A Prototype for the Provision of Mobility-Aware Personalized Services in Wireless Broadband Hotspots

Ian Herwono[1], Joachim Sachs[2], Ralf Keller[2]

[1]Communication Networks, Aachen University (RWTH), Aachen, Germany
[2]Ericsson Eurolab Deutschland GmbH, Herzogenrath, Germany
e-mail: Ian.Herwono@comnets.rwth-aachen.de, {Joachim.Sachs, Ralf.Keller}@ericsson.com

***Abstract*: The Media Point concept aims to facilitate the provision of personalized services to mobile users within WLAN hotspots. In this paper we describe our prototypical implementation of a Media Point network. An insight into our performance evaluation using the prototype is given. Our work has shown that the Media Point system functionalities can already be realized and demonstrated using state-of-the-art technologies. Based on our experience the system performance strongly depends on the level of interworking between the various software modules, e.g., SIP modules, DHCP modules, WLAN device driver, etc. In particular the signalling mechanism within the network by means of SIP doesn't cause any significant delay. Given that the dwell time of the mobile user within the hotspots is in scope of minutes or longer the overall system performance regarding session setup time of 6.3 seconds, wherein 39.03 Mbytes of user data should be pre-cached, and session resume time of 2.8 seconds is found as acceptable.**

1. Introduction

Nowadays many cellular operators begin to extend their cellular networks with WLAN[1] hotspots in public and strategic areas like train and bus stations, shopping malls, campus or downtown areas as they recognize the importance of broadband wireless Internet access for their subscribers. Users with suitable wireless equipments can "hop" from one WLAN island to another to get high-speed Internet access. Due to higher flexibility and mobility of the users the importance of personalized and customized Internet-based services increases. One main issue the corresponding network has to deal with is to be aware of the presence status and location as well as device capabilities of each mobile user. Since the dwell time of the mobile user within a hotspot is limited and the data transmission session could therefore be interrupted, the network should support fast and reliable session re-establishment and continuation at the next hotspot.

The Media Point service concept handles this issue and is introduced in [1], [2], [3]. In this paper we investigate the technical feasibility of the concept for delivery of personalized high-volume data for stationary or very-slowly moving users or terminals by means of state-of-the-art technologies in form of a prototypical network. After introducing the Media Point network architecture and its signaling mechanism we describe the setup of the prototype as well as the functionalities of its software modules. An insight into the experimental evaluation using the prototype is given afterwards.

2. Media Point Network Architecture

As shown in Fig. 1 four types of components form together a Media Point network: the *Media Point Service Control (MPSC)*, the *Media Point Controllers (MPCs)*, the *Access Points* or *Media Points (MPs)*, and the *Mobile Terminals (MTs)*. MPSC is the central logical entity of a Media Point network and provides the interface to cellular network and Internet. It usually functions as a *Presence Server (PS)* as well to monitor the presence status and location of each active mobile user. MPSC supervises all MPCs belonging to that particular network. Each of the MPCs controls a group of MPs which are usually located in close geographical proximity. MPCs should be placed in rooms to ensure appropriate working conditions (temperature, humidity, place for cache storage, etc.) whereas the transceivers (MPs) might be operating in outdoor environment. A group of mobile terminals is finally associated to each of the MPs.

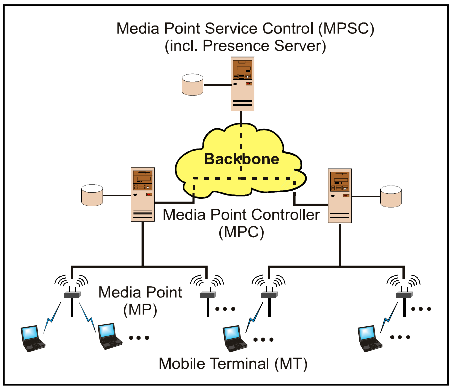

Figure 1: Media Point network architecture

3. SIP-based Signaling in Media Point Network

The mobility and session management within the Media Point network is handled by the Session Initiation Protocol (SIP) [4]. As already mentioned the presence status of each user is stored in the Presence Server (PS) [5] (co-located in MPSC). After successful subscription of the user's presence status at PS (using SUBSCRIBE method), MPSC is notified when the status changes (using NOTIFY method). The change is detected by PS in the way that the terminal (MT) registers itself at an MPC (using REGISTER method) after it is associated with an MP (MPC forwards the

[1] Wireless Local Area Network

registration to PS). Whenever new personalized data is available at MPSC and the addressed user is online, a new push session, which is described using the Session Description Protocol (SDP) [6], is initiated (using INVITE method). The data is transferred completely first to MPC before it is passed to MT. The "smart pull" approach is applied to push the data since it allows a MPC or MT to decide by itself how and when the data should be downloaded depending on its capability, e.g., cache/storage size, computational power, link quality, etc.

