RADAR

Research Archive and Digital Asset Repository

OXFORD BROOKES UNIVERSITY

Younas, M and Awan, I (2013) Mobility management scheme for context-aware transactions in pervasive and mobile cyberspace

Younas, M and Awan, I (2013) Mobility management scheme for context-aware transactions in pervasive and mobile cyberspace. *IEEE Transactions on Industrial Electronics*, 60 (3). pp. 1108-1115.

This version is available: https://radar.brookes.ac.uk/radar/items/a20e6e84-5e82-46a5-4bde-60ae8de21dbe/1/

Available on RADAR: June 2013
Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the postprint version of the journal article. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

(c) 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

WWW.BROOKES.AC.UK/GO/RADAR

Mobility Management Scheme for Context-aware Transactions in Pervasive and Mobile Cyberspace

Muhammad Younas, *Member, IEEE,* and Irfan Awan, *Member, IEEE*

Abstract—**Rapid advances in software systems, wireless networks and embedded devices have led to the development of a pervasive and mobile cyberspace that provides an infrastructure for anywhere/anytime service provisioning in different domains such as engineering, commerce, education, and entertainment. This style of service provisioning enables users to freely move between geographical areas and to continuously access information and conduct online transactions. However, such a high mobility may cause performance and reliability problems during the execution of transactions. For example, the unavailability of sufficient bandwidth can result in failure of transactions when users move from one area (cell) to another. We present a context-aware transaction model that dynamically adapts to the users' needs and execution environments. Accordingly we develop a new mobility management scheme that ensures seamless connectivity and reliable execution of context-aware transactions during mobility of users. The proposed scheme is designed and developed using a combination of different queueing models. We conduct various experiments in order to show that the proposed scheme optimises the mobility management process and increases the throughput of context-aware transactions.**

Index Terms—**Context-aware transactions, mobility management, handover, pervasive, mobile, cyberspace**

Manuscript received April 09, 2011. Revised February 2012. Accepted April 2012.

Copyright © 2009 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org

M. Younas is with the Oxford Brookes University, Oxford OX33 1HX, United Kingdom; (e-mail: m.younas@brookes.ac.uk).

I. Awan is with the University of Bradford, Bradford BD7 1DP, United Kingdom (email: i.u.awan@bradford.ac.uk).

I. INTRODUCTION

Pervasive and mobile cyberspace provides a new open and flexible digital platform in which devices and services are context-aware, adaptive and responsive to users' needs and execution environment. Without regard to time and location it allows users to acquire a variety of services using handheld computing devices and wireless (sensor) networks. For instance, Google Mobile provides users with access to a variety of services from their mobile phones, ranging from simple web pages through to products' prices to driving directions. Similarly, various research prototypes and frameworks have been developed in order to facilitate mobile navigation, searching [1], [2] and service discovery [3], [4].

However, majority of information and services currently available to users are through read only queries such as news bulletin, weather information, and product prices. But in order to fully realize the vision of the pervasive and mobile cyberspace it is imperative to extend service provisioning beyond read only capacity and to allow for update operations (transactions) on data and services. Consider for example a scenario of car insurance services which can be made available to users (drivers and insurance staff) through mobile devices. In case of an accident or a car breakdown, drivers can use their mobile phones to make claims and request for recovery services. In order to process such requests insurance company staff have to complete a number of tasks such as finding information about the car and its driver, police reports, accident information, location of accident or breakdown, etc. In certain cases, the staff are required to physically visit the location and examine the damaged car in order to provide cost estimate for insurance claims.

We believe that transaction management (TM) technology can play a vital role in fulfilling the true vision of pervasive and mobile cyberspace as it has the potential to reliably manage information and services in terms of read only as well as update capacities. Ensuring reliability and performance are very important for different types of applications such as e-business, auctions and web-based ecosystems [22], [23]. In the above scenario, TM can be used to correctly and consistently complete the different tasks involved in the insurance claim. In TM, a transaction represents an abstract view of a sequence of operations involved in the execution of an (pervasive) application.

Existing TM models and protocols [5], [6] are limited to the classical commit procedures which do not give attention to context-awareness and mobility management. However, in the current environment transactions should be managed such that they are context-aware and adapt to the user needs and execution environment. For instance, a transaction can be successfully completed (committed) if it meets the desired context such as location, time, performance, etc. In the above scenario, a transaction for arranging a tow truck should take into account the ‘location’ context wherein a GPS facility can be used to find the nearby available tow truck and send it to the accident location.

According to Dey's and Schilit [7], context is defined as “any information that can be used to describe the situation of people, resources and services in a service oriented environment. It may include all other information that can be considered relevant to the interaction between a user and a service”. Context information can either be directly obtained from the service interface definition using the Context-Based Web Service Description Language (CWSDL) [8] or it can be obtained using external services.

Previous research work proposes context-aware transaction model for pervasive and mobile applications [9]. However, it does not consider mobility management which is one of the most important and challenging problem for context-aware transactions. In such an environment a system must provide seamless movement of users between different geographical areas while simultaneously executing context-aware transactions and accessing services without any disruption of communication. For instance, using a mobile device the insurance staff can process the claim transaction while travelling to the accident location. Such transactions are generally of long duration as they involve different systems which are distributed across pervasive and mobile cyberspace. Thus transactions may start at one site and terminate at other. It is very crucial to ensure a continuous connection during the whole session of a transaction such that it can be handed over seamlessly from one location (or cell) to another without any disruption and loss of information.

We propose an efficient scheme for the mobility management of context-aware transactions. The contributions of the proposed scheme are to: (i) provide a seamless movement of users between different geographical areas while processing context-aware transactions (ii) maximize the throughput of context-aware transactions by reducing the drop rate of transaction requests during the handover process, (iii) minimise the blocking probability of context-aware transactions during the movement of users from one area to another, (iv) improve the efficiency of the mobility management process by minimising the processing overhead.

