A Novel Platform for Context Maintenance and Discovery in a Ubiquitous Environment

Qi Liu and Nigel Linge
Informatics Research Institute, University of Salford
Q.Liu@pgr.salford.ac.uk, N.Linge@salford.ac.uk

Abstract

Context maintenance and discovery, which aim to represent, manage and share context information, have been discussed since context-awareness was presented in the 1990s. These become more challenging when being applied to a ubiquitous environment where dynamic and distributed computing happens anytime, anywhere.

In this paper, a platform for ubiquitous context representation and its distribution is introduced. Five semantic dimensions are used to recognise and organise potential context information. A Context information Base (CiB), a context repository, is integrated in each device of a ubiquitous network to maintain context parameters discovered from neighbour nodes. The CiComm protocol distributes context information and accordingly realises the capabilities of context deployment and retrieval. The resulting architecture is fully extensible being infrastructure-less and based on an ontological model. Simulations carried out using NS-2 have also demonstrated the operations of the protocol and proven through experimental results that it works efficiently in a dynamic ubiquitous environment.

1. Introduction

Ubiquitous computing refers to a collective and cooperative computing environment for anyone, anywhere, anytime operations. A ubiquitous environment utilises embedded, wearable and any other forms of intelligent components. A ubiquitous network can therefore become the combination of mobile computing and ubiquitous computing, where challenges are raised with regard to information processing, interoperability, security, privacy, etc.

Context awareness, on the other hand, has been applied in different fields involving human computer interaction, wireless sensor networks, etc. This is reflected in the growing number of excellent context-relevant models, frameworks, middleware and programs that have been developed or proposed.

A generic platform is presented in this paper for the purposes of context representation discovery and distribution in a ubiquitous network. Context information, defined and represented as parameters is stored in a Context information Base (CiB) in each node. A communication protocol, CiComm is designed for context distribution, discovery and retrieval.

The remainder of this paper is organised into four sections. Related work on context-awareness and its application in ubiquitous computing, as well as service discovery protocols is firstly reviewed. Then, a prototype of a context-driven platform is presented, followed by the context representation, the construction of CiBs, and the design of the associated communication protocol. The simulation section demonstrates scenarios that simulate the platform through which a performance analysis is produced. In the final section, a conclusion and consideration of future work is given.

2. Related Work

2.1 Context Awareness

The concept of context emerged in the early 1990s with the development of Human Computer Interface computing [2]. Context-aware applications were designed using only limited context parameters, e.g. location, time, etc. Popular applications involved Active Badge, a location system that was built to position a user's place and forward appropriate phone calls to him/her [3]; Cyberguide, a tour guide application that directed users both indoors and outdoors [4]; notes assistant applications that detected current time, a user's location and his/her status, and therefore made corresponding behaviours [5, 6]. However, a simple definition of context limits the extension of the concept and therefore gives little generic applicability.

Alternatively in the late 1990s, synonyms that represent abstract partial features from context were employed; typical examples involve the environment of a network [8, 9], a set of transactions [10, 11], and

978-0-7695-3492-3/08 $25.00 © 2008 IEEE
DOI 10.1109/EUC.2008.100

profiles of users or devices [8, 12], etc. These approaches allowed a generic definition for context to emerge, but left context parameters to developers (even users) to define. Consequently, these developer-specified parameters may differ from each other, and therefore lead to conflict and confusion.

In 2001, Dey [2] presented a definition, which identifies all relevant information that is able to describe the "situation of an entity", i.e. its context. Following this definition, recent research activities have aimed to create contextual frameworks and/or models, through which context information is gathered, shared and interpreted [7]. These parameters contain both specific elements and abstract concepts to provide adequate information for their own projects/applications. Furthermore, frameworks were consequently built so that these context parameters could be represented and shared within their own networks.

2.2 Context in Ubiquitous Computing

Ubiquitous computing assumes that computers are pervasive in people's daily life and activities. Information processing becomes embedded into all kinds of networked components and objects; people can therefore experience more intelligence and automation.

In such a ubiquitous environment context awareness has been utilised for information representation and maintenance. A reconfigurable context-sensitive middleware (RCSM) was introduced in [1] for the purposes of context awareness and ad hoc communication within a ubiquitous environment. Bottazzi, et al [17] presented a context-aware middleware solution, AGAPE where context is employed within PDAs to effectively manage an outdoor assistance team and maintain the interaction between team members during emergency activities.

2.3 Service/Resource Discovery Protocols

Web services introduced with the development of the Internet focused on shared resources from a server, involving identification (allocation of IP addresses in DHCP servers, for example), authentication, email services, etc. With the emergence of ubiquitous computing and other applications of wireless networks in recent years, concentration has been moved forward to new protocols and architectures on services discovery and provision.

