A Platform for Device and Computation Management

R. B. Patel
Dept of Computer Engineering
M. M. Engineering College, Mullana-133203
Haryana, INDIA

Nikos Mastorakis
Dept of Computer Science
Military Inst. of University Education / Hellenic Naval Academy
Terma Hatzikyriakou 18539, Piraeus, GREECE

Kuldeep Singh
Dept of Computer Engineering
Harayana Engineering. College, Jagadhari,
Haryana, INDIA

Abstract: Mobile computing extends the horizons of conventional computing model to a ubiquitous computing environment that serves users at anytime, anywhere. Most distributed applications and services were designed with the assumption that the terminals were powerful, stationary and connected to fixed networks. One of the biggest challenges in future application development is device heterogeneity. In the future, we expect to see a rich variety of computing devices that can run applications. These devices have different capabilities in processors, memory, networking, screen sizes, input methods, and software libraries. We also expect that future users are likely to own many types of devices. Depending on users changing situations and environments, they may choose to switch from one type of device to another that brings the best combination of application functionality and device mobility. Applications, middleware, and systems can be measured in a variety of dimensions, including usability, distributability, integration, conformance to standards, extensibility, internationalizability, manageability, performance, portability, scalability reliability- fault tolerance and security. We call these pervasive attributes, since they can apply to the system as a whole, not just to the system's components. In this paper we have designed and implemented a secure and reliable application framework called the Platform for Device and Computation Management (PDCM) system that can both assist developers to build multi-platform applications that can run on heterogeneous devices and allow a user to move/migrate a running application among heterogeneous devices.

Keywords: Mobile Computing, Mobile Host, Base Host, PDCM, Ad hoc network.

1 Introduction

Mobile computing extends the horizons of conventional computing model to a ubiquitous computing environment that serves users at anytime, anywhere [20]. The availability of lightweight, portable computers and wireless technologies has created a new class of applications called mobile applications. These applications often run on scarce resource platforms such as personal digital assistants, notebooks, and mobile phones, each of which have limited CPU power, memory, and battery life. They are usually connected to wireless links which are characterized by lower bandwidths, higher error rates, and more frequent disconnections.

A user also expect to choose to switch computation from one type of device to another in the middle of using an application, in order to access necessary application functionality or to become more mobile, based on changing situations, environments or needs. For example, a user starts planning a vacation online using a desktop computer in his/her office. In the middle of planning, he/she receives an urgent call and must leave the office for a meeting at a remote site. The user would like to continue planning the trip on the bus or during break time between meetings. Given the need for mobility, he/she switches to a lightweight, mobile devices (PDA or laptops) away from the office [7]. Based on this scenario, we believe that when mobile users switch devices, there is a need to allow them to move any running application effortlessly between devices.

There exists a plethora of application scenarios where wireless access to heterogeneous information sources would be of great value. Law enforcement, access to medical information from an ambulance, off shore drilling (Goa, Chennai, etc.) scientific fieldwork, and emergency crisis management such as natural disaster-Tsunami, Sea Storm. Take, for instance, the crisis management involved during a hurricane event like tsunami. Before, during, and after a hurricane hits a region, the crisis management personnel need fast and reliable access to a wide range of information sources.

Most distributed applications and services were designed with the assumption that the terminals were powerful, stationary and connected to fixed networks. Conventional middleware technologies thus have focused on masking out the problems of heterogeneity and distribution [16] to facilitate the development of distributed systems. They allow the application developers to focus on application functionality rather than on dealing explicitly with distribution issues. Different middleware systems such as CORBA [18], and Java RMI [19] have proved their suitability for standard applications- client/server and distributed computing.

However, under the highly variable computing environment conditions that characterize mobile platforms, it is believed that existing traditional middleware systems are not capable of providing adequate support for the mobile wireless computing environment. There is a great demand for designing a new integrated software framework that can support new requirements imposed by mobility. In this paper we discuss architecture of such a system called A Platform for Device and Computation Management (PDCM) under development at Center for Advanced Computing, M. M. Engineering College, Mullana, Ambala, Haryana, India. The PDCM addresses most of the problems in the literature.

Rest of the paper is organized as follows: Section 2 explores limitations of the mobile computing systems available. Section 3 describes requirements of mobile computing systems. Design issues for mobile computing systems are discussed in Section 4. Section 5 elaborates PDCM architecture. Implementation and performance study of the PDCM is presented in Section 6. Related works are given in Section 7 and conclusion and future works are described in Section 8.

