

High performance publish/subscribe middleware for mobile wireless networks

Umar Farooq*, Shikharesh Majumdar and Eric W. Parsons

Department of Systems and Computer Engineering, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada, K1S 5B6

Abstract. Decoupling, flexible, scalable and asynchronous nature of publish/subscribe paradigms makes them a good choice for mobile wireless networks. However, our research shows that current implementations of publish/subscribe systems as well as the traditional solutions for extending publish/subscribe systems to the mobile domain do not perform well in highly mobile and unreliable wireless settings. We present semi-durable subscriptions, a technique that we have developed to overcome the challenges and the performance concerns publish/subscribe systems face in mobile wireless settings. We discuss the architecture of semi-durable subscriptions and study in detail the effect of various mobility parameters on the performance of semi-durable subscriptions to demonstrate their efficacy in mobile wireless settings. We also discuss system parameters for semi-durable subscriptions and with the help of experimental results demonstrate how these parameters can be controlled to configure a system according to a set of desired characteristics.

Keywords: Publish/subscribe systems, mobile computing, middleware for mobile wireless networks, mobile cellular networks, message oriented middleware

1. Introduction

There has been an ever increasing demand from consumers to be able to access all the services accessible from desktops through devices such as cell phones and portable digital assistants (PDAs) without any interruptions and performance degradation. However, user mobility and weakly connected wireless environments make it challenging for the developers to provide services seamlessly to the mobile user. Under these conditions, the concept of middleware, which is well established in fixed networks (herein called *fixed domain*) becomes even more important in the context of mobile networks (herein called *mobile domain*). Middleware in a mobile domain is not meant to deal with the prevalent heterogeneous infrastructures only. In order to ensure seamless provisioning of services to the users the middleware in the mobile domain needs to provide some services additional to those it provides in fixed networks e.g. mobility management, temporary disconnections management, QoS assurance over wireless channels and location dependent services provision.

While middleware in the fixed domain can place high computation loads on the clients' resource rich devices, use synchronous communication in permanent connection environments and can exploit static context to achieve performance, a middleware in the mobile domain deals with limited computing capacity devices which frequently get disconnected from the network and access network from various attachment points. Hence the traditional middleware technologies designed for fixed networks cannot

*Corresponding author. Tel.: +1 613 520 5740; Fax: +1 613 520 5727; E-mail: ufarooq@sce.carleton.ca.

provide the desired behavior in mobile wireless settings. Most popular fixed domain middleware such as CORBA [22] and COM [25] and mechanisms such as Java RMI [24] face challenges such as software size and protocol suitability [4]. For example, they rely mostly on remote method invocations, a synchronous paradigm that couples the two parties in space and time during the whole length of communication. In unreliable mobile settings such coupling cannot be ensured as the mobile client may get temporarily disconnected from the network due to poor wireless channels or it may move from one part of the network to the other. Relying on solutions such as Mobile IP to manage mobility cannot support location and context awareness.

Event based middleware on the other hand use asynchronous communication and can support a publish/subscribe paradigm. A publish/subscribe paradigm consists of *publishers* that are the sources of information called *events* or *messages*, and *subscribers* that show their interest in receiving certain types of events or messages by submitting their *subscriptions* to the system. Whenever a message is published, the middleware delivers it to all the subscribers whose subscription criterion matches the information in the message. The publish/subscribe paradigm is different from the traditional point-to-point communication models in a number of ways. In publish/subscribe systems communication between the end points is anonymous, asynchronous and loosely coupled. In other words, publish/subscribe systems decouple publishers and subscribers in space, time and flow [11]. Decoupling in space allows the subscriber to move from one location to another without *informing* the publisher while decoupling in time allows for disconnected operations of the subscriber. This decoupling in space and time thus makes publish/subscribe systems a good choice for mobile and weakly connected environments as offered by mobile wireless networks. Moreover, it makes publish/subscribe middleware highly scalable by removing all explicit dependencies between the interacting parties. This has been observed in several other studies [2,5,7].

The basic model of a publish/subscribe system is based on a neutral mediator, which is responsible for an efficient and reliable delivery of all the events to interested subscribers. This mediator, also called *message broker* or *broker*, needs to perform a number of tasks such as subscription management, event filtering, event dissemination as well as event routing and processing. In order to provide efficient and reliable services to the client in a mobile environment the broker needs to handle user mobility, manage temporary disconnections and deal with limited bandwidth environments. These additional tasks can put a substantial load on the broker in a dynamic mobile environment with frequent disconnections and connections. The broker may thus become the performance bottleneck limiting the scalability of the system. Broker performance is hence a serious concern in mobile wireless environments and techniques that not only effectively handle mobile wireless clients but also ensure high performance warrant investigation.

Our research is focused on the performance aspects of publish/subscribe extensions to mobile domain. The goal of our research is to extend publish/subscribe systems to a mobile domain in a way that can ensure high performance under high frequency of handoffs and error prone wireless channels. In an earlier paper [15], we have described the performance of a popular implementation of publish/subscribe system for mobile wireless settings. We have also designed and evaluated middleware level handoffs, a well known solution to extend publish/subscribe systems to a mobile domain, and have identified the performance concerns of such solutions. In another paper [13], we have given a high-level description of Semi-Durable Subscriptions, a technique that we have developed to overcome the challenges and the performance concerns publish/subscribe systems face in mobile wireless settings. The contributions of this paper are presented.

- The paper presents a detailed architecture of semi-durable subscriptions.

- The paper studies in detail the effect of various mobility parameters on the performance of semi-durable subscriptions to demonstrate their efficacy in mobile wireless settings.
- The paper discusses system parameters for semi-durable subscriptions and analyzes the effects of parameters such as buffer size and message overwriting policies on their performance.
- With the help of experimental results the paper demonstrates how the parameters in semi-durable subscriptions can be controlled to configure a system according to a set of desired characteristics.

The rest of the paper is organized as follows. Section 2 discusses the background material, provides an overview of publish/subscribe systems in mobile wireless networks and presents the current state of the art in the field of mobile publish/subscribe systems. We discuss the experimental setup used in this research in Section 3. Performance of current implementations of publish/subscribe systems and the traditional solutions to extend them to the mobile domain are presented in Section 4. We present semi-durable subscriptions architecture and compare the performance of semi-durable subscriptions with that of other subscription mechanisms in Section 5. Section 6 studies the effect of mobility and workload parameters, such as frequency of handoffs and message size, on system performance. In Section 7, we study the impact of system parameters, such as buffer size and message overwriting policies, on the performance of semi-durable subscriptions and demonstrate how system parameters can be controlled in semi-durable subscriptions to configure a system according to the needs of the environment. Finally, we conclude the paper and discuss directions for future research in Section 8.

2. Publish/subscribe systems in mobile wireless networks

A publish/subscribe system may have a centralized or a distributed architecture [7]. A centralized architecture has only one message broker. All publishers publish their events to this broker while all subscribers subscribe to this broker for events they are interested in. In such an architecture the broker represents the single point of failure. Also as the number of subscribers and/or publishers increases the broker may become the performance bottleneck, limiting the scalability of the system. A distributed architecture on the other hand, consists of a number of brokers each having a subset of publishers and subscribers. As in a distributed architecture, all publishers and subscribers are not connected to the same broker, in order to ensure that the message published at one broker is received by all the interested subscribers in the system, the brokers need to be interconnected to each other. This gives rise to design issues such as broker interconnection topology, routing algorithm and message processing strategies. The distributed architecture has been studied in several contexts. A design of a Wide Area Notification Service SIENA is discussed in [7] along with the evaluation of several broker topologies, routing algorithms and message processing strategies. Papers such as [1,6,8,20,23] investigate techniques and algorithms for efficient dissemination of information in distributed publish/subscribe systems.

Publish/subscribe extensions for mobile wireless environments can be divided into two classes. The first class includes extensions for those applications which are based on mobile code technology called mobile agents. Mobile agents may execute on different hosts during their life cycle and hence their mobility needs to be managed if they are accessing a publish/subscribe system. The second class includes those applications which run on mobile hosts such as laptops, cellular phones and PDAs. Although both classes are interesting and challenging, in this paper we focus on the second class and extend publish/subscribe system to mobile wireless users.

Although a lot of work has been done on publish/subscribe systems in the fixed domain, there is comparatively little research on the behavior of these systems in a wireless mobile environment. Mobility

management in publish/subscribe systems has been studied in [9] and [16]. These papers focus more on general requirements dealing with design issues, data definitions for publications and query language for subscriptions. The nature of these works is thus broader and more theoretical than the work this paper focuses on. Huang et al. have discussed how publish/subscribe systems can be extended to operate in a mobile environment, giving a basic model for the deployment of these systems on mobile networks and identifying some of the problems that would arise during such a deployment [19]. Although Huang et al. address the issues in the deployment and evaluation of these systems over wireless networks, they have not given any concrete results on the performance behavior of these systems in such environments.

Data dissemination and management are important in the context of many mobile systems including publish/subscribe systems. Yang et al. [30] present techniques for content-based multimedia retrieval in mobile ad hoc networks. Caching and replication of data have been used in mobile systems to improve system performance and reliability. A replica-based data storage mechanism for providing reliable storage and fast dissemination of data is described in [10]. A survey of query processing and cache management in mobile systems is presented in [28]. Although these papers concern important issues such as data distribution and caching techniques they do not focus explicitly on publish/subscribe systems that this research is concerned with.

Concurrent to our research, there has been research to support mobility with SIENA [3] and REBECA [17,31]. These papers describe techniques to retrieve missed messages whenever a user disconnects from the broker and reconnects to either the same broker or a new one. Their proposed techniques in one form or another involve retrieving messages from the previous broker the user was connected to. These techniques are thus similar to Middleware Handoffs discussed in Section 4 of this paper. Our results presented in Section 4, however show that protocols involving two brokers give poor performance under highly dynamic and unreliable mobile wireless settings.

2.1. Publish/subscribe in mobile cellular networks

Figure 1 shows the distributed architecture for a publish/subscribe system in a mobile cellular environment. A client which can act as a publisher or a subscriber or both runs on a mobile host such as a cellular phone, a PDA or a laptop. It connects to one of the message brokers in the system over the wireless link in the cellular network. The message broker may be located somewhere in the cellular network infrastructure or it may be at a fixed node outside the network in which case the client connects to the broker through the wireless gateway (e.g. PDSN).

