

Engineering Mobile Wireless Publish/Subscribe Systems for High Performance

Umar Farooq Shikharesh Majumdar Eric W. Parsons
Department of Systems and Computer Engineering, Carleton University
1125 Colonel By Drive, Ottawa, ON, Canada. K1S 5B6
{ufarooq, majumdar}@sce.carleton.ca, eric.parsons@computer.org

Abstract

Decoupling flexible, scalable and asynchronous nature of publish/subscribe systems makes them a good choice for the mobile wireless domain. Our research investigates the application of publish/subscribe middleware to mobile cellular networks with the focus on their performance in such networks. The results show 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. In order to overcome the challenges faced by publish/subscribe systems in a wireless mobile domain, this paper introduces a novel approach called semi-durable subscriptions. We present the semi-durable subscription approach and through rigorous experimentation prove its efficacy and superiority over traditional solutions.

1. Introduction

A publish/subscribe system consists of *publishers* that are the sources of information called *events* or *messages*, and *subscribers* that receive certain types of events by registering their *subscriptions* with 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 basic model of a publish/subscribe system thus depends 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, provides storage and management of subscriptions and messages. Publish/subscribe systems have numerous applications and are currently being deployed for a large variety of applications such as stock-information delivery, auction systems and air-traffic control systems.

There are many characteristics of a publish/subscribe system that make them ideal for a mobile environment. Publish/subscribe systems decouple publishers and subscribers in space, time and flow. 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 wireless networks. There is hence a pressing need to extend publish/subscribe systems to mobile wireless environments.

Although a lot of work has been done on publish/subscribe systems in the context of fixed networks [1] [3] [4] [13], there is comparatively little research on the behavior of these systems in the context of mobile networks. Due to space limitation only a representative set of related work is described. A more extensive literature survey is available in [6]. Research performed by Cugola [5] and Fenkam [9] studies mobility management in publish/subscribe systems. Their research is more focused on general requirements dealing with design issues, data definitions for publications and query language for subscriptions. The nature of their works is however broader and more theoretical than the work this paper focuses on. Huang and Garcia-Molina [11] have discussed how publish/subscribe systems can be extended to operate in a mobile environment. Although their work addresses the issues in the deployment of these systems over wireless networks, they have not given any concrete results on the behavior of these systems in such environments. Concurrent with our research, there have been researches to support mobility with SIENA [2] and REBECA [10] [15]. However, unlike other works, our research is more focused on the performance of publish/subscribe extensions to mobile domain. The goal of our research is to extend these systems to the mobile domain in a way that can ensure high performance under high frequency of handoffs and error prone wireless channels.

For our research, we have used one of the most widely accepted messaging middleware standard Java Message Service, JMS [14]. In an earlier paper [8], we have characterized a popular implementation of JMS for mobile wireless settings, studying and analyzing the effect of different mobility parameters, which to the best of our knowledge is the first experimental investigation of the

performance of publish/subscribe systems in a mobile wireless domain. In that paper, 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. The results show that such handoff protocols involving two brokers are impractical from a performance perspective under highly dynamic and unreliable mobile wireless settings. In this paper, we engineer performance into mobile wireless publish/subscribe systems through semi-durable subscriptions technique. This technique is developed to overcome the performance limitations of the implementations of existing publish/subscribe systems in the mobile domain.

Although our research is based on JMS, we believe that most of the important conclusions are equally valid for other publish/subscribe systems. The contributions of this paper are as follow:

- The paper identifies the performance limitations, in terms of low throughput or high message loss, of a popular implementation of publish/subscribe system and the traditional solutions to extend them to the mobile domain.
- We introduce a novel technique called *semi-durable subscriptions* to overcome the challenges faced by current implementations and traditional solutions in a mobile wireless environment and through rigorous experimentation prove its quality and efficacy in such settings. We study the effect of workload parameters on the performance of semi-durable subscriptions.
- Through a comparison of the performance of the semi-durable subscription approach with other alternatives the paper provides an experimental validation that it leads to a significant benefit in performance.
- Semi-durable subscriptions provide a rich system parameter space. The experimental results demonstrate how the parameters can be controlled to configure a system according to a set of desired characteristics.
- The paper discusses how the semi-durable subscription technique can be adapted for *regionalism*. Thus, we show how semi-durable subscriptions approach works when messages are not broadcast.