4. Prototype Implementation

A prototypical Media Point network has been realized in order to prove the concept and to evaluate the performance of the proposed protocols experimentally. It is based on IEEE 802.11b WLAN system and demonstrates two types of personalized service, i.e., push of (subscribed) multimedia contents such as video trailers or music files, and e-mail delivery (mailbox update).

4.1. System Setup

As depicted in Fig. 2 our prototypical Media Point network consists of three PCs (i.e., two acting as MPCs and the other as MPSC), two IEEE 802.11b Access Points (APs), and a laptop (user's WLAN terminal). The MPSC is connected to the local area network (Ethernet) and to both MPCs, each of which controls a single AP. MPSC and MPCs build the Media Point core network with two hotspots. Both MPCs are connected via 100 Mbit/s Ethernet link with MPSC. Linux is used as the operating system. In order to avoid interferences both APs are configured to operate on different channels, i.e., channel number 1 and 7.

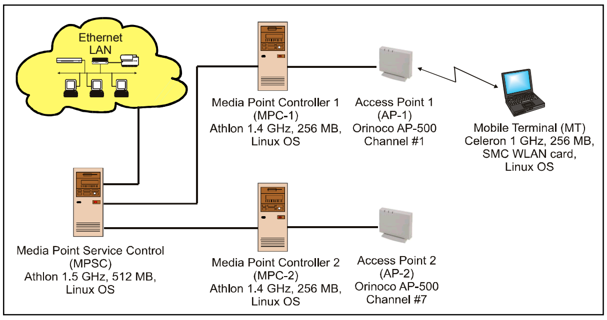

Figure 2: Setup of the Media Point network prototype

4.2. Software Modules and Functionality

In Fig. 3 the software modules of MPSC, MPC, and MT are depicted. The interactions between the modules within one network component are shown as well. The communications between the modules contained in different network components are handled via standard Socket API[2].

The implemented functionality for a complete SIP-controlled push session can be summarized as follows:

1. The user profile is stored in the MySQL database of MPSC. By using a regular Internet connection, the user can configure and update the profile via the web server of MPSC. According to the configured user profile the MPSC updates the personalized data of the user and stores them appropriately.