As a client moves from one part of the network to the other or from one network to another, it may need to disconnect from one broker and connect to another broker in its new location. During this *handoff*, the client may get temporarily disconnected from the network. Also, due to weak connections in wireless environments, the client may get occasionally disconnected from the network even if it remains in the same cell, thus hindering its ability to deliver/receive messages from the system. When a client acts as a publisher, it may not be able to publish messages during such intervals but as soon as it reconnects to any of the message brokers, it can publish some or all of these messages. However, when a client acts as a subscriber it may lose messages published during the time it was disconnected from the network or as it moves from one message broker to the other. Thus client mobility and weakly connected environments in mobile cellular networks pose an additional problem of occasional message loss, generally absent in fixed networks. Limited bandwidth in wireless networks may also prevent or delay the delivery of messages to the subscriber.

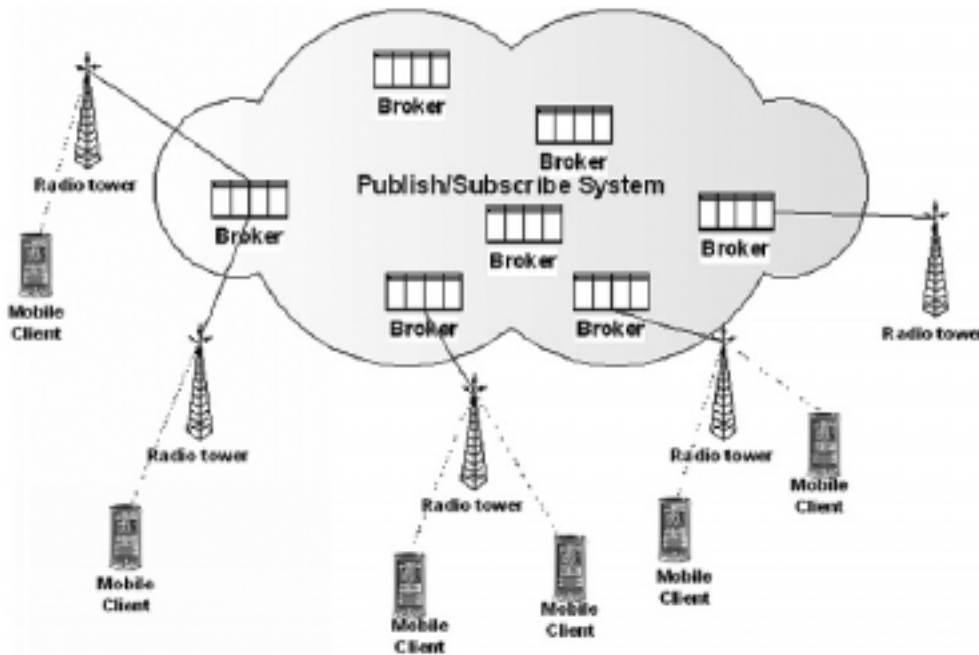


Fig. 1. Publish/subscribe in mobile cellular networks.

2.2. Challenges and performance issues for publish/subscribe systems in mobile wireless environments

As discussed in Section 2.1 publish/subscribe systems face additional challenges in mobile wireless environments that are generally absent in fixed networks. The major challenges are:

- *Managing client mobility:* Clients (both publishers and subscribers) in mobile environments may change their access points to the publish/subscribe system. The system should be able to handle this mobility ensuring that all messages are delivered to all interested subscribers independent of their access points. This will require an implementation of a handoff procedure that takes place whenever a subscriber changes its access point. The handoff procedure should ensure that all active subscriptions of the subscribers are successfully transferred to the new broker and the messages published during the handoff procedure are delivered to the subscriber.
- *Managing temporary disconnections:* Clients in wireless environments may get temporarily disconnected from the network due to poor wireless channels. Publish/subscribe systems should handle these temporary disconnections ensuring that at least no *important* messages are lost by the subscribers. This requires the system to store messages for the disconnected subscribers and deliver them as they reconnect.
- *Managing environments with limited bandwidth:* Clients in wireless environments may have limited bandwidths available to them due to loaded error prone wireless channels. Publish/subscribe systems should efficiently manage such limited bandwidth scenarios. This may require the broker to do message filtering (sending only *important* messages), message simplification (converting a multimedia message to a text message) and message compression for the low bandwidth subscribers.
- *Performance Concerns:* Brokers in publish/subscribe system performs a number of tasks such as subscription management, event filtering, event dissemination as well as messages routing and processing. The additional tasks such as handling mobility, managing temporary disconnections

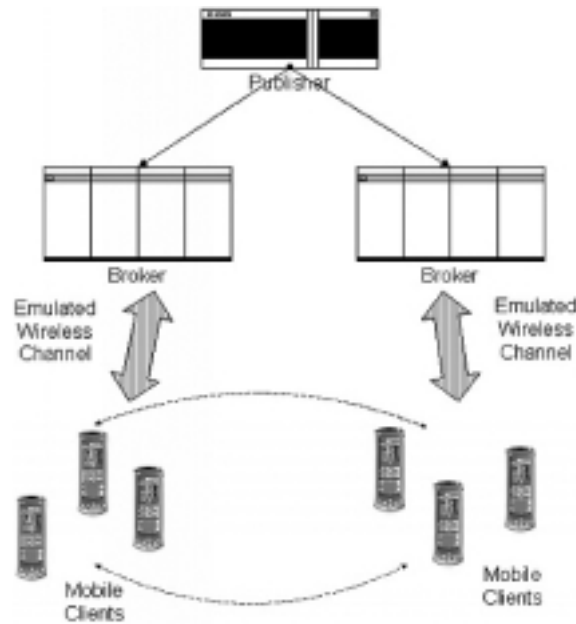


Fig. 2. Logical view of the experimental setup.

and dealing with limited bandwidth environments in mobile wireless environments with frequent disconnections and connections can put substantial load on the broker making it a performance bottleneck.

3. Experimental setup

To study the performance of publish/subscribe systems in cellular networks we have used Java Message Service, JMS [26], as our base publish/subscribe system. JMS developed by Sun and several partner companies, is a Java API that allows applications to create, send, receive and read messages. Figure 2 illustrates our experimental environment consisting of five nodes, one node running a stationary publisher, two message brokers on two nodes while the cellular mobile subscribers running on the remaining two nodes.

Although many open-source and commercial implementations of JMS are available, in our experiments, we have used a widely used commercial solution which is considered a leading JMS implementation in the industry, in terms of its performance, reliability and scalability. In addition to existing software, we have incorporated semi-durable subscriptions technique into the JMS implementation for the mobile wireless environments. During the course of our experiments, two message brokers were run, each on a separate Intel based Pentium 4 machine running RedHat Linux version 7.2. A single publisher running JMS client's library on a separate node publishes messages on a certain topic with different message selector values, to the two brokers. Message selectors allow the subscriber to receive only a subset of messages published on a certain topic. To keep the evaluation simple, each message in our system has only one selector value ranging from 0 to 99 while each subscriber subscribes for messages in a particular selector range (1/10th of the total range). This is reasonable to demonstrate different interests of the subscribers in the system. As it does not make sense to send all the messages on error prone wireless

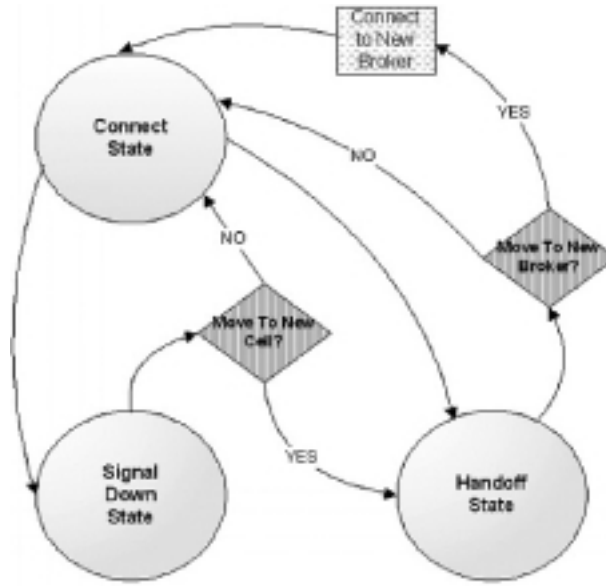


Fig. 3. State diagram for the subscriber mobility model.

channels with limited bandwidths and let the limited capacity devices (such as palmtops) perform the message selection, it was enabled at the broker instead of at the client.

For simulating subscriber mobility a program is written in Java which uses a mobility model illustrated in the state diagram of Fig. 3. In the *connect* state, the subscriber is in a particular cell, connected to one of the message brokers and receiving messages. A subscriber remains in the connect state for an exponentially distributed residence time with a mean t_{CS} , after which it either *handoffs* from one cell to another or goes to the *signal down* state. Signal down state simulates non-graceful handoffs and signal down intervals occurring due to poor wireless channels. It remains in the signal down state for an exponentially distributed duration with the mean t_{SD} . With a probability of 0.9 it either goes back to the same cell or moves to a new cell. The subscriber in the *handoff* state either reconnects to the same broker or connects to the other broker thus simulating that it has moved out of the location covered by the previous message broker. In either case, after the handoff, the subscriber goes back to the *connect* state. Although this is not a detailed mobility model keeping track of a subscriber's position, it is sufficient to simulate the mobility characteristics of terminals required for this investigation, including handoffs and signal down intervals. It should be noted that the model provides a number of parameters that can be varied to simulate a particular mobility environment. In our experiments, we simulated different mobility environments by varying the frequency of handoffs and distribution of signal down intervals.

In our experiments, we have one hundred subscribers running on each of the two machines. In the beginning half of the subscribers are connected to one message broker and the other half to the other but this number changes many times during the course of the simulation due to user mobility. Although one hundred subscribers per broker may seem to be much smaller than the actual number of subscribers connected to one broker in the cellular publish/subscribe systems of the future, the number of messages received per second by each subscriber in our setup is many times higher than that expected in a real world environment. Hence each subscriber in our setup simulates a load of many real subscribers on the broker. At the same time short mean residence time in a cell is used to generate high connection/disconnection requests per second.

We have used an open model publisher whose publishing rate is increased in steps until we reach a stage beyond which, increasing the publisher rate would make the system collapse. This is adequate for determining the highest sustainable system throughputs that is an important performance metric.

We have used the NIST Net network emulator [21] to emulate wireless channel characteristics. NIST Net is implemented as a kernel module extension to the Linux operating system and can be used to emulate a number of system characteristics such as packet delay, congestion, bandwidth limitation, packet loss, packet duplication and packet reordering.

3.1. Performance measures

Following measures are used as performance indicators.

