

Mobile Push: Delivering Content to Mobile Users*

Ivana Podnar

Faculty of EE and Computing
University of Zagreb
Unska 3, Zagreb, Croatia
ivana.podnar@fer.hr

Manfred Hauswirth

Distributed Information Systems Lab
EPFL
CH-1015 Lausanne, Switzerland
Manfred.Hauswirth@epfl.ch

Mehdi Jazayeri

Distributed Systems Group
Technical University of Vienna
Argentinierstr. 8/184-1, Vienna, Austria
M.Jazayeri@infosys.tuwien.ac.at

Abstract

The increasing popularity of information services that rely on content delivery in mobile environments motivates the need for a mobile push service—an efficient and flexible content dissemination service that targets mobile users. We analyze the features of a mobile push service by investigating representative usage scenarios and propose an architecture for mobile content delivery systems. The architecture is based on the publish/subscribe (P/S) paradigm which supports many-to-many interaction of loosely-coupled entities. We define the set of services that need to collaborate with the P/S infrastructure to address the dynamics of mobile environments.

1. Introduction

Recent technology advancements have transformed the Internet into a dynamic environment that supports the development of novel information services targeting mobile users. Content dissemination to mobile users has recently attracted particular attention [4, 10]. Examples of applications that rely on content delivery are notification services for weather or traffic reports, messaging systems for group discussions, or systems supporting the collaboration of mobile employees. Location-based content delivery will be a premier feature in these systems [7]. Additionally, wide acceptance will depend on the delivery of highly personalized and customized content.

In this paper we investigate the features of a mobile push system. We build upon previous work in the Minstrel push system [8]. Minstrel enables P/S style interaction between publishers who announce content and subscribers who have declared their interest in particular content types. The current Minstrel implementation only supports stationary users and a natural further step is to enhance it for mobile users.

*This work was supported in part by the European Commission under contract IST-1999-10288 (OPELIX).

A number of authors argue that P/S is advantageous in mobile environments where devices may be frequently unavailable or disconnected [5, 10]. Publishers and subscribers are decoupled; they interact asynchronously and need not be active simultaneously to exchange messages. The P/S infrastructure is in charge of queuing and delivering the content. Mobility, however, introduces additional requirements: The routing problem is more complex [5]; the system needs to be resilient to frequent disconnections and handle duplicate messages [9].

The problem of routing and message loss is pertinent to the P/S infrastructure in general. Improved routing algorithms and better quality of service will only partly solve these problems. We argue that the P/S infrastructure must address mobility issues explicitly and that it needs to collaborate with supporting services to cope with mobility problems.

Our goal is to clarify the design requirements for content dissemination that are introduced by mobility. We analyze and compare representative usage scenarios that support personal mobility and propose an architecture for mobile push systems based on this analysis. In Section 2 we define the basic principles of content dissemination. Section 3 investigates usage scenarios that have guided the design of the architecture which we describe in Section 4. Section 5 discusses the related work and we present our conclusions in Section 6.

2. Content dissemination

A content dissemination service enables delivery of information from information sources to numerous users across a wide area network. Some authors use the term *push service* to denote that the content is actively pushed to subscribers, as opposed to the user-initiated pull model. The main service task is the timely delivery of possibly large amounts of information to many subscribers.

The service involves two types of entities: *publishers*, and *subscribers*. Publishers are content sources that group

and send data through channels. Subscribers are content destinations that subscribe to a channel and receive the corresponding data. A channel is a logical connector between a publisher and a subscriber. A single channel provides topic-based connections between a number of publishers and subscribers, and offers a coarse level of content classification.

A set of *content dispatchers* (CD) composes the service infrastructure and is responsible for managing channels and sending the content along channels. We assume that the network of CDs is stationary whereas publishers and subscribers are mobile entities: they change their attachment points in the network and can connect to different CDs.