A service discovery protocol can be identified and classified using certain characteristics. Directory, as one keyword used in service discovery protocols (SDPs) represents a logical entity that realises communication between clients and servers. A directory-less SDP requires direct communication between devices, e.g. UPnP [15]. Jini technology [16], on the other hand, is directory based with regard to its lookup service; Jini is also a centralised architecture because of its dependency on the lookup service. Distributed SDPs, e.g. DSDP[13] creates a dynamic backbone for the location and registration of services.

The design of a SDP protocol offers different challenges from context awareness. The former aims at approaches to distributing services; while the latter requires a set of abstract parameters to describe the network environment, its devices and services. However, they also have similarities in terms of ubiquitous networks. First, they both attempt to be adaptive to changes in a network environment, e.g. topology and mobility control. Furthermore, they concentrate on the information processing of networks, i.e. the communication between network components.

3. Context Driven Platform

3.1 Overview of the Platform

The context driven platform for mobile ad hoc networks was introduced in [18], where a ubiquitous device is context driven if its behaviour and attributes can be justified and decided upon by locally retrieved and remotely shared context information. The representation of context in this project takes as a starting point Dey's abstract context entities [2], and evolves from there by defining five basic semantic dimensions. A CiB facilitates the representation and organisation of context information, context exchange between devices, and adaptive configuration and retrieval of context information.

3.2. Context Communication (CiComm) Protocol

3.2.1 Packets Format

Exchange of context information, i.e. the interaction of CiBs is not only a process of transmission and reception, but also of provision and registration. In other words, the exchange shares both context parameters and the underlying information (e.g. services, situation, etc.). A CiB communication protocol has therefore been designed for discovering, sharing, requesting and upgrading context information.

Every packet transmitted by CiComm has a 13-byte packet header, where 6 bytes of source and destination addresses and a 1 byte of flag field are located. In the implementation of CiB and CiComm in NS2, MAC addresses are employed as appropriate CiB addresses; however, the latter are an identifier of context which is integrated into ubiquitous devices, so any form of addressing scheme can be used. When the context information of a neighbour to a node is distributed to

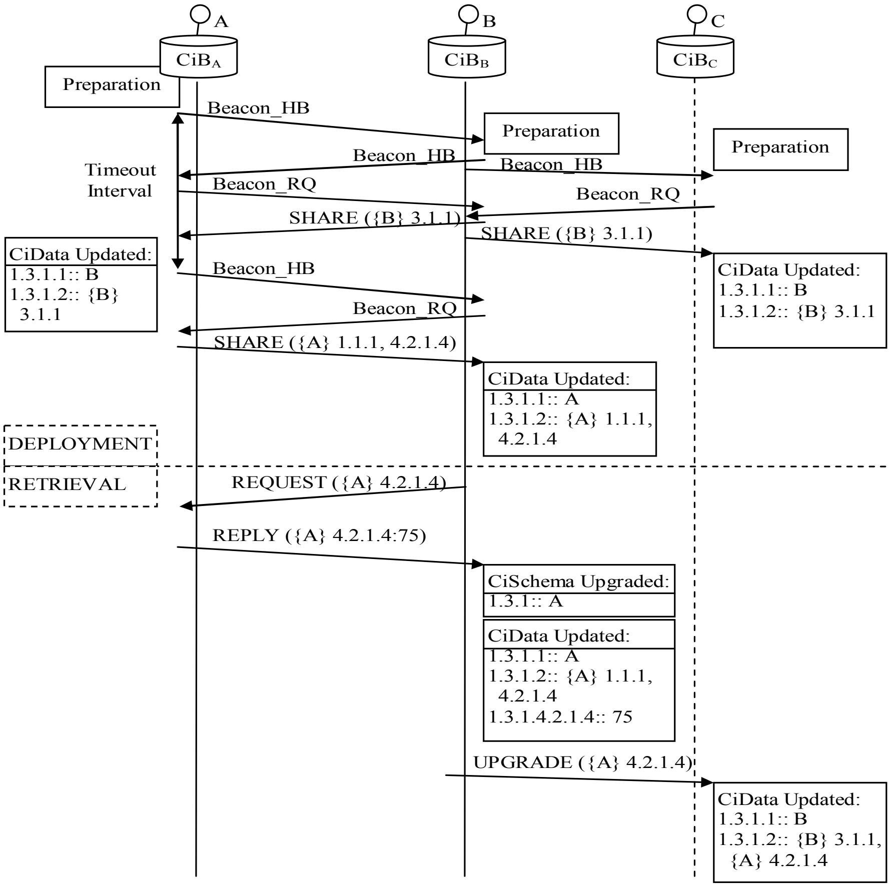


Figure 1: CiComm Packets and the process of communication

other neighbours, the MAC source address will be replaced by the node's, while the CiB source address still comes from the original neighbour node.

