

# A Multimedia Alerting and Notification Service for Mobile Users

Bin Wei, Yih-Farn Chen, Huale Huang, and Rittwik Jana

AT&T Labs - Research  
180 Park Avenue  
Florham Park, New Jersey, 07932, USA.

**Abstract** – Delivering time-critical information to mobile users is extremely important when emergency actions are needed. This paper presents our ongoing research efforts towards providing ubiquitous multimedia alerting services for mobile users. The approach consists of unifying a middleware messaging platform with that of a multimedia service engine. We use the messaging platform to provide user authentication, authorization and accounting. The multimedia engine is used to manage multimedia content including real-time multimedia acquisition and delivery. The system allows users to access the services from inside or outside a firewall. There is also an acknowledgement facility to report whether the alert media content has been received by the end users.

**Keywords:** Wireless, mobile devices, service platform, multimedia messaging, alert and notification.

## 1. INTRODUCTION

With the increased popularity of mobile devices and wireless access, mobile users who are first respondents to emergency situations are beginning to demand the delivery of multimedia based alert content to cope with the crisis more effectively. This paper describes a service platform that strives to deliver such rich and time-critical information to all the relevant mobile users in a secure and timely manner.

For example, when a life threatening situation occurs in a public place (such as bombing in a subway station), it is very important to notify the medical professionals, policemen, local officials, and emergency crew via any means that they may have access to. With real-time video upload capability by mobile users on the site, some site conditions (injuries and damages) can be seen readily on devices owned by doctors or the emergency crew. They will be able to make an assessment and recommend more adequate solutions before they even get a chance to get to the site in person.

To build such a service platform to deliver multimedia alerts, we need to address several research issues:

- *User Management:* How does the system determine to whom the content needs to be delivered?

- *Access Management:* How does the system authenticate and authorize user requests, and allow users access from outside a firewall?
- *Device Adaptation:* How does the system repurpose the multimedia content to mobile users with different mobile devices?
- *Content Acquisition:* Can the system acquire real-time content from mobile devices on the scene and make it available to other relevant mobile users?
- *Quality of Service:* How does the system prioritize requests and ensure that urgent requests always get through with a delivery guarantee?
- *Delivery Control:* How does the system provide a secure method such that the content is only available to those who should receive it? How does the system know whether the content has indeed been delivered and the recipient has viewed it?
- *Scalability:* How does the system support a large number of mobile users in a scalable way?

We study these issues by building a prototype system for multimedia alerting services by leveraging an existing enterprise mobile service platform. There are three components in the proposed system: EMN (Enterprise Messaging Network), AP (Alerting Platform) and MP (Multimedia Platform). In the next section we present the system architecture to address the above issues. Section 3 describes the implementation with the demonstration of a real-time application. Section 4 discusses related work and section 5 concludes the paper.

## 2. SYSTEM ARCHITECTURE

To describe the overall architecture, we first briefly explain each of the three components and discuss how they are integrated to fulfill our goal.

### 2.1 ENTERPRISE MESSAGING NETWORK

EMN [1] consists of gateways and servers, interconnected via message queues. A database system is also used to store application related data as shown in Figure 1. A mobile device always interacts with an EMN gateway to access the services. EMN gateways support different protocol interfaces to

communicate with various mobile devices. We call the interfaces “devlets”. Besides providing a protocol interface, a gateway authenticates a mobile user and puts each service request on the message queue. A set of servers are provided to pick up messages from the message queue in a round-robin fashion and serve the request by invoking corresponding service engines through the interfaces which we call “infolets”. Both EMN gateways and servers communicate with the service profile database, which assists in the transcoding and content delivery process.

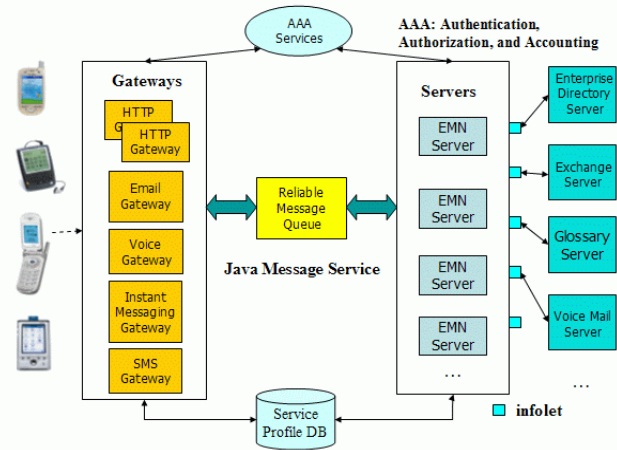


Figure 1 - EMN service platform

## 2.2 ALERTING PLATFORM

An alerting platform (AP) generally consists of an alert management platform (AMP) and an alert dissemination engine (ADE) [15]. Figure 2 shows an instance of the alerting platform with ADE built on top of a customized EMN instance. The AMP is responsible for handling the business logic of alert injection. The ADE functionality is made available through three web service interfaces, namely user provisioning, alert notification and alert reporting. There is also one infolet that offers the alert acknowledgement interface to the user.