Due to the large volume of data and the dynamic nature of publishers and subscribers, efficient content routing in the network of CDs is of major importance. When a publisher submits the data to a CD to be published on a defined channel, the network of CDs needs to route the data to all channel subscribers. One approach is to employ IP multicast [12], but only a limited number of users have access to a multicast network. Another approach is to use point-to-point communication at the network layer and an application-layer network of servers for content routing as is done in *Minstrel* [8].

Minstrel uses a two-phase dissemination approach to address scalability: In phase 1 (“advertising”) the system distributes announcements to advertise content. If the announcement is interesting, a subscriber may request the delivery of the actual content in phase 2 (“delivery”). The *advertising phase* resembles the functionality of notification systems such as SIENA [3], or ELVIN [13], which offer an expressive subscription language for content-based filtering of published events. Minstrel can employ this approach and use content filters to achieve further granularity of channel content.

In the *delivery phase* a user requests the “actual” content. This phase can potentially consume high bandwidth since the user may request a large data item. Thus, Minstrel uses a special protocol for data replication and caching to minimize the network traffic.

3. Mobile push scenarios

This section presents representative usage scenarios that illustrate the issues to be addressed by a mobile push architecture. In the first scenario a user has a stationary host with a fixed IP address. The second scenario enables nomadic users to access the service from different networks using desktop or portable computers via dial-up modem lines or (wireless) LANs. In the third scenario the user can apply various devices and access points to use the service. The difference between nomadic and mobile users is that *nomadic users* connect to the network from arbitrary and changing locations, but do not use the service while moving,

whereas *mobile users* can use the service during movement.

We use the following underlying scenario: Alice lives in the suburbs of Vienna and commutes each day to her downtown office. She uses the traffic notification service which informs her about the current traffic situation.

3.1. Stationary users

Alice accesses the service from her office desktop computer on a LAN. Before leaving the office, she checks the list of received traffic reports for the current traffic situation. If she needs additional information, she can request a detailed map of the particular area with approximate waiting times for the traffic jam areas.

In this scenario Alice is a stationary subscriber to the channel “Vienna traffic.” Whenever traffic problems in the area of Vienna are reported, the service initiates the delivery of the traffic report to all traffic channel subscribers.

The CD hosting the “Vienna traffic” publisher initiates connections to the server running on Alice’s host. For this purpose the CD must know whether Alice is online. The assumption is that she uses a host with a permanent IP address. In case she cannot be contacted, we need a *content queuing strategy* for undelivered reports. Furthermore, the push service stores and manages subscriptions and enables publishers to define their channels and store content using *subscription* and *content management* services.

We have identified personalization as one of the major requirements of a dissemination service. For example, Alice might define several routes between her home and office. In this case the push service would filter the messages for the “Vienna traffic channel” and deliver only those that match her personal routes. Clearly, *content-based filtering* is needed to provide such a personalized service and Alice must also be able to express her preferences via the *user profile service* as a set of rules/filters.

3.2. Nomadic users

In the previous scenario Alice was bound to one location. Naturally, she wants to use this service at home before driving to the office. At home she connects her laptop to the Internet via dialup and thus becomes a nomadic push service user. The service must address the change of the IP address of her host. The publisher’s CD must know the current IP address of all subscriber hosts because if the content is sent to an invalid IP address it might reach the wrong subscriber or the CD might assume that a subscriber is offline.

A nomadic user can frequently change its location in the network, and therefore, the host IP address will change as well. The same problem arises if a network (LAN, PPP) is configured using the Dynamic Host Configuration Protocol (DHCP).

Figure 1 shows a scenario with a CD and a publisher residing on a home network that is dynamically configured. Subscribers can use the LAN in the home network or move to a foreign network and connect to the Internet via wireless LAN. A subscriber can also use the service from home via dial-up. By changing its attachment point, the host IP address will change accordingly. To track this change a *location management service* must map a unique subscriber identifier to the current IP address of the subscriber's host.