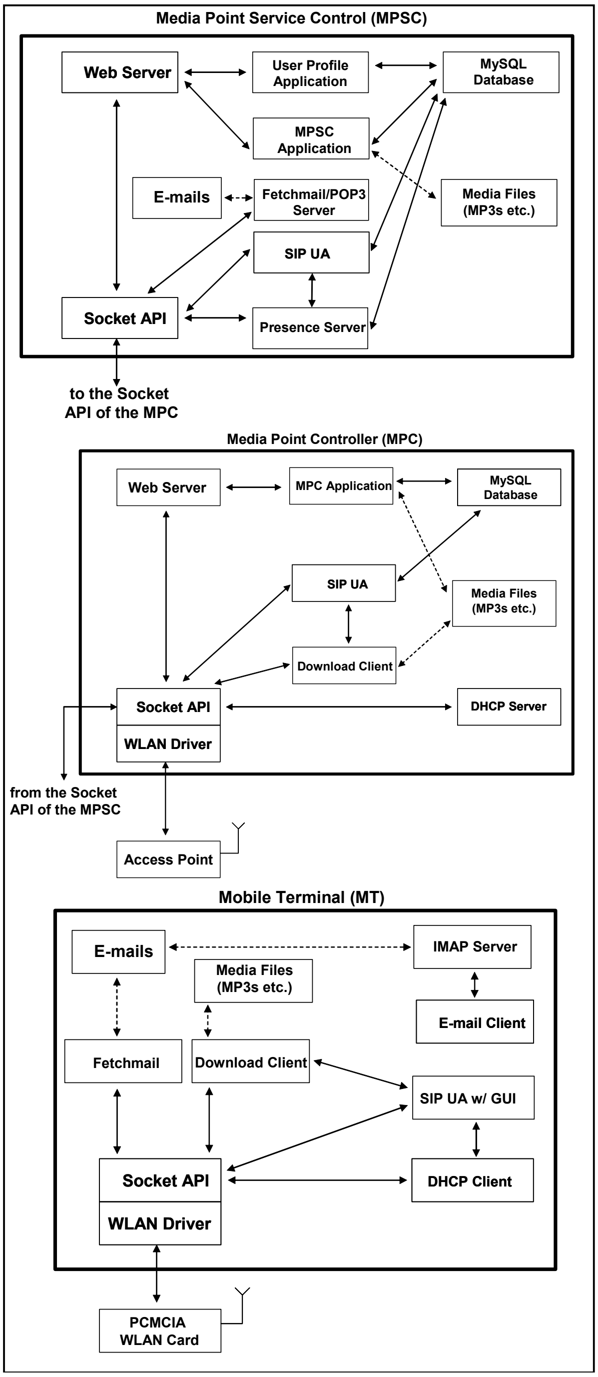

Figure 3: Software modules of MPSC, MPC and MT

2. The SIP Presence server is co-located with the SIP User Agent (UA) in MPSC. Hence an internal SIP communication regarding SUBSCRIBE and NOTIFY between both processes is realized. In order to react, MPSC (its Presence Server) waits for a REGISTER message coming from a SIP user, i.e., the SIP UA client installed in MT.
3. Now, the user using his/her mobile terminal establishes a WLAN connection with one of the Access Points, i.e., AP association. After establishing the physical connection, the DHCP[3]

[2] Application Programming Interface

[3] Dynamic Host Configuration Protocol

client at the MT requests a local IP address from the DHCP server located at MPC. In SIP context MPC functions as a SIP proxy server. MT knows the IP address of the MPC by recognizing it from MPC's response packets.

4. After getting a new IP address, the SIP UA at MT sends a REGISTER message, which is addressed to the Presence Server, to MPC, which in turn forwards the message to MPSC. The IP address of MPSC is usually known and fixed. The Presence Server receives this REGISTER message and responds with an OK message, which is then received by MPC and forwarded to the SIP UA of MT. The registration triggers a presence status change for the user and thus a notification to MPSC. After receiving NOTIFY message from the Presence Server, the MPSC's SIP UA knows via which MPC the user can be contacted and prepares the data to be pushed to MPC. The REGISTER message is sent periodically by MT as long as it remains online.
5. MPSC's SIP UA establishes a SIP session with the MPC by sending an INVITE message and waiting for an OK message which is to be acknowledged.
6. Now the user data available in MPSC must be transferred to MPC. By means of SIP MESSAGE[4] method, arbitrary messages can be pushed to MPC. Since MESSAGE is designed for containing small instant messages of up to 1300 bytes, high volume data should be pushed in another way. As recommended in SIP standard, a mechanism called "content indirection", wherein a list consisting URLs [5] of the data should be included in a MESSAGE, is implemented. MPC can then download the data through the web server at MPSC, i.e., "smart pull" approach. The downloads are carried out in background by using the GNU tool *wget* [6] referred to as the download client. Beside downloads of regular MIME [7] contents (audio, video, images, etc.) via HTTP, *wget* also supports the retrieval and mirroring of web sites as well as resume of interrupted download sessions. These capabilities make *wget* suitable as download client in the system prototype.
7. Hence, in accordance with the current state in MySQL database the MPSC's SIP UA sends a MESSAGE containing a download location (URL) for a particular SIP user and the number of available personalized data (contents). We make use of the capability of *wget* to follow up each URL link, which is included in a HTML document and represents each user data to download. The MPC's SIP UA can download the data in background, acknowledges them and stores them in its own MySQL database. After all data has been acknowledged the SIP session is terminated with a BYE message sent from MPSC.
8. The downloaded data remains stored in MPC for a specific time. It is deleted automatically after the user (MT) has downloaded it or after time out.
9. The same download procedure using the "announce-pull" mechanism is carried out within a SIP session between MPC's SIP UA and the user (MT's SIP UA) too. E-mails are fetched separately from the MPSC's POP3[8] server (or directly from the e-mail server in the Internet). An IMAP[9] server is installed in MT to allow regular e-mail clients to fetch and display new e-mails, that are already stored in the MT's local disk.
10. In case the user leaves the Access Point, and enters the coverage area of a new AP, after getting a new local IP address the MT sends a REGISTER message again towards MPSC and the same procedure as described above is repeated.
11. The MT's SIP UA provides the user with a Graphical User Interface (GUI) to indicate the connection and download status and to display a list of new media files and e-mails. Fig. 4 shows a screenshot of the GUI.
12. For demonstration purpose, a number of sets of data (each set containing different media files, e-mails, etc.) are created and stored in MPSC. A PHP[10] script is used to control the availability of stored data. The databases (in MPC and MPSC) are used to track which data has been downloaded by an MPC, and which one has also been downloaded by the user. An acknowledgment for a specific data is sent by MPC to MPSC, when this data has been downloaded by the user. Fragmentation of each data, e.g., due to download interrupts, is not specially considered during the implementation. However, this feature is supported by the download client *wget*, i.e., the data fragments that have been downloaded to the terminal prior to session/link interruption need not be transferred after session resume anymore. The (cached) data in MPC (and MPSC) is only deleted after it has been acknowledged by the user (by means of a SIP message).