The remainder of the paper is structured as follows: Section II illustrates the basic principles and definitions used in this work. Section III reviews the related work on context-aware transactions and mobility management schemes. Section IV describes the context-aware transaction model and the execution protocol. Section V presents the mobility management scheme. Section VI gives an evaluation of the proposed scheme and presents experimental results. Finally, Section VII concludes the paper.

II. Basic Principles and definitions

A generalized architecture for pervasive and mobile applications is shown in Fig. 1, which contains some of the major components. In such an architecture, services (e.g., insurance claim service) may reside on a fixed host (FH) or on a mobile host (MH) such as a mobile device. MH freely moves from one area to another. Base stations (BSs), controlled by a Base Station Controllers (BSCs), are capable of communicating with mobile devices through wireless networks. BSCs are in turn controlled by the Mobile Switching Centre (MSC) which is connected to the Internet. BS can be associated with home as well as foreign networks. MH is continually accessible from home network using an original address assigned to it. Foreign network is the network to which MH is attached after moving from home network. MH in foreign network is accessible through a newly generated address. Each BS covers a particular area, called a cell.

Handover or handoff happens when a user (mobile device) moves from one cell to another. In this paper we focus on horizontal (inter cell) handover process where an MH moves into an adjacent cell and thus all the connection information is transferred to the new BS. Handover process management is to maintain network connection as the mobile device moves from one location to another and also changes its access point to the wireless network. In general, handover process involves three phases [10]. In the first phase, handover process is initiated whenever a user moves or network condition changes. In the second phase, the wireless network system identifies new channels (or connections) in order to process the handover of requests. In the third and final phase, the data is delivered from the previous location (or connection) to the new location.

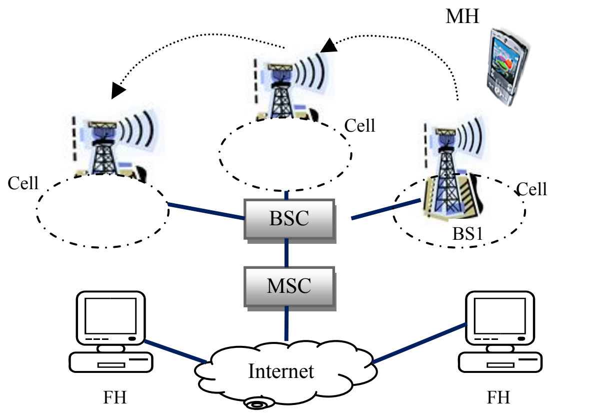

Fig. 1. A generalized architecture of pervasive and mobile computing

It is very crucial that mobile service providers provide users with continuous connections during the whole session of a transaction. Transactions are required to be handed over seamlessly from one cell to another without loss of information and service disruption. This is managed by network based handover control mechanism that redirects the transactions at an appropriate moment to the new mobile node. After the successful handover process, the user communicates through the BS in the new cell. However, some crucial requirements must be taken into account in order to design a mobility management for transactions. For instance, a transaction can be blocked if no channels are available during handover process. Blocking of ongoing transaction can have pessimistic affect on the performance of a system. A frequent handover request can cause network overhead but if the request is delayed for too long then transactions may forcefully be terminated. Thus, an efficient handover scheme is required in order to reduce the blocking of transactions but at a minimum network overhead.

One of the two major approaches to service handover requests is the *channel reservation* approach [11]. In this approach a set of channels is solely reserved to process handover requests (or transactions). For example, when insurance staff processes a claim transaction while travelling in a train that has entered into a new cell, the transaction will be handed over to the reserved channel in the destination cell for completion. This technique gives handover priority over new requests and reduces handover blocking probability. But it wastes the reserved channels in certain situations.

III. Related work

Our literature survey identifies no specific work on mobility management in context-aware transactions. This section therefore reviews work that is generally related to context-aware transactions and mobility management.

Perich et al [12] presented a transaction model for pervasive computing which is based on the concept of neighborhood-consistent. Using active witnesses and epidemic voting protocol this model ensures data consistency and provides higher throughput in terms of successful completion of transactions. This model is interesting and is also one of the first models developed for transaction management in pervasive computing. However, neither it considers context nor it addresses the issue of mobility management. Holanda et al [13] developed an intelligent transaction scheduler using a combination of conservative and aggressive concurrency control protocols. This scheduler is claimed to be context-aware in the sense that it automatically identifies changes in the computational environment and adapts to the appropriate concurrency control protocol. However, this approach is limited to the classical concurrency and commit protocols and does not take into account the context information such as location, time, and so on. Kumar et al [5] proposed a timeout-based commit protocol for mobile transactions. This protocol is a variant of the classical two-phase commit (2PC) protocol. Its objective is to improve performance and throughput of mobile transactions. 2PC protocol enforces classical ACID (atomicity, consistency, isolation, durability) criteria. But 2PC and ACID criteria are inappropriate for context-aware transactions. Lee et al [14] introduced High Commit Mobile Transactions (HiCoMo) model in order to improve the commitment rate of mobile transactions. Feilong et al [15] proposed an adaptive context transaction model for mobile and ubiquitous services. This work develops performance management model. However, unlike our approach, this work does not consider 'context' as a first class correctness property of a transaction.

In summary, the above approaches do not consider the mobility management of (context-aware) transactions. Mobility management and handover schemes are however addressed by the wireless and mobile network research community. Various queuing modelling techniques have been developed to manage mobility and handover processes. For instance, the basic queuing discipline for handover requests is based on First in First out (FIFO) scheme. In this scheme handover requests remain in the queue until channels become available or the signal of the requests drop to a very low level. In the latter situation, the requests are blocked. For example, in the insurance claim transaction if a channel is available and the users are still in the handover area, the channel is allocated to the requests with the highest priority, depending on the type of queuing techniques used. As new requests are not given service until the queue is empty, this guarantees high priorities for handover requests. This prioritisation reduces the probability of forced termination of handover requests at the expense of an increased request blocking probability [16]. In some situation queuing scheme with FIFO policy exhibits performance very close to that of the ideal prioritised handover scheme and provides a quality service [17]. However FIFO is not considered to be the most efficient service discipline to manage handover transactions as it does not take into account the movement of users from one area to another [18]. This paper, therefore, proposes an enhanced scheme for the mobility management and handover process. The novelty of the proposed scheme lies in the combination of different queuing systems.