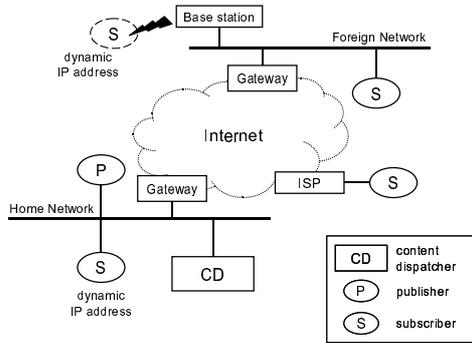


Figure 1: The nomadic user scenario

3.3. Mobile users

In this scenario Alice would like to use the traffic notification service while in motion. Figure 2 depicts an environment where she can use both a PDA and a mobile phone to receive traffic reports. She can use a PDA with wireless LAN connectivity while within the reach of a wireless LAN base station or her mobile phone during outdoor activities.

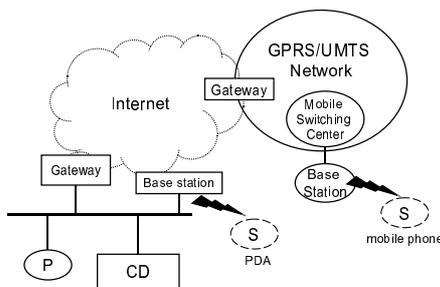


Figure 2: The mobile user scenario

In this setup we need a *location management service* that will map a user to the identifier of the currently used device. This is a one-to-many mapping: A user might register a number of devices, e.g., a mobile phone, a PDA, a desktop, and a laptop computer.

Service personalization is vital in this scenario because a user must be able to define his/her preferences according to the currently used end device.

Due to the variations in network and end-device, *content adaptation and presentation* are essential in this scenario. The content is delivered through various networks that differ in the available bandwidth, and it is displayed on devices with different computational capabilities and screen sizes. For example, Alice can receive high quality maps only on a computer with a high bandwidth connection. When driving home from the office she can re-check the text reports about the changing traffic conditions on her mobile phone.

Table 1 summarizes the required services for each of the described usage scenarios. It is desirable to design and implement generic services that can accommodate the requirements of different scenarios and therefore be applied in all scenarios.

	Stationary	Nomadic	Mobile
subscription management	+	+	+
content management	+	+	+
user profiles	+	+	+
queuing strategy	+	+	+
location management	-	+	+
content adaptation	-	-	+
content presentation	-	-	+

Table 1: Services for stationary, nomadic and mobile users

4. Architecture

Figure 3 depicts the proposed architecture for mobile push systems. It consists of a set of components that provide the features listed in Table 1.

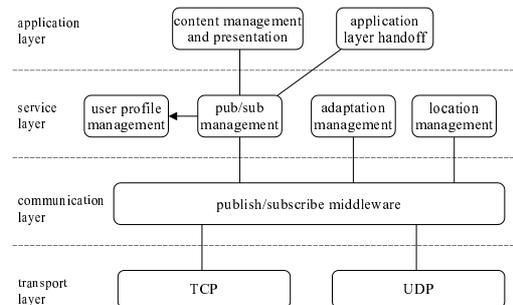


Figure 3: Mobile push architecture

The components are logically divided into the following layers:

- The *communication layer* enables the interaction between publishers and subscribers. It employs the P/S middleware.
- The *service layer* contains the utility services needed by mobile push such as location, adaptation, user profile, and subscription management.

- The *application layer* is the push-specific layer that coordinates other services, manages and stores the device-dependent content, and manages the transfer of information between CDs.

The sequence diagram in Figure 4 shows the interaction between these components for two representative use cases: publish (a publisher releases content to a channel) and subscribe (a subscriber subscribes to the channel). The subscriber sends the request from his/her end device to the P/S management component which resides on a CD. The P/S management component submits the subscribe request together with the user profile to the P/S middleware so that it is routed to the responsible publisher. The user profile contains the unique user identifier and the set of filters for content-based subscription.

