ABSTRACT

SINHA, PRABHAS RANJAN. SIP signaling in Mobile electronic Transaction. (Under the direction of Dr Arne Nilsson)

WAP and SIP are two different technologies to provide telephone services at present. Both technologies employ different approaches to establish connection, transmit data and terminate connection.

This thesis focuses on establishment of SIP signaling for Mobile electronic Transaction and tearing off the connections. WAP and SIP protocols have been analyzed and finally the possible compatibility and pair wise mapping between both have been discussed. Features of both the protocols have been discussed and uses of a service broker between them are outlined.

Department of
Electrical and Computer Engineering
NC STATE UNIVERSITY

SIP Signaling in Mobile Electronic Transaction

Master's thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science
By

Prabhas Sinha

Approved By:

Dr Edward Gehringer
Associate Professor
Department of CSC

Dr Peng Ning
Assistant Professor
Department of CSC

(Chair of Advisory Committee)
Dr Arne Nilsson
Professor
Department of ECE

Raleigh, NC (USA)
07/24/2002

Biography

Prabhas Sinha is a research assistant and a master's student in the Department of Electrical & Computer Engineering at North Carolina State University in Raleigh, USA. He graduated with a BS degree in Electronics and Communication from Regional Engineering College (REC), Jaipur, India in June 1995. From 1995 to 2000, he worked for companies like LEC Denmark, HCL Perot Systems in UK & India and Motorola in India, primarily into object oriented software development & consulting. His main area of interests includes distributed object computing, object oriented design, design patterns and application frameworks and his current research interests focuses on wireless, SIP, WAP, 3GPP related topics.
He was born on February 14th 1971. His parents live in Bokaro Steel City (India) where his father is a lawyer. His sister lives in Madras (India) with her husband and two wonderful daughters. Prabhas has been invited to attend London Business School after graduation and can be reached at prabhas@hotmail.com.

Acknowledgements

First and foremost, I would like to thank my advisor Professor Dr Arne Nilsson for providing me with an opportunity to work on this thesis under his able guidance, providing me with all the resources, for his support, patience, constant encouragement and his confidence in me. I extend my sincere thanks to Vijay Gurbani of Wireless Networks Group/Internet Software and Services at Lucent Technologies, Naperville, Illinois for spending his valuable time in reviewing my work and giving lots of timely feedback. I would also like to extend my sincere gratitude to Dr Edward Gehringer who has put up with me for such a long time and have encouraged me every time during my stay at North Carolina State University. I also express my thankfulness to Dr Peng Ning for agreeing to be on the committee.

Words would seem superfluous if I take it to express my love and thankfulness to my parents for their unconditional love, guidance and continual encouragement since my childhood. I thank them for everything they have done for me. Last but not the least, I would also like to thank my friends who made my stint at NC State more enjoyable. My sincere thanks to Vijay Iyer, Girish Kulkarni, Sandeep Chandra, Prashant Gupta and Warren D’mello.

Table of Contents

Acronyms

API : Application Programming Interface
ASB : Application Service Broker
CSCF : Call Session Control Function
DNS : Domain Naming Service
DTMF : Dual Tone Multi-Frequency
GSM : Global System for Mobile Communications
HSS : Home Subscriber Server
HTTP : Hyper Text Transport Protocol
I-CSCF: Interrogating-CSCF
IMS : IP Multimedia System
IP : Internet Protocol
PAP : Push Access Protocol
P-CSCF: Proxy-CSCF
PTD : Personal Trusted Device
S-CSCF: Serving-CSCF
SDP : Session Description Protocol
SIP : Session Initiation Protocol
SMTP : Simple Mail Transport Protocol
MeT : Mobile Electronic Transaction
TCP : Transport Control Protocol
TLS : Transaction Layer Security
UA : User Agent
UAC : User Agent Client
UAS : User Agent Server
UDP : User Datagram Protocol
URI : Uniform Resource Identifier
URL : Uniform Resource Locator
WAE : Wireless Application Environment
WAP : Wireless Application Protocol
WSP : Wireless Session Protocol
WTP : Wireless Transport Protocol
WTLS : Wireless Transaction Layer Security
WDP : Wireless Datagram Protocol
WTA : Wireless Telephony Applications
WTAI : Wireless Telephony Applications Interface
WWW : World Wide Web
3GPP : Third Generation Partnership Project

List of Tables

List of Figures

1.Introduction

SIP (Session Initiation Protocol) is being used for IP telephony and a great deal of work has been taken up by IETF, ITU and ETSI because it is completely integrated with other Internet and web protocols. SIP is a text-based, lightweight, application layer signaling protocol for multimedia call control. SIP is expected to fundamentally change networking services and the communication habits of the billions of users worldwide in the coming years, with ramifications for the business models of firms in the entertainment, e-commerce and M-commerce industries, in the areas of broadband services, wireless, enterprise applications, network appliances and overlay SIP services. One of these communication services, which will derive immense interoperability with SIP, is MeT (Mobile Electronic Transaction). The advent of wireless telephony, its unprecedented success and millions of PTDs (Personal Trusted Devices) to be marketed in the time to come implies that a huge number of users worldwide will be a party to mobile e-business. MeT will foster a massive growth in mobile e-business arena and we foresee that MeT and SIP will come together and augment their coherent growth across multiple access technologies and seamlessly evolving scenarios. There could be numerous M-commerce applications that would need a session to be established, multimedia transport, and presence etc. where SIP fits very well.
This is a Master’s thesis work done at the Department of Electrical & Computer Engineering, North Carolina State University in Raleigh, USA towards the fulfillment of the award of Master’s degree. The objective of this thesis work is to study SIP, WAP & MeT, understand how these technologies work and propose a framework in which they can interoperate. As these technologies get deployed and penetrate in the market place, it is conceivable that they will merge or adopt the best traits of each other. This report first delved into the work, which has already been done with respect to this master’s thesis topic. To the date of submission of this report there is no explicit reference to any publications stating the interoperability of SIP and MeT. However a lot of work is being done by 3GPP for SIP usage in wireless domain. This thesis work is different in the following aspects:

- It focuses primarily on inter-networking between SIP and WAP protocols3GPP work on SIP is focused more on telephony applications but doesn’t talk about other applications. An effort to converge SIP-WAP protocols will facilitate more benefits to the end users. As a end user one would like to have one wireless device with all applications on it, be it telephony or banking or messaging services or web browsing or mobile electronic transactions.3GPP advocates usage of SIP stack in the wireless device. Later part of this thesis indicates that with SIP-WAP protocol inter-networking it will not be required to have a SIP stack in the wireless device.

2. Wireless Application protocol

Wireless Application Protocol, as the name suggests is a technology that assists in developing applications for wireless devices in order to enable them to access internet, telephony services and even intranet. WAP's architecture is based on the existing OSI model but it does not use all of the layers and instead has defined its own set of protocol layers that are lightweight. The following are some of the constraints of wireless environment:

- Low bandwidth
- High Latency
- Less connection stability

Hence WAP layers have been optimized to meet these requirements.

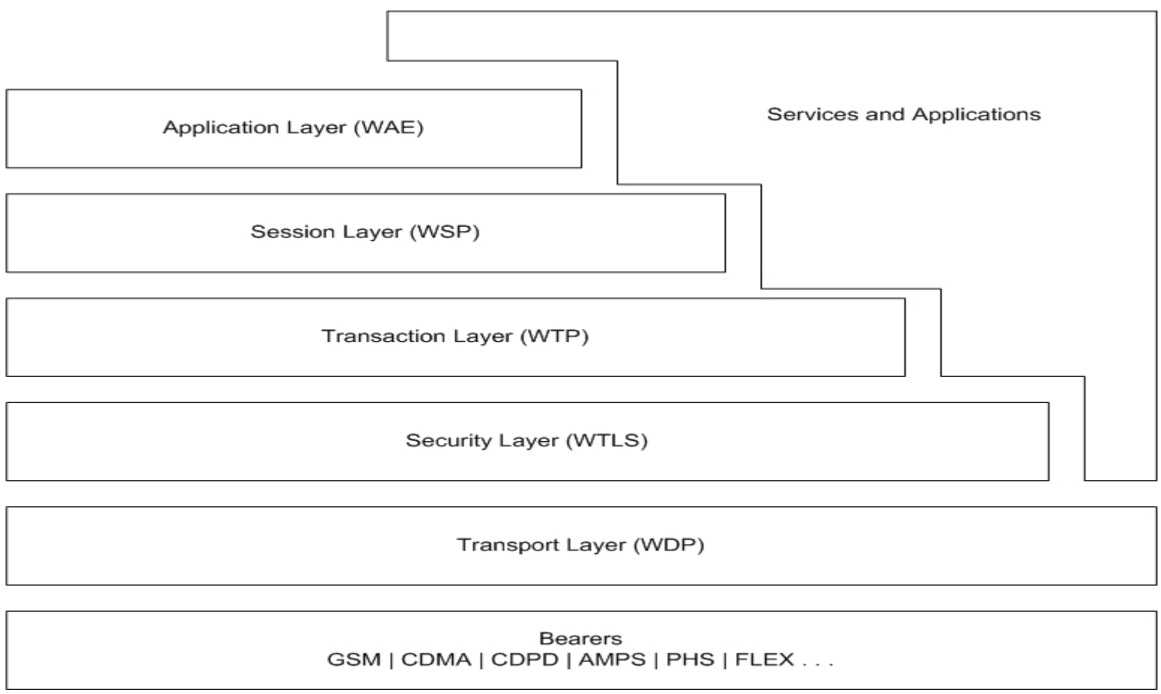


Figure 1 WAP Stack Layer

2.1 Layered Architecture of WAP

WAP has a layered architecture and each is capable of being developed independently. Thus it enables this technology to be manufacturer independent, vendor independent as well as implementation technology independent. Most of these layers are symmetric. It also allows the implementation of only a subset of the available WAP functionality. The layering also allows effective bridging between WAP and the Internet by making use of protocol converters, gateways, proxies and service brokers within the network infrastructure. Layering makes it possible for each layer to provide an interface directly to applications and services in addition to exposing an interface to the layer above it. This means that if an application wants to interact directly with datagram transport layer (WDP) for their services, it can do so. Conceptually these layers can be divided into three groups:

Bearer Layers: encapsulates the various signaling and channel protocols for wireless networks

Service Layers: encapsulates protocols which are primarily responsible for data transmission across the wireless network

Application Layers: microbrowser embedded in every wireless device that forms the user interface

Discussion of Bearer Layer is out of the scope of this thesis but lets look into the Service Layer, which essentially contains the WAP Stack protocols:

1. **Session Layer**: WSP establishes session between the mobile device and the WAP gateway and also allows handshake between applications for data exchange.
2. **Transaction Layer**: WTP supports pairing between client requests and server responses. It provides transaction support and overcomes the unreliable feature of WDP.
3. **Security Layer**: WTLS provides encryption facilities and enhances data security between wireless device and the WAP gateway.
4. **Transport Layer**: WDP forms the transport layer of WAP architecture and sends and receives datagram over the network. It is an unreliable protocol.