The rest of the paper is organized as follows. Section 2 provides an overview of publish/subscribe systems in mobile cellular networks. We discuss the experimental setup used in this research in Section 3. Performance behavior of current implementations of publish/subscribe systems and the traditional solutions to extend them to the mobile domain are presented in Section 4. Then we introduce the semi-durable subscription approach in Section 5 and present an evaluation of semi-durable

subscriptions in Section 6. In Section 7 we discuss how semi-durable subscription approach handles *regionalism*. Finally we conclude the paper and gives direction for future research in Section 8.

2. Publish/Subscribe In Mobile Cellular Networks

Scalable implementations of publish/subscribe systems often require a distributed set of interconnected brokers. In a mobile environment, a client that can act as a publisher or a subscriber or both runs on a mobile host such as a cellular phone, PDA or laptop. It connects to one of the message brokers in the system over a wireless link in the cellular network. The message broker may be located somewhere in a cellular network infrastructure or it may be at a fixed node outside the network in which case the client connects to the broker through a wireless gateway.

As a client moves from one part of the network to the other or from one network to the other, 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 the messages generated during the time it was disconnected. However, when a client acts 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 poses an additional problem of occasional message loss that is usually absent in fixed networks. Limited bandwidth in wireless networks may also prevent or delay the delivery of messages to the subscriber. The major challenges for publish/subscribe systems in mobile cellular networks are thus managing clients mobility, handling temporary disconnections and dealing with limited bandwidth environments.

3. Experimental Setup

To study the performance of publish/subscribe systems in cellular networks we have used Java Message Service, JMS [14], 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 1 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.

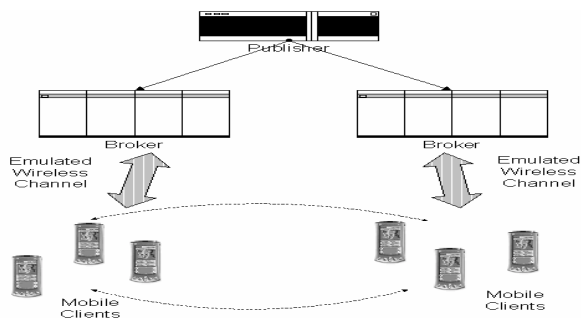


Figure 1. Logical View of Experimental Setup (Initial Number of Subscribers per Broker Fixed at 100)

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

Mobility model and mobility related parameters: For simulating subscriber mobility a program is written in Java which makes each subscriber go through a mobility model illustrated in the state diagram of Figure 2. 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.

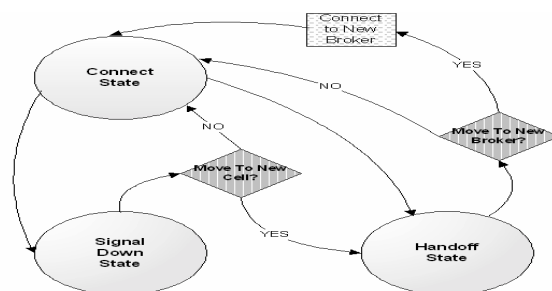


Figure 2. State Diagram of Subscriber Mobility Model

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 future cellular publish/subscribe systems, 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 [12] 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 performance scenarios such as packet delay, congestion, bandwidth limitation, packet loss, packet duplication and packet reordering. Most of our experiments were done for three mobility/bandwidth environments, namely fixed wireline, mobile with a 10MB/second total bandwidth and mobile with a 220kB/second total bandwidth. The value of 10MB/second is chosen to ensure that it is the server that becomes the bottleneck (and not the network), and represents what we term the *server bottlenecked region*. The value of 220kB/seconds is chosen to ensure that the bandwidth becomes the bottleneck, representative of what we term the *bandwidth bottlenecked region*.