To publish content, the publisher defines it via the content and presentation component. In Figure 4 we assume that the content is already defined and that it matches the subscriber's end device. The publisher sends a publish request to the P/S management which submits it to the P/S middleware. The P/S middleware publishes the content to the P/S management component which tries to submit the content to the user's device. In our scenario a user has changed his/her location and thus the P/S management queries the location management component for the current location. Then it performs its internal handoff procedure: the subscriber's queued content is transferred from the old CD to the new one that is now responsible for the subscriber. The new CD will send the queued content to the subscriber and update the subscription data in the P/S middleware component. After receiving a notification, a user decides to request more information using the received URL and enters the delivery phase.

4.1. Communication layer

P/S is the base interaction style between content publishers and subscribers. It can also serve as a distribution channel for location management events and environment-related events that guide service adaptation.

The P/S middleware component enables the *subject-based subscription* policy to support channels and possibly offers *content-based filtering* for further content granularity. It has a *distributed architecture* to address scalability and implements a *routing algorithm* that supports user mobility. The design of an efficient routing algorithm in the mobile setting is still an open research problem [5].

4.2. Service layer

The *P/S management* component is a mediator between the application layer services and the P/S middleware. It manages subscriptions and advertisements. Subscriptions

consist of a unique subscriber identifier and a list of subscribed channels. Advertisements contain a publisher identifier and a list of channels on which it delivers content.

The P/S management is also distributed. It implements a *flexible queuing policy*, and can be thought of as a subscriber's proxy that will deliver notifications to his/her device, or queue them until the subscriber reconnects. The simplest queuing strategy is to drop all content for unreachable subscribers. A more complex one would store undelivered content for later attempts and enable a subscriber to define properties such as priorities and expiry dates for each channel.

The *location management* component is responsible for locating the currently active user terminal. It supports a one-to-many mapping of a unique user identifier to a number of end devices. It could also be extended to track and store the user's geographical position. It should have a distributed architecture to scale well and support multiple name spaces (e.g., telephone numbers and IP addresses). A user could update the host information each time he/she starts to use it and to provide his/her credentials with a time-to-live period for the current connection.

It is possible to implement the mobile push service without a location management component. The P/S management would then be responsible for (un)subscribing to/from the P/S component each time a user changes the access point. This solution would increase the network traffic and would not scale for the mobile user scenario in which a user frequently changes the location, but it seems considerably simpler because we do not have to design a special location service. However, if we assume that an adequate location service is available, it would free the P/S management from the burden of tracking the user location.

Domain Name System (DNS), mobile IP, and mobility management in telecommunication networks are at the moment standard naming and location services. DNS is a distributed database that maps host names to IP addresses, but it cannot handle multiple name spaces and it does not scale well if name-to-address mappings change frequently. Mobile IP is a network layer protocol that allows a single host to move across the Internet. Mobility management in mobile telecommunication networks enables a network to locate a device for call delivery and maintain a connection despite the changes of the access point. This solution supports only one-to-one user-device mapping and does not support the multiple devices scenario. Therefore, further work is needed to design an appropriate location service for mobile push.

User profile management stores and manages user profiles and enables a subscriber to define rules/filters to customize the service. A subscriber can decide what subscriptions would apply to a particular end-device, current location, or time of day. Content can thus be queued for later