2 Limitations of Mobile Computing Systems

There are mainly three common factors that affect the design of the software infrastructure required for mobile computing: mobile hosts (MHs), network connection, and mobility. MH varies from one to another in term of resource availability. Devices like laptops can offer fast CPUs and large amount of RAM and disk space while others like pocket PCs and phones usually have scarce resources. It is either impossible or too expensive to augment the resource availability. Hence, software framework should be designed to achieve optimal resource utilization. Limited bandwidth, high error rate, higher cost, and frequent disconnections due to power limitations, available spectrum, and mobility characterize network connection in mobile scenarios. Many wireless and mobile networks such as Wireless local area network (WLAN) are organized into geographically defined cells, with a control point called a base host (BH) [17] in each of the cells. Hosts within the same cell share the network bandwidth, means bandwidth rapidly decreases whenever a new host joins the cell. MHs may move around different areas with no coverage or high interference that cause a sudden drop in network bandwidth or a loss of connection entirely. Unpredictable disconnection is also a common issue that frequently occurs due to the handoff process or shadowed areas. Most wireless network services charge a flat fee for their service, which usually covers a fixed number of messages. Additional charges are levied on per packet or per message basis. In contrast, the cost for sending data over cellular is based on connection time instead. This forces mobile users to connect for short period of time.

Physical host mobility can greatly affect network connection, which accordingly has to adapt to user mobility by reconnecting the user with respect to a new location. MHs may interact with different types of networks, services, and security policies as they move from one area to another. This requires applications to behave differently to cope with dynamic changes of the environment parameters. Due to these limitations, conventional software framework technologies designed for fixed distributed systems are not prepared to support mobile systems. They target a static execution platform where the host location is fixed, the network bandwidth does not fluctuate, and services are well defined. We next identify a number of important requirements that must be provided by software framework for mobile computing.

3 Requirements for Mobile Computing Systems

During the system lifetime, the application behavior may need to be altered due to dynamic changes in infrastructure facilities, such as the availability of particular services.

1. Dynamic reconfiguration is thus required and can be achieved by adding a new behavior or changing an existing one at system runtime. Dynamic changes in system behavior and operating context at runtime can trigger re-evaluation and reallocation of resources. Infrastructure supporting dynamic reconfiguration needs to detect changes in available resources and either reallocate resources, or notify the application to adapt to the changes.
2. Adaptivity is also part of the new requirements that allows applications to run efficiently and predictably under a broader range of conditions. Through adaptation a system can adapt its behavior instead of providing a uniform interface in all situations. The infrastructure needs to monitor the resource supply/demand, compute adaptation decisions, and notify applications about changes.
3. Asynchronous interaction tackles the problems of high latency and disconnected operations that can arise with other interaction models. A client using asynchronous communication primitives issues a request and continues operating and then collects the result at any appropriate time. The client and server components do not need to be running concurrently to communicate with each other. A client may issue a request for a service, disconnect from the network, and collect the result later on. This type of interaction style reduces the network bandwidth consumption, achieves decoupling of client and server, and elevates system scalability.
4. Context–awareness is an important requirement to build an effective and efficient adaptive system. The context of a mobile unit is usually determined by its current location which in turn defines the environment where the computation associated with the unit is performed. The context may include device characteristics, user's activities, services, as well as other resources of the system. Context-awareness is used by several systems; however, few systems sense execution context other than location. The system performance can be increased when execution context is disclosed to the upper layer that assists middleware in making the right decision.
5. Lightweight middleware needs to be considered when constructing software framework for mobile computing. Current middleware platforms like CORBA are too heavy to run on devices with limited resources. By default, they contain a wide range of optional features and all possible functionalities, many of which will be unused by most applications. For example, invoking a method on a remote object involves only client side functionality and either Dynamic or Static Invocation Interface. Most of the existing ORB implementations provide either a single or two separate libraries for the client and server sides that contains all functionality. This forces the client program to be glued with the entire functionality without having a choice to select a specific subset of this functionality.