From a provisioning and alert management perspective the end user interacts directly with the AMP system block and never with ADE. The provisioning interface allows the AMP to add, delete or edit recipients and their devices. Consequently, relevant user profile information is already available to ADE at the time of the alert submission in order to efficiently perform the dissemination. The user interacts directly with the ADE system during the acknowledgement process.

A typical alert can be injected by a personnel interacting with the AMP from a desktop browser and then disseminated to end recipients via the ADE. Alternatively, a user can initiate a multimedia alert from a mobile device and have the content manager and the delivery manager in the multimedia platform participate in the distribution, which is discussed in the following section.

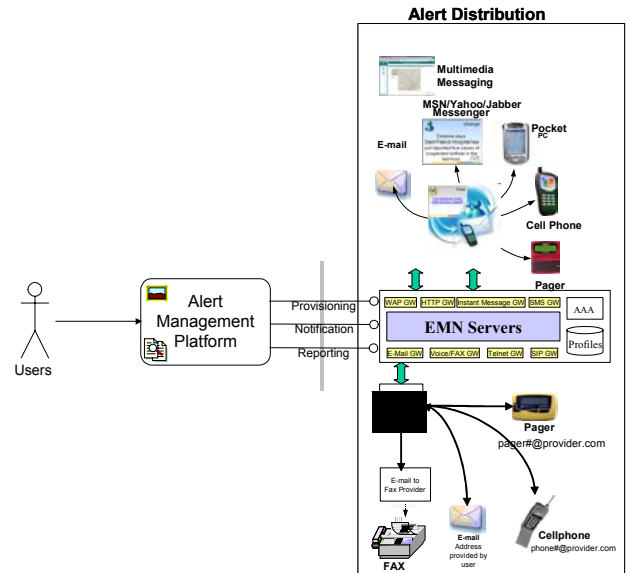


Figure 2 – Alert Management Platform and Alert Dissemination Engine

## 2.3 MULTIMEDIA PLATFORM

The multimedia platform provides the functions for collecting and storing multimedia content, performing transcoding when necessary, retrieving content from the storage system, and streaming the multimedia content to mobile devices in appropriate formats. Our design principle in building a multimedia platform is to separate *control flow* from *media flow*. Thus the multimedia platform consists of a *content manager* and a *delivery manager*, which handle the control flow and media flow respectively. The major functions that the content manager provides are:

- *Video storage*: Manage a storage system for collecting multimedia content.
- *Video acquisition*: Acquire real-time multimedia streams when needed.
- *Video transcoding*: Transcode or transrate video streams for delivery purpose.
- *Content Selection*: Send the multimedia content in the proper format for the target device to the delivery manager for streaming or downloading.

- *Content Addressing*: Generate access information for mobile users to access multimedia content directly from the delivery manager.

The delivery manager is responsible for transferring multimedia content to mobile devices as requested in real-time. The separation of delivery manager from the content manager helps us address the scalability issue more effectively. There can be many delivery servers serving the role of the delivery manager because streaming protocols may vary among different mobile devices. Each server can handle streaming in a particular format and more servers can be added for service scalability.

It is the content manager's responsibility to provide an end user the means to access the multimedia stream and also to transfer the proper stream to the delivery manager which handle the final media flow. Not all mobile devices are equipped with streaming capability and hence we should also allow mobile devices to download the video file and play it locally. The real streaming or downloading process happens directly between the delivery server and end users. Figure 3 is a diagram of the multimedia platform.

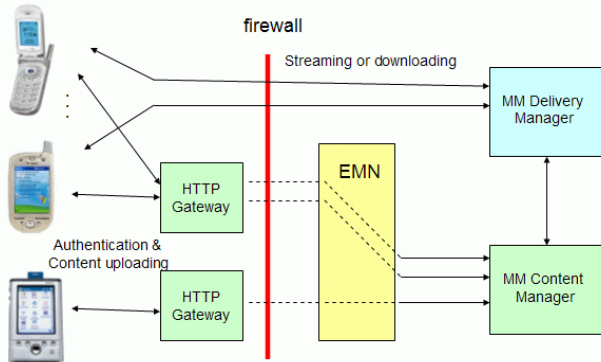


Figure 3 - System architecture for multimedia services

## 2.4 MULTIMEDIA ALERTING SERVICE

We have described the three major pieces for providing a multimedia alerting and notification service. In what follows, we describe how to integrate them together to provide the multimedia alerting service and to address the issues we raised at the beginning.