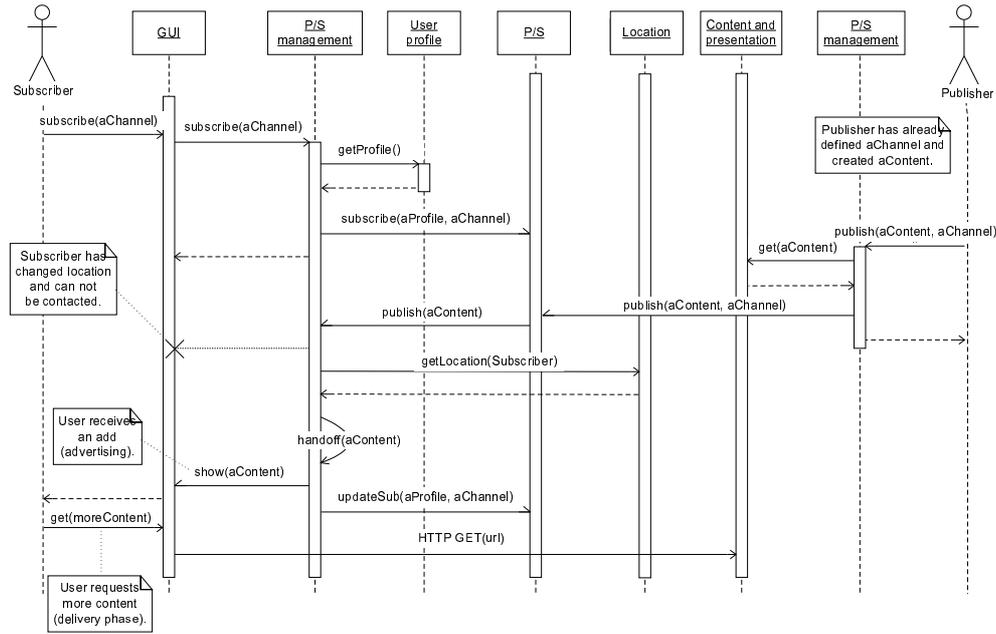


Figure 4: Sequence diagram for publish and subscribe use cases

delivery to a suitable device according to user preferences. The open problems are related to security and privacy: will the profile be stored on user devices, or will a CD store a copy, and who can access and change a user profile.

Content adaptation deals with the problem of client and network variability in mobile environments. Data compression and data conversion are standard techniques for client and network variability adaptation. For example, an image must be transformed into a new format to be displayed on a mobile phone, or a smaller and lower quality image is sent over a low-bandwidth connection. Dynamic adaptation can be used for mobile push: the system monitors the environment, and acts upon changes, such as low bandwidth, or battery consumption. The P/S middleware can be used for distributing events about environment changes.

4.3. Application layer

The *content management and presentation* component enables a publisher to create and manage device-dependent content which will be published on different channels. The publisher needs to adjust the content format to end devices to suit different display sizes and to deal with input limitations. Currently, XML and related technologies are used to create and manage flexible user interfaces [11]. The presentation-related problems, such as content structuring and partitioning, and simple input techniques are still open research topics.

The *application-layer handoff* component controls the

transfer of content between CDs. The content consists of large data items that are requested in the delivery phase of the mobile push scenario. We can adapt the existing Minstrel protocol [8] for data replication and caching to distribute the content in the mobile setting with minimal traffic and response times.

5. Related work

To our knowledge, an architecture or an implementation of a content dissemination system supporting both nomadic and mobile users has not been published yet. ELVIN [13] is the only notification system that implements limited support for mobile users. The proposed solution puts a proxy server between the ELVIN server and a mobile device to queue messages for non-active users. The presented solution implements a queuing strategy with time-to-live expiry, but it is not clear how location management and distribution are handled.

Reference [9] discusses the operation of mobile P/S systems and analyzes the adaptation of a centralized and a distributed architecture to mobility. The presented ideas and identified problems have motivated our work. However, we analyze the system from the user's perspective as opposed to their system-centric view and present an initial system architecture.

CEA [2] and JEDI [6] are P/S middleware systems that offer mobility support. CEA uses a mediator which receives

notifications on behalf of a subscriber during disconnections. The mediator can register interest in a subscriber's location, get a notification when it reconnects, and then deliver the queued messages to the new location. JEDI [6] offers two operations: `moveIn`, and `moveOut`. A subscriber uses `moveOut` to disconnect from a CD and `moveIn` to reconnect to a new CD. The old CD stores events on behalf of the subscriber during the disconnection and transmits them to the new CD upon reconnection.

Both CEA and JEDI solve the queuing problem, but the routing problem remains open. Reference [5] proposes a solution with a dynamic dispatching tree that has a leader responsible for subscribers with the same subscription. This solution requires a complex protocol and further study is needed to evaluate it.