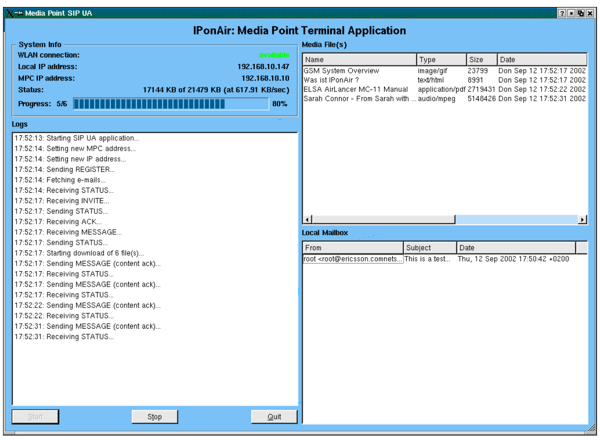

Figure 4: GUI of the Media Point demonstrator application

[4] The SIP MESSAGE method is usually used for sending instant messages.
[5] Uniform Resource Locators
[6] http://wget.sunsite.dk
[7] Multipurpose Internet Mail Extensions

[8] Post Office Protocol 3
[9] Internet Message Access Protocol
[10] Hypertext Preprocessor

4.3. SIP Implementations

Totally four different SIP program modules are implemented in the prototype: SIP UA and SIP Presence Server at MPSC, SIP Proxy at MPCs, and SIP UA at MT. An open source C++ SIP library provided by *Vovida.org* (version 1.4.0b) is chosen for the implementation as it supports most of the SIP methods and extensions. The SIP Proxy software for each MPC is identical. Due to co-location of SIP UA and Presence Server in MPSC an inter process communication (IPC) between both modules is implemented. Furthermore the source code of the used DHCP client [11] is slightly modified in such a way that an IPC between DHCP client and SIP UA both at MT can be realized. In this way information about IP address of the serving MPC and the currently assigned IP address can be exchanged between both modules. It triggers SIP UA to send a REGISTER message to MPC, which in turn forwards it to Presence Server. The user's presence information included in NOTIFY messages (generated by Presence Server) is coded in XML [12] in accordance with SIP standard for instant messaging [7]. An additional XML parser is implemented in MPSC's SIP UA in order to evaluate the information value, i.e., whether or not the subscribed user is online and via which MPC the user can be contacted.

4.4. Database Implementations

A MySQL database containing entries of personalized data for SIP users/subscribers is created for MPSC. By using a web-interface the database can be configured or modified, and new contents (e.g., song files, videos, etc.) can be manually added. Each file can be made accessible to a single user or to all or a group of users. Furthermore a PHP-based download script is implemented for retrieving the data available to a user, which is identified by his/her global SIP address. The resulting HTML file (generated by the script) contains a list of data to be downloaded by *wget*. A similar MySQL database is created for each MPC as well.

4.5. E-mail Delivery Implementations

In the prototype e-mails are delivered to a user (MT) independent of push sessions carried out between MPSC, MPC and MT. As shown in Fig. 5 the user's e-mails are retrieved periodically by the *fetchmail* [13] daemon of MPSC from the corresponding Internet e-mail servers and are stored in the local POP3 server repository. Each time MT has established a new WLAN link and been assigned an IP address, its *fetchmail* client retrieves the e-mails from MPSC using POP3 protocol and stores them in its local IMAP server repository. The end user can read the e-mails using regular IMAP-capable clients such as *Netscape Mail*. IMAP was chosen for implementation because it allows for not transferring complete e-mail messages to client unnecessarily (i.e., only the headers) as both IMAP client and server are co-located in MT. In case that a new e-mail arrives at MPSC while the user is online, MPSC sends an announcement to MPC using SIP MESSAGE method. MPC forwards this announcement to MT which can fetch the e-mail from MPSC's POP3 server by itself.