IV. Context-aware Transaction Model

This section first presents the context-aware transaction model [9]. It then describes the execution process and handover mechanism of context-aware transactions.

A. Context-aware Transactions

Context-aware transactions can access a wide range of services which are distributed across pervasive and mobile cyberspace. In the proposed model, a context-aware transaction, denoted by CA_T, is defined as an execution of a (pervasive and mobile) application which can be divided into component service transactions, denoted by cst_i, where $(1\leq i \leq n)$. The execution of CA_T aims to acquire services (required by a user) and to maintain the correctness and consistency of those services and their data sources. Formally, CA_T is defined as a tuple, $CA_T = (cst_i, <)$; cst_i are executed to acquire the desired pervasive and mobile services (MS_i) and the context information, and $<$ is a partial ordering of the cst_i which determines the order of execution of cst_i. Each of the cst_i is a sequence of operations which is executed in order to collect service context and acquire services, i.e., $cst_i = \{operation_i, context_i\}$. For instance, in the car insurance scenario a CA_T will comprise different cst_i: e.g., cst_1 for finding accident location, cst_2 for getting police report, and cst_3 for dispatching tow truck to the accident location. cst_i may have different types such as compensatable, vital and non-vital. cst_i is compensatable if its effects can be semantically undone by executing an compensating action. All vital cst_i must be successfully executed in order for the CA_T to commit successfully. If any of the vital cst_i fails, the CA_T will be aborted. Failure of non-vital cst_i may not result in the abortion of the CA_T. Each context-aware transaction, CA_T, must maintain the following RACCD (Relaxed Atomicity, Consistency, Context, Durability) properties:

- ***Relaxed Atomicity (RA):*** It allows partial commit of CA_T in that individual cst_i may commit unilaterally and the failure of non-vital cst_i may not result in the abortion of the CA_T. Relaxed atomicity requires that all of the (vital) cst_i must be successfully executed in order for CA_T to be committed. In the above example, either all or none of the vital cst$_1$, cst$_2$, and cst$_3$ will be executed in order for CA$_T$ to maintain the RA property. Since RA allows unilateral commit, it is possible that some cst$_i$ execute successfully while others fail. In that case, it is required to compensate the effects of the executed cst$_i$ via compensation actions in order to maintain the RA property and also the consistency of data.

- ***Consistency:*** Consistency requires that the data remains consistent after the execution of CA$_T$. But with the relaxed atomicity, the consistency property can be enforced only at the component service level. The traditional notion of consistency and isolation of ACID criteria cannot be enforced in pervasive and mobile cyberspace.
- ***Context:*** It requires that CA$_T$ must fulfill the requirements of the service context. That is, CA$_T$ can be committed only if all its (vital) cst$_i$ are successfully executed and their contextual requirements are met.
- ***Durability***: It requires that effects of a committed CA$_T$ must be made permanent in the respective data sources even in the case of failures.

B. Execution of Context-aware Transactions

The main components of the proposed system are shown in Fig. 2. Users submit context-aware transactions (CA$_T$) to the *Main Coordinator* (MC). MC submits the component services transactions (cst$_i$) to the *Component Coordinators* (CC$_i$) which are associated with various *M-services (*MS$_i$*)*. Each CC$_i$ executes cst$_i$ in order to collect context information and acquire MS$_i$ services.

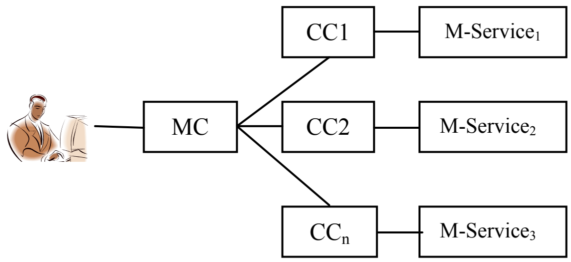

Fig. 2. The main components of the proposed system architecture

Referring to Fig. 1, we assume that MC and CC can be deployed at fixed hosts where MS_i may be deployed at fixed as well as mobile hosts. MC and CC are assumed to be deployed at fixed hosts as they are considered to be more reliable than mobile hosts. CA_T is coordinated by the MC which ensures that it complies with the rules set by the RACCD criteria. MC also maintains the execution order between $cst_i \in CA_T$. For instance, MC maintains execution order between cst_1 (finding accident location), cst_2 (getting police report) and cst_3 (dispatching tow truck to the accident location). Each CC_i executes cst_i and sends the result to the MC.

MC and each CC implement the execution protocol for context-aware transactions in order to enforce the RACCD criteria. The protocol is implemented in the following phases. In the first phase, component service transactions are executed in order to collect the context information. The second phase concerns the commit process of context-aware transactions.

Context Gathering Phase:

In this phase, the Main Coordinator, MC, and each Component Coordinator, CC, acquire the required context. The objective is to enforce the property of ‘Context’ of the RACCD criteria. Algorithm 1 is designed to collect the required context information.

