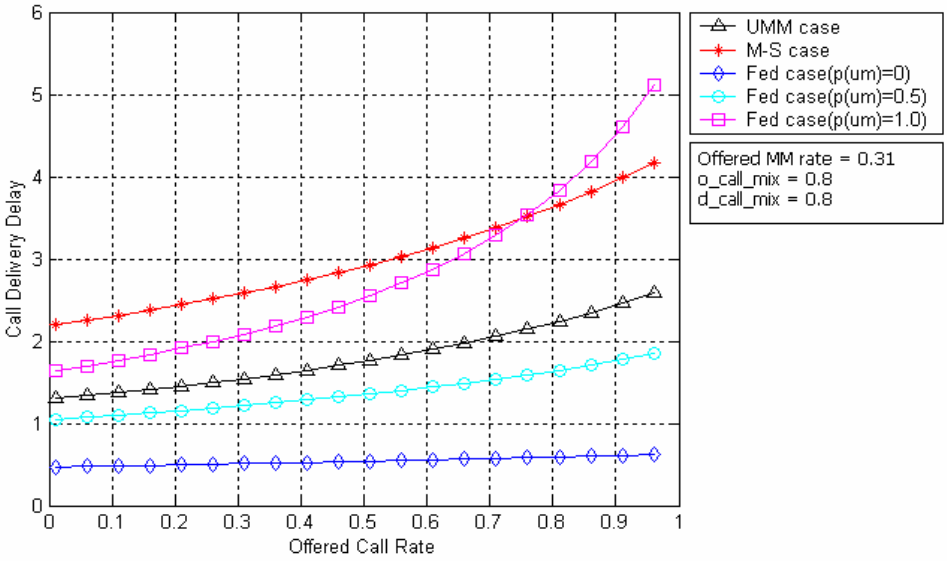
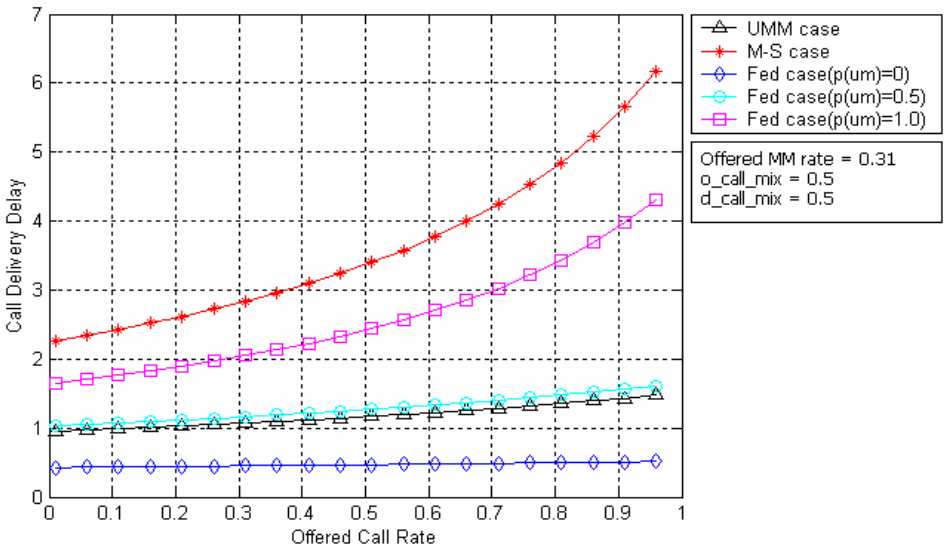