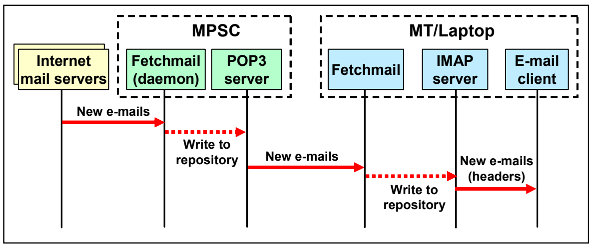

Figure 5: E-mail delivery implementation

In the future, e-mails may be handled as regular personalized MIME contents that can be transferred to user's terminal together with other multimedia contents (e.g., video, audio, etc.) within SIP-controlled push sessions. Specific encapsulation and decapsulation modules must then be implemented in involved network components.

5. Performance Evaluation

The main goal of our performance evaluation is to determine the session setup and resume times using the prototypical Media Point network. The *session setup time* is defined as the time span between the beginning of AP association (WLAN connection setup) and the time point where the user's terminal is able to download the first data segment (from MPC). Further a session between a MPC and a user's terminal might be interrupted due to lost connectivity with the AP, for example. In order to resume the session at another AP (or at the same AP) which is still controlled by the same MPC, some procedures described in Section 4.2 must be repeated (assumed that the data is still cached in MPC [14]). The needed time is defined as *session resume time*.

5.1. Evaluation Scenario

The evaluation is carried out with a single stationary user's terminal (laptop) and under condition of an optimal wireless link quality (net data rate ~500 kbyte/s) between terminal's WLAN card and the APs. Cross traffic induced by other users as well as co-channel interference between both APs (or other APs in close proximity) can be excluded. The average values for session setup and resume times are based on at least five independent measurements, i.e., the measurement is ended after five trials when no significant differences can be observed between the ascertained values.

The following scenario is used for the measurements: the user has subscribed to several data services like a music hit list and announcements of movie trailers. During the test session he/she receives one MP3 music file and one movie trailer. In total this results in a data amount of 39.03 Mbytes which are to be transferred to the terminal during the session. In addition to that he/she receives a new e-mail of 1,375 bytes.

[11] A DHCP client implementation provided by ISC (Internet Software Consortium), http://www.isc.org.
[12] Extensible Markup Language
[13] http://www.catb.org/~esr/fetchmail/

[14] The data will be deleted from MPC after a timeout.

5.2. Session Setup and Resume Time

An initial session setup procedure takes place when user's terminal (MT) establishes a connection with one of the APs for the first time. The procedure thus comprises the following sub-procedures:

- AP association,
- IP address assignment via DHCP,
- SIP registration at MPC and MPSC,
- Caching of user data in MPC, and
- Initiation of push session with MT.

AP association and DHCP

Fig. 6 shows the message exchange between MPC, AP and MT during AP association and DHCP procedure. The corresponding message sizes and elapsed times are indicated. AP initialization or restart time is the time needed by an AP to go to standby after it is powered on. Powering on and off an AP is done during the measurements to simulate the situation where MT is within or outside an AP's coverage area. The AP restart time is measured from the time point the AP is powered on until it sends an ARP [15] packet towards the corresponding MPC to re-confirm its IP address (since each AP is configured with a fixed IP address).

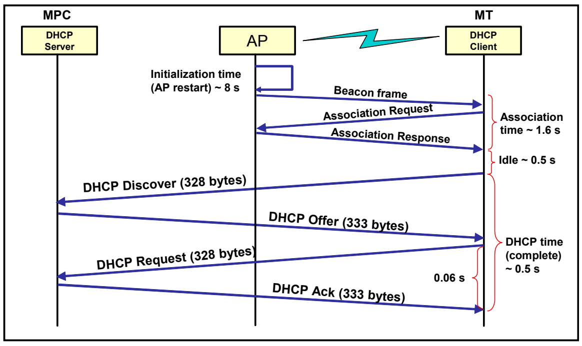

Figure 6: AP association and DHCP time

Being in standby mode the AP periodically broadcasts its cell and transmission parameters within a beacon frame on its operating frequency. After MT receives such beacon frame it starts an association procedure by sending an Association Request message to AP, which then responds with an Association Response message in case that MT is authorized to use the AP's service[16]. In average about 1.6 seconds are needed to complete such association irrespective of current link quality.

After completing the association MT remains in an idle state until it broadcasts a DHCP Discover message. The idle time depends on the operation mode of DHCP client in MT. Two types of operation mode are investigated for the prototype: *daemon* and *manual* mode. In daemon mode the client periodically broadcasts a DHCP Discover with an interval of around 1.0 second as long as no response is received from a DHCP server. In average an idle time of 0.5 seconds is measured. Operating in daemon mode may lead to minimum idle time but cause inordinate load on server (i.e., MPC) and especially on AP. The manual mode aims to overcome this problem. In this mode a DHCP Discover message is first broadcast when a new connection to an AP is established. By tracing the information provided by the WLAN card, such "new AP detected" event can be captured. Since the used DHCP client implementation doesn't support such manual triggering, after a new AP is detected the client daemon program must first be shutdown and then restarted immediately. The idle time corresponds to the time needed to shutdown and restart the client. We observed that relatively much time of up to four seconds are needed to initialize the client program, i.e., for memory allocation and writing, reading the configuration file, etc.