2.2 Wireless Telephony Application Services

Since this thesis focuses on mobile electronic transactions so lets look at the WAP Wireless Telephony Application Interface. WTAI is a set of services to create telephony applications. These interfaces are part of WTAI functions libraries accessible via WMLScript. They can be classified as:

Public services: includes simple features for third party applications using standard WAE user-agent. No WTA events are associated with this library. Moreover the telephony functions provided by this library are a blocking one hence it will not be discussed here, as it does not go in line with this research.

Network-common services: The most common features available in all the networks accessible only through WTA user-agent. The WTA events in these services are related to voice calls, network messages, phonebook, call logs and miscellaneous. For this research, we are interested only in voice call control, which is served by the **WTAVoiceCall** library. The table below shows the various WTA events of WTAVoiceCall library.

Table 1 Listing of WTAVoiceCall's WTA events

#.	Event Name	Event ID	Description
1	Incoming Call	wtaev-cc/ic	Indicates an incoming voice call
2	CallAlerting	wtaev-cc/cc	Indicates an outgoing voice call alerting the called party
3	CallConnected	wtaev-cc/co	Indicates an incoming or outgoing voice call has been set up
4	CallCleared	wtaev-cc/cl	Indicates an incoming or outgoing voice call has been terminated
5	OutgoingCall	wtaev-cc/oc	Indicates an outgoing call being established
6	DTMFsent	wtaev-cc/dtmf	Indicates the DTMF sequence has been sent on a voice call

The table below shows the various functions of WTAVoiceCall library. All these functions are non-blocking and WTA events signal the call progress.

Table 2 Listing of WTAVoiceCall's functions

#.	Function Name	Method	Description
1	Call Setup	WTAVoiceCall.setup()	Initiates mobile originated call
2	Call Accept	WTAVoiceCall.accept()	Accepts an incoming call
3	Call Release	WTAVoiceCall.release()	Terminates a voice call
4	Call Status	WTAVoiceCall.callStatus()	Retrieves status of a call in progress
5	Send DTMF	WTAVoiceCall.sendDTMF()	Sends a DTMF sequence through a voice call
6	Call List	WTAVoiceCall.list()	Retrieves the call handle of a voice call

Network Specific services: specific features available in select networks or operators, which is beyond the scope of this thesis.
Thus Network-common WTA services can place calls, receive calls, terminate it and get information about voice calls. There is two call models: originating call model and terminating call model.
The originating call model describes the protocol call states of an originating call from a mobile device. The diagram below shows the originating call model. It shows the various call states, actions by the user and the network events generated.

Figure 2 Originating Call Model

The terminating call model describes the protocol call states of an incoming call to a mobile device. The diagram below shows the terminating call model. It shows the various call states, actions by the user and the network events generated.

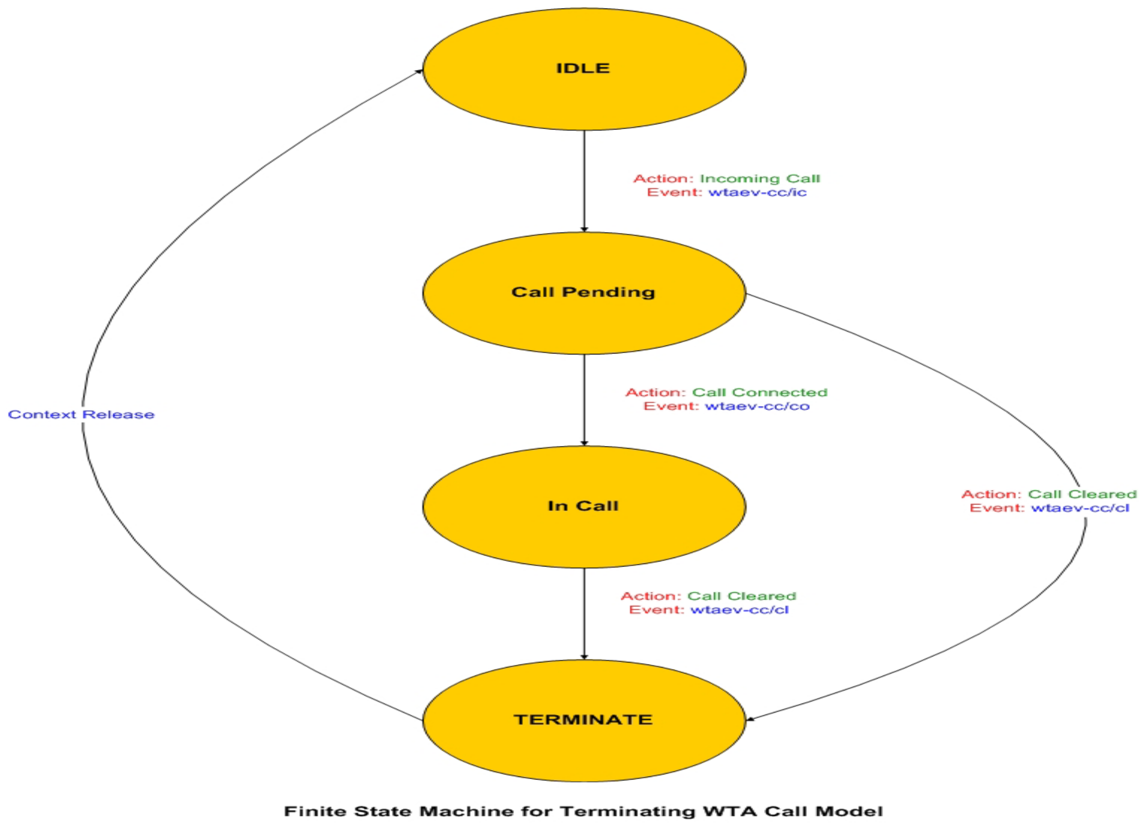


Figure 3 Terminating Call Model

If the caller and called party belong to the same WAP gateway, the originating call model is assigned to the caller and the terminating call model is assigned to the called party. If the call get routed through multiple gateways to get to the destination, each of the participating gateways will run both the call models, with the destination gateway's terminating call model providing services to the called party.
WTA applications must generate WTA events as shown in the call models.

WTA applications are developed using WML and WMLScript. The figure below illustrates the end-to-end system for a WTA application.

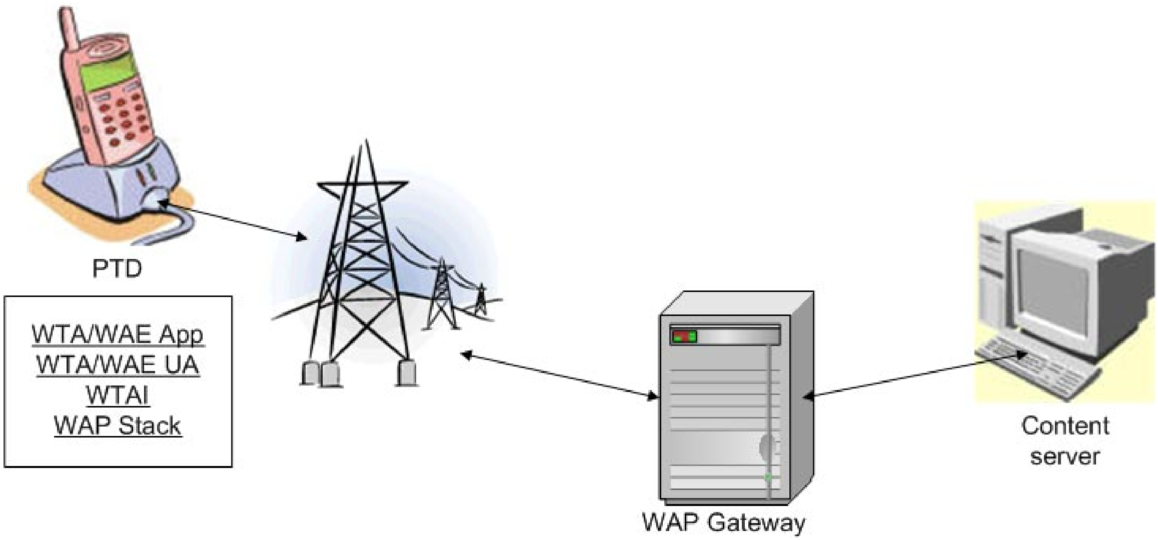


Figure 4 WTA Application end-to-end systems

WTA user agent, which supports WTAI libraries, executes the WMLScript and runs the WTA application. This application uses the WAP protocols. WSP requests and delivers the application, WTLS ensures the security and PAP pushes the content to the mobile device.

3. Mobile Electronic Transaction

Mobile electronic Transaction, popularly known as MeT, is an initiative started by Ericsson, Motorola and Nokia to establish and promote a framework for secure mobile transactions using a mobile device. Interoperable mobile transactions are the objective of this initiative. With WAP it has become possible to access mobile Internet services and execute mobile e-commerce transactions.

It has been predicted that mobile e-business will very soon become a substantial activity and it has been estimated that by the end of 2003 there will be more mobile devices than PCs connected to the Internet. The mobile phone is evolving into a Personal Trusted Device (PTD), which will not be just a wireless telephone but will have the capabilities to support services and applications like banking, payments, ticketing and secure access-based operations. MeT architecture describes how these applications can be accessed over the PTD in different environments.

3.1 Service Interfaces

MeT has defined three interfaces:
3.1.1 Service registration interface: This interface exists between the issuer and the PTD and it is primarily used for registering the PTD and loading the service certificates onto the PTD. The issuer may be a bank or a store’s credit department.
3.1.2 Service execution interface: This interface exists between the content server and the PTD and is used for executing secure transactions with the content server.
3.1.3 User interface: This interface basically represents the user interaction in a MeT transaction like presenting the transaction information to the user, getting the user input and accepting the same and passing it to the transaction routines.

3.2 Key Elements

The MeT core specifications include key elements like a
PTD: the user’s mobile phone
Content server: provides the transaction content
Acquirer: provides a single point of contact between the

issuers and the content providers and controls the business rules applicable to a transaction.
Issuer: provides the service certificate for an account to identify the user.

This diagram shows the various key elements, interfaces and the reference model of MeT system.