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 (L) is percentage of the messages lost by the subscribers.*

4. Performance of JMS in Mobile Wireless Environments

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 3 shows the effect of mobility/bandwidth on the performance of nondurable subscriptions. These results

were obtained with $t_{CS} = 108$ sec, $t_{SD} = 12$ sec and the size of the message body equal to 50 Bytes.

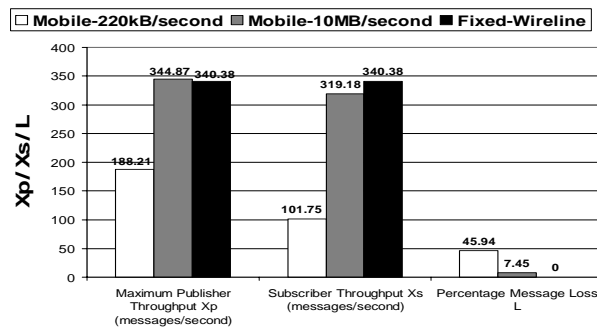


Figure 3. Performance of Nondurable Subscriptions

The figure shows that for nondurable subscriptions in the 10MB/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 [8].

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 [8]. Moreover, in a highly mobile environment, each broker would end up having a large number of inactive subscribers that may 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 [8] out of which middleware level handoff seems to be the most natural. We implemented a prototype of middleware level handoffs and our results [8] 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 Figure 4. They were obtained using the same parameters as in Figure 3. 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 Figure 3 and Figure 4 shows that throughput of durable subscriptions in mobile environments can be as low as 5% of that with nondurable subscriptions.

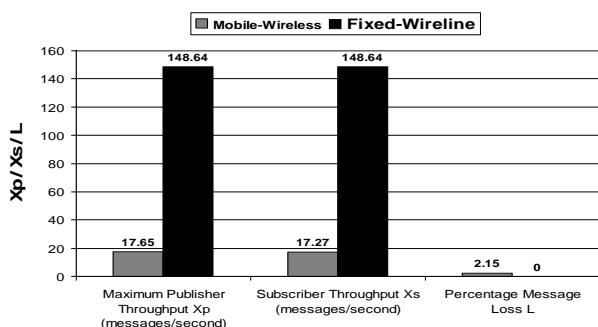


Figure 4. Performance of Durable Subscriptions

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 in their current form are not suitable to be deployed in mobile environments. Although middleware level handoffs with durable subscriptions can prevent message loss, their subscriber throughput in mobile environments, with the given range of parameters in our research, can be as low as 5% of that with nondurable

subscriptions. Such a high cost-to-benefit ratio can 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 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 any where between the two extremes.

In order to tackle the challenges mentioned above, this paper introduces 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) and hence 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 [13] 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 Section 7.

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 broker does not store messages on a subscriber by subscriber basis, its performance does not depend on the number of inactive subscribers it has.

The details of different components in semi-durable subscriptions and the interaction between them is available in [7] but is not being presented here due to space limitation. In semi-durable subscriptions every message published to the broker is cached in the buffer. *Semi-durable Buffer Manager* organizes the size and internal partitions of the buffer according to the inputs from the administrator. When the buffer gets filled the *Semi-Durable Buffer Manager* uses the *Message*

Overwriting Policy to decide which messages to overwrite.

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 not need to be uncoupled from the broker. Every subscriber keeps track of the ID of the last message it received in a log file. 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. 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* which in turn delivers them to the subscriber. 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. 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.