There are several procedures that constitute an end to end multimedia alerting service:

- User and device provisioning
- User authentication

- Real-time video uploading
- Alert posting
- Alert notification
- Multimedia delivery.

EMN, as a middleware platform, provides a front-end interface to authenticate mobile users before they can interact with the platform or receive service results. A mobile user typically interacts with an AMP (Alert Management Platform) to request the privilege to post a video alert.

To integrate the multimedia platform with EMN and utilize AMP functions, we implemented a multimedia infolet (MMIlet) on EMN servers. MMIlet acts as a bridge between mobile users and the multimedia content manager. MMIlet provides two basic functions: the first is to allow a mobile user to upload a multimedia alert and consequently generate the alert to end users via the AMP and ADE. The second function is to pass the access information to end users for them to retrieve video alerts.

In the case of uploading video to the multimedia platform, the content manager either provides an upload page to allow mobile devices to push the media content to the content manager, or communicates to a server on the mobile device to pull the media content. In the first situation, the browser on the mobile device must support the upload function to a web server. In the second situation, a server that supports file transfer must be running on the mobile device. In either situations, the video is stored and transcoded when needed.

The content manager also provides users the interface to post a multimedia alert through AMP and also keep the mapping of alert with the media file. Once the content is ingested and ready for use, it is transported to the delivery manager. On the other hand, AMP manages the alert posting according to a delivery plan which contains a set of recipients to be reached. The alert messages which have text, alert priority and alerting time, also include the access information on how to access the alert video stream. This access information can be a dynamic URL or a piece of code. The access information is based on what the content manager provides so that the information obtained through AMP matches the content available at the delivery manager for the same alert.

. After a multimedia alert is posted, relevant users are notified according to their preferred notification profiles, followed by an acknowledgement process that tracks the outbound notifications. When a mobile user follows the instructions in the notification to view the multimedia content, he needs to authenticate himself to EMN. Once authenticated, the content manager generates respond to the user's multimedia request. It communicates to EMN servers and the delivery manager directly. On the one hand, the content manager informs the delivery manager where and how to place the multimedia

content for mobile users; On the other hand the content manager, through the EMN server, informs mobile users about the access means for viewing the video content. With this information, mobile users can retrieve or receive the multimedia content directly from the delivery manager. If users are outside a firewall, the access information can be encrypted when it is delivered to mobile users. The delivery manager is also associated with a time-to-live attribute. When the user session expires, the content is no longer available to users.

Now let us consider the research issues we discussed at the beginning. In EMN, there is a database that maintains user and device profiles. For *user management*, we use the information from the database (in conjunction with the alert management platform) to determine the set of valid or authorized recipients of the multimedia content. For *access management*, we use AAA (Authentication, Authorization, and Accounting) services in EMN to authorize user requests. For *device adaptation*, depending on the user's access mechanism through an EMN gateway and the user's profiles in the database, the multimedia content manager repurposes the content in a proper format which is suitable for the target mobile devices. For real-time *content generation*, the multimedia content manager can communicate to a mobile device and acquire the captured video stream if available in real-time. The video content is then passed to the delivery manager that makes the video accessible for the recipients' mobile devices. For *delivery control*, we provide dynamically generated access information to mobile users. For scalability, we need to consider both the number of users and the variety of the devices that the system needs to support. In other words, we must be able to dynamically adjust the number of content managers and the delivery servers to uphold an agreed quality of service.

### 3. IMPLEMENTATION

As we described earlier, we use the EMN platform for authentication and profile matching, the alert management platform to handle the business logic of alert posting, the multimedia content manager for media storage, control, and processing, and the delivery manager for final content dissemination to authenticated users.

Based on the implementation techniques of EMN and AMP, the content manager, delivery manager, and MMilet are also implemented in JAVA technology [17].

MMilet provides the web interface for users to send multimedia related requests after passing EMN authentication process. Because the browser on a PDA tends to have no support for uploading, we provide a server on a PDA device to support uploading video to the content manager. Thus in the current implementation, MMilet provides users a web page

which has two basic functions: one is for upload; one is for displaying the alert video.