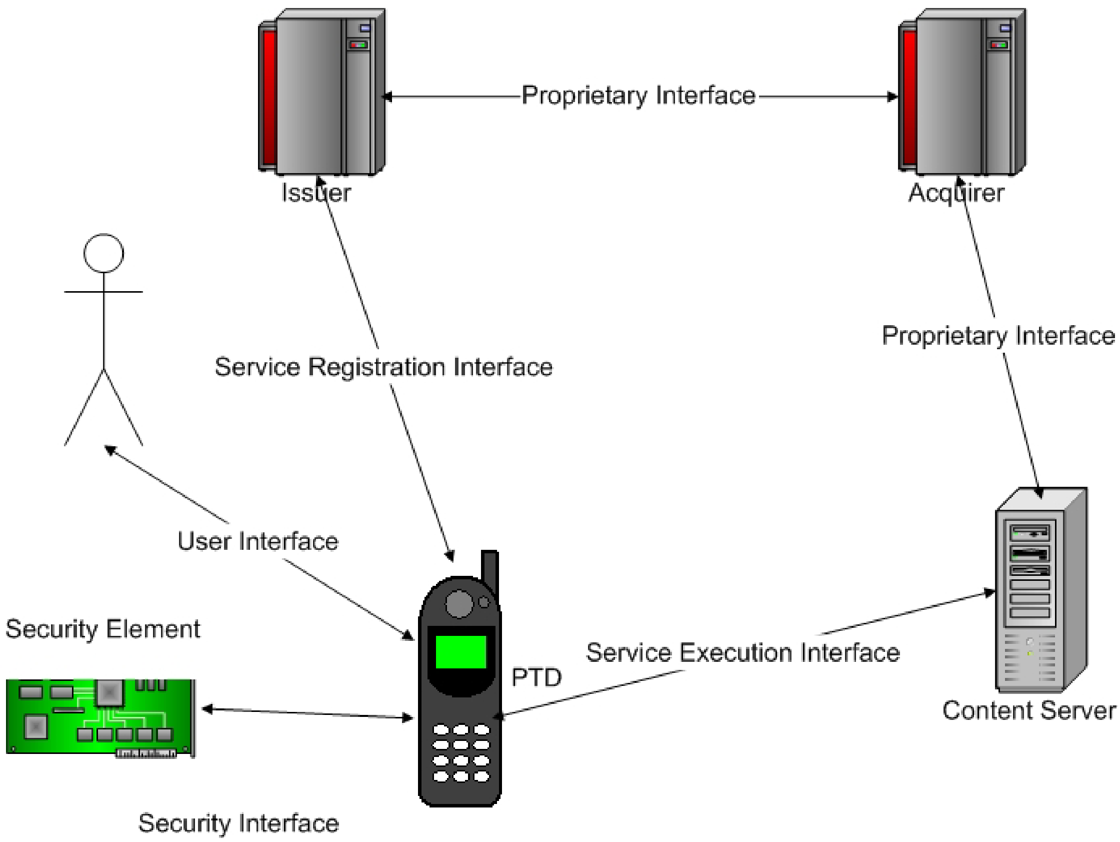


Figure 5 MeT reference model

3.3 Operating Environments

MeT applications can be accessed over the PTD in three environments:
3.3.1 Remote environment: the mobile Internet world via WAP services like WAP shops or WAP banks. Figure 4 shown previously can serve as an example of a remote environment.

3.3.2 Local environment: possibly over Bluetooth like payment services in a shop or ID services in an office.

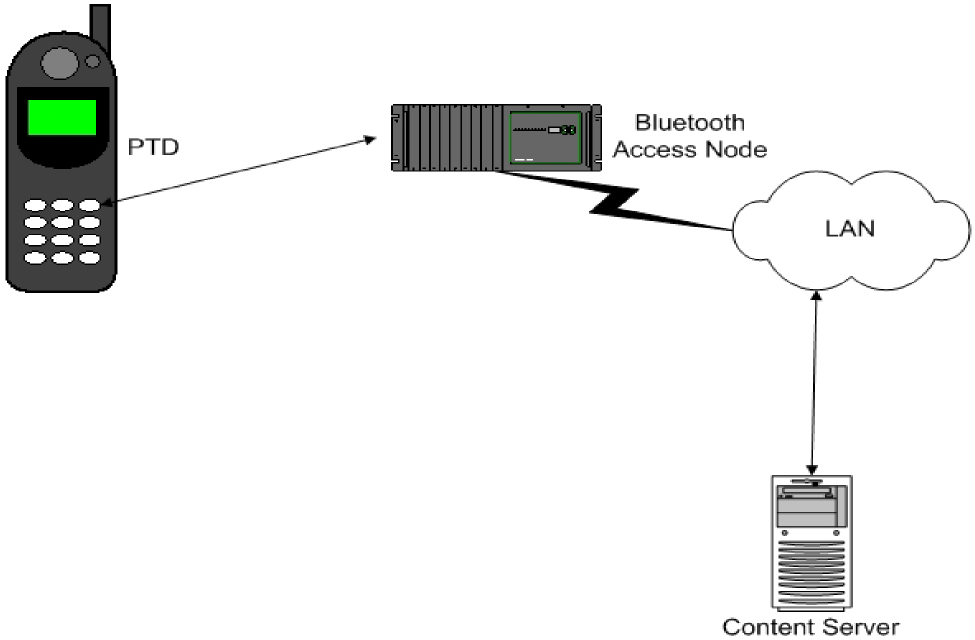


Figure 6 MeT in a local environment

3.3.3 Personal environment: possibly over Bluetooth to perform transaction in association with a PC.

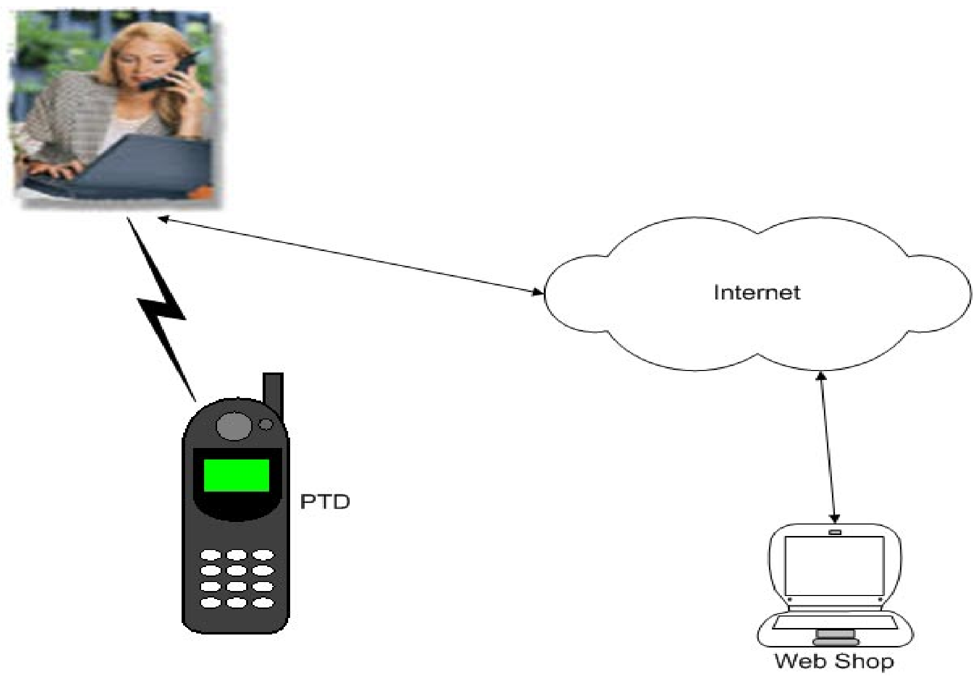


Figure 7 MeT in a personal environment

3.4 MeT core functions

The core functions in a MeT transaction are as follows:

3.4.1 Initialization: equipping the security element with the initial private-public key pairs and root CA certificates or SIM/WIM cards necessary to execute secured MeT transaction initializes the PTD.

3.4.2 Registration: The PTD shall have a service account with a service provider so that it can be identified. Here a user service certificate is associated with the service and assigned to its key pairs. Multiple service certificates from different service providers can be assigned to a common key pair. It is possible to combine initialization and registration functions.

3.4.3 Secure session establishment: confidentiality, data integrity and security of a transaction are very important to be established between the PTD and the content server. WTLS is used for it however WTLS operates only between the PTD and the WAP gateway and then SSL or TLS is used between the WAP gateway and the content server. So it is not an end-to-end security. Thus the WAP gateway must be secure and trusted.

3.4.4 Authentication: the service provider to authenticate the PTD user uses it. However not all MeT services requires it. The WAP gateway establishes the identity of the user by means of the user service certificate assigned to the key pairs in the PTD. So the server challenges a PTD and the user enters the access PIN over the PTD to authenticate himself.

3.4.5 User authorization: this is a means to avoid non-repudiation. The service provider ensures that the user accepts a transaction contract by signing it digitally using a private key associated with a user service certificate.

4. Understanding Session Initiation Protocol

SIP is a new signaling protocol which is being used for IP telephony and multimedia call control. It can establish, modify and terminate a multimedia session over the Internet. SIP has received a lot of attention recently because it is completely integrated with other Internet and web protocols. SIP derives its features from HTTP (Hyper Text Transport Protocol) used over the web and SMTP (Simple Mail Transport Protocol) used for e-mails. SIP also interacts with other Internet Protocols like IP, TCP, UDP and DNS. The client-server paradigm and the use of URLs (Uniform Resource Locators) has been imbibed from HTTP whereas text-encoding and header style like *To*, *From*, *Date*, *Subject* etc. has been borrowed from SMTP. It is a text-based, lightweight application layer signaling protocol. It makes use of the layered Internet Multimedia Protocol stack. These layers provide functions for communication processes. We are indeed referring to TCP/IP protocol suite here. SIP can use TCP as well as UDP for transport.

4.1 TCP/IP Suite

It consists of four layers as shown here:

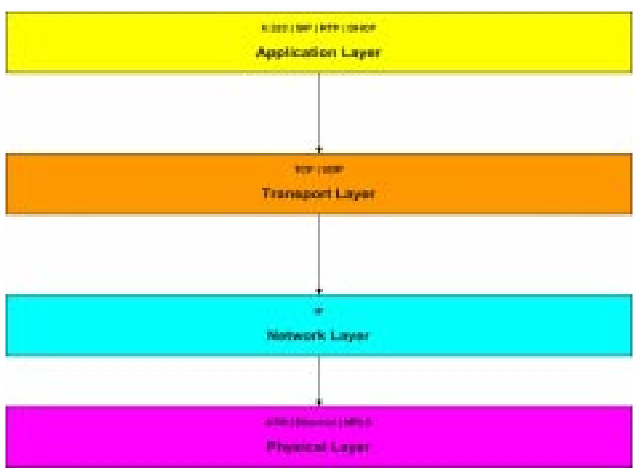


Figure 8 TCP/IP Protocol stack

4.1.1 Physical Layer: this layer accesses the physical medium and provides interfaces for symbol exchange, frame synchronization and Token Ring, Ethernet, serial-parallel cabling for data transmission.