6. Evaluation of Semi-Durable Subscriptions

We conducted extensive sets of experiment to study the behavior of different subscription mechanisms in the mobile wireless environments. However, due to space limitation only a representative set of results are presented here. For details the interested reader is referred to [6] [7].

6.1 Comparison with Other Subscription Mechanisms

Performance of semi-durable subscriptions is compared to that of nondurable and durable subscriptions using the same parameters as in Figure 3. 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. A least recently published overwriting policy is used in which when the buffer is full and a new message arrives, the least recently published message(s) is (are) overwritten.

6.1.1. Throughput in the Server Bottlenecked Region.

The maximum sustainable throughputs of the three subscription types in the server bottlenecked are presented in Figure 5. The figure shows that the maximum sustainable publisher throughput X_P for semi-durable subscriptions is about 82.5% of that for nondurable subscriptions while the subscriber throughput X_S is about 90% of that of nondurable subscriptions. This loss in throughput capacity 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% in semi-durable subscriptions. 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. This strongly suggests that bringing the message loss, occurring in mobile wireless environments, to a negligibly small value through the use of semi-durable subscriptions is cost effective but bringing it down to zero with durable subscriptions is too expensive.

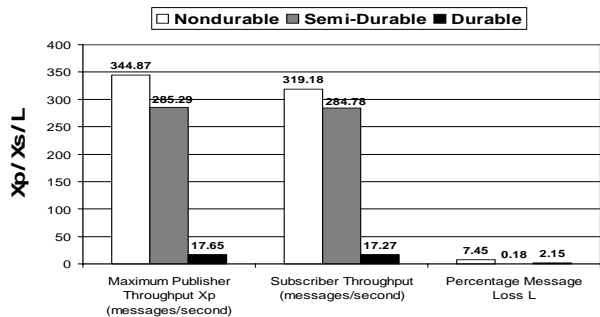


Figure 5. Comparison of the Three Subscription Types in the Server Bottlenecked Region

In our implementation a significant amount of message loss in semi-durable subscriptions is due to the *leaky* implementation where some messages are lost during the transfer of a client from the Handoff Manager to the broker. Integrating the broker and the semi-durable managers into a single entity can prevent this message loss.

6.1.2. Throughput in the Bandwidth Bottlenecked Region.

The throughputs for durable subscriptions are too low to saturate even a small bandwidth, thus the maximum sustainable throughput of semi-durable subscriptions in a bandwidth bottlenecked region has been compared to nondurable subscriptions only. Figure 6 shows that in the bandwidth bottlenecked region, the maximum sustainable publisher throughputs X_P for both nondurable and semi-durable subscriptions are almost equal. This is because the throughput is limited by the network bandwidth which is kept the same for both subscription types. However, contrary to the intuition, the subscriber throughput X_S of semi-durable subscriptions is significantly higher than that of nondurable subscriptions. The reason is that the implementation of JMS does not perform well in limited bandwidth environments and generates significant overhead traffic (For details see [8]). The messages delivered by the broker thus consume higher bandwidth than actually required, reducing the total message carrying capacity of the channels. On the other hand, the Handoff Manager in semi-durable subscriptions is designed in such a way that it does not generate such additional traffic. Moreover, the Handoff Manager sends all the stored message in the same TCP session, unlike the newly published messages which are sent message by message by the broker.

The comparison of Figure 5 and Figure 6 shows that message loss is significantly higher in limited bandwidth regions than that in the higher bandwidth regions for both nondurable and semi-durable subscriptions. The reason for higher message loss again lies in the fact that the JMS implementation does not perform well in limited bandwidth environments. Although semi-durable subscriptions significantly reduce the percentage message loss, it is still much higher than that in higher bandwidth regions. However, our detailed study indicates that almost no message is lost due to a limited buffer size or during the delivery of messages through the Handoff Manager. All the loss occurring in semi-durable subscriptions was during the period when the subscribers were connected to the broker.