For the user who wants to post an alert, he uses the upload function and provides the device's IP address and the port number for the content manager to communicate to the server on the PDA for uploading the video. Once the alert video is successfully uploaded to the content manager, MMilet will direct the user to a web page for posting a multimedia alert. The user can then follow the link to post the alert with certain emergency priority. The content manager associate the alert with the media file in order to provide the proper video streams later.

For the user who wants to view the alert video, he follows the displaying function to send the request to the content manager through MMilet. According to the device profile, the content manager selects the proper stream and performs transcoding if necessary. It then generates a random file name for the alert stream and pushes it to the delivery manager. Obtaining the device information for the content manager is important for generating correct access information. For example, streaming support for PocketPC devices is different from that of Palm OS devices. Incompatible streaming support across different operating systems is an issue today and therefore we also provide download options for those devices that can only play video files locally.

MMilet can construct the URL based on the file name provided by the content manager and make the URL available to the users. With the URL, the user can obtain the video stream directly from the delivery manager. In addition to the randomly generated file name embedded in the URL, the delivery manager also removes the file after certain interval. These can prevent the abusive and unwanted access of the alert video stream.

To keep our system secure, we use a *reverse proxy server*. All the requests from mobile users go through the proxy first. The proxy server redirects the requests to MMilet which is inside the firewall

A Microsoft Windows Server 2003 machine was used for the delivery manager. As a result, streaming support for Media 9 [2] was tested. With current setup, we only experiment with PocketPC 2003 devices for streaming video. We plan to experiment with other streaming protocols, such as QuickTime [3] and RealNetworks [4] later on.

For mobile media capturing, a Toshiba e740[5] was used with a compact flash (CF) camera. For uploading, we wrote a server on the PDA to help transfer the real-time video to the content manager. All the PDAs connect to the servers through a 802.11b wireless network which is outside the corporate firewall.

#### 4. DISCUSSIONS

Although our goal is to provide a general multimedia service platform for interactive users and peer-to-peer applications, we have identified some system limitations.

PocketPC devices were used in our experiments with the current implementation. There is no streaming support for other PDA devices, such as Palm OS devices. The framework allows the content manager to control and provision additional delivery managers for streaming video to different kinds of devices.

Besides the support of various devices, we would also like to further investigate issues about security and scalability,

*Security:* There are many digital rights management (DRM) providers that offer content providers and retailers a platform for the secure distribution of digital media files. Ongoing work is being addressed to investigate how to interface EMN with such a third party DRM provider.

*Scalability:* EMN supports a scalable and open architecture. The number of EMN servers and gateways can be scaled automatically depending on the sustained load of the system. The delivery manager can also support multiple delivery servers. Ongoing performance measurement and QoS (Quality of Service) work is being addressed to define how to dynamically adjust the number of servers or gateways to meet fluctuating demands.

#### 5. RELATED WORK

In our previous paper [6], we highlighted the contributions of other researchers that have recognized the importance of integrating multimedia services with a service platform. The service platform in that paper was based on a proxy server which had difficulty accommodating a large number of user requests in real time, each user demanding a different encoded version of the same media. The service platform discussed in this paper is a new design of its predecessor to address scalability and other research challenges.

Recently, multimedia messaging service (MMS), a communication technology developed by 3GPP [7], is becoming a new global messaging standard that allows users to exchange multimedia content among capable mobile phones and other devices. Jun et al presented an interactive multimedia messaging service platform in [8]. They designed an MMS server as a middleware engine to allow interactions between senders and receivers. The design is fundamentally bound to the MMS standard. It would be difficult for users to access the service with non-MMS-capable devices. Our

approach is to build a service platform that allows multimedia content delivery to different mobile devices and not necessarily just MMS-capable devices.

Along with the Microsoft Media Series 9 server, Microsoft has provided a Digital Rights Management solution [10] and Windows Media-on-Demand Producer [11]. While we currently use our own content manager, we do not exclude the possibility of using a third-party content management solution. On the other hand, currently, the Microsoft solution focuses on transcoding most media formats to the Windows media format, while our platform does not exclude other transcoding possibilities.

Microsoft recently announced .NET alerts - a message and notification service that can be used to deliver customer communication to desktops, cellular phones and personal digital assistants [12]. This solution is more powerful in the Intranet since it makes use of Microsoft related products deployed within an enterprise for example, Microsoft Active Directory, SQL Server notification services and .NET Framework v1.1. Individuals use the service to subscribe to certain types of notifications and applications use the service to send notifications to all subscribed users. This service is mainly used for pushing outbound messages with media content. It supports both on demand and broadcast modes of distribution.