4.1.2 Network Layer: this is the Internet Layer and IP is used for packet routing across the network using the destination IP addresses which are four octets long, also known as “dotted-decimal notation” (e.g., 152.161.1.11). IP is a connectionless protocol and provides best-effort packet delivery.

4.1.3 Transport Layer: it verifies whether the network layer is working efficiently or not and employs two well-known protocols, namely TCP and UDP, to deliver the datagram to correct applications using two-octet port number. Some port numbers are called “well-known” port numbers like 80 for HTTP, 5060 for SIP, 5004 for RTP. Other applications are assigned port numbers from a pool of available port numbers known as “ephemeral” port numbers that ranges between 49152-65535.

1. **TCP**
   Transmission Control Protocol is a connection-oriented, highly monitored reliable transport protocol over IP, which provides error-correction, and flow control. TCP uses sequence numbers and acknowledgements to implement flow control and other services over IP.
2. **UDP**
   User Datagram Protocol is a connectionless, simple and unreliable transport protocol. Both UDP and IP are connectionless, best-effort delivery protocol then why need IP ? All IP does is to send datagrams from one host to another. It doesn't provide any application interfaces or management services whereas UDP does provide these services. UDP also takes care of application multiplexing/demultiplexing services through the use of port numbers thus allowing many application protocols to be used on a single host. Doing these with IP would require a lot more transport protocols and it would be very inefficient.

4.1.4 Application Layer: this is the top layer and includes signaling and media transport protocols and provide interfaces to end-user application protocols. A few examples are HTTP, RTP, SIP, H.323, and POP3 etc.

SIP has the capabilities to implement reliability in transporting SIP messages hence it can use unreliable UDP. If TCP is used then these mechanisms are not used.

4.2 SIP

SIP is a lightweight, text-based, application layer signaling protocol for multimedia call control. It is used to create, modify and terminate sessions with one or more participants. These sessions include, but not limited to, Internet multimedia conferences, IP telephony etc. Signaling is achieved based on entities like SIP clients and servers, well-defined SIP request messages, SIP response messages, included SIP headers, related protocols like SDP, RTP etc.

4.2.1 SIP Entities

The following primary entities have been defined in RFC 2543:

User Agent Client (UAC): UAC is a SIP-enabled end-device that initiates a session. It maintains call states, which it initiates or participates in. Even if a session has been terminated, call state must be maintained by the UAC for at least 32 seconds to accept responses incase of lost messages in the call termination.

User Agent Server (UAS): UAS is a SIP-enabled server application that contacts a user when a request is received and responds on behalf of the user. It can accept, reject or redirect a request. UAS too maintains the call state as mentioned for the UAC.

Proxy, Proxy server: It is an intermediary entity that acts as both client and server. It can make requests on behalf of other clients and service a received request internally or by passing them to other proxies possibly after translation. A proxy server has no media capabilities. It can either be a stateless server or a stateful server. A stateless server never stores any information about a message once it has been parsed, processed or responded. All the processing is done sheer based on the message contents. It never uses any timers nor retransmits a message. A stateful proxy server, on the contrary, maintains information of all the requests and responses and uses it for future processing. It is capable of retransmitting messages and maintains timers. A proxy is used only to help the caller user agent to locate the called party (if they don't know each other apriori) and once a connection is established then proxy does not participate in the call. The two parties can communicate directly.

Redirect server: It receives a SIP request and returns zero or more possible locations for the called party. No message forwarding is done. It neither initiates a request unlike the UAC nor accepts a call unlike the UAS.

Registrar: A registrar server accepts SIP REGISTER requests. It can be co-located with a proxy server or a redirect server. It provides the information about the registered SIP user agents to other SIP servers within the same administrative domain. A user agent must register with the registrar if it intends to receive calls. Registration is not necessary for making outgoing calls.

4.2.2 SIP Requests

There are six defined methods:
INVITE: is used to set up multimedia sessions between two parties
REGISTER: is used by a user agent to register it with the registrar notifying it of its contact address where it can receive calls.
BYE: is used to terminate an established session and is always initiated by the participating user agents.
ACK: is used to acknowledge the final responses to INVITE methods. Other final responses are never acknowledged.
OPTIONS: is used to find out the capabilities and availability of a UA or a server. It is always initiated by a UA and never by a proxy.
CANCEL: is used to teardown a pending INVITE.

New methods like **INFO**, **PRACK**, **REFER**, **SUBSCRIBE**, **NOTIFY** are being discussed and have been described in separate Internet-Drafts.

4.2.3 SIP sub-protocol

The following sub-protocol have been mentioned in the RFC 2543:
Registration (2 Phase Protocol): To register a UA with a registrar server. The UA may register to a local SIP server

by sending a request to a multicast address "sip.multicast.net" (224.0.1.75). Same UA can register from different locations. Third party registration is also allowed. The requests are processed in the order they are received.

Session Invitation (3 Phase Protocol): To establish a multimedia session with another UA for a two-party call or a multiparty conference. The UAC sends an INVITE request possibly with a message body in SDP format. The proxies can send 1xx informational response and the UAS sends the final responses (2xx - OK or non-2xx - reject). The 2xx response contains the media capability of the receiver. The UAC sends an ACK to acknowledge this 2xx response. If a non-2xx response is generated then the first proxy that receives this response from the UAS sends an ACK to the UAS and forwards it to the caller. Caller on receipt sends an ACK that gets terminated at the first proxy.

Session Modification (3 Phase Protocol): To inform all the participants of a session about the change in parameters of an existing session possibly by sending a re-INVITE request.

Session Termination (2 Phase Protocol): To terminate an established session by sending a BYE message. Either caller or the callee can initiate this request. The UA agent receiving a BYE request must stop transmitting data.

Invite Request Cancellation (2 Phase Protocol): To cancel a pending request other than ACK and CANCEL e.g., REGISTER, INVITE, OPTION but typically it is used to cancel a pending INVITE request.

4.2.4 SIP Responses

There are six classes of responses in SIP signaling – 1xx to 6xx. 1xx – 5xx has been borrowed from HTTP (especially x00 – x79 whereas new responses for SIP starts from x80) and 6xx have been defined specifically for SIP. The table below shows the various response classes:

Table 3 Listing of SIP's response classes

Response Class	Description	Examples
1xx – Informational	Indicates call progress	100 – Trying 180 – Ringing 181 – Call Forwarding 182 – Queued
2xx – Success	Indicates successful responses for a request	200 - OK
3xx – Redirection	Asks to retry the request at a different location	300 – Multiple locations 301 – Moved Permanently 302 – Move Temporarily 305 – Use Proxy
4xx – Client error	Indicates error in the client request	400 – Bad Request 401 – Unauthorized 403 – Forbidden 404 – Not Found 486 – Busy
5xx – Server error	Indicates server error	501 – Not Implemented 502 – Bad Gateway 505 – Unsupported version
6xx – Global Failure	Indicates a global failure	600 – Busy everywhere 603 – Decline 606 – Unacceptable

4.2.5 Session Description Protocol

Like SIP, SDP is text-based protocol and is a part of the Internet Multimedia Conferencing protocol suite. SDP is used to carry a message body in SIP. The calling user agent lists their media capabilities in either INVITE or ACK message body whereas the called party lists their media capabilities in the final successful response (200 OK) to the INVITE. The table below enlists the various SDP fields:

Table 4 Listing of SDP's attributes

Field	Description	Mandatory ?	Remarks
v	SDP version	Yes	
o	Session originator	Yes	Not used by SIP but included for compatibility
s	Subject	Yes	Not used by SIP but included for compatibility
i	Session Info	No	
u	URI	No	
e	Email id	No	
p	Phone Number	No	
c	Connection Info	Yes	Used by SIP to set up sessions between UAs
b	Bandwidth Info	No	
t	Start & end time for session	Yes	Not used by SIP but included for compatibility
r	Repetition	No	
z	Timezone Info	No	
k	Encryption Key	No	
a	Attribute	No	Used by SIP to set up sessions between UAs
m	Media Info	Yes	Used by SIP to set up sessions between UAs

4.2.6 SIP Headers

SIP Headers have been classified in four different categories: *general, request, response and entity.* The table below shows the various headers in each category.

Table 5 Listing of SIP's Headers

Header Type	Description	Associated Headers
General	Includes all the mandatory headers in a SIP message (request as well as response). In most of the cases these headers are generated by the UA and can not be modified by proxies	Call-ID Contact Content-Encoding Content-Length Content-Type Encryption From To Subject Via Cseq etc.
Request	Usually inserted by a UAC in a SIP message to give additional information about the request	Accept Accept-Contact Accept-Encoding Accept-Language Authorization Record-Route Max-Forwards Session-Expires Response-Key etc.
Response	Usually inserted by a UAS or server in any response to give additional information about the response	Unsupported Server Proxy-Authenticate Warning WWW-Authenticate Rseq etc.
Entity	Used to provide more information about the SDP message body	Allow Content-Encoding Content-Length Content-Type Expires MIME-Version etc.

5. Proposing the framework for SIP Signaling in MeT

The key to this framework is to derive conceivable messaging interoperability between WAP and SIP protocols. This section of the thesis focuses on:

- IMS Subsystem architecture for SIP in 3GPP
- Interpreting WTA calls for the SIP environment
- Mapping WTA call model into SIP state machine
- Mapping WTA calls into SIP messages and vice versa
- Defining SIP/WTA extensions if necessary

5.1 3GPP requirements on SIP

The 3GPP outlines the requirements, specifications and framework for 3G mobile systems that is essentially based on GSM networks. The 3G networks comprises of three domains:

- Circuit-switched domain : employs circuit-switched network to carry voice and multimedia as in 2G systems
- Packet-switched domain: employs IP technology to provide internet access to mobile devices
- IP multimedia domain: employs SIP for call signaling to render IP-based multimedia services (popularly known as IMS).

Packet-switched network is primarily an access technology. IMS specifications have been laid to harness this access technology and also provide multimedia services to wireless devices. IMS subsystem would be aware of the characteristics of the real-time data to be transported, codec to be used, IP addresses alongwith the port numbers, apply error correction, header compression, required QoS etc. beyond that can be achieved over the packet-switched domain.

IMS subsystem consists of various SIP proxies referred to as Call Session Control Functions (CSCF). The IMS specifications describes three types of CSCFs:

- Proxy-CSCF (P-CSCF): connects the wireless device to the IMS subsystem. SIP requests and SIP responses passes through it.
- Serving-CSCF (S-CSCF): provides services to the users, which they subscribe to. Each wireless device is

associated with an S-CSCF on SIP registration. The incoming and outgoing sessions passes through it. Home Subscriber Server (HSS) maintains the information about S-CSCF and user's association. HSS knows the users location and subscribed services.

- Interrogating-CSCF (I-CSCF): locates the associated S-CSCF for a user and routes the request to it.