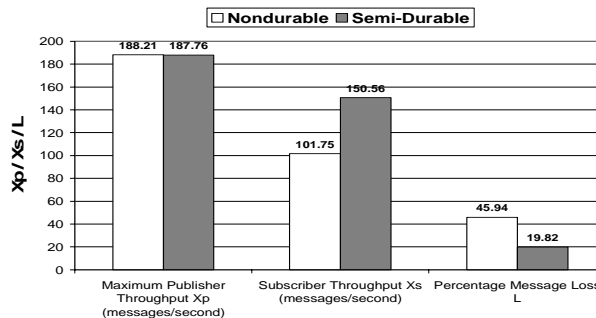


Figure 6. Comparison of Nondurable and Semi-Durable Subscriptions in the Bandwidth Bottlenecked Region

6.1.3. Comparison at a Given Publishing Rate. We also evaluated 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 show that at publishing rates equal to or lower than the maximum publishing rate semi-durable subscriptions can sustain in a server bottlenecked region with the given values of the parameters, semi-durable subscriptions give higher subscriber throughput than even that achieved with nondurable subscriptions. The results are available in [7].

6.2. Effect of Workload Parameters on the Performance of Semi-Durable Subscriptions

The effects of parameters such as message size, frequency of handoffs and variability in signal down intervals on the performance of semi-durable subscriptions were also studied in detail. For these experiments, the values of the fixed parameters were the same as in earlier sections. The results are discussed in detail in [7], but will be reviewed only briefly here due to space limitation.

6.2.1. Effect of Frequency of Handoffs. To study the effect of frequency of handoffs, f_H , the proportion of total time in signal down state T_{SD}/T is varied keeping t_{SD} constant. Here T_{SD} is the average time a subscriber spends in a signal down state during the simulation and T is the total time of the simulation. It has been shown in [8] that

$$f_H = T_{SD}/(T*t_{SD}) \quad (1)$$

The results show that frequency of handoffs has a profound effect on the performance of both nondurable and semi-durable subscriptions. When frequency of handoffs is varied, X_S is determined by factors such as connections/disconnections overheads, effective number of subscribers connected to the broker and the increased load of message acceptance and selection.

For nondurable subscriptions, the percentage message loss given in Figure 7 increases linearly with a slope of 1, with an increase in T_{SD}/T . Increase in message loss with f_H in semi-durable subscriptions is much lower compared to that in nondurable subscriptions. The results show that for the mobility parameters used in the experiments, semi-durable subscriptions give slightly lower maximum throughput than non-durable subscriptions but 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.

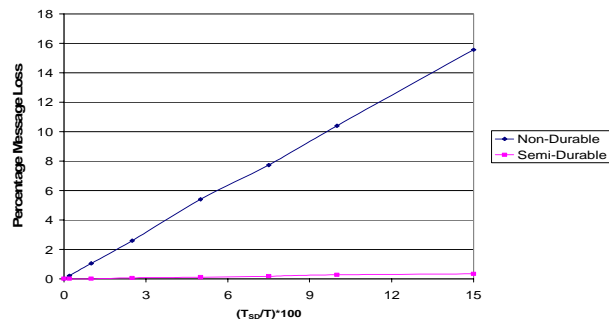


Figure 7. Effect of Frequency of Handoffs on Percentage Message Loss

6.2.2. 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 using a bi-phase hyper exponentially distributed signal down intervals. The coefficient of variation C_V of the duration of the signal down intervals is varied as an input parameter. 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 [7].

6.2.3. Effect of Message Size

To study the effect of message size, the size of the message body S is varied keeping the size of message header and properties constant. The results show that X_P and X_S decrease slightly with the increase in message size for both nondurable and semi-durable subscriptions. However, semi-durable subscriptions are less affected by increased sizes of the messages as they handle network resources better than nondurable subscriptions. The percentage message loss for both nondurable and durable subscriptions does not change with the change in message size. The results also show that the number of bits/second delivery capacity of the broker increases with the increase in message size. This suggests that if possible information should be batched together and then sent as a larger message.