In the area of telecommunications the problems related to mobility are well-understood. However, wireless networks offer limited possibilities for the design, implementation, and deployment of services. The third generation partnership project (3GPP) [1] is trying to solve this problem by offering an API specification to enable third parties to design and deploy services using the network provider infrastructure. 3GPP has recently specified the functional capabilities of the Multimedia Messaging Service (MMS), the successor of the Short Message Service (SMS). MMS will provide non-realtime multimedia messaging in 3G networks. The MMS specification focuses on one-to-one usage scenarios, but considers one-to-many communication scenarios that will offer functionality similar to mobile push.

6. Conclusion

Content dissemination is an increasingly popular service in environments that support user mobility. In this paper we have analyzed and compared representative scenarios and proposed an initial architecture for a mobile push system.

The P/S interaction scheme is the basis for our architecture. It is a well-established solution for the asynchronous interaction between frequently unavailable devices. However, the existing P/S solutions must be extended to be applicable in mobile environments. Location management and efficient routing are among the significant open research issues. Solutions to message queuing and the problem of duplicate messages have already been addressed. We have identified content presentation and adaptation as vital components that deal with client diversity and network variability. There are existing solutions for both presentation and adaptation. What remains a challenge is the question of how to integrate such solutions into a mobile push system. User profile management and content-based filtering offer means for service customization and personalization. Open issues are related to security because user profiles store sensitive personal data.

Our future work will be directed to the design and implementation of a complete system for mobile push. We plan to focus on designing a solution for location management and routing and to use a flexible integration technique for system components.

7. References

- [1] 3GPP website. The Third Generation Partnership Project, 2001. <http://www.3gpp.org/>.
- [2] J. Bacon, et.al. Generic Support for Distributed Applications. *IEEE Computer*, **33**(3):68–76, March 2000.
- [3] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and Evaluation of a Wide-Area Event Notification Service. *ACM Trans. on Computer Systems*, **19**(3):332–383, August 2001.
- [4] A. Carzaniga and A. L. Wolf. Content-based Networking: A New Communication Infrastructure. *NSF Workshop on an Infrastructure for Mobile and Wireless Systems*, (Scottsdale, AZ.), October 2001.
- [5] G. Cugola and E. Di Nitto. Using a publish/subscribe middleware to support mobile computing. *Advanced Topic Workshop on Middleware for Mobile Computing*, (Heidelberg, Germany), November 2001.
- [6] G. Cugola, E. Di Nitto, and A. Fuggetta. The JEDI Event-Based Infrastructure and Its Application to the Development of the OPSS WFMS. *IEEE Trans. on Software Eng.*, **27**(9):827–850, September 2001.
- [7] A. Dornan. Can M-Commerce Find a Place in Your Network? *Network Magazine*, November 2001.
- [8] M. Hauswirth. *Internet-Scale Push Systems for Information Distribution—Architecture, Components, and Communication*. PhD thesis. Distributed Systems Group, Technical University of Vienna, October 1999.
- [9] Y. Huang and H. Garcia-Molina. Publish/Subscribe in a Mobile Environment. *Proc. 2nd ACM Int. Workshop on Data Eng. for Wireless and Mobile Access (MobiDE'01)*, pages 27–34, (Santa Barbara, Calif.), May 2001.
- [10] H.-A. Jacobsen. Middleware Services for Selective and Location-based Information Dissemination in Mobile Wireless Networks. *Advanced Topic Workshop on Middleware for Mobile Computing*, (Heidelberg, Germany), Nov. 2001.
- [11] E. Kirda, C. Kerer, and M. Jazayeri. Supporting Multidevice Enabled Web Services: Challenges and Open Problems. *Proc. 10th IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, IEEE CS Press, Los Alamitos, Calif., pages 49–54, (Boston, MA.), June 2001.
- [12] T. Liao. Global Information Broadcast: An Architecture for Internet Push Channels. *IEEE Internet Computing*, **4**(4):16–25, July/August 2000.
- [13] P. Sutton, R. Arkins, and B. Segall. Supporting Disconnect-ness - Transparent Information Delivery for Mobile and Invisible Computing. *Proc. IEEE Int. Symposium on Cluster Computing and the Grid*, IEEE CS Press, Los Alamitos, Calif., pages. 277–285, (Brisbane, Australia), May 2001.