- *Maximum Publisher Throughput: Maximum Publisher Throughput (X_P) is the maximum number of messages per second that the publisher is able to deliver to the system without making the system collapse.*
- *Subscriber Throughput: Subscriber Throughput (X_S) is the total number of messages received per second by all the subscribers in the system divided by message fanout. As there can be more than one subscriber interested in the same message, message fanout indicates the extent to which each message is duplicated by the broker. The total throughput of the subscribers is divided by message fanout for direct comparisons with publisher throughputs.*
- *Percentage Message Loss: Percentage message loss is the percentage of the messages lost by the subscribers.*

3.2. Parameters

The following are the workload parameters of interest.

- *Bandwidth: Total wireless bandwidth available is an important system parameter which may affect the overall system performance significantly. Most of our experiments were done for three bandwidth/mobility environments: fixed wireline, mobile with 10 MB/seconds total bandwidth and mobile with 220 kB/seconds total bandwidth. Examples of systems with approximately 220 kB/seconds bandwidth include current generation cellular networks based on standards such as CDMA2000 1xEV-DO RelA standard, also known as IS-856-A [27]. IS-856-A offers average packet data rates of up to 2 Mbps in CDMA2000 networks. Examples of systems with 10 MB/seconds bandwidth are future WiMAX (Worldwide Interoperability of Microwave Access) technology based networks with data rates of up to 75 Mbps. IEEE 802.16 [29] is one of the most important standards currently being developed for WiMAX technology based wireless metropolitan area networks (WMAN).*

In our experiments, the total bandwidth of 10MB/seconds ensures that it is the server that is the bottleneck (and not the bandwidth), representative of what we refer to as the *server bottlenecked region*. In the 220 kB/seconds total bandwidth environment the bandwidth becomes the bottleneck, representative of what we term the *bandwidth bottlenecked region*. It is to be noted that we express bandwidth in Bytes/second instead of bits/second, the commonly used unit of bandwidth. This is because the NIST Net network emulator used in the experiments takes a value in Bytes/second for setting the required bandwidth.

- *Number of Connection/Disconnection Requests per Second: The total number of connection/disconnection requests that each subscriber makes to the brokers per second can be divided*

into those occurring during graceful handoffs of the subscriber from one broker to the other (f_{GH}) and those occurring because of signal down intervals and non-graceful handoffs (f_H). If a Markov chain is built from the mobility model of this paper with each state of the Markov chain representing a particular state in the mobility model, we can determine the mean duration t_C between any two consecutive requests made by each subscriber to the broker. The following equation shows the relationship between t_C , f_{GH} and f_H .

$$t_C = 1/(f_{GH} + f_H)$$

Although both f_{GH} and f_H are related to t_C , it was found that more theoretical insights could be gained by treating them separately. Hence we call the frequency of connection/disconnection requests occurring due to signal down intervals and non-graceful handoffs as Frequency of Handoffs, f_H . The frequency of graceful handoffs of the subscriber from one broker to the other is called Frequency of Graceful Handoffs, f_{GH} . Both f_H and f_{GH} are varied by varying t_C .

As discussed earlier, we used short cell residence times to generate higher connection/disconnection requests per second. The default value of t_C calculated by solving the Markov chain discussed above is 154.32 seconds.

- *Distribution of Signal Down Intervals:* Mean signal down interval t_{SD} is the mean length of the interval during which no signal is available to the subscriber. Variability in signal down intervals can potentially affect overall system performance. So in addition to the exponentially distributed signal down intervals we also used bi-phase hyper-exponentially distributed intervals. Co-efficient of variation C_V of signal down intervals is varied as an input parameter.

The default value of t_{SD} used in this research is 12 seconds. Such a high value would stress test the mechanisms that store the messages for disconnected subscribers. This is consistent with other studies such as [3]. The default distribution of signal down intervals in this research is exponential with $C_V = 1$.

- *Total Time in Signal Down State:* Although total time in signal down state (T_{SD}) is not an independent parameter and can be given by

$$T_{SD} = f_H * t_{SD} * T \tag{1}$$

where T is the total time of the simulation. It was found that varying T_{SD}/T as an input parameter gives additional insights into the dynamics of the system. For example, for studying the effect of f_H , T_{SD}/T was varied keeping t_{SD} constant. The default value of T_{SD}/T obtained by solving the Markov chain with default values of other parameters is 0.0742.

- *Message Size:* In JMS, the total length of a message after the removal of IP and TCP headers is the sum of the lengths of the message header, the message properties and the message body. As all the messages have almost the same sized header and properties, the size of the message body (S) was varied as an input parameter and system performance was evaluated for different values of S .

Because we expect small messages (after potential message compression and/or simplification) to be sent over error prone wireless channels with scarce bandwidth in real environments, we used a small value equal to 50 Bytes as a default value of S .

Table 1 summarizes the workload parameters of the paper.

Table 1
Summary of workload parameters

Parameter	Default Value	Levels
Size of the Message Body (S)	50 Bytes	50,100,200,500 Bytes
Total Bandwidth	10 MB/second	220 kB/seconds, 10 MB/second
Proportion of the Total Time in Signal Down State (T_{SD}/T)	0.0742	0, 0.002, 0.01, 0.025, 0.05, 0.0742, 0.075, 0.1, 0.15
Mean Duration between consecutive connection requests (t_C)	154.32 seconds	Different levels corresponding to each level of T_{SD}
Frequency of Handoffs (f_H)	0.00617 per second	Different levels corresponding to each level of T_{SD}
Frequency of Graceful Handoffs (f_{GH})	0.000309 per second	Different levels corresponding to each level of T_{SD}
Mean Signal Down Interval (t_{SD})	12 seconds	12, 24 seconds
Probability Distribution of Signal Down Intervals	Exponential	Exponential, Hyper- Exponential
Co-efficient of Variation of Signal Down Intervals (C_V)	1	1, 1.5, 2, 2.5

4. Performance of JMS in cellular mobile networks

We conducted extensive sets of experiment to study the behavior of JMS in the mobile wireless environments. A representative set of results are presented in this Section. Further details can be found in [12].

JMS defines two types of subscriptions, *nondurable* and *durable*. Nondurable subscriptions are less reliable and can be used only for applications that can afford to miss messages. If a nondurable subscriber disconnects from the network for some reason, all the messages published during the time it was disconnected are lost. Nondurable subscriptions, however, give rise to high performance as there are no overheads involved in storing messages for the inactive subscribers.

Figure 4 shows the effect of mobility/bandwidth on the performance of nondurable subscriptions. These results were obtained using the default values of the parameters as in Table 1. The figure shows that for nondurable subscriptions in the 10 MB/second mobile case when the server is the bottleneck, subscriber throughput X_S is close to that in the fixed networks but it suffers from bursts of message losses as the subscribers lose all messages published during the time they were disconnected from the network. Such message loss in bursts may not be acceptable to a number of applications. Although nondurable subscriptions are not supposed to deliver messages published during the time the subscriber was voluntarily disconnected from the network, there should at least be a way to prevent message loss occurring due to user mobility and due to temporary disconnections in weakly connected wireless environments. The results also show that the system give poor performance in low bandwidth environments. The results are discussed in detail in [15].

Durable subscriptions are highly reliable and are used for applications that cannot afford to miss messages. In fixed networks, even if a durable subscriber is disconnected from the broker, it does not lose messages, as the broker preserves all the messages published during the time it was disconnected and delivers them as the subscriber reconnects. However, this storing and delivery of messages for the inactive subscribers impose a substantial overhead on the broker. As a result the performance of durable subscriptions is very poor in comparison to nondurable subscriptions. This suggests that even in the fixed networks, durable subscriptions should be restricted to a small number of subscribers that cannot afford to miss a message.

In the absence of any support service, durable subscriptions face additional challenges in mobile environments such as message loss and duplicate delivery of messages [15]. Moreover, in a highly mobile environment, each broker would end up having a large number of inactive subscribers that may

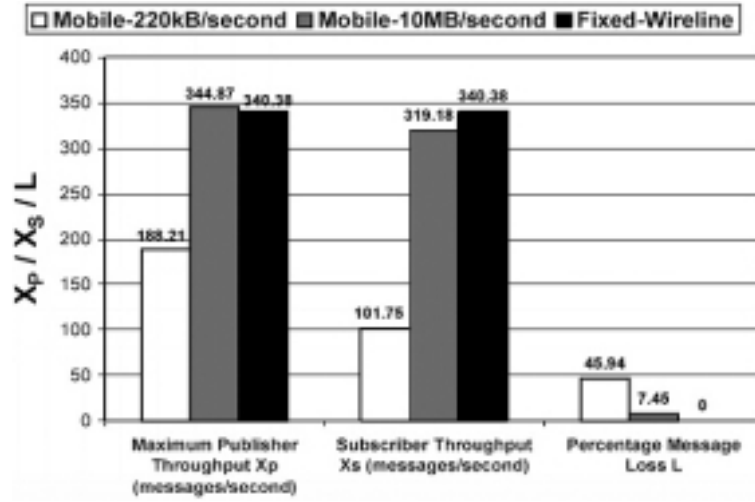


Fig. 4. Performance of nondurable subscriptions.

never reconnect to that broker, imposing on the broker a substantial overhead and making its performance increasingly worse to the point of saturation.

Several techniques can be considered to extend durable subscriptions to the mobile domain [15] out of which middleware level handoff seems to be the most natural. We implemented a prototype of middleware level handoffs and our results [15] show that durable subscriptions with middleware level handoffs perform poorly in mobile cellular networks where disconnections are frequent compared to fixed networks where interruptions in communications are rare. A set of representative results are shown in Fig. 5. They were obtained using the default values of the parameters. The figure shows that with durable subscriptions subscriber throughput X_s in mobile wireless environments can be as low as 11% of that in fixed networks. The reason is the overhead incurred in keeping track of and storing messages for the inactive subscribers as well as the overhead of delivering the stored messages during handoffs. Comparison of Figs 4 and 5 shows that throughput of durable subscriptions in mobile environments can be as low as 5% of that with nondurable subscriptions.

Poor performance of durable subscriptions in mobile environments can be attributed to a tight coupling between the subscriber and the broker. In fixed networks where the subscriber usually connects to the same broker, a tight coupling may not degrade performance, but in highly mobile environments this introduces a substantial overhead.