6.3. Tradeoff between System Throughput and Message Loss

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. This section presents an analysis of the tradeoff between system throughput and message loss in semi-durable subscriptions using buffer size as a control parameter. The results shown in Figure 8 are obtained for

two values of t_{SD} : 12 seconds and 24 seconds. The values of other mobility parameters are the same as in the previous sections.

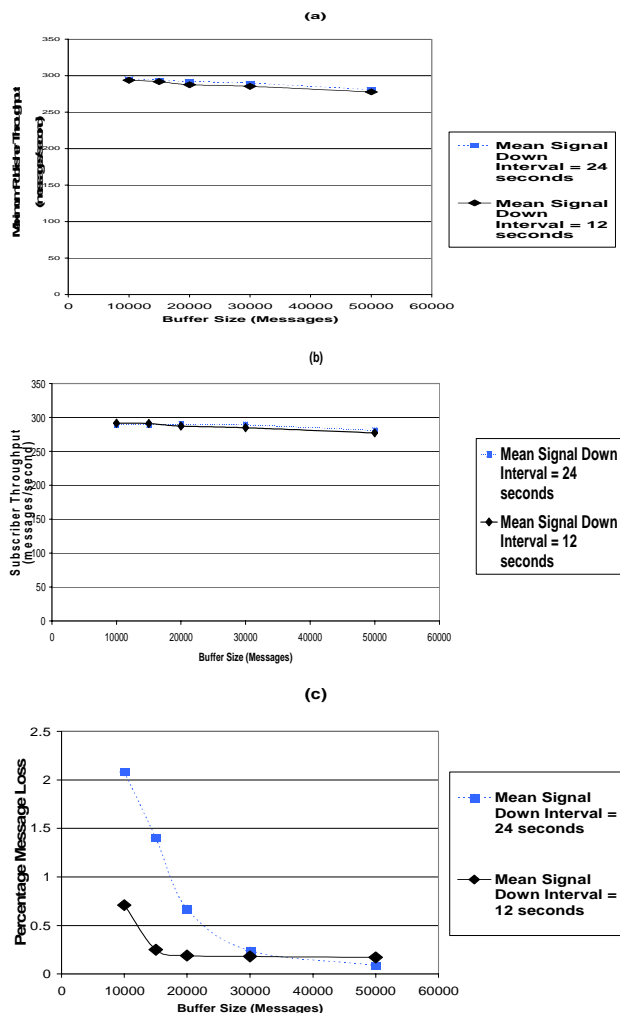


Figure 8. Tradeoff between Throughput and Message Loss (a) Maximum Throughput of Publisher (b) Throughput of Subscribers (c) Proportion of Messages Lost

Figure 8(a) shows that as the buffer size increases, the maximum sustainable publisher throughput X_P decreases. The reasons are the increased overheads of larger buffer sizes in terms of increased CPU time of delivering more messages to subscribers, frequent garbage collection due to an increased heap memory size of JVM and increased search time of the searching algorithm. However, this decrease in X_P pays off in terms of decreasing the message loss in Figure 8(c). Thus when the buffer size is increased from 10000 to approximately 16000 messages there is a significant reduction in message loss for t_{SD} equal to 12 seconds. On the other hand, for t_{SD} equal to 24 seconds the buffer size needs to be increased to

approximately 28000 messages or more to bring the message loss to a very small value. Why this message loss is not reduced to zero has already been discussed in the previous section. Thus for a given mobility environment there is a certain threshold buffer size at which the message loss is reduced to an acceptable value. Increasing the buffer size beyond the knee of the percentage message loss vs. buffer size curve does not produce any significant improvement.

Subscriber throughput X_S in Figure 8(b) does not change significantly until the threshold value of the buffer is reached. With an increase in buffer size, X_P decreases which tends to decrease X_S . But at the same time the increase in buffer size decreases message loss which increases X_S . Thus X_S more or less remains constant until the threshold value of the buffer is reached. Increasing buffer size beyond the threshold value has a negative effect on X_S as it decreases X_P without decreasing the message loss.