Figure 9. Federated Call Flow Example: ANSI-41 to SIP





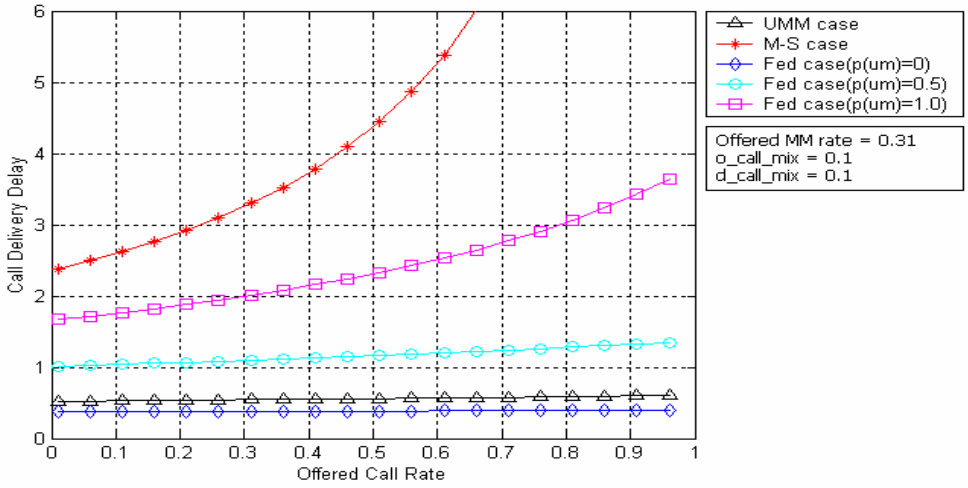


Figure 10. Call Delay vs. Call Arrival Rate – top(a); middle (b); bottom (c)