4 Design Issues for Mobile Computing Systems

Mobile computing technology has produced a variety of devices ranging from simple pager to palmtops and laptops. When combined with wireless technology, they operate as a part of distributed system. The issues in mobile computing are different from conventional distributed systems due to the unique characteristics of the mobile device and wireless technology [15]. When it is required to shift applications from one mobile device to another in heterogeneous network then the task application must be adaptive and reliable or computing system should provide reliability to the application. If we consider reliability for an application in heterogeneous network several issues arise which are as under:

1. If a mobile application needs information on the current status for internal logic adaptation, how does the infrastructure convey such information? It is desirable for the infrastructure to expose a set of generic interface for status subscription and query.
2. The mobile application needs to specify the requirements for infrastructure services, such as the type of service (ToS), and the quality of service (QoS). Again, a comprehensive set of abstracted interface needs to identify for negotiating the required infrastructure services.
3. How is the mobile infrastructure knows under what situation should it carry out adaptation? What are the combinations of parameters for infrastructure services that should be chosen to best adapt to the current status [11,12], while fulfilling the requirements of the mobile application?

4. Data Management issues–where to keep the data, when to transmit it, whether to use caching and where in the network, how to optimize data placement, pull/push approach, transactional services, location-independent queries and how the network recognize the stage of disconnection of mobile applications. Another issue is which applications can be implemented and used effectively regarding the often-narrow bandwidth, limited computing power, small memory and battery capacity and other resources of the terminals?

To address these issues, we have identified several requirements that PDCM system supports:

- A transparent view of the user's dynamically changing computing and communication environment.
- A stack of protocols are integrated in it to support mobile computing and computation. The protocols support interoperation between many kinds of infrastructures (e.g. wire-line, wireless).
- Ability to deal with unpredictability of user behavior, network capability and computing platforms.
- A fault tolerance mechanism for providing reliability to the opened channels between two communicating parties (viz., client-server, server-server or client-client).
- Scaling regarding the heterogeneity, address space, quality of service (QoS), bandwidth, geographical dimensions, number of users, etc.
- Integrated and ad-hoc access to services.
- Maximum independence between the network and the application from both the user's viewpoint as well as from the development viewpoint.
- Ability to match the nature of what is transmitted to the bandwidth availability (i.e., compression, approximation, partial information, etc).
- Cooperation among system elements such as sensors, actuators, devices, networks, operating system, file system, middleware, services and applications.

Thus, there are several components which are identified for developing a secure, and reliable mobile computing infrastructure that should support safe heterogeneity management of mobile devices on open networks. We have designed and implemented PDCM in keeping above issues and components in mind which is discussed in the next section.

5 Architecture of PDCM

In a mobile computing environment where mobile applications suffer from the limitation and variation of system resources availability, it is desirable for the applications to adapt their behaviors to resource limitations and variations. In order for mobile applications to operate efficiently in a mobile environment, mobile computing infrastructure should provide solutions to the earlier discussed requirements and issues.

The general architecture for mobile computing environment is shown in Fig. 1. It is assumed that mobile computing system consists any number of MHs connected through one or more BH (called Mobile Service Stations (MSSs)) also known as servers, over some wireless network like infrared. MH will often be disconnected for prolonged period of time due to the low power of battery or unreachable of signal but they will also frequently reallocate between different BHs at different time. Mobile computing environment no longer requires users to maintain a fixed and universally known position in the network and enables unrestricted mobility of the MHs. There may be any numbers of BHs, which communicate through the existing wireless Ad-hoc network infrastructure. It is suggested that there should be a BH in between Internet and other network for providing mobility to the heterogeneous mobile device among heterogeneous networks.

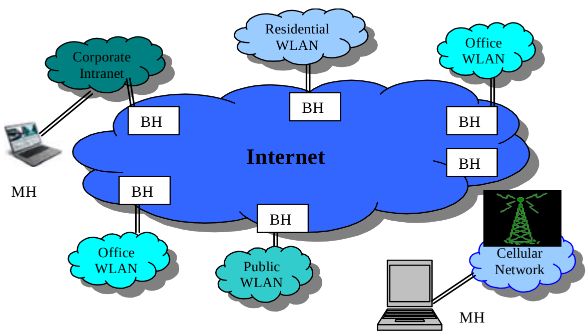

Fig. 1. General Architecture of Mobile Computing Environment

We have designed PDCM for the BHs for secure and fault tolerance mobile computing is discussed next.

Modular organization of different functionalities provided by the PDCM is shown in Fig. 2.

PDCM serves users at anytime, anywhere. It is the integration of independent modules, and the synergy provided by a well established, unified baseline architecture that promotes the development of efficient, secure and fault tolerance mobile computing applications. PDCM consists of the manager modules and the Kernel. The Kernel is the basic utility module, which lies below the manager modules and is responsible for driving the PDCM, by ensuring proper co-ordination between the various managers and making them work in tandem.