Hence, since it has been shown by daemon mode that a fast IP address (re-)assignment can be realized, the combination of both daemon and manual (user-triggered) modes would be the optimal solution whereby the idle time can be minimized without causing high load on MPC and AP. This requires extensive source code modification of the client program we used for the prototype.

As indicated in Fig. 6 the actual DHCP procedure can be completed within another 0.5 seconds. By using daemon mode a total time of around 2.6 seconds is needed for AP association and IP address assignment.

SIP registration at MPC and MPSC

In order to update its current presence status MT sends a REGISTER message to the corresponding MPC, which in turn forwards it to SIP Presence Server in MPSC, as shown in Fig. 7. SIP PS sends then a NOTIFY message to MPSC's SIP UA in case that the user's presence/location status has changed since last time. A confirmation message (OK) is sent back by MPC immediately after receiving the REGISTER message, without waiting for the confirmation (OK) message coming from MPSC. This allows MT to proceed with the next procedure, e.g., mailbox update.

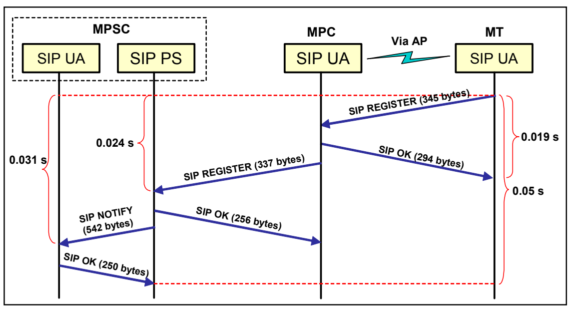

Figure 7: SIP user registration at MPC and MPSC

We observed that the delays caused by SIP registration are negligible for the total session setup time. In average after 0.019 seconds the registration is complete for MT. The total time needed until MPSC's SIP UA acknowledges the change of user's presence status, averages to 0.05 seconds. MT resends the same REGISTER message every 120 seconds to refresh its presence status in SIP PS. If there is no status change,

[15] Address Resolution Protocol

[16] In the prototype the authorization is based on MAC address filtering only. The standard WLAN security protocols are deactivated.

SIP PS needs not to resend a NOTIFY message to MPSC's SIP UA.

Caching of user data in MPC

As already mentioned in Section 4.2. the user data must be pre-cached in MPC before it can be pushed to MT. As depicted in Fig. 8 a push session is first initiated between MPSC and MPC by means of INVITE method. In average this takes 0.012 seconds. Another 0.017 seconds are needed to send the download information (i.e., download URL) by means of MESSAGE method to MPC. After around 0.04 seconds the session initiation is completed and the download process can be started.

According to our evaluation scenario two independent data files with a total of 39.03 Mbytes should be cached, which requires around 3.45 seconds to complete. The established push session is terminated by MPSC by sending a BYE message to MPC. This termination sub-procedure only needs a small fraction of time, i.e., around 0.013 seconds.

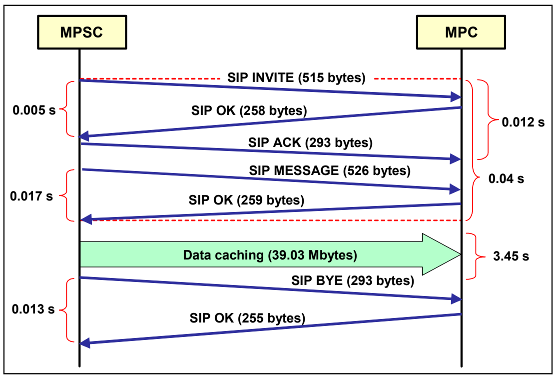

Figure 8: Caching of user data in MPC

Initiation of push session with MT

In this sub-procedure a push session is now established between MPC and MT. As indicated in Fig. 9 it takes 0.212 seconds to complete. The subsequent MESSAGE provides the download URL of the data cached in MPC. In average 0.27 seconds are required to prepare the push session before MT can start download user data from MPC.