On this line, a typical registration message flow in IMS network has been shown as:

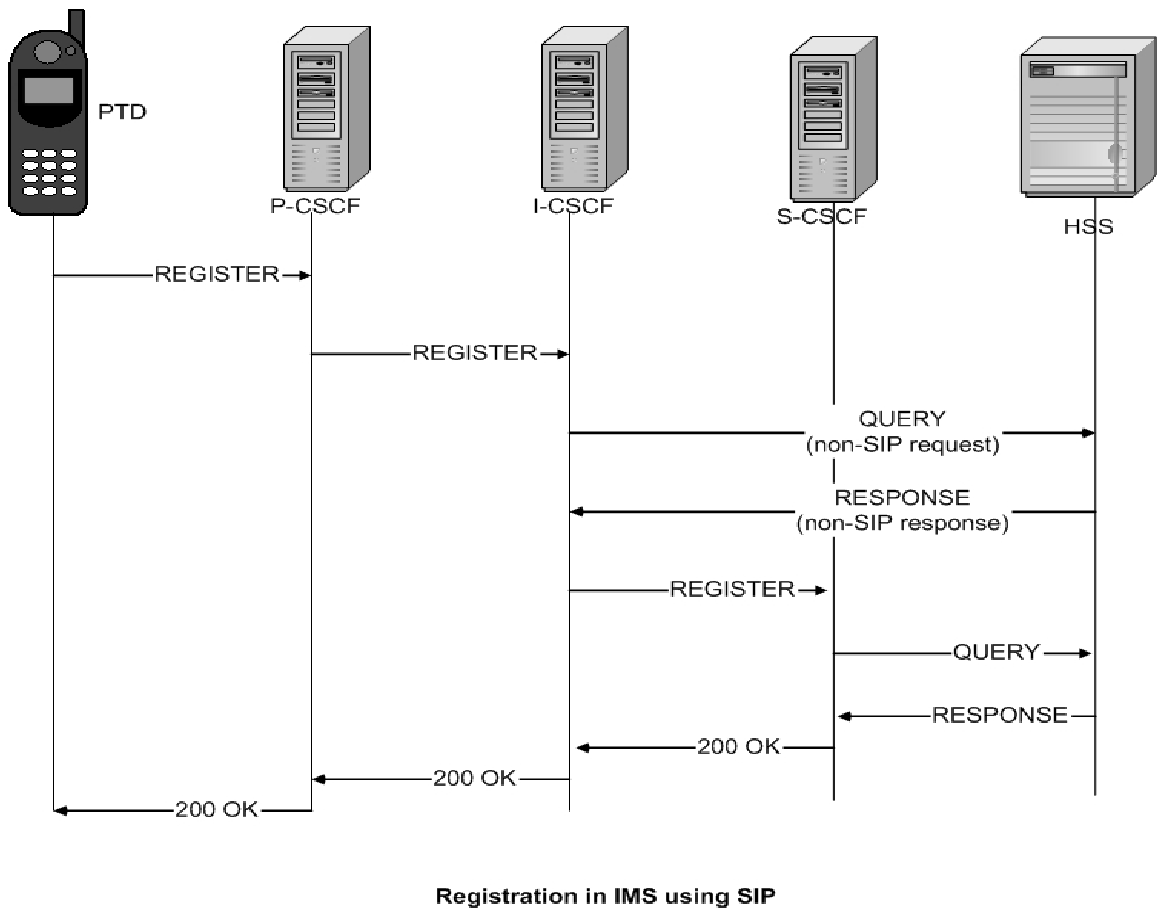


Figure 9 Registration in IMS Subsystem

Here all the CSCFs can be in the home network. Incase of roaming, wireless device will talk to P-CSCF of visited network and it will talk to the I-CSCF in its home domain so the call flow still remains as shown above. In the roaming scenario, each user will contact P-CSCF in their respective visited domain. That P-CSCF will contact the I-

CSCF and S-CSCF subsequently in their respective home domains to facilitate the intended communication. This has been described below:

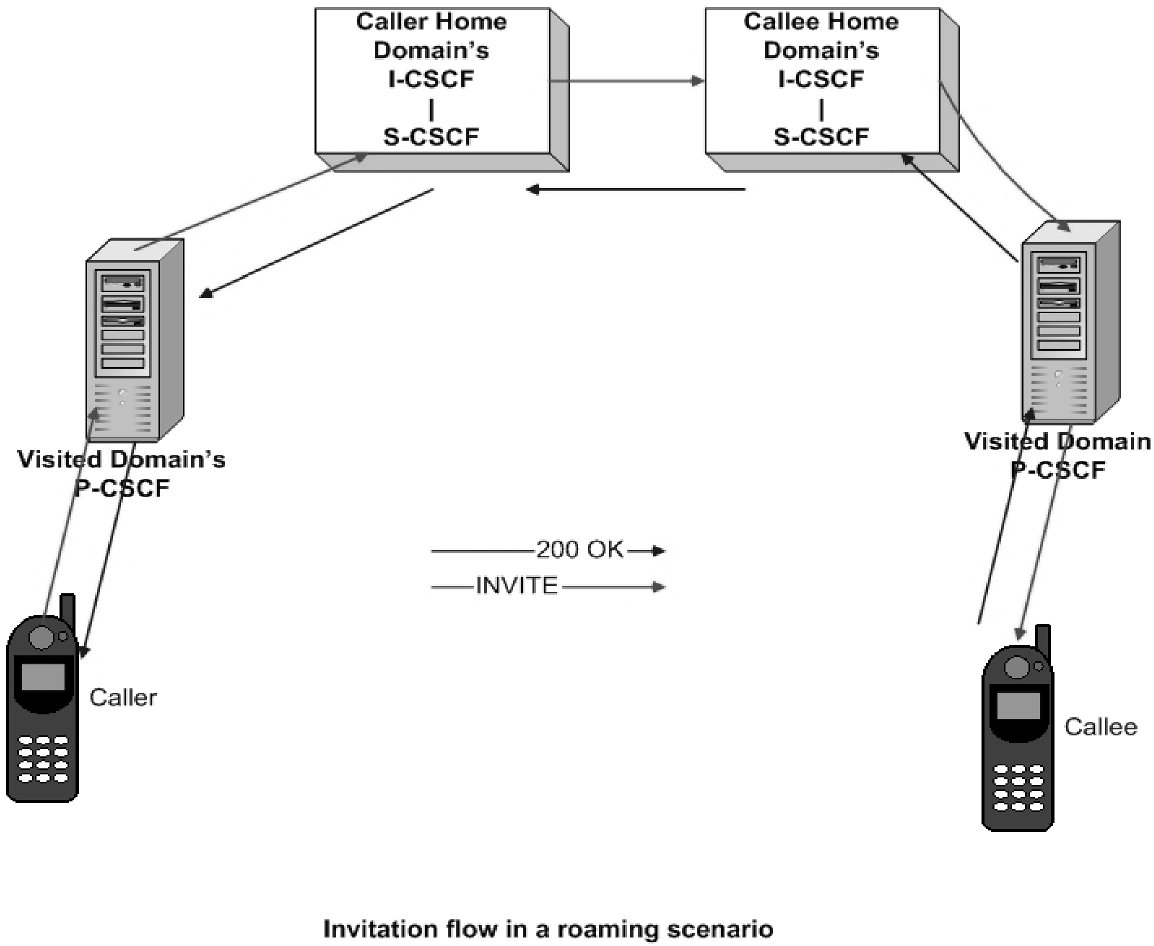


Figure 10 Invitation in IMS Subsystem

IMS specifications rely on GPRS technology for transport. Seeing the fast data transmission rate as promised by the GPRS technology it is particularly well suited as a bearer for WAP communication. WAP will become easier and faster to use as it works across the GPRS Service. The PTD user can be continuously connected to a WAP portal and will no longer pay for the entire time they stay connected, but only for the amount of data they send or receive. Both Nokia and Ericsson have announced that they consider WAP as an effective protocol for wireless Internet access using the GPRS phones. The 3GPP requirements on SIP have been detailed in [9]. As far as the network requirements for SIP signaling in MeT are concerned, most of the features applicable to the usage of SIP protocol in cellular networks and 3GPP IMS [9] shall be valid.

Some of the important requirements for this thesis topic are:

- Minimum session establishment time
- Minimum software/hardware requirement for the PTD
- Un-interrupted services while moving from home network to visited network
- IPV6 addresses
- Shifting an active session from one device to another.
- User authentication on the PC while shifting the transaction: SIP allows more than one user to log in from the same PC.
- Consistency of the transaction: no loss or no duplicate information should be offered to the user when a transaction is shifted to another device.

The IMS network architecture [10] as shown in figure 9 and 10 shall be applicable to implement SIP signaling in MeT over WAP.

5.2 Identification of SIP messages for MeT

I propose the following SIP protocols for its use in MeT:

1. Registration
2. Session Invitation
3. Session Modification
4. Session Termination
5. Invite Request Cancellation

Registration

This is in addition to *service registration* as mentioned in 3.1.1 and might be done together to minimize the session establishment time. This registration will associate SIP URLs with its various contacts. As described in [9], these registration requirements hold good:

“The user must be reachable for terminating sessions and services”

“The user is pre-authenticated early, so that authentication does not contribute to post-dial delay”

“The user is assigned a particular serving proxy. The serving proxy downloads the service profile for that user to trigger services”

It should also provide transparent registration for PTDs in home network as well as visited network so that un-interrupted services are provided while moving from home network to visited network. All the other network requirements related to location of registrar, Qos, resource allocation, user-authorization, and authentication etc. shall be in conformance to [9] as described in sections 6.3 and 6.4 and is out of scope of this thesis.

Session Invitation
A session must be established with the content server to participate in the mobile electronic transaction.

Session Modification
In Mobile electronic Transaction, it is quite evident that session integrity is important and yet we are looking forward to effective mobility. With mobility we mean that a user should be able to shift an active session or transaction between PTD and other devices like a PC or a laptop. This is the crux of the situation. Let us consider this scenario:
A user has logged into his PC and initiated some transaction and then a need arises wherein he has to be mobile. So he would very much like to shift this transaction onto his PTD without loosing any bit of the dynamic information or vice versa i.e., he realizes that his PTD is running low on the power or probably he is using the PTD during peak hours and would very much like to shift the transaction to his PC ones he is in his office/home.

Here we are looking forward to shifting an active session with respect to the ongoing transaction. All the involved devices can have the same address so that device switching should not result in loss of transaction details. Usage of SIP becomes handy in this scenario because it allows multiple registration (several devices) using the same URI. Once the session has been established over the PTD, a laptop or a PC can be brought into the same session by Session Modification protocol.
The REFER SIP message and REPLACES header can be used to achieve this. The procedure has been defined in [11].