The secure communication pipe provides a hosting facility to the carrier agent which regularly monitors requests issued by the Kernel of the local host or from remote to receive the request, to transfer requests to/from other PDCMs. Various manager modules help to perform functions like- execution to mobile code, communication, mobility, name services, adaptation, fault tolerance, etc. These managers are discussed next.

Architecture is described as below: when any MH want to communicate with some BH for the fulfilling of the requirement for some required resources (viz., data file) which may be available on that BH. To make this type of communication user friendly User Interface (UI) is provided which run on the server machine and enables the mobile applications to communicate with the BH. UI is platform independent and adapts variety of MHs displays. To establish the communication link among the MHs and BHs UI calls the Communication Manager (CM), which manages the data exchange between the different components of the PDCM. Usually, the MHs use different types of network connections to access the necessary information. The major task of CM is to decide which network component will perform desired task efficiently at lower cost. CM also supports the adaptivity of the system in case of bandwidth fluctuation.

Mobility Manager (MM) performs two types of task Location Management (LM) and Handoff Management (HM) of MHs. LM and HM modules manage the two type of roaming for mobile hosts (MHs) in wireless system: intra-system (intra-domain) and intersystem (inter-domain) roaming. Intra-system roaming refers to moving between different cells of the same system. Intra-system mobility management techniques are based on similar network interfaces and protocols. Inter-system roaming refers to moving between different backbones, protocols, technologies, or service providers. Based on intra- or inter-system roaming, the corresponding LM and HM modules can be further classified into intra- and inter-system location module and handoff module (IILM and IIHM): LM enables the system to track the locations of MHs between consecutive communications. It includes two major tasks. The first is location registration/location update, where the MH periodically informs the system to update relevant location databases with its up to date location information. The second is call delivery, where the system determines the current location of the MH based on the information available at the system databases when a communication for the MH is initiated. For inter-system roaming, the design of LM techniques has the following objectives: (1) Reduction of latency of service delivery. (2) Quality of service (QoS) guarantees in different systems. (3) When the service areas of heterogeneous wireless networks are fully overlapped:(a) through which networks an MH should perform location registrations. (b) In which networks and how the up-to-date user location information should be stored. (c) How the exact location of an MH would be determined within a specific time constraint. General architecture of MM comprises of Mobile Client Managing Agent (MCMA), Virtual Single Account Agent (VSAA) and Secure Mobility Gateway Agent (SMGA) which are discussed next.

The MCMA is mainly responsible for creating and maintaining a mobile IPsec tunnel between the user's system and the corporate network over the best available wireless network. It interacts directly with, and controls, the available wireless, interfaces and Modems.

The VSAA provides several functions. It stores every authentication credential used to access wireless networks and the intranet. It also serves as a back-end authentication server for the SMGA and provides an interface for system administrators to manage each user's access rights and authentication credentials. It also provides authentication-credential-updating services to the MCMAs. The VSAA stores access credentials in a VSAA record, which contains a user's single sign on VSAA certificate, an intranet profile, a cellular profile, and several WLAN profiles. Multiple profiles are needed because we assume the user will need to use various networks that can be managed by different entities, and that will typically have different configuration parameters. The intranet profile contains the user's authentication credential for accessing the corporate intranet; the cellular profile contains the commands and parameters needed to establish a cellular data connection. A WLAN profile contains configuration parameters, access parameters, and an authentication credential.

The SMGA is a special IPsec gateway deployed between the public Internet and the corporate intranet. It authenticates a user's system with the VSAA's help, tracks the system's location with the MCMA's help, and relays IP packets between the Mobile device and other IP nodes. The IP packets transmitted between the SMGA and the Mobile device's current location are encrypted and encapsulated.