ALGORITHM 1: CONTEXT GATHERING PHASE

```
$CA_T$ = {$cst_1$, $cst_2$,..., $cst_n$} // $CA_T$
with n cst
context [n];// store context-
```

```
replies from $CC_i$
count=0: //number of successful context-replies
n: number of $CC_i$ and $cst_i$
-----------------------------------------------
User: initiates $CA_T$ and submits it to the MC
MC: receives $CA_T$ request from user
MC: logs necessary information about $CA_T$
MC: collect context information from $CC_i$
  For i = 1...n
    MC: sends context-$req_i$ to $CC_i$
    CCi: process context-$req_i$ {
       get($context_i$)
         If $context_i$ = success
           send context-$rep_i$ (success) to MC
         Else if $context_i \neq$ success
           send context-$rep_i$ (error) to MC
     }
   End For
MC: receives context-$rep_i$ from $CC_i$
   For i = 1...n
```

```
      If context-rep$_i$ = success
        context[i] = success;
         count = count + 1;
      Else if context-rep$_i$ ≠ error
        context[i] = error;
        cancel CA$_T$
         terminate CA$_T$
    End For
 MC: starts commit process (phase
 2), if count = n
```

In the above algorithm, users submit CA_T to the MC which logs necessary information about CA_T and its component services transactions, cst_i. MC then starts collecting context information. e.g., if CA_T has to process an insurance claim transaction, it needs to collect context information about the insurance services. Upon receiving the request, each CC starts collecting the context information by executing cst_i. For the sake of simplicity, CC is designed such that it is capable of collecting both internal as well as external context of the required service. Each CC then sends the required context information to the MC, (e.g., sending the tow truck location). MC checks (with feedback from users) the context information received from CC. If the information received does not meet the desired context then MC has to cancel the CA_T as it cannot preserve the property of 'Context'. Otherwise, MC starts the commit phase provided no more context information is required. If further context information is required, the above steps for context gathering are repeated.

Commit Phase:

Algorithm 2 illustrates the commit process. In this process, MC contacts each CC regarding the processing of cst_i. Each CC starts processing its cst_i. If cst_i is committed (i.e., locally committed), CC sends a commit message to MC, or sends an abort message to MC, otherwise. MC checks the messages received from each CC. If all the messages received are 'commit', MC informs each CC about the commit decision. Each CC then marks its cst_i as globally committed. MC logs the commit decision of CA_T, terminates it and starts a new CA_T (if any).

If any of the messages received is 'abort', MC has to abort CA_T in order to maintain (relaxed) atomicity. MC sends abort message to each CC that has locally committed its cst_i. Each CC executes compensating action in order to cancel the effects of cst_i.

ALGORITHM 2: COMMIT PHASE

```
$CA_T$ = {$cst_1$, $cst_2$,..., $cst_n$} // $CA_T$ with n cst
vote [n];// store commit or abort vote
n: number of $CC_i$ and $cst_i$
----------------------------------------
MC: contacts each $CC_i$ to start committing $cst_i$
  For i = 1...n
   MC: sends commit-$req_i$ to $CC_i$
   $CC_i$: process commit-$req_i$ {
        If $cst_i$ = commit
           send $vote_i$(commit) to MC
        Else if $cst_i$ = abort
           send $vote_i$(abort) to MC
   }
  End For
```

```
MC: receives commit-rep$_i$ from CC$_i$
   For i = 1...n
      If commit-rep$_i$ = commit
       vote[i] = commit;
      Else if commit-rep$_i$ $\neq$ abort
        vote[i] = abort;
        MC sends abort to CC$_i$
        CC$_i$ executes compensates committed cst$_i$
        CC$_i$: marks cst$_i$ as aborted
   End For

MC: commits CA$_T$ if all the votes received are commit
   For i = 1...n
       MC: sends global-commit to CC$_i$
       CCi: marks cst$_i$ as globally-committed.
   End For
```

C. *Handover in Context-aware Transactions*

A general mobility scheme employed in context-aware transactions is depicted in Fig. 3. Assume that a user (insurance staff) starts processing CA$_T$ (e.g., a claim transaction) using his/her mobile device (or MH). As described above the processing of CA$_T$ involves different steps and generally takes longer time to complete. Thus it is possible that the user starts CA$_T$ at one cell (home network) and moves to another cell (foreign network) while CA_T is still in progress. Steps involved in such mobility and handover process are described as follows:

a. The BS1 (for home network) continuously sends messages to the user's mobile device or mobile host (MH) which is used in processing CA_T.

b. User's mobile device, MH, receives messages from BS1 and determines whether it is in the home network (BS1) or is in the foreign network (BS2).

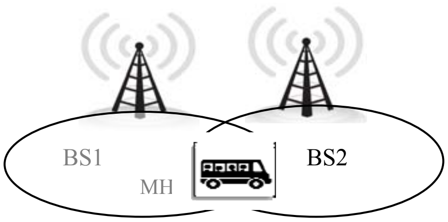

Fig. 3. Mobility during processing of CA_T

c. If MH finds that it is in the home network then the handover process will not be started. User's transaction will continue to be processed through the facilities provided by the home network.

d. If MH finds that it is in the foreign network, then it needs to get a new address in the foreign network so that it can still be able to continue processing CA_T. It also needs to inform its home network about the new address.

e. Once MH is moved from home network (BS1) and is registered with foreign network (BS2), any communication (regarding CA_T) with it will be directed to the foreign network.

V. The Mobility Management Scheme

In this section we present the proposed mobility management scheme. We first describe the classification of the requests of context-aware transactions. We then explain the queuing system developed for modeling those requests.

A. Classification of Requests

The proposed scheme is based on enhanced priority queuing mechanism. It is, therefore, necessary to devise a classification scheme that facilitates the categorisation of requests (related to CA_T) into different classes. We classify the requests into the following types:

*Handover transaction requests (HTR)****:*** These requests are part of an ongoing CA_T. That is, the requests generated by roaming users from one area (cell) to another. In the above example, when insurance staff moves from one area to another while processing the claim transaction, the requests are called handover requests.

New transaction requests (NTR): These are requests that belong to a newly created CA_T. For example, an insurance staff starts a new transaction to process the claim for another car incident.

In the proposed scheme both HTR as well as NTR have access to the (wireless network) channels which are available for the processing of transactions. Both HTR and NTR are treated equally once they occupy the channels. That is, both requests have same service rate but distinct arrival rates.