Session Termination
There could be two scenarios:

a) Graceful session termination: a SIP session being released by the user, either the calling party or the

callee disconnects
b) Ungraceful session termination: the PTD ran out of the battery power or the wireless connection fails

Session termination shall be done in conformance to [9] as described in its section 6.14.

Invite Request Cancellation
A user should be able to cancel an INVITE request if it is still pending.

5.3 Support from WAP
In WAP layered architecture, it is possible for each layer to provide an interface directly to applications and services in addition to exposing an interface to the layer above it. This means that if an application wants to interact directly with WDP layer for their services, it can do so. Even though there is no pair wise compatibility between WAP and Internet/WWW layers, this layered architecture allows the bridging between them by making use of protocol converters, gateways and proxies residing within the network infrastructure. However, this bridging can occur only at certain layers especially datagram transport layer (WDP), session layer (WSP) and application layer (WAE). Since each layer in WAP architecture has a well-defined service abstraction for upper layer and can get a particular service abstraction from lower layers, it is possible that non-WAP services like SIP can use WAP services as long as they provide the desired service abstraction. With the advent of WAP2.0 and its new enhanced features, SIP signaling in WAP environment is a definite possibility.

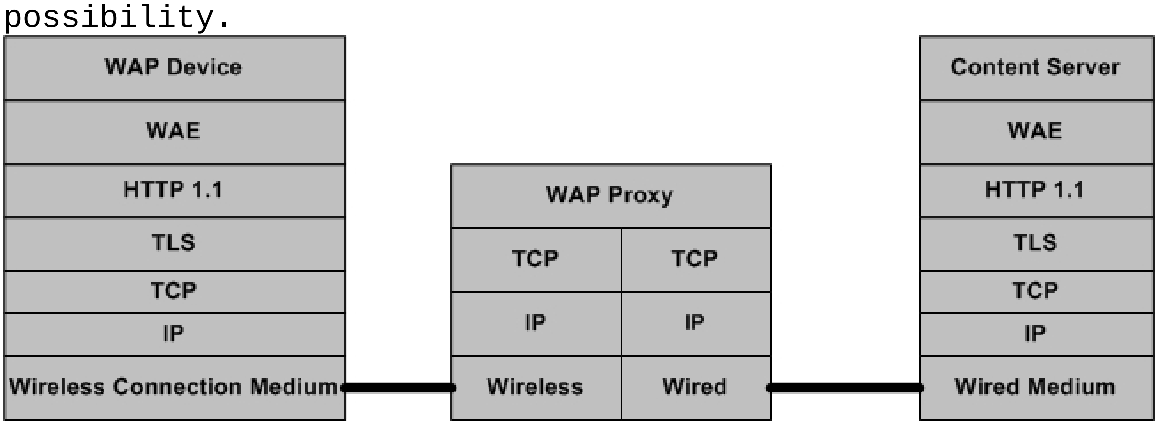
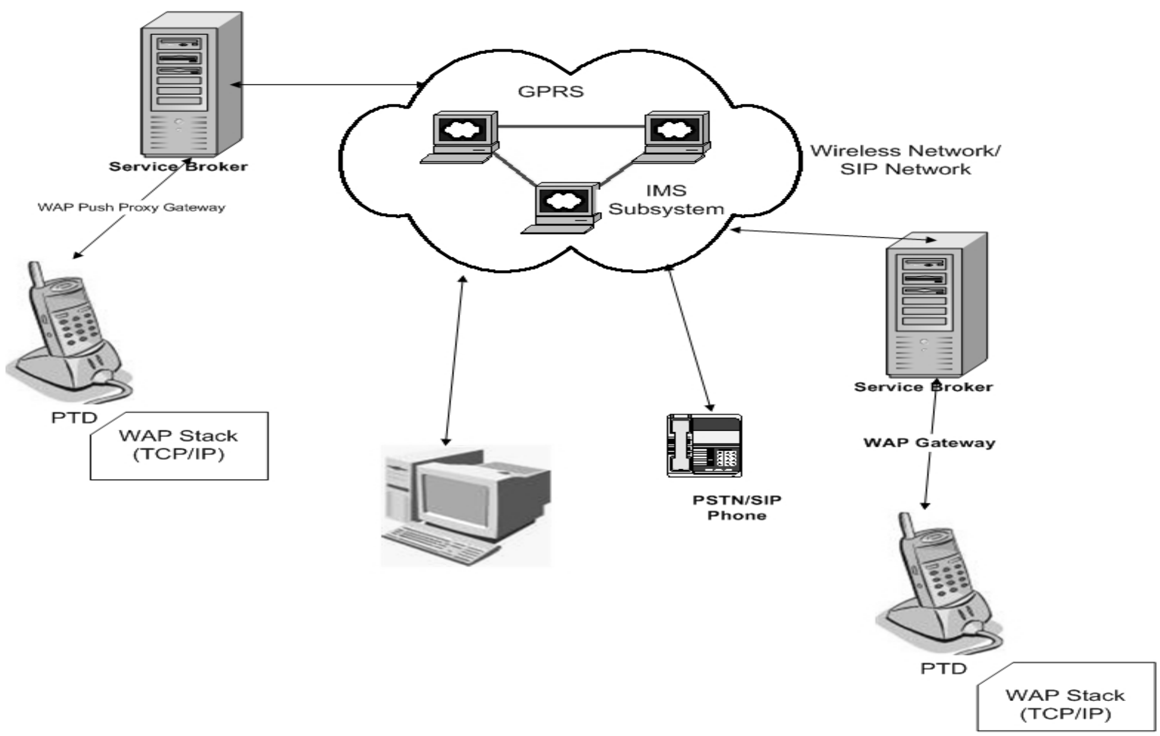


Figure 11 WAP2.0 Stack

The following new features in WAP protocol shall assist in SIP signaling:

- Additional support for mobile profile of IP, TCP and TLS
- WAP protocols can facilitate client-server communication using HTTP/1.1 without the mandatory need of a WAP Proxy
- User Agent profiling service can be explored to exchange client capabilities, which is one of the very important need of SIP signaling
- Looking at the WTAI's event model, scripting interfaces and the finite state machine there is a possibility of developing a pair wise compatibility between SIP messages and network common libraries.

The following reference model is being proposed for SIP signaling in WAP domain.



SIP signaling in WAP domain

Figure 12 Reference model for SIP signaling in WAP

The WTA messages shall be invoked by the PTD and sent to the WAP Gateway. The WAP Gateway can convert the WAP content into text and send it to the ASB over the GPRS. The ASB shall do the mapping between WTA call model and SIP state machine and can be configured to interact with the SIP Proxies in the IMS network to connect the caller to the content server (e.g., a bank) and pass the received response to the WAP Gateway. The gateway will again convert the information into WAP content and send it to WAP Push Proxy gateway, which can deliver it to the PTD.

5.4 Service Broker

The proposed concept of Application Service Broker (ASB) has the capabilities to manage call control, call state, brokering media platforms, advanced SIP capabilities and logging SIP messages shall foster SIP signaling in MeT over WAP. The Application Services Platform is a multilayered, component based software that has been designed over SIP servers which fosters the speedy creation and deployment of next-generation converged services. It consists of following layers:

Application Layer: User agents

Service Layer: Application Services Brokers

Call Control Layer: SIP servers, soft switches, Media Gateway Controllers

Switching & Routing Layer: IP, Frame Relay, ATM

Transmission Layer: Electrical, Optical, and Wireless

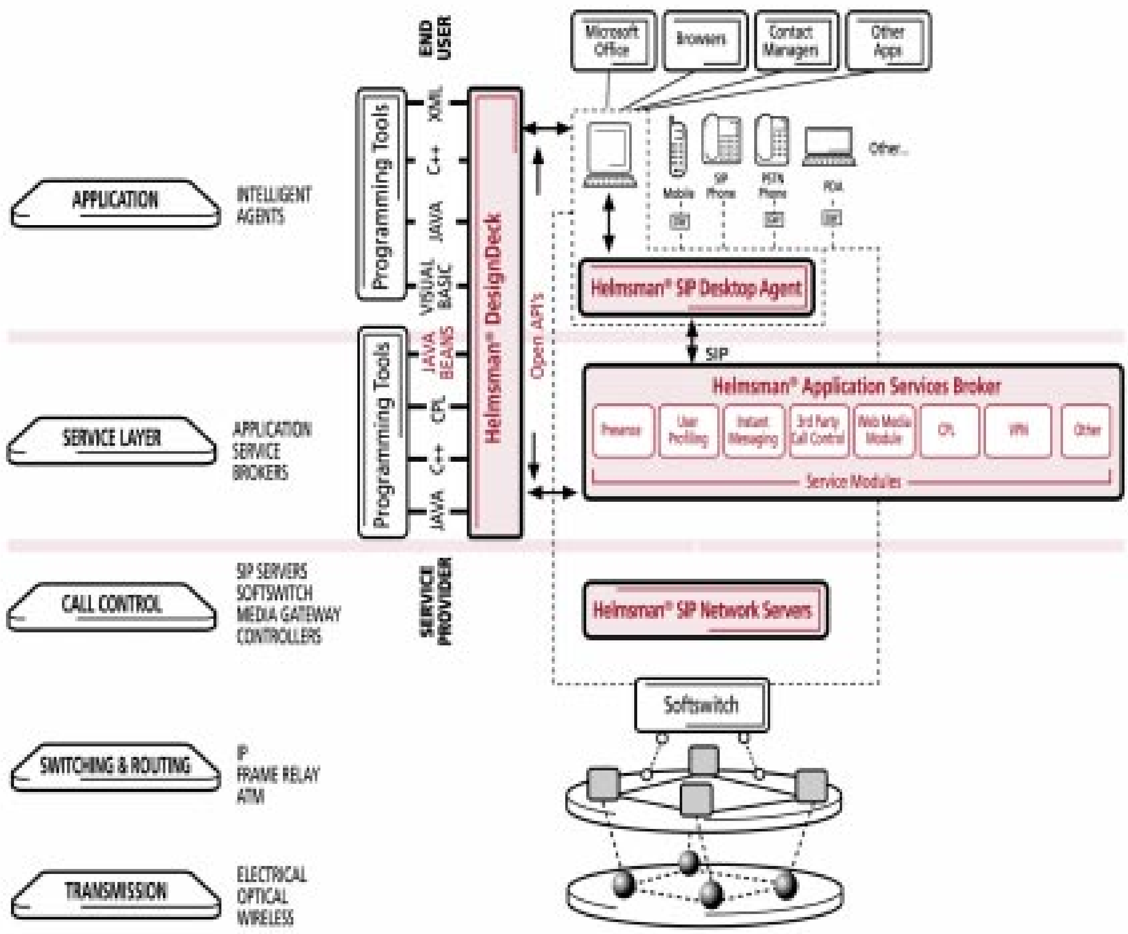


Figure 13 Application Services Platform Ref [8]

The heart of this system is Application Services Broker (ASBs). It supports services like Presence Management, Third-Party Call control, Conferencing and many others. It is flexible and scalable. The layered architecture allows a vendor to create a new service by defining the appropriate APIs that are supported by the service modules from the lower layers.