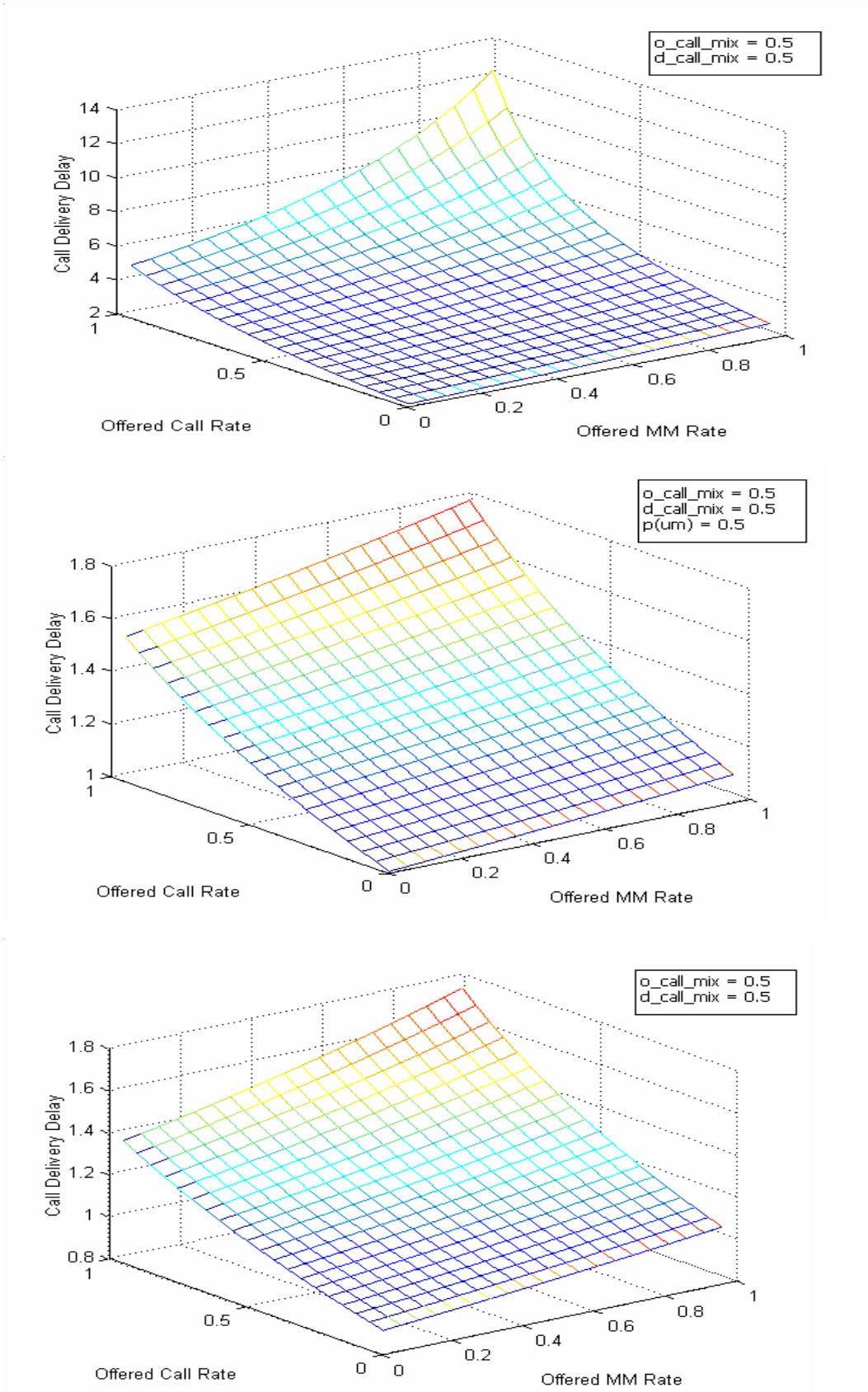


Figure 11. Call Delay vs. Call Mobility Ratio – top (a): Master-Slave; middle (b): Federated; bottom (c): UMM