B. Modelling of HTR and NTR Requests

One of the key objectives of the proposed mobility management scheme is to improve the efficiency and throughput of context-aware transactions. In order to achieve this objective we aim to evaluate the performance of the loss (or blocking) probability of HTR and NTR. The crucial factor in such evaluation is that the HTR and NTR (traffic) rate has to be estimated accurately under realistic assumptions. In this context, we consider that the cellular traffic is bursty in nature; meaning that requests tend to arrive in bulks.

We first considered the possibility of modelling such requests using Poisson distribution [19], which is the most commonly used distribution to model the number of events occurring within a given time interval. However, a Poisson arrival process does not adequately characterise arrival traffic for cellular networks as it assumes one arrival at random time. Though it is applicable in a non-blocking environment such as telephone networks it does not fit well the growing traffic of the internet and pervasive and mobile cyberspace. In cellular networks, request blockage occurs when insufficient channels (resources) are available and when requests tend to arrive in bulks. We therefore use the Generalised Exponential (GE) Distribution [20] to model the arrival of such requests.

The GE Distribution: The GE distribution is a mixed interevent-time distribution of the form:

$$F(t) = P\,(X \leq t) = 1 - \tau e^{-\sigma\, t},\; t \geq 0, \quad (1)$$

$$\text{where, } \tau = 2/(C^2 + 1),\; \sigma = \tau\upsilon \qquad (2)$$

X: is the interevent time random variable (rv). $\{1/\upsilon, C^2\}$ are the mean and the Squared Coefficient of Variation (SCV), the ratio of the variance to the square of the mean, of the interevent time distribution X, respectively.

All distributions can have a SCV higher than 1 except the exponential distribution which has a coefficient of 1. SVC provides an important measure of the variability of the distribution.

The counting process of the GE distribution is a Compound Poisson Process (CPP) with parameter $2\upsilon/(C^2 + 1)$ and a geometrically distributed batch sizes with mean $1/\tau = (C^2 + 1)/2$ and SCV $(C^2 - 1) / (C^2 + 1)$.

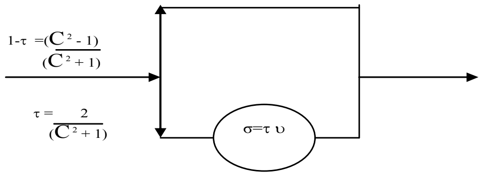
$1-\tau = \frac{(C^2 - 1)}{(C^2 + 1)}$
$\tau = \frac{2}{(C^2 + 1)}$
$\sigma = \tau\,\upsilon$

Fig. 4. The GE distribution

The GE distribution, depicted in Fig. 4, has a memory-less property like Poisson distribution. The GE distribution is considered to be the most appropriate distribution to model the burstiness of the inter-arrival times of requests characterised by various values of SCV greater than one.

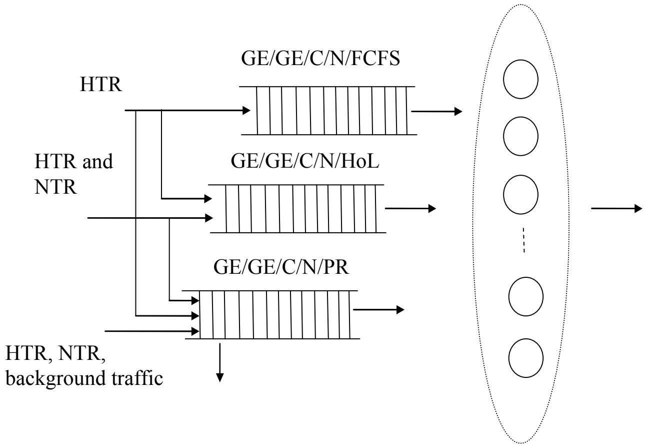

Fig. 5. The proposed modelling scheme

Based on the GE distribution, our work develops a new scheme for modelling the HTR and NTR requests. The new scheme, shown in Fig. 5, integrates different queueing modelling systems of GE/GE/C/N/FCFS, GE/GE/C/N/HoL and GE/GE/C/N/PR with Push Out priority queue. The objective of the combination of the above models is to provide a pool of communication channels for data loss sensitive applications such as context-aware transactions – wherein communication of data over the wireless networks can tolerate delay but is intolerant to any loss. This mechanism will ensure a seamless connectivity of highest priority loss sensitive context-aware transactions running on Secure Socket Layer.

In the proposed scheme, GE/GE/C/N/FCFS is used to model a dedicated single class queue that manages only the HTR requests. All the HTR requests have similar priority and thus they are served according to the FCFS discipline. The upper queue in Fig. 5 shows the HTR requests that are modelled using GE/GE/C/N/FCFS.

The lower queue in Fig. 5 models the HTR and NTR requests as well as background traffic such as email and web browsing. That is, a multiple class (GE/GE/C/N/PR) Push Out priority queue is used to handle background traffic (e.g., emails, web browsing, etc) and serve both HTR and NTR requests. It also manages HTR requests if the other two queues are full. The arriving requests will be served under pre-emptive resume (PR) service discipline. Arrival of high priority transactions will push out the background traffic in the case of traffic congestion.

The middle queue in Fig. 5 shows the HTR and NTR requests that are modelled using GE/GE/C/N/HoL queueing system [11]. As shown in Fig. 6, GE/GE/C/N/HoL is used to model a cell with HTR and NTR inter-arrival and service times at each channel having GE distribution. C represents the total number of channels available in the cell and N is the total capacity of the queue to temporarily hold the incoming HTR and NTR requests.