The tradeoff between message loss and throughputs exists before the threshold value of the buffer is reached. For example, if a higher maximum publisher throughput is more desirable than low message loss the system can use a buffer size value less than the threshold value. Using a buffer size value corresponding to the knee of the percentage message loss vs. buffer size curve is recommended: it not only brings down the message loss to a very small value but it also achieves a subscriber throughput that is close to the maximum possible value.

7. Semi-Durable Subscriptions with Regionalism

Regionalism is a phenomenon in which subscribers in different regions have different interests. As a result not all messages need to be broadcast to all brokers in the system. However, it is worth mentioning here that research such as [13] shows that due to the overhead involved in frequent computing of the multicast trees, in a mobile environment all messages should be broadcast to all the brokers. Even if the broadcasts on a global scale are too expensive, regional broadcasts can be used. If we assume that messages are not broadcast, semi-durable subscriptions can be extended to deal with regionalism. Here we show how semi-durable subscriptions can be extended for regionalism if a hierarchical topology of brokers is used. Extensions for systems using other topologies are discussed in [7].

In a hierarchical topology, the brokers are connected in a hierarchical fashion as shown in Figure 8. The advantage of using a hierarchical topology is the *grouping* of subscriptions as we move upward in the hierarchy, thus simplifying the filtering algorithm. Also as opposed to other architectures, there may be a substantial bandwidth saving on the links between the brokers because in a

hierarchical topology the message is duplicated only in the lower layers. Thus a message for which there are thousands of subscribers may not be duplicated until very low in the hierarchy. This is because the subscriptions of all the clients at each level are grouped together and thus multiple subscriptions for the same message appear as a single subscription at a higher level.

Consider a hierarchical broker topology shown in Figure 8 where each broker covers a particular region. With semi-durable subscriptions in such a topology, message buffering at the brokers in each layer of the hierarchy can be limited to active subscriptions in the region covered by the broker. The messages missed by the subscribers moving from one region to another can be recovered from the broker at a higher level in the hierarchy which covers a wider region. For example, in Figure 9, let S_E and S_F be the total subscription set of all the clients connected to Broker E and Broker F respectively which means the total subscription set of Broker C is the union set of S_E and S_F . With semi-durable subscriptions, Broker C can limit its message buffering to only those messages belonging to the difference of subscriptions sets S_E and S_F while Broker E and Broker F can limit their buffering to the messages belonging to subscription set S_E and S_F respectively. Hence as a client moves from Broker E to Broker F, if the subscriptions of the client belong to the subscription set S_F the messages missed by the subscriber can be recovered from the messages buffered at Broker F. Otherwise, they can be recovered from the messages buffered at Broker C. Similarly, Broker A would limit its message buffering to the difference of subscription sets of Broker B and Broker C. Note that with this scheme semi-durable subscriptions do not generate any overhead network traffic and at the same time maintain the loose coupling between the subscribers and the broker necessary to achieve high performance in dynamic mobile environments. Moreover, this scheme reduces the buffering overhead at the brokers by reducing the number of messages to be buffered by each broker.