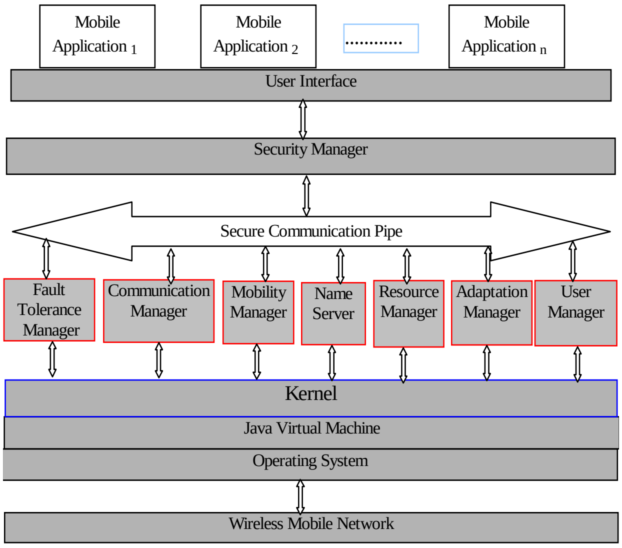

Fig. 2. Architecture of PDCM

The HM keeps MH connection active when it moves from one access point to another. The handoff process can be intra or inter-system. Intra-system handoff is the handoff in homogeneous networks. The need for intra-system handoff (or horizontal handoff) arises when the signal strength of the serving BH deteriorates below a certain threshold value. The need for inter-system handoff (or vertical handoff) between heterogeneous networks may arise in the following scenarios: (1) When a user is moving out of the serving network and will enter another overlaying network shortly. (2) When a MH is connected to a particular network, but chooses to be handed-off to the underlying network for its future service needs. (3) When distributing the overall network load among different systems are needed. At this level Resource Manager (RM) manages the resources of the system (viz., system memory used by MH, files, etc.), and also controls the admission of mobile applications that subscribe the services. Main duties include: authentication, Type of Services (ToS) and QoS negotiation, service subscription and un-subscription. To simplify the functionalities RM is subdivided into two another modules Resource Allocator (RA) and Replication Controller (RC). RA allocates and controls the resources for newly subscribed services. Resources are allocated fairly among mobile applications, at the same time fulfilling individual application's requirement. RC keeps the global state of the distributed resources consistent among all local resources based on a given coherence strategy. We have assumed *n* number of mobile application (MHs), and the coordination among all the applications are managed by a User Manager (UM), which collaborates with Name Server (NS) [17] because names are symbolic ways of referring to objects across a network. Names are detached from their corresponding objects: one may possess a name without having immediate access to any object of that name. To enable mobility and disconnected operation, all objects across a wireless network should be denoted by unique names. Name Server manages the process of naming and identification. During this process there is very much risk of security and inconsistency of the data due disconnection and security. Security, trust and privacy must be addressed from the very beginning of system design and on all levels such as hardware, operating system, protocols, and architecture. So the Security Manager (SM) is provided which has the duties of (1) Protecting BH against unauthorized modifications, (2) Program validation/verification (what an uploaded/downloaded piece of software really does), Trust modeling, (3) How fragments of information can be efficiently shared in a controlled manner, Key/certificate management, and (4) Implications of ad-hoc communities (what can be done without trusted servers). Fault Tolerance Manager (FTM) is implemented for recovering the loss of information during disconnection while performing the mobile computations. It also provides fault tolerance to the communication channels on failure. The execution environment for both public architecture services (one set of services that are shared among all mobile application) and application specific architecture services (each mobile application has its own set of services) is provided by the core module of the architecture, which is known as kernel with the help of different managers running just above.

To achieve the secure and fault tolerance mobile computing, it is redundant and inefficient for mobile applications to maintain the required resource availability independently. It is also not appropriate for application developers to check the status of all related applications exactly. Instead, a more generalized and abstracted description of the current application would be adequate for a mobile application to work effectively. Another problem is how mobile applications utilize the underlying system service to adapt to the current status of the system. This is a challenging task for application developers since mobile applications have to implement their own adaptation mechanism to the system level. It is also necessary to exploit optimal application performance. However, adaptation mechanisms by mobile applications usually suffers from the problem of unfairness to other applications, in contrast, adaptation by the operating system focuses more on the overall system performance, while neglecting the needs of individual applications. Hence, the adaptation task is best coordinated by a Adaptation Manager in the PDCM that is able to cater for individual application's need on a fair ground, while maintaining optimal system performance. This is achieved by the PDCM that sits in between the mobile application and the operating environment.

6 Implementation and Performance Study

The performance indices chosen for studying the handoff latency for the PDCM are detection time, address configuration time, registration time and packet forwarding time. Detection Time – the time between handoff occurring and the MH receiving a router advertisement (RA) from the new network access point (AP). Address Configuration Time – the time between the RA and getting a care of address (COA) from the new AP, Registration time – the time from getting a COA to registering the new COA with the home agent (HA) and corresponding hosts and getting the binding acknowledgements and Packet Forwarding Time – is the time from the last binding acknowledgement to the first data packet from the corresponding host. We have used two methods to trigger handoff (1) by varying access point power levels and (2) by alternating MH's service set identifier (SSID) association.