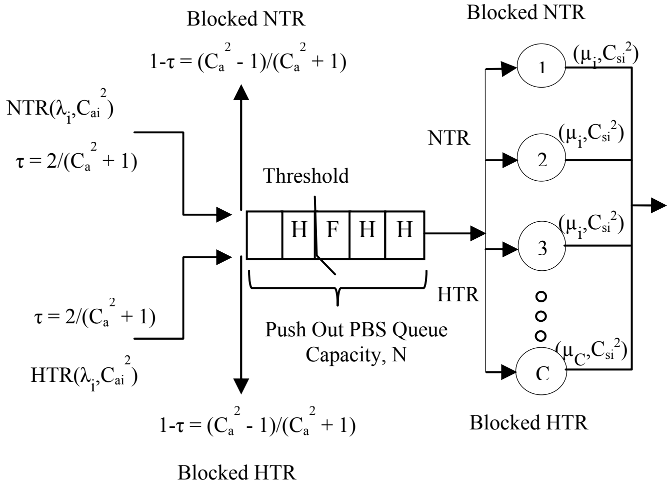

Fig. 6. Modelling HTR and NTR using Push Out Priority Queue

Free channels are assigned to the requests in the queue according to HoL scheduling discipline. This adds to the unique features of the model from the existing ones. Each cell is considered to be a queue and each channel as a server. Restricted number of requests capacity in the queue mimics the limited number of channels available due to scarce frequency spectrum availability. Let, i represents the classes of requests (i.e., HTR & NTR), λ_i and μ_i represent the mean arrival and service rate, C_{ai}^2 and C_{si}^2 are the SCVs of the inter-arrival and service times for i^{th} class, respectively. We consider that values of μ_i and C_{si}^2 for all types of requests are the same as we assume that all channels have same service rates.

When a GE arrival process with rate λ_i is sampled with probability τ, if $\tau = 2/(C_{ai}^2 + 1)$, the GE arrival rate ($\sigma = \tau\ \lambda_i$) of the request will be $2\lambda_i /(C_{ai}^2 + 1)$ and, the GE inter-arrival time is $(C_{ai}^2 + 1)/ 2\lambda_{i.}$ Likewise, for the GE service process with rate μ_i, the probability that the request will receive service is $2\mu_i/(C_{si}^2 + 1)$ and, the GE inter-service time is $(C_{si}^2 + 1)/ 2\ \mu_{i.}$

VI. Experimental results

In order to evaluate the proposed mobility management scheme we conducted various simulation experiments using QNAP-2 [21]. QNAP2 is a powerful tool for describing, handling and solving large and complex systems such as data communication networks and computer systems.

We considered various scenarios in the experimentation where several users want to execute context-aware transactions simultaneously, for example, making an insurance claim, applying for bank loans or booking travel arrangements using their mobile devices such as iPhone, PDA, etc. These users can issue context-transactions which contain two classes of requests HTR and NTR as discussed above. That is, when a transaction starts and user moves from one cell to another it will generate HTR requests. Similarly NTR requests are generated when user starts a new transaction within the same cell. We conducted three different types of experiments in order to evaluate the (i) impact of the load of HTR requests on the connection dropping probability, (ii) effect of HTR requests load on queue occupancy, and (iii) impact of HTR requests with burstiness traffic property on the connection dropping probability, queue occupancy and channel utilisation.

Experiment 1: This experiment evaluates the impact of HTR requests load on the connection dropping probability. It shows that the proposed scheme reduces the dropping probability of the (HTR) requests.

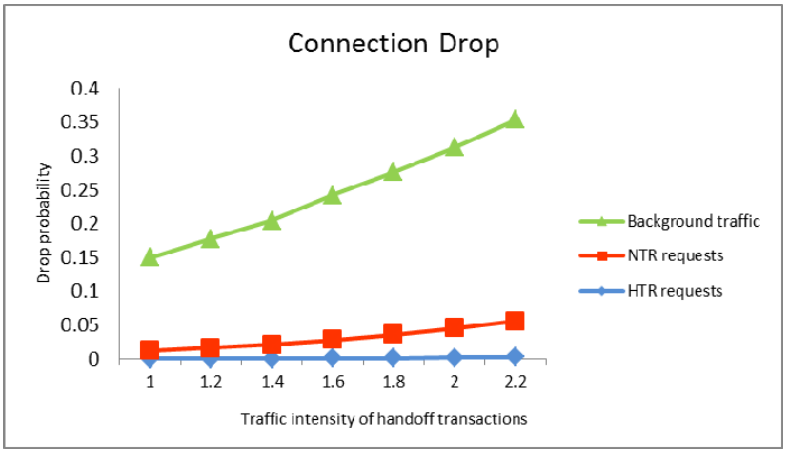

Fig. 7. Impact of non-bursty HTR requests load on the connection dropping probability

Fig. 7 shows the effect of increasing HTR requests intensity on the dropping probability when the arriving HTR requests follow Poisson distribution. The effectiveness of the proposed scheme is evident from the very minor increase in the dropping probability of the loss sensitive HTR requests as compared to the background traffic such as web browsing and other newly generated transactions, i.e., NTR requests. This is mainly due to the privileged access to the other buffers when the dedicated buffer becomes full. This rerouting mitigates the dropping effect on the HTR requests. This experiment shows that HTR requests are not dropped frequently but are serviced by the system and are provided with seamless connectivity during the migration of user from one cell to another.

Experiment 2: The effectiveness of proposed scheme is also clearly shown in Fig. 8, where mean queue occupancy for the newly generated transaction requests (i.e., NTR) and background traffic (e.g., email, web browsing, etc) are not affected when the intensity of HTR requests continuously increases. The increasing mean queue length for the handoff loss sensitive transactions show their easy access to the other buffers when the dedicated buffer is completely full.