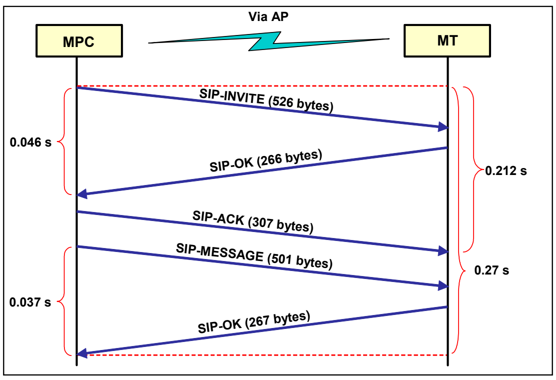

Figure 9: Initiation of push session with MT

Summary

The initial session setup time for the considered scenario, wherein 39.03 Mbytes of user data must be pre-cached, has been found to be in average 6.3 seconds. After a temporary loss of connectivity, the average session resume time takes 2.8 seconds if the new AP belongs to the same MPC that has already pre-cached the user data. If the new AP belongs to another MPC the mean session resume time increases to between 2.8 and 6.3 seconds, depending on when during the push session the connectivity was lost, i.e., on how much data that has already been transmitted to MT in the previous (interrupted) session. The performance has been found to be primarily effected by three procedures: AP association, IP address assignment via DHCP and data caching in MPC. The impact of SIP signaling is insignificant.

5.3. E-mail Delivery

Considering the e-mail delivery implementation in the prototype it is interesting to experience how long it would take MPSC for announcing a new e-mail to an online user. As shown in Fig. 10 the announcement by means of SIP MESSAGE method can be received within 0.025 seconds. MT retrieves the e-mails from MPSC's POP3 server afterwards.

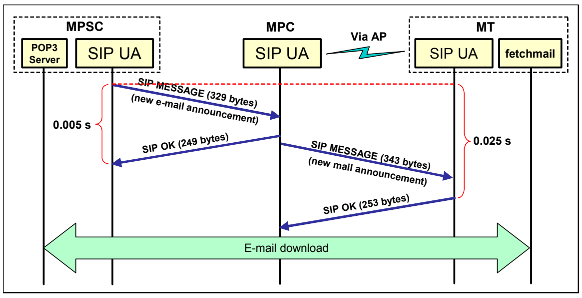

Figure 10: Announcement of new e-mails

The retrieval of a single e-mail of 1,375 bytes takes in average 3.2 seconds. As depicted in Fig. 11 a TCP connection must first be set up between MT and MPSC since POP3 is a connection-oriented protocol. Next MPSC performs the Identification Protocol (ident) to determine the user identity for the established TCP connection [8]. After determining the capability (supported features) of POP3 server, user's e-mail account information (i.e., username and password) is provided by MT and the number of new e-mails in user's mailbox (in server) is indicated. MT can then update its local mailbox. In case of one new e-mail it takes in average 0.1 seconds to query the mailbox content status and another 3.1 seconds to retrieve the e-mail.

In the retrieval phase a list of available e-mails with the corresponding size in bytes is transferred first. Depending on the number of e-mails the message size (IP packet size) may thus vary. This list download procedure needs around 3.05 seconds to complete. The number of available e-mails do not vary the completion time significantly . As indicated in Fig. 11, almost three seconds of processing delay in client (MT) are observed. This delay might be implementation-specific, which is not analyzed further.

Each e-mail, that is included in the list, can then be requested by the client. The list allows the client to select which e-mail(s) to retrieve. In case that several e-mails are available, each e-mail will be downloaded one after another (each with a preceding request message). Depending on the client configuration, retrieved e-mails can be removed from the POP3 server. The time needed for requesting each e-mail or deleting it from the server is size-independent and negligibly small, i.e., 0.037 seconds or 0.0026 seconds, respectively.