We have taken 1000 handoff distribution samples shown in Fig. 3, each case recorded with MH switching simultaneously between two APs, method 2 results in a lower mean handoff value (631 ms compared to 825 ms shown in Table 1). MHs keep a list of all recent RAs from all nearby subnets in a cache known as the router advertisement cache (RAC). It also records signal strength, time, etc of the RAC. When the signal strength of the current attachment falls below a predetermined threshold, the MH looks into the RAC and chooses the best subnet for handoff. The MH does not have to wait for a new RA after handoff, so handoff detection time almost zero.

802.11b handoff deals with all 3 phases - detection, search and execution, search phase was the most significant contributor to the handoff latency. The type of wireless card firmware can have a large impact. Our measured handoff is generally longer than reported in other literature. We measure the entire time it takes for actual Ethernet level bridging to successfully resume after re-association with a new (or previous) AP.

Table 1. Results for method-2 both AP on same or different channels

	Mean 802.11b handoff (ms)	Longest handoff (ms)	Shortest handoff (ms)
Both AP1 and AP2 on channel 10	631	781	506
AP 1 on channel 4 and AP 2 on 10	667	825	506

Table 2. Handoff including link layer handoff

	Average handoff from home to foreign network (sec)	Average handoff from foreign back to home network (sec)
BH is PDCM controlled (RA interval: 30 ms – 70ms)	4.75	3.638
BH is not PDCM controlled (RA interval: 30ms – 70ms)	4.770	3.779

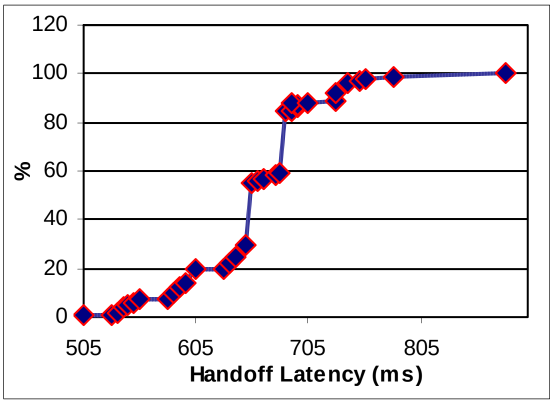

Fig. 3. Cumulative distribution of 802.11bhandoff samples

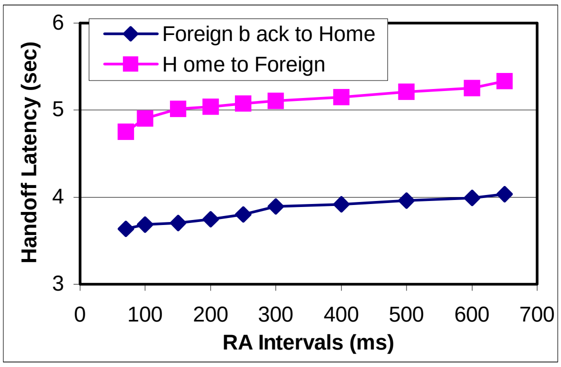

Fig. 4. Handoff times versus RA intervals

We experimentally trigger handoff events and measure the time period during which connectivity was lost shown in Fig. 4. We found that real-world 802.11b handoffs were typically completed in less than 700 ms. The IP level disruption due to 802.11b and PDCM controlled handoff together was significantly higher around 4.75 and 3.638 sec. But handoff time for IP level disruption due to 802.11b is higher in comparison to PDCM controlled {Table 2}. Tuning the RA intervals from 30-70ms (the default) to 500-700ms not significantly degrade these handoff times. Short RA intervals may, in practice, not be worth the transmission overhead as shown in Fig. 4. Default PDCM controlled handoff is highly disruptive to real-time and interactive applications during handoff events, even if the underlying link layer handoff was instantaneous. How simple implementation bugs can cause substantial increases in the handoff latencies, regardless of the actual PDCM system itself.