5. Semi-durable subscriptions

The discussion in previous section shows that nondurable subscriptions cannot prevent message loss in a mobile environment occurring due to temporary disconnections of the subscribers whereas durable subscriptions result in extremely low throughputs. Durable subscriptions with middleware level handoffs can thus only be justified for the sensitive applications such as air traffic control that cannot afford to miss a single message. However, there are many applications where occasional message loss is tolerable, in particular where further subsequent updates are likely to occur (such as game scores, stock information delivery and auction systems). Relying on durable subscriptions for disseminating information for such applications is too expensive from a performance perspective. On the other hand, nondurable

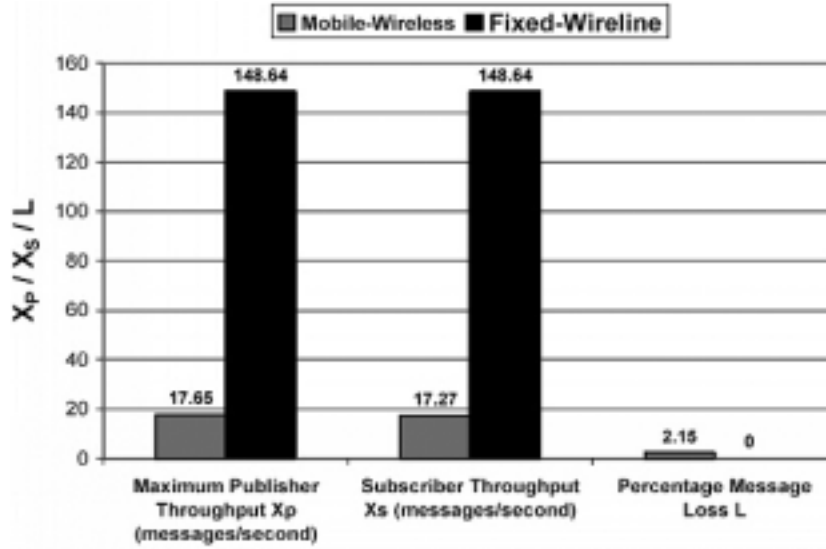


Fig. 5. Performance of durable subscriptions.

subscriptions cannot produce the desired behavior since frequently occurring message losses in bursts are not tolerable. Under these conditions, we need a new technique which is the midway between the two extremes: nondurable and durable. It may incur occasional message losses but should achieve high throughput. Moreover, it should be possible to configure the system anywhere between the two extremes.

In order to tackle the challenges mentioned above, we present a *semi-durable subscription* approach. All the mobility extensions of publish/subscribe systems discussed in the first section involve uncoupling and retrieval of messages from the previous broker the user was connected to in one form or the other (such as in middleware level handoffs). Thus they incur substantial performance overheads in dynamic environments. The semi-durable subscription technique on the other hand tends to recover the messages lost from the local buffer at the new broker. This assumes that the messages missed by the subscriber are available at the new broker. Thus, in semi-durable subscriptions each broker buffers all the messages published irrespective of its current active subscriptions. On the arrival of a connect request, the broker sends all the messages missed by the subscriber from its buffer. The ID of the last message the subscriber received is transmitted as a part of the subscription request and the broker sends messages published after the message with this ID.

It is worth mentioning that messages are broadcast in most distributed publish/subscribe systems so that they can be delivered to all the interested subscribers connected to the different brokers in the system. Moreover, research papers such as [23] show that in highly mobile environments, due to the practical limit on the number of multicast addresses and the overheads of frequent computing of multicast trees, the most feasible strategy is to broadcast all messages to all the brokers in the publish/subscribe system. Thus if a broker buffers messages irrespective of its current active subscriptions, messages will be available to be sent in case a new subscriber moves from another broker to this one. Even if we assume that not all the messages are published to all the brokers in the system, semi-durable subscription can handle this phenomenon of regionalism. This is discussed in detail in [13,14].

The advantage of message buffering at each broker is that it maintains the loose coupling between the subscriber and the broker. The broker thus does not incur overheads in maintaining and managing the

list of its inactive subscribers. Moreover, the subscriber can now disconnect from the broker without a need to be uncoupled and the handoff of subscriber now involves only one broker.

The disadvantage is that unlike middleware level handoffs the subscriber can now occasionally lose messages. This is because each broker can buffer only a limited number of messages due to the practical limits on the size of the buffer. However, it is worth mentioning here that semi-durable subscriptions are meant to deal with message losses occurring due to temporary disconnections of the subscribers in mobile wireless environments due to poor wireless channels and during handoffs; they are not an alternative to durable subscriptions. The losses in semi-durable subscriptions are typically associated to those subscribers who wandered in areas of very long disconnections. For other connectivity environments semi-durable subscriptions do not suffer any message loss. As the results discussed in the later sections show, a moderate buffer size can recover almost all of the messages missed by the subscriber due to temporary disconnections in a mobile environment even if we assume a very high publishing rate and fairly long disconnections of the subscribers.

One point worth mentioning here is that even if we assume that there is an infinite buffer at each broker and all the messages are buffered, semi-durable subscriptions differ from durable subscriptions in a number of ways. For example, while storing messages there is no concept of messages being stored for a certain subscriber. The message stored may be delivered to more than one subscriber or may not be delivered at all. Also unlike durable subscriptions, no particular broker is now storing messages for a certain subscriber. The subscriber can receive messages from any broker in the system. This flexibility in the message source in semi-durable subscriptions is extremely important in achieving high performance in a mobile domain. Since the broker does not store messages on a subscriber by subscriber basis, its performance does not depend on the number of inactive subscribers it has.

5.1. Operation of semi-durable subscriptions

We briefly discuss here the operation of semi-durable subscriptions. A more detailed discussion of the design is presented in [14].

The interactions between various system components in semi-durable subscriptions are shown in the collaboration diagram of Fig. 6 using standard COMET notation [18]. In this notation each sequence of interactions N is depicted by a sequence of messages $N.I$, $N.(I + 1)$ and so on where $N.I$ precedes $N.(I + 1)$. A lower case letter represents a concurrent sequence. For example, sequence of interactions N would be concurrent with the sequence of interactions N_a . The symbol $*$ shows repetitions. In order to explain the details of the system operations, each sequence of interactions is briefly reviewed.

Sequence 1: First the system initialization is performed. The system administrator provides the *Subscriptions Policy File*, the *Buffer Organization File* and the *Message Overwriting Policy File* along with the desired size of the buffer as inputs to the *Semi-durable Subscriptions Manager*. In accordance with the *Subscriptions Policy File* the *Buffer Subscriptions Manager* subscribes to the broker that in turn notifies the publish/subscribe system of its desired subscriptions and starts receiving messages on them. At the same time *Semi-durable Buffer Manager* organizes the size and internal partitions of the buffer according to the inputs from the administrator. *Buffer Subscriptions Manager* forwards every message to *Semi-durable Buffer Manager* which writes it to the buffer. When the buffer gets filled the *Semi-Durable Buffer Manager* uses the message overwriting policy to decide which messages to overwrite.

Sequence 2: Every subscriber in semi-durable subscriptions receives messages from the broker just as a normal nondurable subscriber. Hence if it gets disconnected from the broker for some reason it does

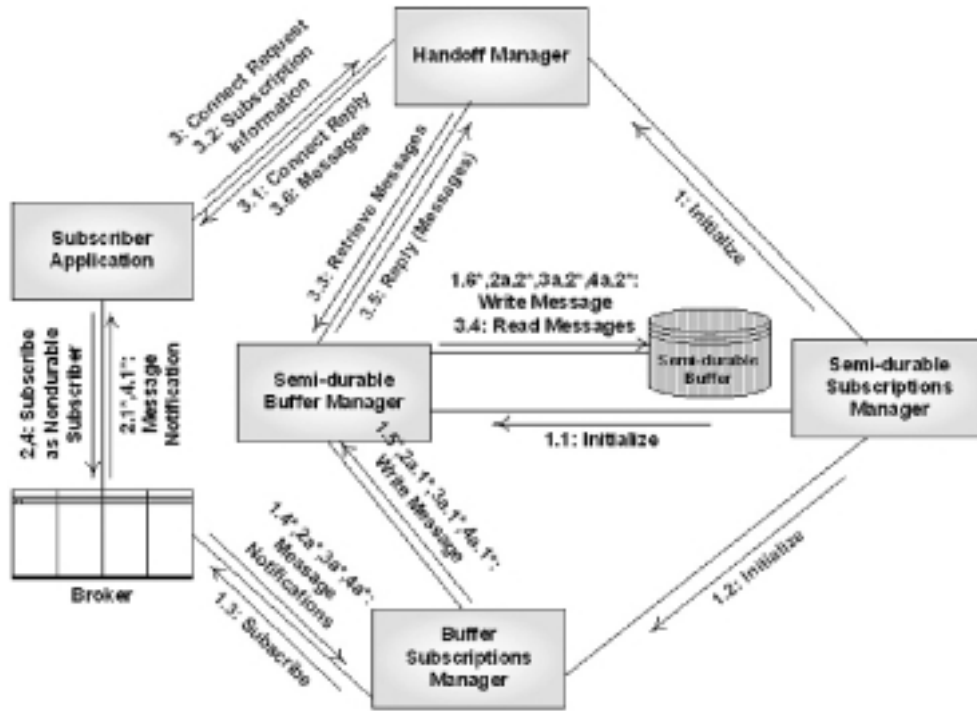


Fig. 6. Operation of semi-durable subscriptions: Collaboration diagram.

not need to be uncoupled from the broker. The broker does not maintain the list of its inactive subscribers or store messages on a subscriber by subscriber basis.

Sequence 3: Every subscriber keeps track of the ID of the last message it received in a log file. This could be the JMS Message ID or any other uniquely identifiable message ID used by the system. When the subscriber reconnects to the network after a disconnection, irrespective of whether it connects to the same broker or to the new broker (if it has come to a new location), it first contacts the *Handoff Manager* instead of contacting the broker directly. It passes its subscription information to the *Handoff Manager* along with the ID of the last message it received on a certain topic. The *Handoff Manager* in turn contacts the *Semi-durable Buffer Manager* to retrieve the lost messages. *Semi-durable Buffer Manager* tries to locate the message with that ID in its buffer. There are two alternative sequence of interactions 3A and 3B depending upon whether the *Semi-durable Buffer Manager* finds the message with that ID in the buffer.

Sequence 3A: If the *Semi-durable Buffer Manager* finds the message, it sends all the messages matching the subscriber's subscription criterion published after that message to the *Handoff Manager*. The *Handoff Manager* in turn delivers them to the subscriber. The subscriber then goes to sequence 4.