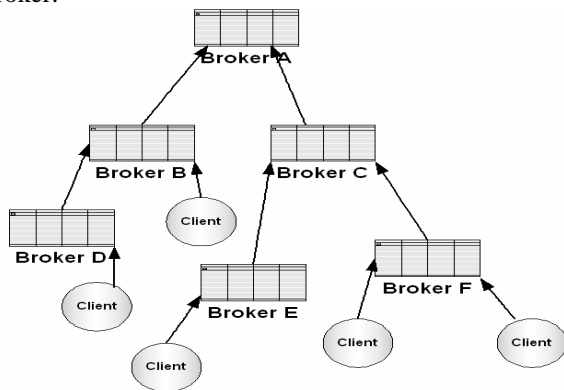


Figure 9: Hierarchical Topology of Brokers

8. Conclusions

There are many features of publish/subscribe systems that make them a good choice for the mobile wireless domain. However, a lot of work still needs to be done to achieve high performance in mobile wireless publish/subscribe systems. Although nondurable subscriptions produce almost as much throughput for mobile users as for fixed users, they suffer from message loss in bursts. Durable subscriptions can prevent message loss but give rise to extremely low throughputs. The poor performance of durable subscriptions in mobile environments can be attributed to a tight coupling between the broker and the subscriber.

To overcome the problems faced by nondurable and durable subscriptions this paper introduces semi-durable subscriptions. The semi-durable subscription approach ensures loose coupling between the broker and the subscribers and hence does not incur substantial overheads. At the same time almost all of the messages missed by the subscribers during temporary disconnections in a mobile wireless environment are recovered. It incurs a very nominal cost in terms of a small reduction in X_p of the system as compared to nondurable subscriptions. Bringing message loss to exactly zero with the use of durable subscriptions can be very expensive as the subscriber throughput, with the given range of parameters in our research, can be as low as 6% of that for semi-durable subscriptions. At all publishing rates equal to or lower than the maximum rate semi-durable subscriptions can sustain, semi-durable subscriptions give rise to higher subscriber throughput than even nondurable subscriptions. All these features make them an effective choice for mobile cellular networks.

The semi-durable subscription approach also provides a rich parameter space to configure the system. The paper demonstrates how the parameter buffer size can be controlled to exploit the inherent tradeoff in throughput and message loss with semi-durable subscriptions. We are currently investigating the effects of the internal organization of the buffer and message overwriting policies on system performance.

9. References

- [1] G. Banavar, T. D. Chandra, B. Mukherjee, J. Nagarajao, 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 (ICDCS'99)*, Austin, TX, May 1999, pp. 262-272.
- [2] 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, January 2003.
- [3] A. Carzaniga and A.L. Wolf, “Content-Based Networking: A New Communication Infrastructure,” in *Proceedings of NSF Workshop on an Infrastructure for Mobile and Wireless Systems* in conjunction with the 10th IEEE International Conference on Computer Communications and Networks (ICCCN 2000), Scottsdale, AZ, October 2001, pp. 59-68.
- [4] A. Carzaniga, D. S. Rosenblum and A. L. Wolf, “Achieving Expressiveness and Scalability in an Internet-Scale Event Notification Service,” in *Proceedings of the 19th ACM Symposium on Principles of Distributed Computing (PODC 2000)*, Portland, OR, July 2000, pp. 219-227.
- [5] 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, November 2001.
- [6] 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, August 2003.
- [7] 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, July 2003.
- [8] 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 (WOSP 2004)*, Redwood City, CA, January 2004. pp. 278-289.
- [9] 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 (WETICE’02)*, Pittsburgh, PA, June 2002, pp. 23-28.
- [10] 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 (Middleware 2003)*, Rio de Janeiro, Brazil, June 2003, pp. 103-134.
- [11] 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 (MobiDE’01)*, Santa Barbara, CA, May 2001, pp. 27-34.
- [12] National Institute of Standards and Technology, *NIST Network Emulation Tool*, <http://snad.ncsl.nist.gov/itg/nistnet/index.html>
- [13] 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 (Middleware 2000)*, New York, NY, April 2000, pp. 185-207.
- [14] Sun Microsystems Inc., *Java Message Service*, <http://java.sun.com/products/jms>
- [15] A. Zeidler and L. Fiege, “Mobility Support with REBECA,” in Proceedings of the 1st International Workshop on Mobile Computing Middleware (MCM’03) in conjunction with the 23rd International Conference on Distributed Computing Systems (ICDCS’03), Providence, RI, May 2003, pp. 354-361.