7 Related Works

In [1] authors described the design of a platform which was implemented using ANSAware platform to support collaborative multimedia applications in a mobile environment. The platform provides a programming interface compatible with emerging open systems standards, and includes services for processing multimedia information. In addition it provides feedback to applications and users on the state of their communications infrastructure - an important requirement in mobile environments. The services provided by the platform have been used to develop a collaborative multimedia application designed to support a specific class of mobile worker, i.e., field engineers. But platform and application works over a specific range of wireless network types, poor utilization of bandwidth and QoS. It works with a limited range of wireless communications technologies. In [2] authors proposed ALICE architecture which does not rely on Mobile IP, but addresses the mobility problems at a higher level (the session level in the ISO protocol stack), and the same holds for the OnTheMove project [3]. The MobiWare project [4] offers a CORBA based middleware toolkit for mobile services providing alternatives to existing network architectures, this approach again deviates widely from existing standards in the area. In [5, 6] author developed the CRAS (Client Representative Agent Server) architecture together with a location and query management strategy. Each client or MH has an associated representative (R) which lies on the fixed network. The connection between MH and R is wireless. The CORBA ORB is used as the means of communication between the different components which is again a heavy weight middleware that restrict the mobility and this architecture needs to be extended in a variety of ways to accommodate the components needed to allow transparent access to distributed information sources and to take changes in the way mobile computing is handled and also proposed another architectural model which allow the support of architectural model [7] which allows the support to mobile users in accessing heterogeneous information sources. But this model also relies heavily on the usage of CORBA middleware. In [8] authors presented the challenges, design, and implementation of the Roam system. The Roam system is a seamless application framework for building seamless applications that can migrate at runtime across heterogeneous devices. The Roam system provides adaptation strategies at the component level, including dynamic instantiation, offloading computation, and transformation. In ROAM with the existing SGUI toolkit it is difficult to customize a device-independent representation for a particular device. When the developers change the device-independent model at a later time, they may also have to update transformation rules that are affected by the change. There is no secure fault tolerance support for seamless application migration for real-time applications such as video conferencing. For example, a user may want to migrate a video conferencing from a mobile phone to a car navigation system when he/she is entering a car. This places a realtime constraint on migration latency and there is no way to make sure that the interruption time is minimized during application migration. The system need the support of repository to save execution state of the service where a user can save a Roam application on one device, and restore it at a later time on any device so that application migration cannot be suspended. And this should be extended to each user level such that every user can save and restore application execution state individually without affecting other users. In [9] work deals with reconfigurable control functions and protocols for supporting mobile computing applications in heterogeneous wireless systems like cellular networks and WLANs. The control functions are implemented in a software module named-reconfigurable access module for mobile computing applications (RAMON), placed in mobile and/or base stations. RAMON operates on abstract models of the main communication functions of wireless systems (e.g., transmission over the radio channel, coding end error recovery, capacity sharing and packet scheduling, handover, congestion control, etc.). But it does not providing fault tolerance and security to communication channel. In [10] author proposed the application-aware adaptation that supports a collaborative relationship between application and system. However, the prototypal system is only based on network-aware application. It also raises the question on how to structure systems that can support pervasive applications. Many other systems that provide an adaptive behavior to mobile applications were also analyzed in [10,13,14]. In [11] authors describes a dynamic service reconfiguration model where the proxy is composed of a chain of service objects called mobilets (pronounced as mo-be-lets), which can be deployed onto the network actively. This model offers flexibility because the chain of mobilets can be dynamically reconfigured to adapt to the vigorous changes in the characteristics of the wireless environment, without interrupting the service provision for other MHs. Furthermore, mobilets can also be migrated to a new proxy server when the MH moves to a different network domain. We have realized the dynamic service reconfiguration model by crafting its design into a programmable infrastructure that forms the baseline architecture of the WebPADS system. However, a significant difference between existing systems and PDCM architecture is the first requires all resources be explicitly bound and leased, while PDCM support dynamic adjustment to new state based on system-application behavior. In literature existing architectures also do not take into account the specific characteristics of mobile applications. Thus, in order to make them ideally suitable for the support of mobile users, a number of adaptations are necessary. None of these approaches aims at supporting access to heterogeneous information sources. In PDCM architecture we have considered every phase of existing problems.

8 Conclusion and Future Works

The proliferation of mobile devices in wireless environments has put special requirements on the ability to access the Web seamlessly. To addresses the impact of varying contextual characteristics on mobile access, we developed the PDCM framework, which can actively deploy new mobile services to provide an optimal set of functionality. PDCM can also dynamically reconfigure and migrate its services to adapt to the vigorous changes in the wireless ad hoc environment. In the PDCM architecture we are investing security, fault tolerance, naming and locating, mobility management. Currently we are working on location management of MHs. Further we are in the process of testing the PDCM architecture for-handoff, naming and location, security, and fault tolerance of the system.