Sequence 3B: If the *Semi-durable Buffer Manager* does not find the message, there are two possibilities; either the new broker has not yet received that message due to network delay or the message has been overwritten. If the *Semi-durable Buffer Manager* finds a message published after the message with the ID supplied by the subscriber, it means that the message has been overwritten. Otherwise, the message has not yet reached the broker. In case the message has not yet reached the broker, the system waits for

the arrival of the message before disconnecting the subscriber from the Handoff Manager (to prevent duplicate message delivery). In case the message has been overwritten, Semi-Durable Buffer Manager delivers all the messages available in the buffer, matching the subscriber's criterion (and published after the message with the ID supplied by the subscriber) to the Handoff Manager which sends them to the subscriber.

Sequence 4: A control message from the Handoff Manager is used to inform the subscriber of the number of outstanding messages. After receiving the last message the subscriber disconnects from the Handoff Manager and subscribes to the broker just as a normal nondurable subscriber.

One point worth mentioning here is the message loss occurring in semi-durable subscriptions. In sequence 3B if the Semi-durable Buffer Manager finds that the earliest message in its buffer was published later than the message with the message ID passed by the subscriber, it means that some of the messages missed by the subscriber have been overwritten and hence are not available to be recovered. This loss of messages can be reduced by running sophisticated message overwriting policies, which can guarantee that the subscriber will receive at least all the *important* messages.

The prototype implementation of semi-durable subscriptions required minor modifications in the subscriber's application. If one were to have the source code of the JMS implementation the scheme could be implemented at the middleware level. This would hide all the implementation details from the applications and would make the integration of fixed and mobile domains seamless.

5.2. System parameters for semi-durable subscriptions

Semi-durable subscriptions provide a rich parameter space to configure the system according to the needs of the mobility environment. The important system parameters for semi-durable subscriptions are described in this section.

- *Buffer Size:* Buffer Size is an important system parameter for semi-durable subscriptions. Using large buffer size will reduce message loss but will impose more overheads by increasing the search time of the searching algorithm and increasing the task of delivering greater number of messages. Large buffer size also increases the heap memory size of the JVM and hence may cause more frequent garbage collections. Buffer size thus should be set according to the desired output of the system or the needs of the mobility environment. Size of the buffer can also be set dynamically according to the publishing rate.
- *Internal Organization of the Buffer:* Our experience with semi-durable subscriptions shows that it makes sense to divide the buffer into a number of partitions. It decreases the search time of the searching algorithm and at the same time allows several requests to be served concurrently. However, parameters such as number of partitions, size of each partition need to be determined. Moreover, different partitions can have the same size or they may have different sizes. The buffer thus can be organized according to the subscription characteristics and the needs of mobility environment hence providing control on the system behavior.
- *Message Overwriting Policies:* The message overwriting policy is the most important system parameter as this parameter decides which messages should be overwritten. A number of overwriting policies can be designed but the goal of each should be to ensure that at least all *important* messages are delivered to the subscribers. The *importance* of a message may have different meanings in different contexts and can give rise to different message overwriting policies. The simplest of all the policies is the *least recently published overwriting policy*. In this policy when the buffer is full and a

new message arrives, the least recently published message(s) is (are) overwritten. The justification behind this policy is the fact that recently published messages are typically more important than others in the sense that they carry the latest information. More sophisticated policies are also possible (for details, see [14]).

5.3. Comparison with other subscription mechanisms

Performance of semi-durable subscriptions is compared to that of nondurable and durable subscriptions using the default values of the workload parameters listed in Table 1. For semi-durable subscriptions, the buffer was divided into 10 equal partitions corresponding to different message selector ranges. Each partition has a fixed capacity of 3000 messages. The least recently published overwriting policy is used.

The results given in [13] shows that for server bottlenecked regions the maximum sustainable subscriber throughput X_S for semi-durable subscriptions is about 90% of that of nondurable subscriptions. This loss in throughput of semi-durable subscriptions is due to the overhead of running the semi-durable managers, buffer maintenance and recovering the missed messages to subscribers. But this loss in throughput pays off by reducing the message loss from 7.45% in non-durable subscriptions to only 0.18%. X_S for durable subscriptions, where we can expect the message loss to be zero, is merely 6% of that achieved with the semi-durable subscriptions. The results also show that semi-durable subscriptions give much better performance than other subscription mechanism in the bandwidth bottlenecked region.

We also compare the performance of the three subscription approaches at a given a publishing rate for a system in which the bandwidth is not the bottleneck. The results were obtained with different publishing rates equal to or lower than 285.29 messages/second (the maximum publishing rate semi-durable subscriptions can sustain in a server bottlenecked region with the values of the parameters used). The results with the publishing rate of 285.29 are shown in Fig. 7. The discussion is equally valid for all publishing rates lower than this message publishing rate.

Figure 7 shows that at the given rate, the subscriber throughput X_S for semi-durable subscriptions is approximately 7.5% higher than that achieved with non-durable subscriptions. This corresponds to the messages delivered by the Handoff Manager in the system with semi-durable subscriptions. These messages are lost in a system with nondurable subscriptions. The X_S for durable subscriptions is zero, as the system with durable subscriptions cannot withstand such a high publishing rate.

6. Effect of workload parameters on the performance of semi-durable subscriptions

This section investigates the effect of different workload parameters on the performance of semi-durable subscriptions. For this investigation the default values of the fixed parameter shown in Table 1 were used. For semi-durable subscriptions the Least Recently Published message overwriting policy is used with the buffer divided into 10 equal partitions each having a fixed size of 3000 messages. The results for each set of experiments also include the corresponding graph for nondurable subscriptions for comparison purposes.

6.1. Effect of frequency of handoffs

To study the effect of frequency of handoffs f_H on the performance of semi-durable subscriptions, the proportion of total time in signal down state T_{SD}/T is varied keeping t_{SD} constant. The results are shown in Fig. 8.

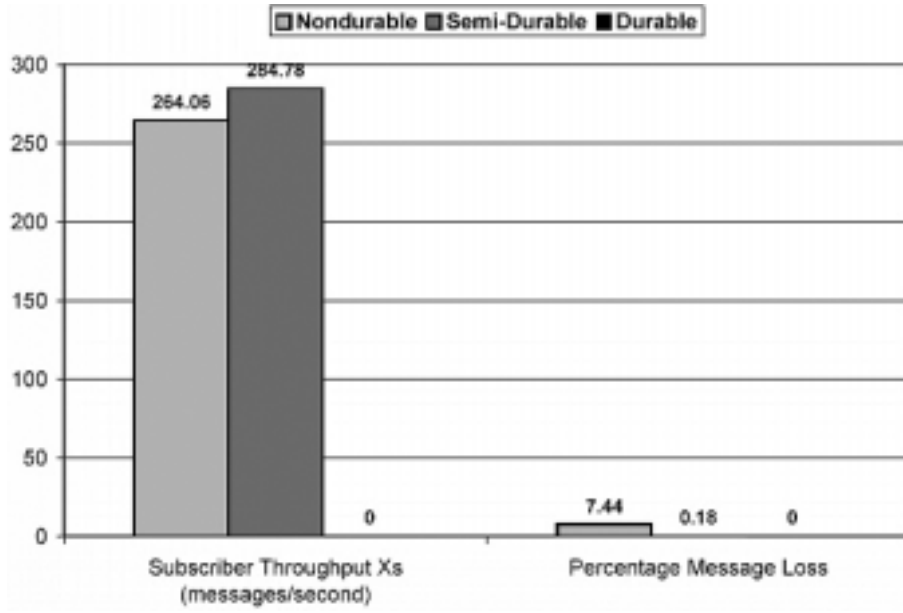


Fig. 7. Performance of three subscription types at a publishing rate of 285.29 messages/second in the region where bandwidth is not the bottleneck.

Results for nondurable subscriptions are discussed in detail in [15]. For semi-durable subscriptions, Fig. 8(a) shows that as the frequency of handoffs is increased from zero (fixed subscribers), the maximum publisher throughput X_P decreases sharply at first. The reasons are the connections/disconnections and handoffs overheads which are imposed on the system each time a subscriber disconnects and/or reconnects to the system. Also for nonzero f_H the number of subscribers per broker may temporarily become greater than the average number of subscribers per broker. The increased load of subscribers on the broker is another cause of a lower maximum sustainable throughput. As the frequency of handoffs and hence T_{SD}/T increases further, the total time of disconnection of subscribers increases. This decreases the average number of active subscribers per broker. This decrease in load of subscribers on the broker increases X_P . The behavior of semi-durable subscriptions is the same as of nondurable subscriptions until about T_{SD}/T equal to 0.05. However, as f_H increases further, unlike nondurable subscriptions, the X_P for semi-durable subscriptions does not increase further and more or less remains the same. The reason is that in semi-durable subscriptions the connections/disconnections overhead is much greater than in nondurable subscriptions, as every subscriber reconnecting to the broker first connects to the Handoff Manager to recover the missed messages. Hence as T_{SD}/T and thus f_H increases, this overhead increases, offsetting the benefit gained in performance due to the decreasing number of subscribers per broker.

Subscriber throughput X_S for semi-durable subscriptions in Figure 8(b) follows the same pattern as X_P (unlike nondurable subscriptions). The reason is that due to small message loss the X_P and X_S in semi-durable subscriptions are almost the same. Figure 8(c) shows that message loss percentage for semi-durable subscriptions increases with the increase in frequency of handoffs. This is because of the *leaky* implementation in which some messages are lost in semi-durable subscriptions when the subscriber is transferred from the Handoff Manager to the broker and hence with the increase in f_H the frequency of transfers increases which in turn increases the message loss. Also with increasing f_H the probability that some subscribers will disconnect from the network for a period long enough to overwrite