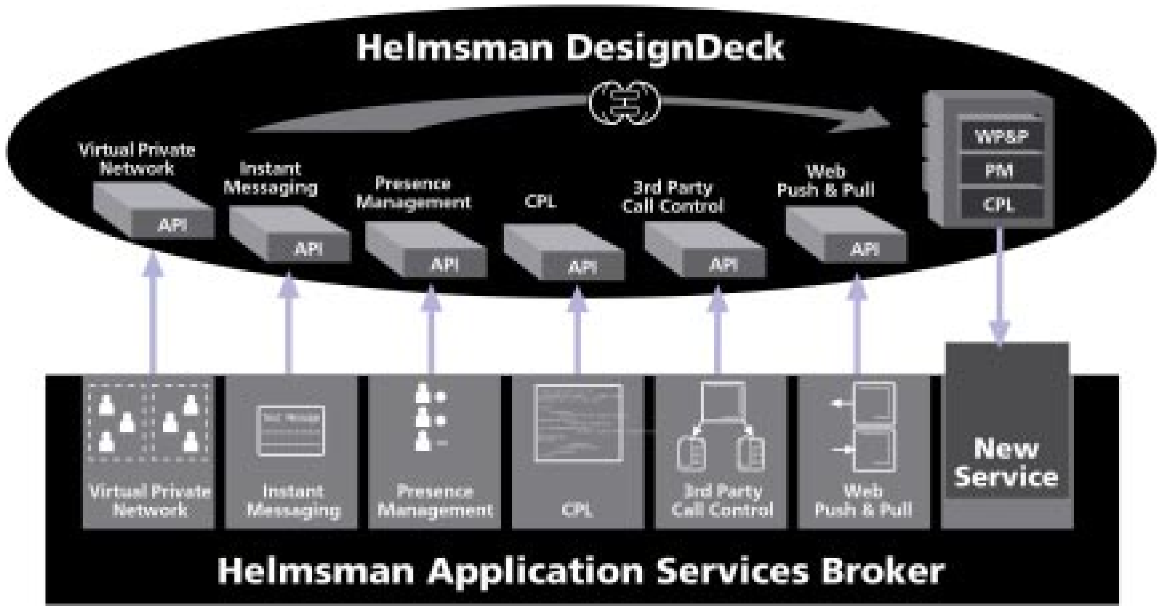


Figure 14 Application Services Broker Ref [8]

Thus, the application layer and the service layer's functionality is of our interest here where software can be implemented to facilitate SIP signaling in MeT and the same can be supported by the call control, switching & routing and transmission layer. Looking at the ASB block diagram in figure 14, it is evident that a component can be developed on the designdeck which can do the pair wise mapping between WTAI calls and SIP messages and can be supported by SIP call control services component in the ASB. The message format translation might be required from WAP content to text format and vice versa either in the WAP gateway or service broker. The detailed discussion of this is beyond the scope of this document.

5.5 Mapping SIP messages to WTA Event Model

This section establishes the mapping between SIP state machine and the WTA event model. As shown in figure 2 & 3, the event model is divided into two halves – an originating event model (O_WTAem) and a terminating event model (T_WTAem). Both the event models are comprised of a few states through which the transition falls through.

5.5.1 Originating WTA event model

This diagram below shows how the states of SIP protocol come into play when WTA initiates call and reaches the layer running the WTA event model. This layer will create the O_WTAem state machine, initialize it to IDLE and then subsequent states are mapped to the SIP message.

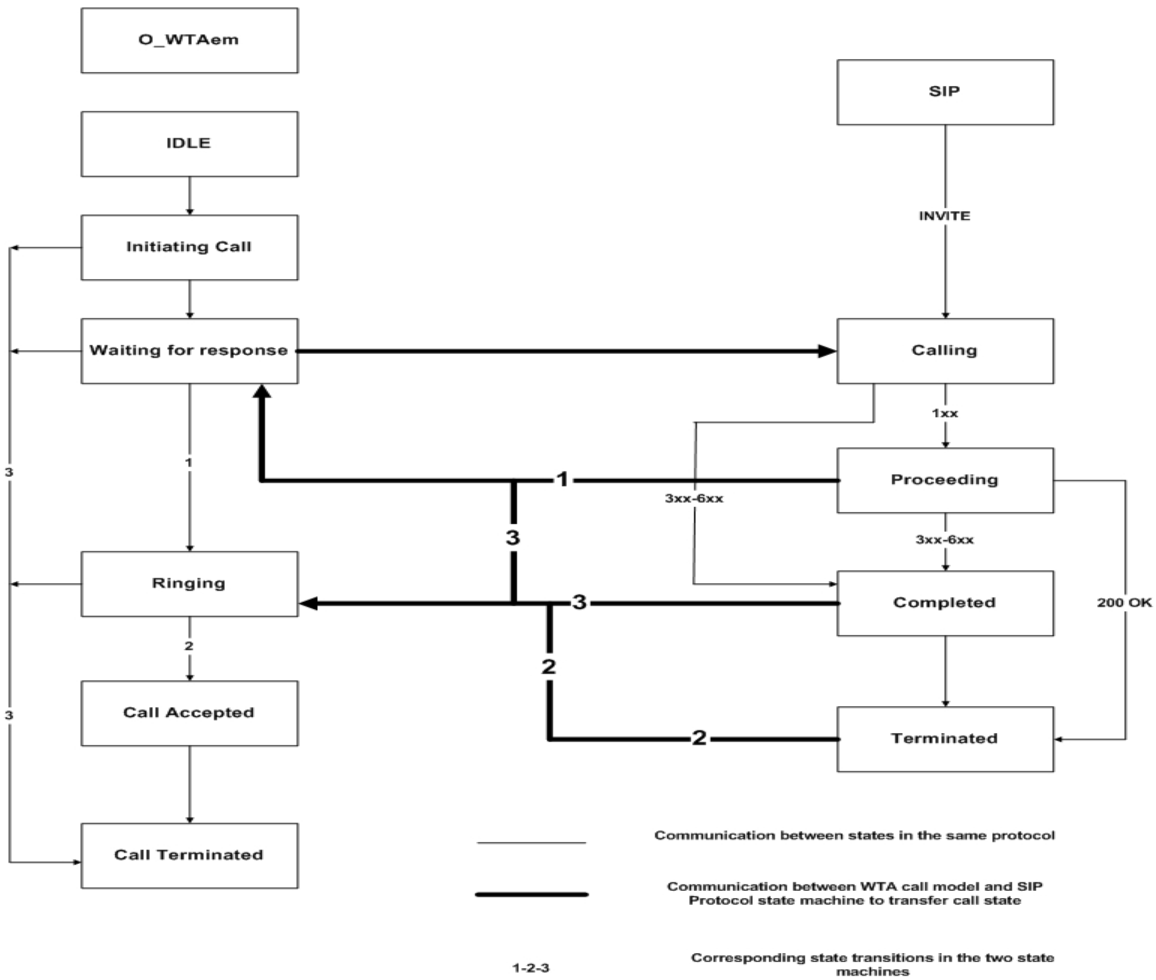


Figure 15 Mapping between O_WTA state machine and SIP state machine

The O_WTAem has the following states:
IDLE: starting state

INITIATING CALL: the user places. Any required authorization or validation of dial string or call screening (e.g., which may be required for freephone number translation, differentiating between local dialing and international dialing) is done. If it succeeds then the state machine moves to the next state otherwise an exception is generated and the call is cleared.

WAITING FOR RESPONSE: At this stage, the SIP state machine shall be invoked which then performs the signaling. If it succeeds then the state machine moves to the next state otherwise a non-2xx response is generated by SIP state machine and the call is cleared.

RINGING: at this stage the call has been sent and the called party is being alerted. The response will inform whether the callee has been reached or not. While all these state transitions are being executed, a 100 SIP response can be generated on receiving call-connecting event (wtaev-cc/cc). If the callee cannot be reached then the response shall be mapped to an appropriate 3xx SIP response and the O_WTAem call model goes to CALL TERMINATED state and the call is cleared. If the callee has been reached then the O_WTAem call model is blocked. The call-connected event (wtaev-cc/co) can be mapped to a 200 OK SIP response and the O_WTAem call model transitions to CALL ACCEPTED.

CALL ACCEPTED: Once the call has been accepted, the state machine shifts to this state. A busy response received enables the generation of appropriate 4xx SIP response. At this point the resources for the call are released in the O_WTAem call model.

CALL TERMINATED: call cleared. Call can be terminated in the following manner:
1. the wireless connection fails – ungraceful termination
2. the called party disconnects – graceful termination
3. the calling party disconnects – graceful termination
If (1) happens then the control is transferred to WTA state machine after processing suitable exception. NO need to send a BYE request here. Both the state machine would just release all the resources. If (2) happens then the SIP state machine goes to COMPLETED state and subsequently to TERMINATED state. This results in WTA state machine going to CALL TERMINATED state too. If (3) happens then the O_WTAem will send call-cleared event and wait for the final response. The SIP state machine should send a BYE request and wait for final response. This final response shall be

passed to the WTA state machine. The O_WTAem call model releases the entire context and initializes itself to IDLE state and even the SIP state machine would free all its resources.

5.5.2 Terminating WTA event model

This diagram below shows the mapping between SIP protocol state machine and the terminating half of the WTA event model. This layer will create the T_WTAem state machine, initialize it to IDLE and then subsequent states are mapped to the SIP message.