References

[1] Nigel Davies, Gordon S. Blair, Keith Cheverst, Adrian Friday, Supporting collaborative applications in a heterogeneous mobile environment, Computer Communications 19 (1996) 346-358.

[2] M. Haahr, R. Cunningham, V. Cahill, Supporting CORDA Applications in A Mobile Environment, in Proceedings of Mobicom'99, Seattle, WA, August 15-20, 1999.

[3] E. Kovacz, K. Rohrle, M. Reich, Mobile Agents on The Move integrating an Agent System into The Mobile Middleware, in Proceedings of the ACTS Mobile Summit 1998, Rhodos, Greece, June 8-10. 1998

[4] O. Angin, A. Campbell, M. Kouvanis, R. Liao, The Mobiware Toolkit: programmable support for adaptive mobile networking, IEEE Personal Communications Magazine, Special Issue on Adapting to Network and Client Variability, 5(4), 1998.

[5] N. Pissinou, K. Makki, M. Hong, L. Ji and A. Komar, An Agent Based Mobile System, in Proceedings of 16th International Conference On Conceptual Modeling, Los Angels, CA, pp. 361–374, Springer, New York, 1997.

[6] N. Pissinou, M. Hossain, K. Makki, On the Design of a Location and Query Management Strategy For Mobile and Wireless Environments, Computer Communications Journal 22 (1999) 651–699.

[7] Niki Pissinou, Kia Makki, Birgitta Konig-Ries, Mobile users in heterogeneous environments with middleware platform, Computer Communications 26 (2003) 700–707.

[8] Hao-hua Chu, Henry Song, Candy Wong, Shoji Kurakake, Masaji Katagiri, Roam, a seamless application framework, The Journal of Systems and Software 69(3): 209–226, 2004.

[9] Carla-Fabiana Chiasserini, Francesca Cuomo, Leonardo Piacentini, Michele Rossi, Ilenia Tinirello, Francesco Vacirca, Architectures and protocols for mobile computing applications: a reconfigurable approach, Computer Networks 44 (2004) 545–567.

[10] B. Noble, System Support for Mobile Adaptive Applications, IEEE Personal Computer Systems, pp. 44-49, Feb. 2000.

[11] S.N. Chuang, A.T.S. Chan, J. Cao, Ronnie Cheung, Dynamic Service Reconfiguration for Wireless Web Access, in Proceedings of the Twelve International World Wide Web Conference, Budapest, Hungary, May 2003, pp. 58-67.

[12] S.N. Chuang, A.T.S. Chan, J. Cao, Ronnie Cheung, Actively Deployable Mobile Services for Adaptive Web Access, IEEE Internet Computing, 8(2): 26-33, March/April 2004.

[13] G. Chen, D. Kotz, A Survey of Context-aware Mobile Computing Research, Technical Report TR2000-381, Dartmouth Computer Science, 2000.

[14] Anurag Acharya, M. Ranganathan, and Joel Saltz, Sumatra: a Language for Resource aware Mobile Programs, in Proceedings of Second International Workshop on Mobile Objects Systems (MOS'96), Linz, Austria, July 1996, Mobile Objects Systems: Towards the Programmable Internet, Jan Vitek and Christian Tschudin (Eds.), Springer-Verlag LNCS 1222, pp. 111-130, April 1997.

[15] Tomasz Imielinski and B.R. Badrinath, Data Management for Mobile Computing, in SIGMOD RECORD, 22(1): 34-39, March 1993.

[16] P. A. Bernstein, Middleware: A Model for Distributed System services, Communications of the ACM, 39(2): 86-98, Feb. 1996.

[17] R. B. Patel, Nikos Mastorakis, K. Grag, Mobile Agent Location Management in Global Networks, WSEAS TRANSACTIONS on COMPUTERS, Issue 7, Vol. 4, 697-710, July 2005.

[18] R. Ben-Natan, CORBA - A Guide to Common Object Request Broker Architecture, McGraw-Hill, 1995.

[19] Sun Microsystems, Java Remote Method Invocation–Distributed Computing for Java. White Paper, March 1998.

[20] Akyildiz I., Jiang X. and Mohanty S, A survey of mobility Management in Next Generation All-IP Based Wireless Systems, IEEE Wireless Communications, 11(4): 16-28, August 2004.