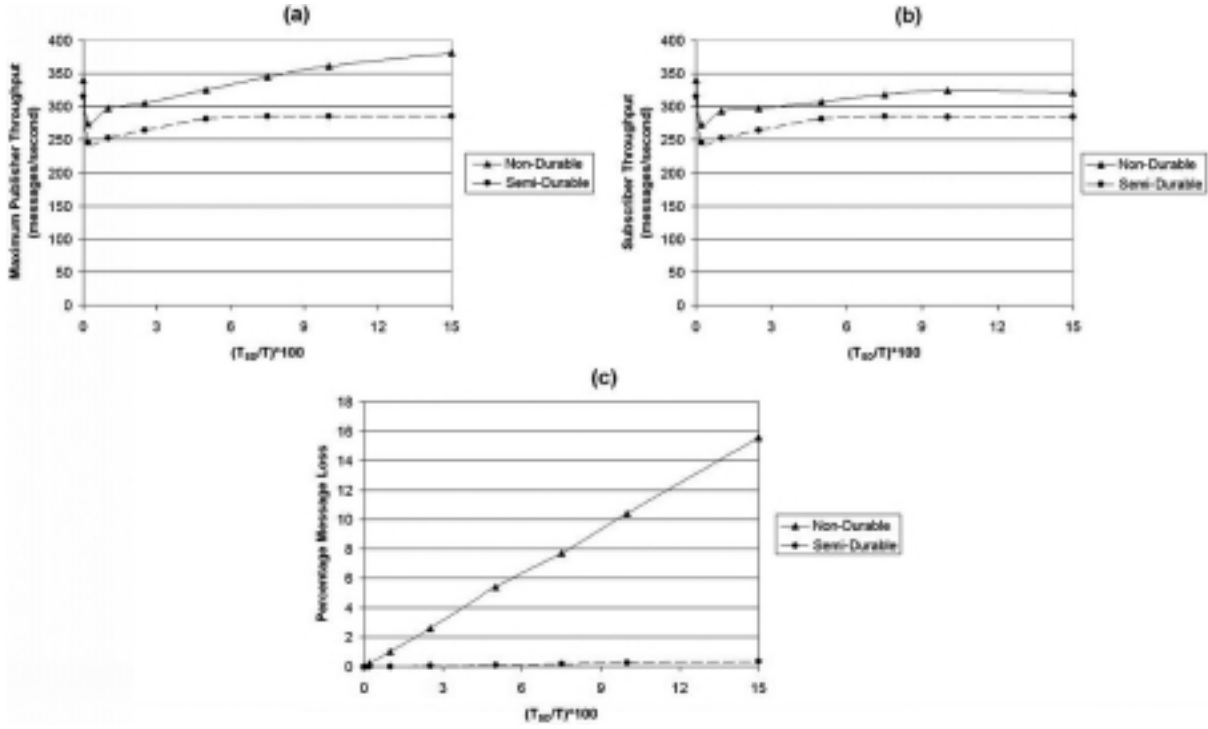


Fig. 8. Effect of frequency of handoffs on the performance of semi-durable subscriptions ($f_H = ((T_{SD}/T)*100)*(1/(100*t_{SD}))$; $t_{SD} = 12$ seconds) (a) Maximum throughput of publisher (b) Throughput of subscribers (c) Proportion of messages lost.

the semi-durable buffer, increases. However, the rate of increase in message loss with f_H in semi-durable subscriptions is much lower compared to that in nondurable subscriptions.

These results show that for the mobility parameters used in the experiments, although semi-durable subscriptions give slightly lower maximum throughput than non-durable subscriptions, they effectively bring down the message loss to almost zero. The difference between the message loss of the two subscription mechanisms increases with the increase in f_H without any change in the difference between the throughputs. This shows that semi-durable subscriptions are a much better choice for mobile environments than nondurable subscriptions.

6.2. Effect of variability in signal down intervals

The effect of variability in signal down intervals on the performance of nondurable and semi-durable subscriptions is studied by varying the co-efficient of variation C_V of the duration of the signal down intervals. The results show that probability distribution of signal down intervals affects X_P , X_S and percentage message loss slightly. The throughputs decrease while percentage message loss increases as the C_V of signal down intervals increases. The results are available in [14].

6.3. Effect of message size

To study the effect of message size on the performance of semi-durable subscriptions, the size of the message body (S) is varied keeping the size of message header and properties constant. The results are shown in Fig. 9.

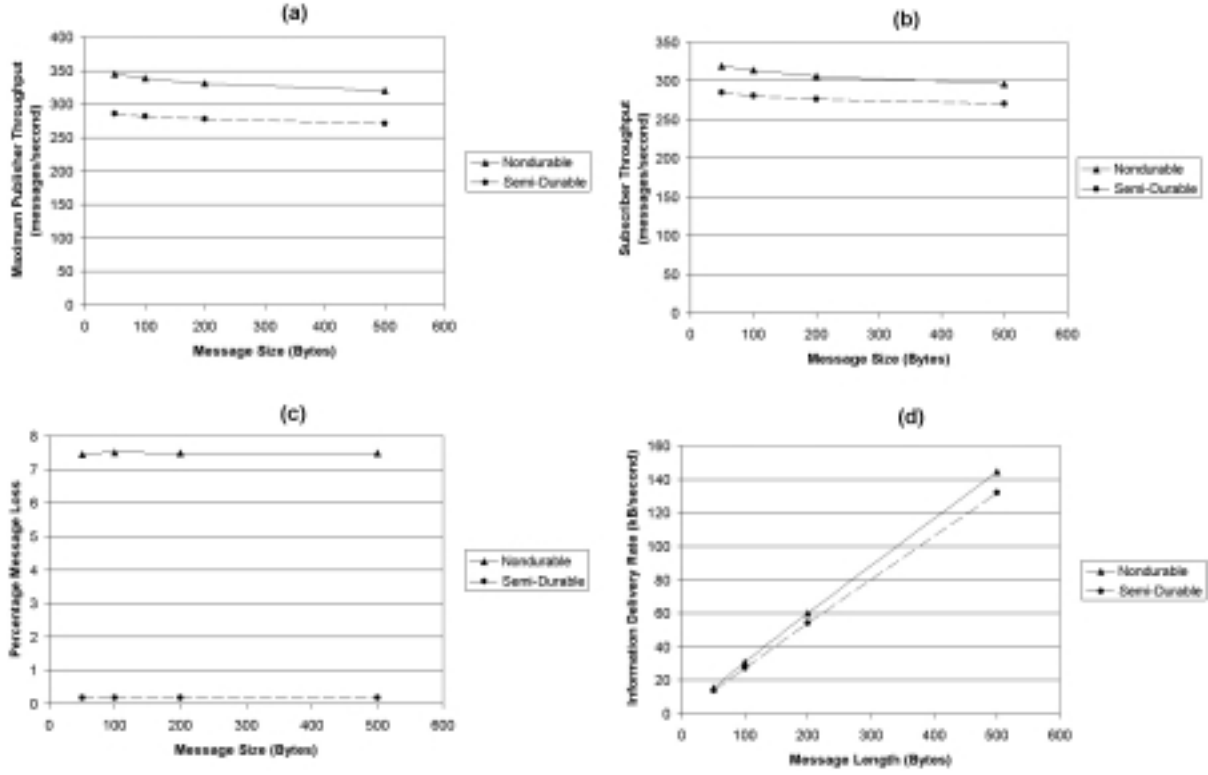


Fig. 9. Effect of message size on the performance of semi-durable subscriptions (a) Maximum throughput of publisher (b) Throughput of subscribers (c) Proportion of messages lost (d) Maximum delivery rate of information.

Figure 9(a) and 9(b) show that for both nondurable and semi-durable subscriptions, the system throughput decreases with the increase in message size. The reason is that the broker has to do more work on larger messages as compared to smaller ones. This includes receiving the message from the publisher, storing it in a buffer, delivering it to all interested subscribers and then reclaiming the space from the buffer in memory. Also the size of heap memory increases with the increase in message size and the broker may need to do garbage collection more often than with small messages. Moreover, for large messages there is more contention for resources on the network. Figure 9(c) shows that the message size has no effect on message loss percentage.

Figure 9(d) shows the effect of message size on the maximum sustainable information delivery rate of the broker. Information delivery is given by

$$\text{Information delivery rate} = S(\text{in kB}) * X_S \quad (2)$$

It reflects the amount of application level data delivered by the broker per second. The figure shows that information delivery rate increases almost linearly with the increase in message size. This suggests that in the server bottlenecked region information should be batched together and then sent as a larger message. In the bandwidth bottlenecked region too, using large messages is expected to increase the information delivering capacity. This is because sending more message contents in the same message saves bandwidth by reducing the overhead of the JMS and lower layer headers of several smaller messages.

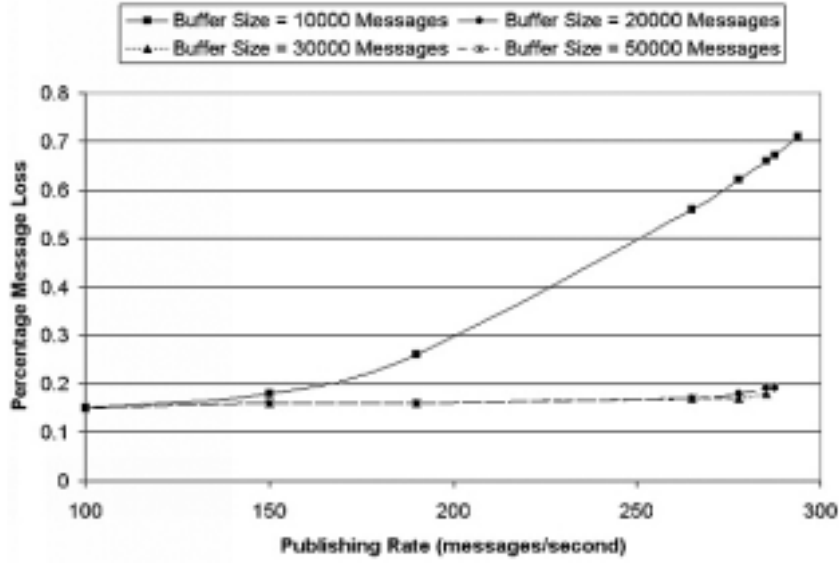


Fig. 10. Effect of publishing rate on message loss for different buffer sizes ($t_{SD} = 12$ seconds).

7. Effect of system parameters on the performance of semi-durable subscriptions

In this section we study the effect of system parameters such as buffer size and message overwriting policy on the performance of semi-durable subscriptions. We demonstrate how these parameters can be controlled to configure the system according to the desired characteristics. A number of experiments were conducted for this purpose using the default values of fixed workload parameters presented in Table 1 unless otherwise mentioned. For semi-durable subscriptions 10 partitions of the buffer were used with Least Recently Published message overwriting policy as the default policy.

7.1. Effect of publishing rate on message loss

This section presents a study of the message loss occurring in semi-durable subscriptions at different buffer sizes at a given publishing rate. The first set of results shown in Fig. 10 is generated using t_{SD} equal to 12 sec. To study the effect of longer signal down intervals, a second set of results shown in Fig. 10 uses a t_{SD} equal to 24 sec. For the second set duration between consecutive connection requests to the broker t_C is increased to 308.64 sec (from 154.32 sec in first set) to keep T_{SD}/T equal to the default value of 0.0742.

Publishing rate is increased until the maximum sustainable throughput is achieved. Since smaller buffer sizes can sustain higher publishing rates, the results given in Figs 10 and 11 show more readings for smaller buffer sizes as compared to larger ones.