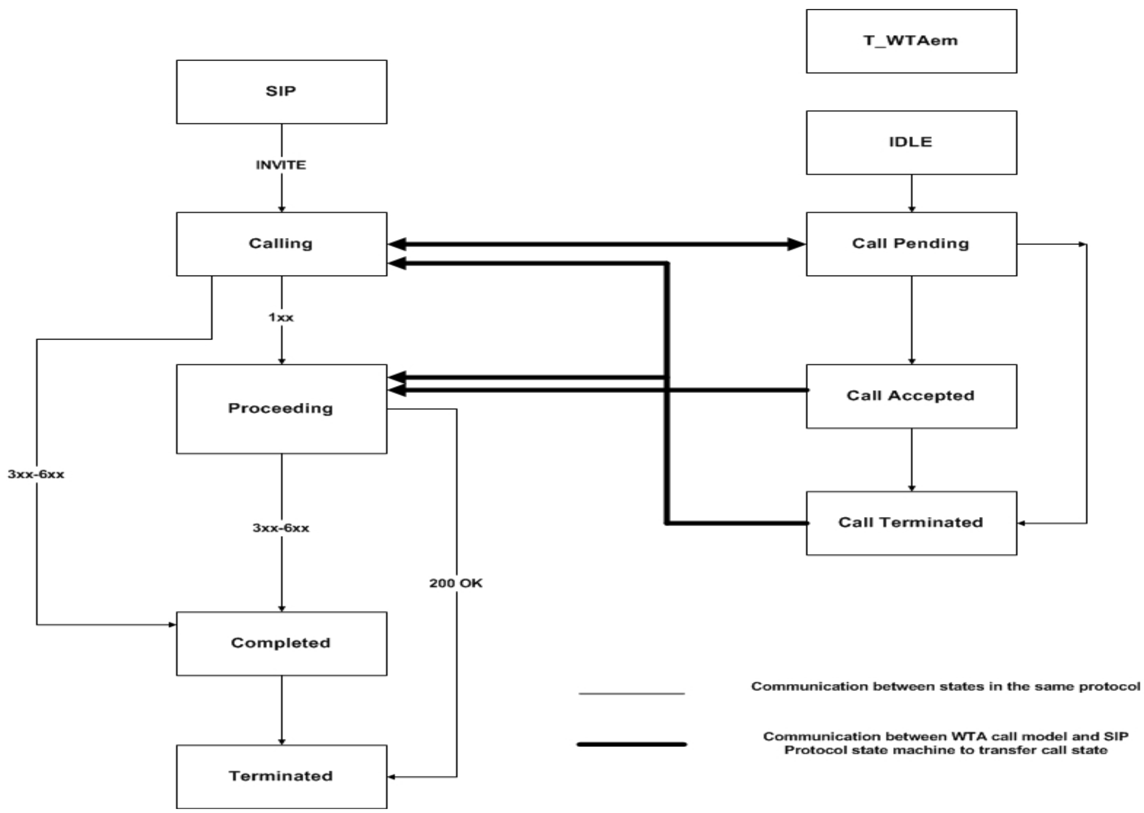


Figure 16 Mapping between SIP state machine and T_WTA state machine

IDLE – This state is initialized when the terminating end creates the call at the
WAP layer.

CALL PENDING: In this state, the capabilities of the called party could be established as per the calling party's requirement and the call is being presented by simply by ringing the called party. If the called party cannot support the required capability or could not be alerted then (CALL TERMINATED) is invoked and the call control is transferred to the SIP protocol state machine. The SIP protocol state machine can send a non-2xx final response. If the conditions are met and the call was successfully presented then at this point, the called party is being "alerted". Control now passed momentarily to the SIP protocol state machine, so it can generate and send a "180 Ringing" response to its peer. Furthermore, since network resources have been allocated for the call, timers are set to prevent indefinite holding of such resources. The expiration of the relevant timers results in the call control being transferred to the SIP protocol state machine. The SIP protocol state machine can send a non-2xx final response. So on success the state transitions to the next state i.e., SIP state machine goes from "Calling" to "Proceeding" and WTA state machine goes from Call Pending to Call Accepted.

CALL ACCEPTED - The call is now active. Once this state is reached, the call may be terminated only under one of the following three conditions: (1) the network fails the connection, (2) the called party disconnects the call, or (3) the calling party disconnects the call. Event (1) results in the processing of (CALL TERMINATED) and call control is transferred to the SIP protocol state machine. Since the network failed, there is not much sense in attempting to send a BYE request; thus both the SIP protocol state machine and the WTA call layer should release all resources associated with the call and initialize them to the IDLE state. Event (2) results in the processing of CALL TERMINATED state in WTA state machine and the SIP state machine transitions to the next state, TERMINATED. Event (3) would be caused by the receipt of a BYE request at the SIP protocol state machine. Resources for the call should be deallocated and the SIP protocol state machine must send a 200 OK for the BYE request.

CALL TERMINATED - In this state, the disconnect treatment associated with the called party's having disconnected the call is performed at the WAP layer. The SIP protocol state machine sends out a BYE and awaits a final response for the

BYE.

5.5.3 Call Flow

A typical WAP-SIP signaling sequence will be like this:

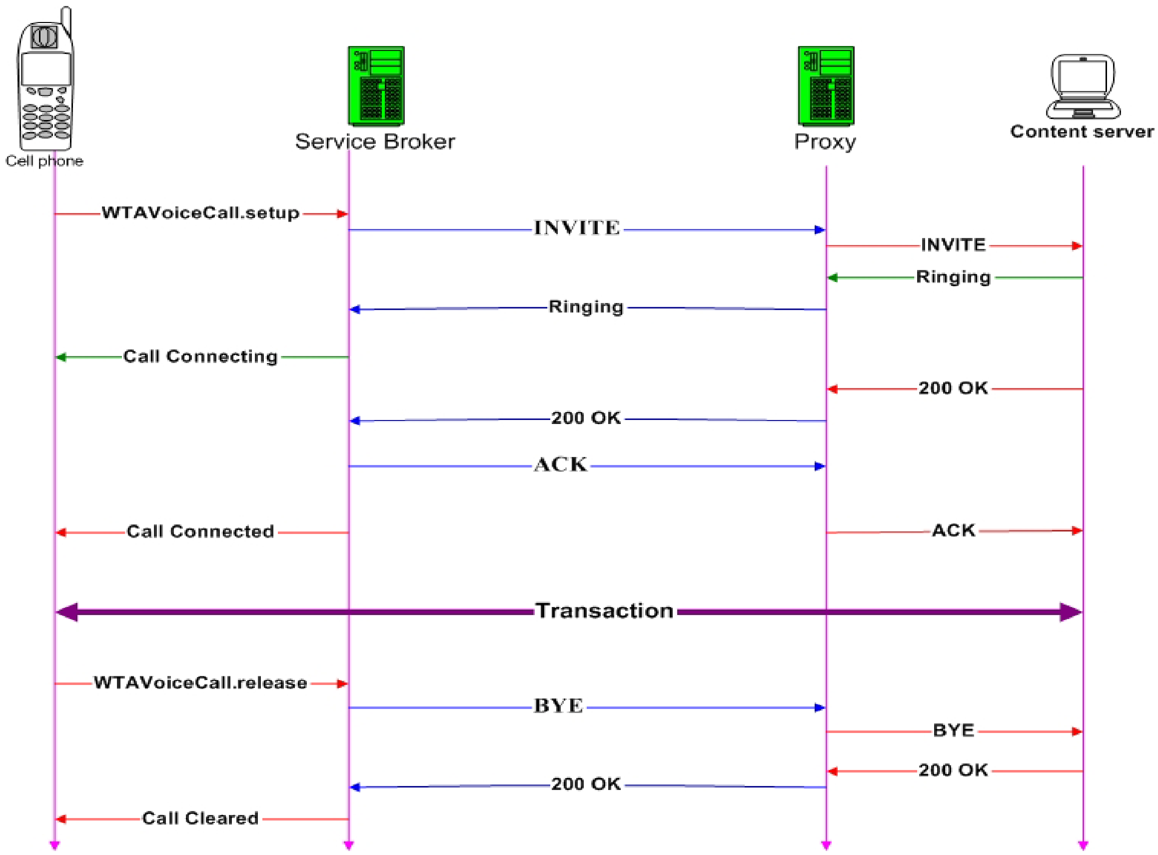


Figure 17 WAP-SIP signaling sequence diagrams

5.6 Defining SIP/WTA extensions

It is envisaged that the service broker shall have to support a component to facilitate the message conversion between SIP and WAP domain. This study shows that message mapping between SIP & WTA is a possibility and it might call for polymorphing the available WTA functions or even adding a few new ones (e.g., corresponding to SIP's REFER, REPLACE etc.) to the WTAVoiceCall library. The design & implementation aspects of this component are beyond the scope of this document.

6. Conclusions

Usage of SIP in 3GPP is focusing primarily on telephony issues (including the Instant messaging or SMS). The working group advocates the use of SIP stack in the wireless devices. Embedding SIP stack in the wireless devices is an issue worth pondering at and that this device may not provide rich experience of WAP applications to its end users because it appears that the 3GPP working group has not thought about any other applications like banking, mobile electronic transactions, simple web browsing etc on the same device other than just the telephony application using SIP signaling.

With this proposal, we can eliminate the need of the SIP stack in the wireless devices. This proposal still stands good for telephony, IM as well as SMS & MMS and many other features of this nature. In addition, WAP enabled devices allow its user to do the WAP Banking, Mobile electronic Transaction and other WAP related applications too. This also eliminates the need of constant updates of the SIP stack in the wireless devices. Thus we are expanding the user benefits and experience in addition to the telephony applications being proposed by 3GPP working group.

References

[1] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg -- RFC 2543 on SIP: Session Initiation Protocol, IETF/Network Working Group March 1999
[2] "MeT Core Specification" Version 1.0 (21 February 2001), http://www.mobiletransaction.org
[3] "MeT PTD Security Requirements" Version 1.0 (21 February 2001), http://www.mobiletransaction.org
[4] "MeT WAP Banking" Version A (21 February 2001), http://www.mobiletransaction.org
[5] "MeT Account-Based Payment" Version A (21 February 2001), http://www.mobiletransaction.org
[6] “WAP WTAI (ANSI 136), Version 7 July 2000, http://www.wmlscript.com/wap_spec/WAP-172-WTAIIS136-20000707-a.pdf
[7] “SIP Enhanced Mobile Network Services“ Ver 3.0 Ubiquity Software Corp, http://www.ubiquity.net/pdf/Mobile_Services_Whitepaper.pdf
[8] “SIP Service Architecture” Release 1.0, May 2001, Ubiquity Software Corp, http://www.ubiquity.net/pdf/SIP-Service-Arch-WP1_0.pdf
[9] “3GPP requirements on SIP”, Garcia et.al. http://www.ietf.org/internet-drafts/draft-garcia-sipping-3gpp-reqs-03.txt
[10] 3GPP TS 23.228 "IP Multimedia Subsystem (IMS) (Stage 2) - Release 5". Version 5.3.0, ftp://ftp.3gpp.org/Specs/2001-12/Rel-5/23_series/23228-530.zip
[11] “The SIP Replaces Header” http://www.ietf.org/internet-drafts/draft-ietf-sip-replaces-00.txt

ADDITIONAL READINGS:
[1] "Internet Communications enabled by SIP", Henry Sinnreich, WCOM, August 2000, http://www.sipforum.org/whitepapers/sinnreich.htm
[2] “The Wireless Application Protocol” by Sandeep Singhal et al. Addison-Wesley Publication
[3] “SIP – Understanding the Session Initiation Protocol” by Alan B Johnston, Artech House Publication.
[4] “SIP Demystified” by Gonzallo Camarillo McGraw-Hill Publication
[5] WAP Technical Specifications, www.wapforum.com
[6] SIP-IN Interworking Protocol Architecture and

Procedures, http://www.iit.edu/~gurbvij/I-D/draft-gurbani-sin-00.txt