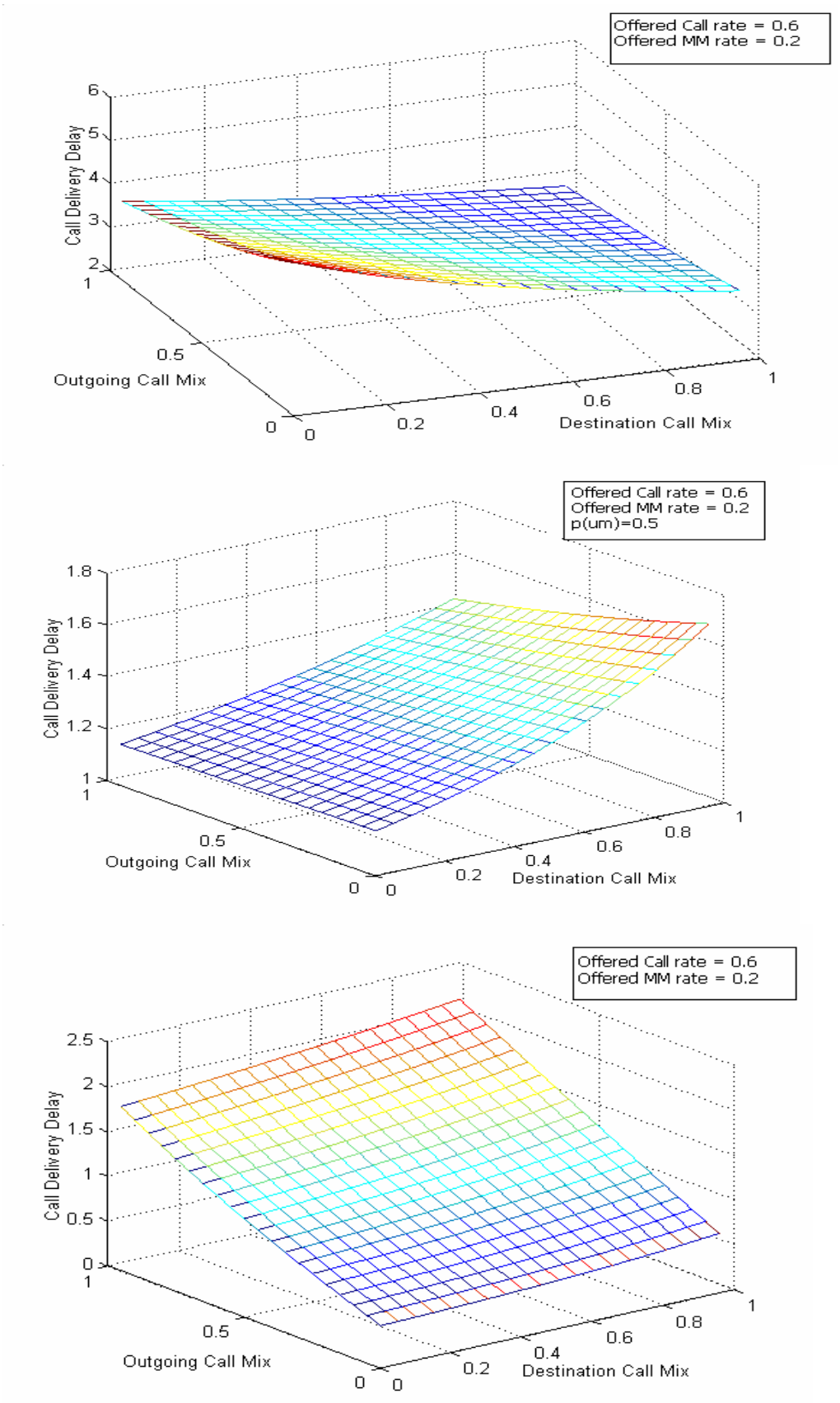


Figure 12. Call Delay vs. Call Mix – top(a): Master-Slave; middle (b): Federated; bottom (c): UMM

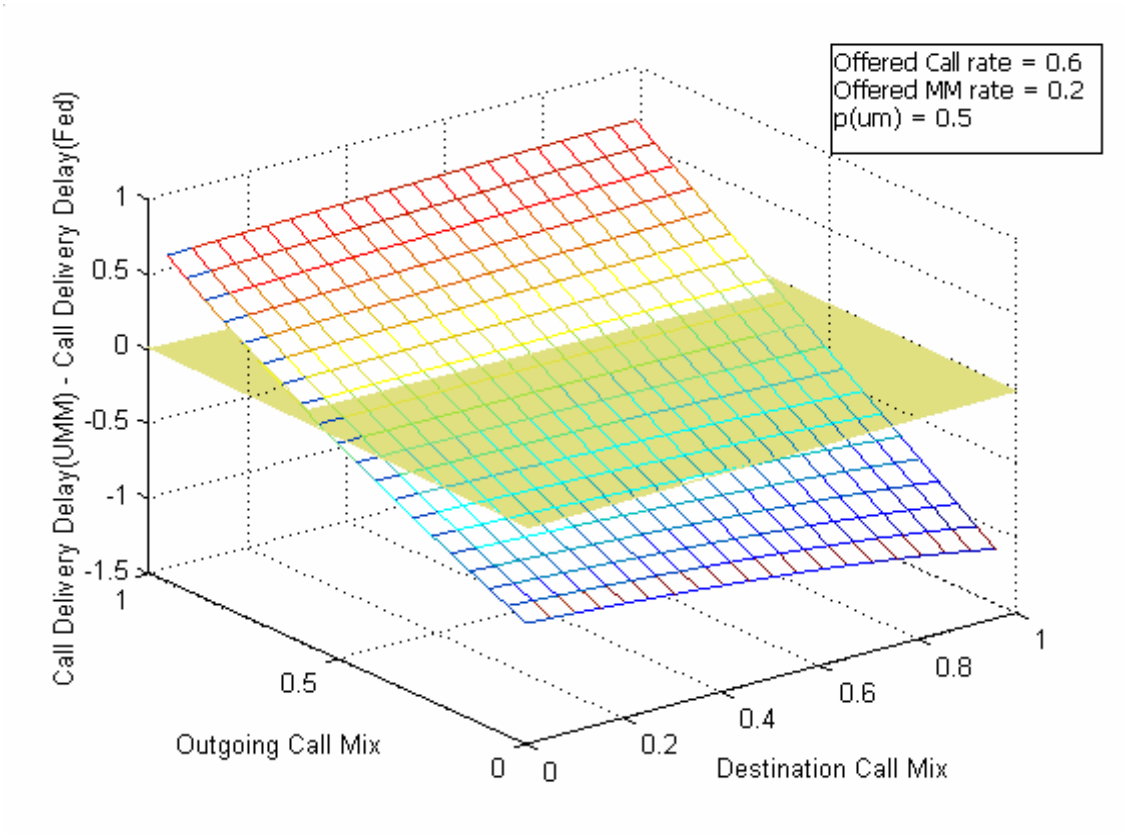


Figure 13. Comparison of UMM vs. Federated Mobility Management

Entity	Function	ANSI-41	UMTS / GSM	SIP
Mobility manger		HLR / AC	HLR / AC	SIP server
Originating CCP	Registration	New VLR		Proxy
	Call Delivery	Home MSC	Gateway MSC	Proxy
Destination CCP	Registration	Previous VLR		NA
	Call Delivery	Serving MSC		GW Broker