The results show that as the buffer size increases, message loss for a given publishing rate decreases. However, beyond a certain buffer size the percentage message loss does not decrease further with the increase in buffer size. The reason is that not all the loss occurring is due to the messages getting overwritten. A significant portion of loss occurs in this prototype implementation of semi-durable subscriptions during the transfer of subscribers from the Handoff Manager to the broker. This portion of loss increases with the increase in publishing rate. This is because increased publishing rates increase the probability that a message will be missed in the duration in which the subscriber transfers from the

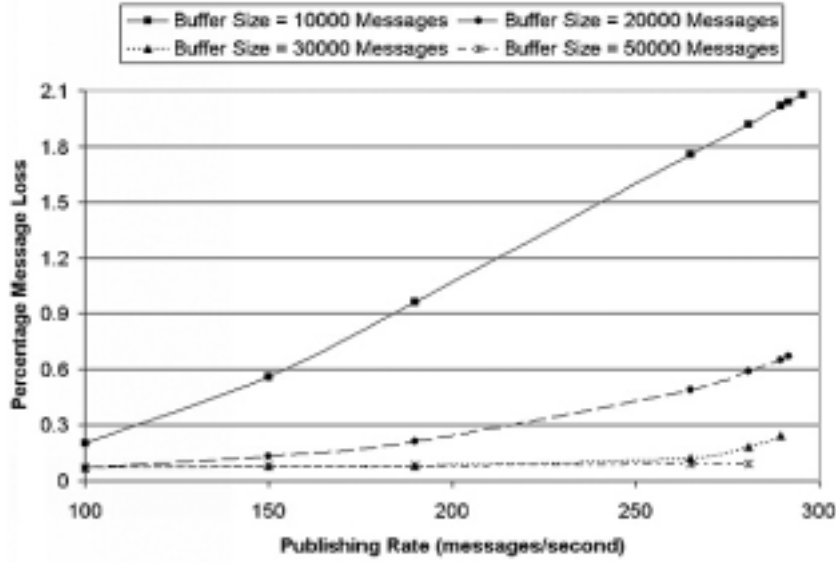


Fig. 11. Effect of publishing rate on message loss for different buffer sizes ($t_{SD} = 24$ seconds).

Handoff Manager to the broker. Contrary to the intuition, this portion of the loss is higher for t_{SD} equal to 12 seconds than for t_{SD} equal to 24 seconds. The reason for this is that for t_{SD} equal to 24 seconds t_C is equal to 308.64 seconds instead of 154.32 seconds (that corresponds to t_{SD} equal to 12 seconds). A higher t_C means a lower number of connections/disconnections per second and thus for t_{SD} equal to 24 seconds this portion of message loss is almost half of that for t_{SD} equal to 12 seconds.

If we subtract the portion of the message loss occurring during the transfer of subscribers (in integrated implementations it will not be present at all) from the total loss, the results show that at a given publishing rate with the increase in buffer size message loss decreases until it reaches zero. Increasing buffer size beyond this critical value just increases the overheads on the system without any reduction in message loss. The critical value of buffer size depends on the publishing rate as well as on the mobility environment. For example, in Fig. 10 ($t_{SD} = 12$ seconds) at a publishing rate of 250 messages/second a buffer size of 20000 messages is sufficient to reduce the message loss to a small value but in Fig. 11 ($t_{SD} = 24$ seconds) a buffer size of 30000 messages or more is desirable. The buffer size thus should be carefully selected according to the given mobility environment as well as the publishing rate.

7.2. Effect of buffer size on the performance of semi-durable subscriptions

We have analyzed the effect of buffer size on the performance of semi-durable subscriptions. The results are given in [13]. The results show that there exists a tradeoff between maximum publisher throughput and message loss which can be controlled by buffer size. Both the maximum sustainable publisher throughput and message loss decrease with an increase in buffer size. However, there exists a certain threshold buffer size for the given mobility environment that most suits the system in a sense that not only it brings down the message loss to a very small value but it also maintains the maximum subscriber throughput. In most of the cases it will make sense to set the buffer size to that threshold value.

7.3. Effect of message overwriting policy on the performance of semi-durable subscriptions

In this section we study the effect of three message overwriting policies on the performance of semi-durable subscriptions. For this study, we assign each message an index value between 1 to 10 that corresponds to the importance of message, with higher values representing more importance. The publisher randomly assigns each message an index between 1 to 10 with a uniform probability of 0.1. The importance of message in our study is static. Hence once assigned the index of the message does not change.

We define a new performance metric *percentage index loss* E as the percentage of index lost when messages are sent from publishers to the subscribers. E is a measure of the percentage of *important information* lost. It is calculated by first adding index of all the messages received by the subscribers and dividing it with message fanout. The resulting number I_S is subtracted from the sum of index of all messages published by the publisher I_P . The percentage is then taken by dividing $I_P - I_S$ by I_P and multiplying by 100. Thus,

$$E = ((I_P - I_S)/I_P) * 100 \quad (3)$$

For the experiments described in this section we use a high value of t_{SD} equal to 24 seconds to stress test the overwriting policies. Default values of other workload parameters as listed in Table 1 were used. The buffer was divided into 10 partitions each having a total size of 2000 messages. We investigate three overwriting policies in our study:

- *Random (RAN)*: When the buffer gets full, this policy randomly picks up a message from the buffer and overwrites it.
- *Least Recently Published (LRP)*: As discussed in Section 5.2 this policy overwrites the least recently published messages.
- *Policy with Priorities (PwP)*: This policy divides each partition of the buffer into two unequal parts. The first part has a size of 80% of the total size of the partition and uses LRP. The second part keeps the important messages and overwriting here takes place on the basis of a message index, where messages with higher indices are given priority over messages with lower indices. Whenever a message in the first part of the buffer partition is selected to be overwritten, it is first written to the second part (if there exists a message with a lower priority). Each message in the second part is given a maximum lifetime of 48 seconds (twice the mean signal down interval) after which it is marked to be overwritten.

The results are shown in Fig. 12. Figure 12(a) shows that running more sophisticated policies reduces the maximum throughput of the system due to the computing overheads involved. Hence in Fig. 12(a), X_S is decreased from 289.79 for LRP to 258.63 for PwP. Random overwriting policy gives lower throughput than LRP because unlike LRP messages are no longer in their publishing order for RAN and hence for every subscriber reconnecting the whole buffer is searched for the messages meeting the subscriber's criterion. For both RAN and LRP, percentage message loss L and percentage index loss E are equal. This is because for both of these policies, the probability of message loss does not depend on the message index value. For PwP, E is significantly lower than L showing that most of the messages lost had lower index value and hence were less important.

In Fig. 12(b), we compare L and E obtained with different policies for an X_S value of 258.63 messages/second (maximum subscriber throughput PwP can sustain). The results show that we get the least message loss L with LRP. With PwP we get slightly higher L than LRP because PwP prevents loss

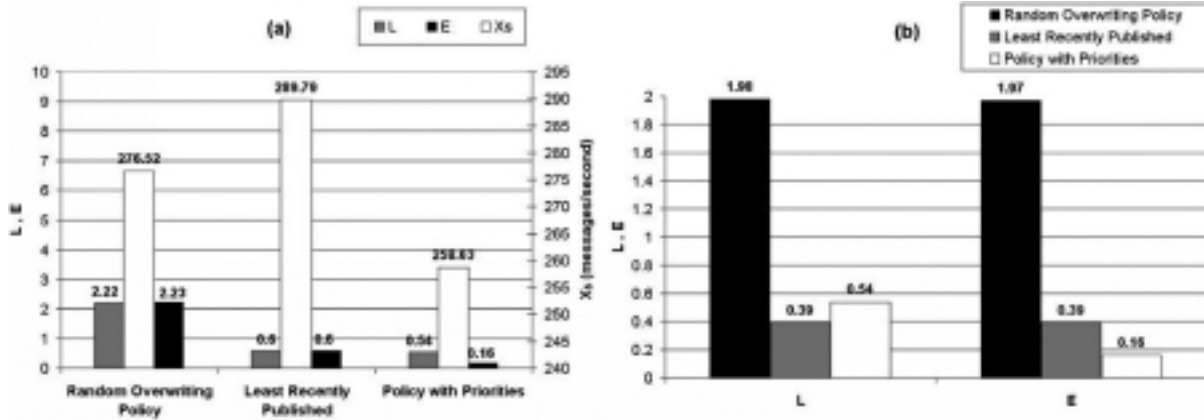


Fig. 12. Effect of the message overwriting policy on the performance of semi-durable subscriptions (a) Performance of the different overwriting policies (b) Comparison of L and E for different overwriting policies at $X_S = 258.63$ messages/second.

of important messages at the cost of higher loss of less important messages. However, with PwP we get least percentage index loss E showing that most of the important messages were delivered to the subscribers. With the RAN policy both L and E are substantially higher than both PwP and LRP because with RAN almost every subscriber reconnecting to the system loses a part of the messages randomly overwritten by RAN.

Note that in order to reduce the important message loss to a percentage index value of 0.11 with LRP (as in PwP) at $X_S = 258.63$ messages/second, buffer size is needed to be increased from 20000 messages to almost 40000 messages (See Fig. 11). This shows that by running a sophisticated policy we can reduce the amount, and hence cost, of buffering needed.

8. Conclusions

There are many features of publish/subscribe paradigms that make them a good choice for the mobile wireless domain. However, adapting current publish/subscribe systems to the dynamic and unreliable mobile wireless environment is a challenging task. This research extended a popular implementation of publish/subscribe system to achieve high performance under high frequency of handoffs and error prone wireless channels. Some of the insights of this research are summarized.

Current implementations of publish/subscribe systems as well as the traditional solutions for extending publish/subscribe systems to the mobile domain do not perform well in highly mobile and unreliable wireless settings. Although nondurable subscriptions produce almost as much throughput for mobile users as for fixed users, they suffer from message loss in bursts. Durable subscriptions with middleware level handoffs give rise to very low throughputs. The poor performance of durable subscriptions in mobile environments can be attributed to a tight coupling between the broker and the subscriber. The semi-durable subscription approach ensures loose coupling between the broker and the subscribers and hence does not incur substantial performance overheads. As discussed in Section 5.3, semi-durable subscriptions incur a nominal cost in terms of 10% reduction in X_P compared to nondurable subscriptions but recovered 97.5% of the messages lost in nondurable subscriptions. For the experiments described in that section X_P for durable subscriptions is merely 6% of that of semi-durable subscriptions. Results

presented in Section 5.3 also show that at all publishing rates equal to or lower than the maximum semi-durable subscriptions can sustain, semi-durable subscriptions give rise to higher subscriber throughput than even nondurable subscriptions.