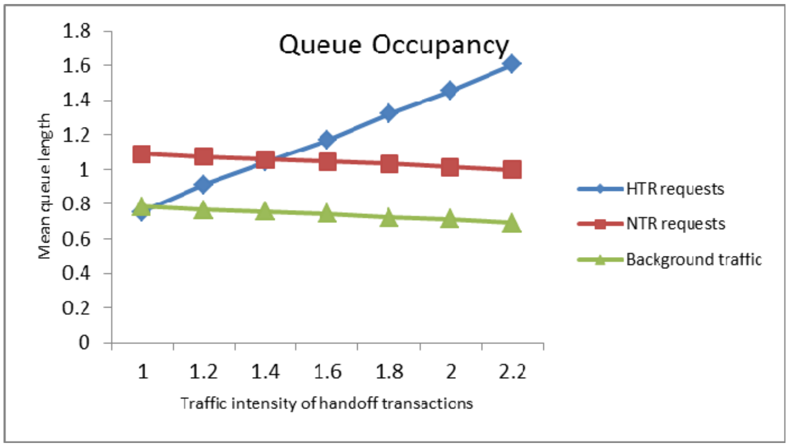

Fig. 8. Effect of HTR requests load on queue occupancy

Experiment 3: Figs. 9-11 show the impact of increasing HTR requests load on the system when the actual traffic is bursty.

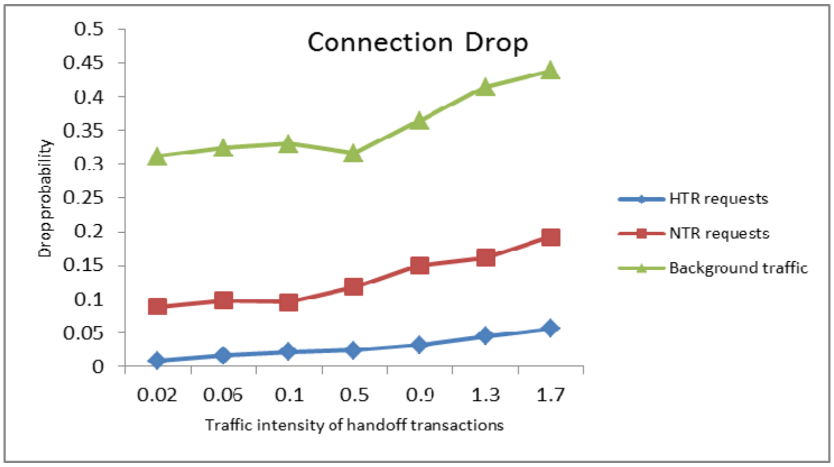

Fig. 9. Impact of HTR requests with burstiness traffic property on the connection dropping probability

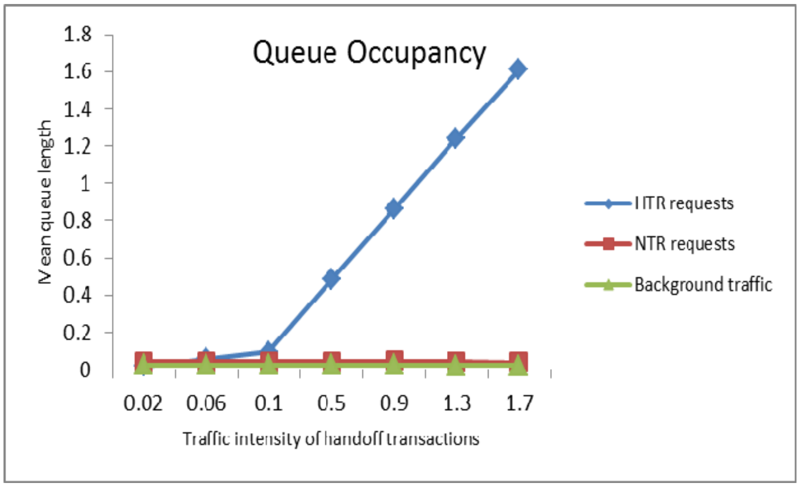

Fig. 10. Impact of bursty HTR requests on queue occupancy

Fig. 9 clearly shows better performance achieved in terms of call drops than Poisson traffic (shown in Fig. 7).

It is also interesting to see that increasing HTR load increases mean queue length (Fig. 10) and channel utilization (Fig. 11) but maintains a low call drop (Fig. 9).

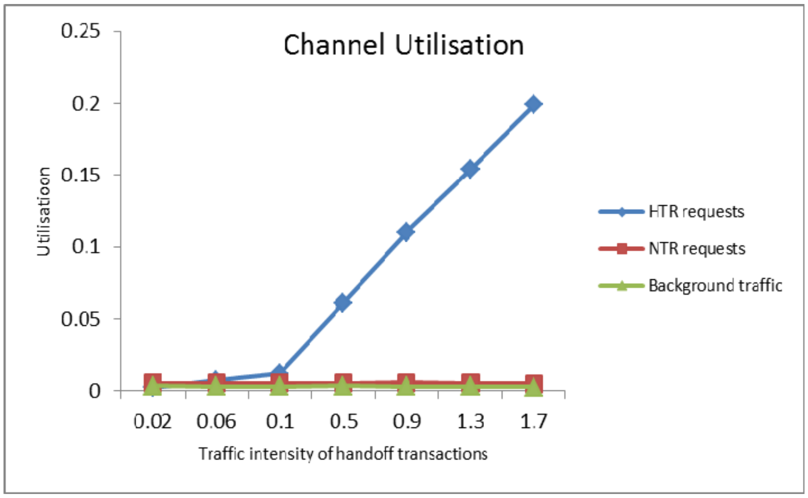

Fig. 11. Impact of bursty HTR requests on channel utilisation

In summary, all the above experiments show that the proposed scheme can be used as an effective tool for simple and bursty traffic during processing of context-aware transactions.

VII. CONCLUSION

This paper investigated into the mobility management of context-aware transactions in pervasive and mobile cyberspace. In such an environment transactions are important as they provide consistency of data and reliability of the applications in the case of communication failures or when users move from one area to another. We have developed a new scheme for the mobility management of context-aware transactions which is based on a combination of different queueing models. Such scheme provides an efficient and reliable execution environment where users can freely move from one cell to another while processing context-aware transactions. The proposed scheme reduces the dropping of context-aware transactions during handover process and also increases their throughput.