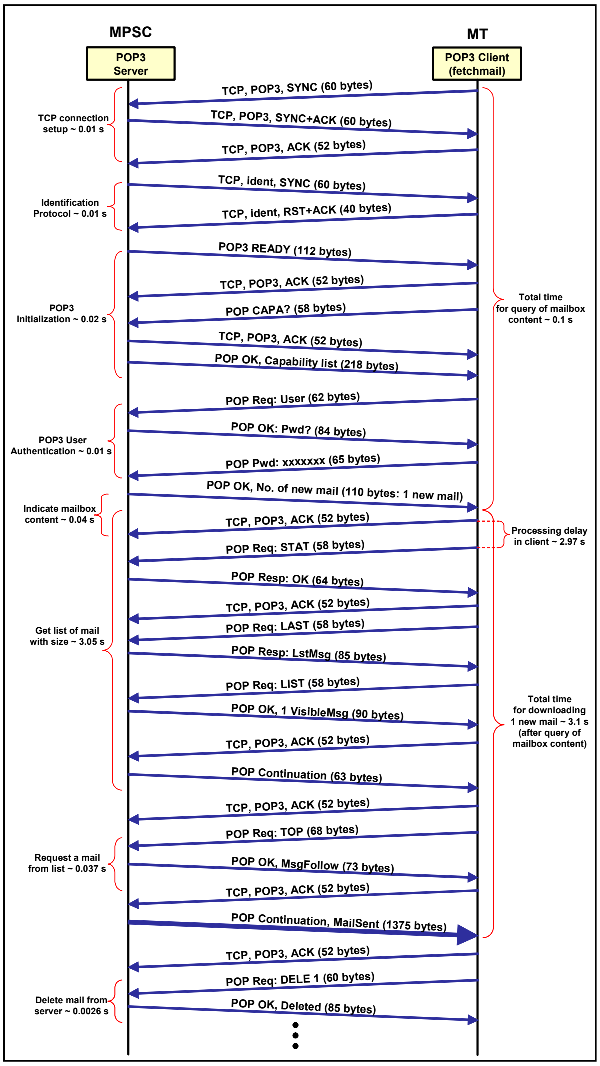

Figure 11: Retrieval of a single e-mail of 1,375 bytes via POP3

6. Conclusions

Our work has shown that the Media Point system functionalities can already be realized and demonstrated using state-of-the-art technologies. We observed that the system performance is strongly dependent on the level of interworking between the various software modules, e.g., SIP modules, DHCP modules, WLAN device driver, etc. In particular the signaling mechanism within the network by means of SIP doesn't cause any significant delay. The overall system performance regarding session setup and resume times is found as acceptable when assuming that the dwell time of mobile users within WLAN hotspots is in scope of minutes or longer. However since the initial session setup time increases with the volume of user data, more sophisticated pre-caching mechanism and strategy should be developed and applied for the Media Point network. One simple approach might be to allow the data transfer to user's terminal begin immediately after a specific amount of data is already buffered in MPC, i.e., before all the data is completely cached in MPC (as it is now the case in our current prototype). Furthermore it has to be noted that no evaluation with a high number of simultaneous mobile users in a large scale Media Point network has been carried out yet. We expect that this will have major impact on net data rate during the transfer of user data over the air interface, i.e., after the push session with terminal is initiated. But it might not impact the session setup or resume time significantly because relatively small amount of signaling data must be exchanged over the air during the session setup or resume procedure[17].

Finally we have observed that our current e-mail delivery implementation may cause significant delays especially when several e-mails should be retrieved at once. It should be deeper investigated whether or not the encapsulation of e-mails as regular multimedia (MIME) contents to be pushed to users along with other contents (e.g., video, audio, etc.) can achieve better performance.

ACKNOWLEDGMENT

The work presented in this paper has been supported by the German Federal Ministry of Education and Research in the project "IPonAir" (01BU163)[18].

REFERENCES

[1] B. Walke, J. Habetha, I. Herwono, R. Pabst, D.C. Schultz. "*The Wireless Media System: A Mobile Broadband System with Invisible Infrastructure and Low Radio Exposure of Humans*". In ANWIRE 1st International Workshop on "WIRELESS, MOBILE & ALWAYS BEST CONNECTED", University of Strathclyde, Glasgow, April 2003.

[2] B. Walke, I. Herwono, R. Pabst. "*Service Architecture for Infrastructure based Multi-hop Networks based on SIP*". In Proceedings of ICCT 2003 International Conference on Communication Technology, April 2003.

[3] G. Plitsis, R. Keller, J. Sachs. "*Realization of a Push Service for Media Points based on SIP*". In Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks, Stockholm, Sweden, September 2002.

[4] J. Rosenberg, et al. "*SIP: Session Initiation Protocol*". RFC 3261, June 2002.

[5] A. Houri, T. Hiller, T. Hansen. "*SIP/SIMPLE Based Presence and IM Architecture*". SIMPLE Group Internet Draft, June 2003.

[6] M. Handley, V. Jacobson. "*SDP: Session Description Protocol*". RFC 2327, April 1998.

[7] H. Sugano, et al. "*Common Presence and Instant Messaging (CPIM) Presence Information Data Format*". Internet Draft, May 2002.

[8] M. St. Johns. "*Identification Protocol*". RFC 1413, February 1993.

[17] Refer to Section 5.2.

[18] The authors are responsible for the content of this publication.