Results described in Section 6 show that mobility parameters such as frequency of handoffs, mean and co-efficient of variation of probability distribution of signal down intervals can significantly affect performance of publish/subscribe systems. Hence all these factors should be taken into account while deploying them in mobile wireless networks. Message size is another important factor that can affect the performance. Section 6.3 shows that increasing the message size decreases X_S but increases the information delivering capacity of the broker. Hence information batching before transmission can improve performance in mobile wireless environments.

Semi-durable subscriptions provide a rich parameter space to configure the system into the desired mode. Buffer size, buffer organization and message overwriting policies can be used to exploit the inherent tradeoff between message loss and throughput in semi-durable subscriptions. There exists a certain threshold buffer value for a given mobility environment that most suits the system in a sense that not only it brings down the message loss to a very small value but it also maintains the maximum subscriber throughput. Using sophisticated message overwriting policies can not only reduce the volume of important information lost but can also decrease the amount, and hence cost, of buffering. Figure 12(b) shows that although L with the PwP policy is slightly higher than that with the LRP policy, using PwP reduces E by 76% compared to that achieved with LRP. Results obtained in Fig. 11 show that for the parameters used in the experiments such a reduction in E when using LRP can be achieved only by a 100% increase in buffer size.

The paper used a peer-to-peer topology of brokers. However, other topologies such as a hierarchical topology are possible. It is an important direction for future research to compare the behavior of different topologies for dynamic mobile wireless environments with high frequency of connections and disconnections. If using a hierarchical or hybrid (combination of peer-to-peer and hierarchical) topology of brokers, the relationship between semi-durable buffers at each layer of hierarchy and their corresponding message overwriting policies makes an interesting problem to investigate.

Acknowledgements

Financial support for this research was provided by Natural Sciences and Engineering Research Council of Canada (NSERC).

References

- [1] G. Banavar, T.D. Chandra, B. Mukherjee, J. Nagarajarao, R.E. Strom and D. C. Sturman, An Efficient Multicast Protocol for Content-based Publish/Subscribe Systems, in: *Proceedings of the 19th IEEE International Conference on Distributed Computing Systems*, Austin, TX, 1999, 262–272.
- [2] L.F. Cabrera, M.B. Jones and M. Theimer, Herald: Achieving a Global Event Notification Service, in: *Proceedings of the 8th Workshop on Hot Topics in Operating Systems*, (Elmau, Germany, 2001), pp. 87–94.
- [3] M. Caporuscio, A. Carzaniga and A.L. Wolf, Design and Evaluation of a Support Service for Mobile, Wireless Publish/Subscribe Applications, Technical Report CU-CS-944-03, Department of Computer Science, University of Colorado, Boulder, CO, 2003.
- [4] L. Capra, W. Emmerich and C. Mascolo, Middleware for Mobile Computing, in: *Advanced Lectures on Networking – Networking 2002 Tutorials*, (Pisa, Italy, 2002), pp. 20–58.
- [5] A. Carzaniga, *Architectures for an Event Notification Service Scalable to Wide-Area Networks*, PhD Thesis, Politecnico di Milano, Milano, Italy, 1998.

- [6] A. Carzaniga, D.S. Rosenblum and A.L. Wolf, *Content-Based Addressing and Routing: A General Model and its Application*, Technical Report CU-CS-902-00, Department of Computer Science, University of Colorado, Boulder, CO, 2000.
- [7] A. Carzaniga, D.S. Rosenblum and A.L. Wolf, Design and Evaluation of a Wide-Area Event Notification Service, in: *ACM Transactions on Computer Systems* **19**(3) (2001), 332–383.
- [8] A. Carzaniga, D.S. Rosenblum and A.L. Wolf, Interfaces and Algorithms for a Wide-Area Event Notification Service, Technical Report CU-CS-888-99, Department of Computer Science, University of Colorado, Boulder, CO, 1999. Revised 2000.
- [9] G. Cugola and E. Di Nitto, Using a Publish/Subscribe Middleware to Support Mobile Computing, in: *Proceedings of Workshop on Middleware for Mobile Computing*, Heidelberg, Germany, 2001.
- [10] M.K. Denko and H. Lu, Replica Dissemination and Update Strategies in Cluster-Based Mobile Ad Hoc Networks, in: *Mobile Information Systems* **2**(4) (2006), 193–209.
- [11] P.T. Eugster, P. Felber, R. Guerraoui and A.M. Kermarrec, *The Many Faces of Publish/Subscribe*, Technical report DSC ID: 2000104, Swiss Federal Institute of Technology, Lausanne, Switzerland, 2001.
- [12] U. Farooq, High Performance Middleware for mobile wireless publish/subscribe systems, M.A.Sc. Thesis, Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, 2003.
- [13] U. Farooq, S. Majumdar and E.W. Parsons, Engineering Mobile Wireless Publish/Subscribe Systems for High Performance, in: *the proceedings of the 12th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Volendam, The Netherlands, 2004, pp. 295–305.
- [14] U. Farooq, S. Majumdar and E.W. Parsons, Semi-Durable Subscriptions: A Technique to Achieve High Performance in Mobile Wireless Publish/Subscribe Systems, Technical Report SCE-03-17, Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, 2003. Available from <http://www.sce.carleton.ca/~ufarooq/publications.html>
- [15] U. Farooq, E.W. Parsons and S. Majumdar, Performance of Publish/Subscribe Middleware in Mobile Wireless Networks, in: *Proceedings of the 4th International Workshop on Software and Performance*, San Francisco, CA, 2004, pp. 278–289.
- [16] P. Fenkam, E. Kirda, S. Dustdar, H. Gall and G. Reif, Evaluation of a Publish/Subscribe System for Collaborative and Mobile Working, in: *Proceedings of the 11th IEEE International Workshop on Enabling Technologies*, Pittsburgh, PA, 2002, pp. 23–28.
- [17] L. Fiege, F.C. Gartner, O. Kasten and A. Zeidler, Supporting Mobility in Content-Based Publish/Subscribe Middleware, in: *Proceedings of ACM/IFIP/USENIX International Middleware Conference*, Rio de Janeiro, Brazil, 2003, pp. 103–134.
- [18] H. Gomaa, *Designing Concurrent, Distributed and Real-Time Applications with UML*, Addison Wesley, 2000.
- [19] Y. Huang and H. Garcia-Molina, Publish/Subscribe in a Mobile Environment, in: *Proceedings of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access*, Santa Barbara, CA, 2001, pp. 27–34.
- [20] G. Muhl, L. Fiege, F.C. Gartner and A.P. Buchmann, Evaluating Advanced Routing Algorithms for Content-Based Publish/Subscribe Systems, in: *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Fort Worth, TX, USA, 2002, pp. 167–178.
- [21] National Institute of Standards and Technology, NIST Network Emulation Tool, <http://snad.ncsl.nist.gov/itg/nist-net/index.html>
- [22] Object Management Group, Common Object Request Broker Architecture (CORBA/IIOP) v 3.0.2, <http://www.omg.org>.
- [23] L. Opyrchal, M. Astley, J.S. Auerbach, G. Banavar, R.E. Strom and D.C. Sturman, Exploiting IP Multicast in Content-Based Publish/Subscribe Systems, in: *Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, New York, NY 2000, pp. 185–207.
- [24] E. Pitt and K. McNiff, *Java.RMI: The Remote Method Invocation Guide*, Addison Wesley, 2001.
- [25] D. Rogerson, *Inside COM* (Microsoft Press, 1997).
- [26] Sun Microsystems Inc., Java Message Service, <http://java.sun.com/products/jms>.
- [27] Telecommunications Industry Association, 1xEV-DO RelA: TIA-856-A cdma2000 High Rate Packet Data Air Interface Specification, 2004.
- [28] A.B. Waluyo, B. Srinivasan and D. Taniar, Research in Mobile Database Query Optimization and Processing, in: *Mobile Information Systems* **1**(4) (2005), 225–252.
- [29] Wireless MAN Working Group, <http://wirelessman.org/>.
- [30] B. Yang and A.R. Hurson, Similarity-Based Clustering Strategy for Mobile Ad Hoc Multimedia Databases, in: *Mobile Information Systems* **1**(4) (2005), 253–273.
- [31] A. Zeidler and L. Fiege, Mobility Support with REBECA, in: *Proceedings of the 1st International Workshop on Mobile Computing Middleware*, Providence, RI, 2003, pp. 354–361.

Umar Farooq is a Ph.D. candidate at the Department of Systems and Computer Engineering at Carleton University in Ottawa, Canada. He received his Masters in Electrical Engineering from Carleton University in 2003 and his Bachelors in Electrical

Engineering from the University of Engineering and Technology in Lahore, Pakistan in 2001. As a student, Umar has received several highly competitive scholarships, awards and scholastic honors, including Pakistan's national award, Medal of Excellence, given by the President of Pakistan. Umar is working as a graduate research assistant in Real-Time and Distributed Systems Laboratory at Carleton University since 2001. His work there has resulted in several refereed conference and journal papers. His current research interests include large-scale distributed systems, pervasive computing and wireless and sensor Grids. Umar is a member of IEEE.

Shikharesh Majumdar is a Professor and the Director of the Real Time and Distributed Systems group at the Department of Systems and Computer Engineering at Carleton University in Ottawa, Canada. He holds an M.Sc. and a Ph.D. degree in Computational Science from University of Saskatchewan, Saskatoon, Canada. Before his graduate studies in Canada he did a Bachelor of Electronics and Telecom Engineering and a Post-Graduate Diploma in Computer Science (hardware) from Jadavpur University in India and completed the Corso Di Perfezionamento from Politecnico Di Torino in Italy. Dr. Majumdar has worked at the R&D Wing of Indian Telephone Industries (Bangalore) for six years. His research interests are in the areas of grid-based systems, operating systems, middleware and performance evaluation. He has received two awards related to publications in these areas. Dr. Majumdar is a member of ACM and IEEE. He was a Distinguished Visitor for the IEEE Computer Society from 1998 to 2001.

Eric W. Parsons is a presently a strategic advisor at Nortel, in the area of wireless and wireline networks, and is an Adjunct Professor at the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. He holds an MMath from the University of Waterloo, Waterloo, Canada, and a Ph.D. from the University of Toronto, Toronto, Canada. His research interests are in the areas of wireless systems, grid-based systems, multiprocessor systems, and performance evaluation. Dr. Parsons is a member of the ACM and IEEE.