REFERENCES

[1] G. Zhao, K. Xuan, W. Rahayu, W, D. Taniar, M. Safar, M. Gavrilova, B. Srinivasan "Voronoi-Based Continuous k Nearest Neighbor Search in Mobile Navigation," *IEEE Trans. Industrial Electronics*, vol. 58. no.6, pp. 2247-2257, June 2011

[2] W. Lee, C. Leung, J. Lee "Mobile Web Navigation in Digital Ecosystems Using Rooted Directed Trees" *IEEE Trans. Industrial Electronics,* vol. 58, no. 6, pp. 2154-2162, June 2011

[3] H. Dong, F. Hussain, E. Chang "A Service Search Engine for the Industrial Digital Ecosystems," *IEEE Trans. Industrial Electronics*, vol. 58, no.6, pp. 2183-2196, June 2011

[4] H. Dong, F. Hussain "Focused Crawling for Automatic Service Discovery, Annotation and Classification in Industrial Digital Ecosystems," *IEEE Trans. Industrial Electronics*, vol.58, no.6, pp. 2106-2116, June 2011

[5] V. Kumar, N. Prabhu, M. Dunham, Y.A. Seydim "TCOT - A Timeout-based Mobile Transaction Commitment Protocol", *IEEE Trans. Computers*, vol. 5, no. 1, pp.1212-1218, 2002

[6] M. Younas, I. Awan, K-M Chao “Network-centric strategy for mobile transactions” *Interconnection Networks*, vol.5, no.3, pp. 329-350, 2004

[7] A. Dey, G.. Abowd, D. Salber, “Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications” *Human-Computer Interaction*, vol.16, pp. 97–166, Dec. 2001

[8] S. Kouadri, Z. Maamar, and N.C Narendra ‘Mobile middleware for context-aware service composition’, *Mobile Middleware*, Boca Raton, September 2006.

[9] M. Younas, S. K. Mostéfaoui "A New Model for Context-Aware Transactions" *Journal of Personal and Ubiquitous Computing,* Springer, vol.15, no. 8, pp. 821-831, 2011

[10]I. Akyildiz, J. Ho, and W. Wang, “Mobility management in next-generation wireless systems,” *Proceedings of the IEEE*, vol.87, no.8, pp. 1347-1384, Aug. 1999

[11]I. Awan, S. Singh "Performance evaluation of e-commerce requests in wireless cellular networks" *Information and Software Technology*, vol. 48, no. 6, pp.393-401, 2006

[12]F. Perich, A. Joshi, Y. Yesha, T. Finin “Neighborhood-Consistent Transaction Management for Pervasive Computing Environments” in *Proc. of 14th Int. Conf. on Database and Expert Systems Applications*, LNCS 2736, 2003, pp. 276–286.

[13]M. Holanda, A. Brayner, S. Fialho “Introducing Self-Adaptability into Transaction Processing” *ACM SAC’08*, March 16-20, 2008, Fortaleza, Ceará, Brazil

[14]M. Lee, S. Helal “HiCoMo: High Commit Mobile Transactions” *Distributed and Parallel Databases*, vol.11, no.1, pp. 73–92, 2002

[15]F. Tang, M. Guo, M. Li, I. You “An Adaptive Context-Aware Transaction Model for Mobile and Ubiquitous Computing” *Computing and Informatics*. vol. 27, pp. 785–798, 2008

[16]S. Tekinay, B. Jabbari “An effective prioritization scheme for handovers in cellular networks”, in *Proc. of the 1st Int. Conf. Universal Personal Communications*, September 1992, Dallas, Texas, USA

[17]R. Fantacci **“**Performance evaluation of prioritized handoff schemes in mobile cellular networks*”*, *IEEE Transactions on Vehicular Technology*, Vol. 49, No. 2, 2000, pp. 485–493

[18]A. E. Xhafa, O. K. Tonguz "*Dynamic Priority Queueing of Handover Calls in Wireless Networks: An Analytical Framework*" *IEEE Journal on Selected Areas in Communications*, vol. 22, no.5, pp.904-916, 2004

[19]A. H. Ahmed **"**Characterization of beta, binomial, and Poisson distributions*", IEEE Trans. Reliability*, vol.40,no.3, pp.290-295, 1991

[20]I. Awan, D. D. Kouvatsos "Entropy maximization and open queuing networks with priorities and blocking", *Performance Evaluation,* 2003, vol. 51, no. 2-4, pp.191- 227

[21]M. Veran, D. Potier "QNAP -2: a portable environment for queuing systems modeling" *in D. Potier (Ed.), Modelling Techniques and Tools for Performance Analysis*, North HoLland, 1985, pp.25-63.

[22]W. W. K. Lin, J. H. K. Wong, A. K. Y. Wong "A Novel Real-Time Traffic Sensing (RTS) Model to Improve the Performance of Web-Based Industrial Ecosystems" *IEEE Trans. Industrial Electronics*, vol. 58, no. 6, pp.2147-2153, June 2011

[23]A. Waluyo, W. Rahayu, D. Taniar, B. Srinivasan, "A Novel Structure and Access Mechanism for Mobile Broadcast Data in Digital Ecosystems" *IEEE Trans. Industrial Electronics*, vol.58, vo.6, pp. 2173-2182, June 2011

Muhammad Younas (M'02) received the PhD degree in Computer Science from University of Sheffield, UK, in 2001. He is currently a Senior Lecturer in Computing at Oxford Brookes University, Oxford, UK. He has published more than 100 papers in international journals and conferences. He is on the editorial and advisory boards of international journals. He is also involved in the steering, organizing and program committees of refereed international conferences and workshops.

Irfan Awan (M'02) is a Professor of Computer Science in the Department of Computing, University of Bradford UK. He received his PhD from University of Bradford, UK in 1997 and joined the Department of Computing in the same university in 1999 as a Lecturer. He has published over 200 papers in prestigious international journals and refereed conference. His research interests include performance modelling and evaluations of networks and communication systems.