Table I. Entities corresponding to mobility manger and originating/destination CCP

Feature Name	Mobility management function	Step in Fig. 2	ANSI-41 Message	GSM / UMTS Message	SIP Message	COPS Message
Registration	Register the new location of a user	Step 1	Registration Notification	Update Location	REGISTER	RT: Register Terminal
	Inform the CCP of the departure of a user from its serving area	Step 2	Registration Cancellation	Cancel Location	N/A	CTR: Cancel Terminal Registration
Call Delivery	Interrogate the current location of a mobile user	Step 1	Location Request	Send Routing Information	INVITE	RL: Request Location
	Retrieve a temporary routing information from the serving system (destination CCP)	Step 2	Routing Request	Provide Roaming Number	ALLOCATE	RRI: Request Route Information

Table II. COPS messages and the equivalent messages in various networks

Variable	Description
λ_{call}	Offered call arrival rate to system including calls of all types
λ_{mm}	Offered location update arrival rate to system including terminals of all types
o_call_mix	Percentage of calls originated from an ANSI-41 network
d_call_mix	Percentage of calls to ANSI-41 destinations
P(um)	Percentage of calls delivered via a user address (as opposed to terminal address)

Table III. Analysis Parameters

Appendix A

Component	Functions included in processing time	Normalized Processing Time (usec)	Method
SS7_TIME	Lower layers of SS7	0.0357	Measured
ISUP_TIME	ISUP Processing	0.0179	Inferred – half the lower layers of SS7
SIP_TIME	SIP protocol processing	0.0357	Inferred – similar to lower layers of SS7
ANSI_PDLS_TIME (including TCAP)	Higher layers of SS7 (TCAP) plus call processing	0.1659	Measured
SIP_PDLS_TIME	Call processing	0.0553	Inferred - ANSI PDLS minus TCAP
CLS_TIME	Call processing	0.0200	Measured
COPS_TIME	Messaging	0.0059	Measured
UMM_DBCOM_TIME	Messaging to the UMM database	0.0082	Measured
UMM_DB_READ_TIME	Reading UMM database	0.0762	Measured
UMM_DB_WRITE_TIME	Updating UMM database	0.1879	Measured
FED_DB_TIME	Accessing Fed database	0.0687	Inferred based on the fact this should be slightly simpler than the UMM database
ANSI41_MSC_TIME	All processing in MSC	0.1310	Inferred – see equation A.1
FED_UMC_TIME	All processing in UMC	0.1659	Similar to ANSI PDLS Processing
BROKER_SIP_TIME		0.0553	Similar to SIP PDLS Processing
GATEWAY_TIME	Slave MM processing	0.0891	Inferred – see equation A.2
SIP_PROXY_TIME	All processing	0.0868	Inferred – see equation A.3

Table A.1 Processing Times

The formula used to calculate the processing time for each message in an MSC is shown in equation A1. The first term accounts for processing associated with receiving signaling messages and performing the related processing. The MSC receives six signaling messages per call, five of which are relatively simple (those that have a processing time similar to *SIP_PDLS_TIME* + *CLS_TIME* with additional ISUP processing time) and one that requires TCAP processing (*ANSI_PDLS_TIME* + *CLS_TIME*). Therefore, we take this weighted average as the processing time for these messages. We multiply this term by 1.1 to account for the extra service logic that executes in MSCs. The second term accounts for the fact that the MSC must access an internal database one time for each call setup (requiring about the same time as *DF_FED_TIME*). We divide this cost over the six messages to arrive at an average processing time per message.

Equations A2 and A3 are similarly derived.

$$\frac{(5(ISUP\_TIME+SIP\_PDLS\_TIME+CLS\_TIME)+(ANSI\_PDLS\_TIME+CLS\_TIME))x1.1}{6} + \frac{FED\_DB\_TIME}{6} \quad (A.1)$$

$$(SIP\_PDLS\_TIME + CLS\_TIME) + \frac{FED\_DB\_TIME}{5} \quad (A.2)$$

$$(SIP\_PDLS\_TIME + CLS\_TIME) + \frac{FED\_DB\_TIME}{6} \quad (A.3)$$