#### 6. CONCLUSION AND FUTURE WORK

In this paper, we described research efforts in providing a multimedia alerting and notification service for mobile users. We enhanced the EMN middleware platform with a content and delivery manager to collect multimedia content and handle user requests. By dynamically generating the access point at the delivery manager, we allow mobile devices to retrieve multimedia content directly from the streaming server securely. A prototype system was developed with PocketPC devices for real-time video acquisition, transcoding and delivery to end devices. Our system supports access from both inside and outside the firewall. Our approach also has the flexibility of adding support for new protocols and devices.

For future work, we would like to utilize sensor information and image processing techniques to allow alerts to be triggered automatically. We would also like to extend the content manager with video indexing, searching, and querying capabilities with technologies such as those found in eClips[13]. Finally, we would like to extend the platform with Quality of Service technologies [14] under development in our research team to prioritize and guarantee service delivery for critical alerts to important recipients.

## 7. ACKNOWLEDGEMENTS

David Gibbon gave us many valuable comments on the draft of the paper. The authors would like to thank people of the whole team working on EMN and AMP/ADE projects, David Belanger and Phong Vo for their management support in this study.

## REFERENCES

- [1] Y.Chen, H. Huang, R. Jana, T. Jim, S. Jora, R. Muthumanickam, and B. Wei, "iMobile EE – An Enterprise Mobile Service Platform", *Wireless Networks*, Vol. 9, No. 4, pp. 283-297, July 2003.
- [2] Microsoft, <http://www.microsoft.com/windows/windowsmedia/>, Windows Media 9 Series.
- [3] Apple, <http://developer.apple.com/darwin/projects/streaming/>, Darwin Streaming Server.
- [4] RealNetworks, <http://www.real.com/>, Real.
- [5] Toshiba, [http://www.csd.toshiba.com/cgi-bin/tais/pc/pc\\_prodList.jsp?comm=CS&plin=Toshiba%20Pocket%20PC](http://www.csd.toshiba.com/cgi-bin/tais/pc/pc_prodList.jsp?comm=CS&plin=Toshiba%20Pocket%20PC), Pocket PC devices.
- [6] Y. Chen, H. Huang, R. Jana, S. John, S. Jora, A. Reibman, and B. Wei, "Personalized Multimedia Services Using a Mobile Service Platform", *IEEE Wireless Communication and Networking Conference*, Orlando, FL, March 2002.
- [7] 3GPP, <http://www.3gpp.org>.
- [8] S. Jun, Y. Rong, S. Pei, and S. Song, "Interactive Multimedia Messaging Service Platform", *Proc. ACM Multimedia*. pp. 464-465, Berkeley, CA, November 2003.
- [9] Eric Manning, "The Utility Model for Multimedia Servers and Networks: A Retrospective, *Keynote Address, ICCIT*, December 2002.
- [10] Microsoft 9 Series Digital Rights Management, <http://www.microsoft.com/windows/windowsmedia/9series/drm.aspx>.
- [11] Microsoft Windows Media-on-Demand Producer, <http://www.microsoft.com/technet/prodtechnol/netshow/downloads/wmodp.msp>.
- [12] Microsoft .NET alerts v6.0 - <http://msdn.microsoft.com/msdnmag/issues/03/12/XMLFiles/default.aspx>
- [13] D. Gibbon, L. Begeja, Z. Liu, B. Renger, and B. Shahraray, "Creating Personalized Video Presentations using Multimodal Processing," **Handbook of Multimedia Databases**, Edited by Borko Furht, CRC Press, pp. 1107-1131, June 2003, ISBN 0-8493-7006-X.
- [14] Jun He, Matti A. Hiltunen, and Richard D. Schlichting, [Customizable QoS for Mobile Service Platforms](#), [The International Conference on Dependable Systems and Networks \(DSN 2004\)](#), Florence, Italy, Jun 2004. (to appear).
- [15] S. Jora, R. Jana, Y. Chen, M. Hiltunen, T. Jim, H. Huang, A. Singh, R. Muthu, "An alerting and notification service on the AT&T Enterprise Messaging Network", AT&T Labs Research Technical Report, TD-5TXMG4, 5<sup>th</sup> Dec 2003.
- [16] Han Chen, Kai Li, and Bin Wei, "Memory Performance Optimizations For Real-Time Software HDTV Decoding", *Journal of VLSI Signal Processing, Special Issue on Media and Communication Applications on General Purpose Processors: Hardware and Software Issues*, 2004.
- [17] Sun, <http://java.sun.com>, JAVA Technology.