The Flag field carries compulsory attributes of a CiComm packet, including packet type, beacon type (valid when the packet type is BEACON) and direction flag. Table 1 depicts these flags and their hexadecimal value. Full functionalities of each type of packets will be discussed in the Operation section.

3.2.2 Operation

Two phases, Deployment and Retrieval are utilised in the process of communication of context information. No strict dependencies are set on either of the phases; that is either of two phases can be started without the completion of the other one. However, partial completion of the Deployment Phase in a node can cause its neighbours to fail to retrieve values of the node because of insufficient provision of the context information.

TABLE I
FLAG OF CICOMM PACKETS

Flag Type	Packets Type	Hexadecimal Value
cctype	BEACON	0x1000
	SHARE	0x0001
	REQUEST	0x0010
	REPLY	0x0011
	UPGRADE	0x0100
bctype	Beacon_Heart_Beat	0x01
	Beacon_Request	0x10
cftype	Direct	0x01
	Forward	0x10

In the Deployment Phase each node distributes context parameters to its active neighbours. Beacon and SHARE packets are transmitted in the Deployment Phase. A Beacon_HB packet is generated periodically (e.g. the interval is set to once per second in NS2). When a Beacon_HB is received, the host checks whether it knows this neighbour; if not, a Beacon_RQ will be replied. Once the Beacon_RQ packet is received, a SHARE packet containing all local context information and those of registered neighbours will be sent.

The Retrieval Phase retrieves exact values of particular context information from specified nodes and then distributes this to other nodes. A REQUEST packet is generated to acquire detailed context information. A REPLY packet is then generated with the requested value. The reception of the REPLY packet will update the CiB and generate UPGRADE packets to the nodes' other neighbours.

Figure 1 depicts how three nodes with different active times can discover and share their context information with each other. According to [18], each context parameter is represented as a reference number and its value. In this scenario, the context in Node A, "4.2.1.4", is discovered and shared to Node B and C.

4. Simulation: Construction and Analysis

4.1 Platform construction in NS-2

The context driven platform is simulated using NS-2. Mobile nodes with the CMU monarch's wireless extensions are utilised to represent devices in a ubiquitous network. A complete schematic of mobile nodes is shown in Figure 2, where the platform is integrated within the Logic Link module. As the figure depicts, the platform offers its interface (cib_) to the Logic Link only; meanwhile it gets instances from the Mobile Node, the Logic Link and the MAC for the purposes of context collection and addressing resolution.

To present a complete simulation pattern and prevent influence on simulation results from certain settings, randomness is employed in CiB and CiComm simulations; it includes random positions and movements of mobile nodes, random CiB loading and releasing time, and random selection of local context information and requirement of remote context.

4.2 Functionalities: a particular scenario simulation

The platform applies the full features of CiB and CiComm to NS2 involving the construction of CiB in each node and both Deployment and Retrieval phases of CiComm. Additionally, more configuration settings are performed by the platform for the better integration to the NS2 environment; these include construction of the CiComm packet header, registration of the CiComm packet type, and initial parameters in CiB and CiComm.

In this section, a designated ubiquitous environment is simulated to demonstrate the feasibility and adaptability of the platform and its functionalities in the deployment and retrieval phase. Eleven nodes are simulated in the scenario. 10 nodes are static at their initial positions numbered from 1 to 10; Node 0 moves through the network from ([17.22, 19.62]) to ([1.59, 1.35]). Every node is randomly designated with a local and requested context parameter respectively in order to prevent simulations on narrow scenarios (e.g. same context information requested, etc.).

The initial topology of the network is shown in Figure 3. Nodes' proximity is expressed by enclosing directly connected nodes together. The results of context discovery in appropriate nodes are represented such that the first line reveals where the context information is from, and the second line, having double colons, shows the exact requested parameter.

The movement and CiData of the mobile node, i.e. Node 0 at critical times are demonstrated in Figure 4. Four critical times are shown according to the mobile node's position, and its interaction with its neighbours. The contents of CiData are listed next to the corresponding topology figures. In each CiData, local context information shown differs from the remote one; also, any modifications at the particular time are revealed italics.

4.3 Performance analysis

To examine the performance of the platform, a set of simulations has been conducted in NS-2. All simulations are achieved in a 50 x 50 metres square with the total simulation time of 30 seconds. Different

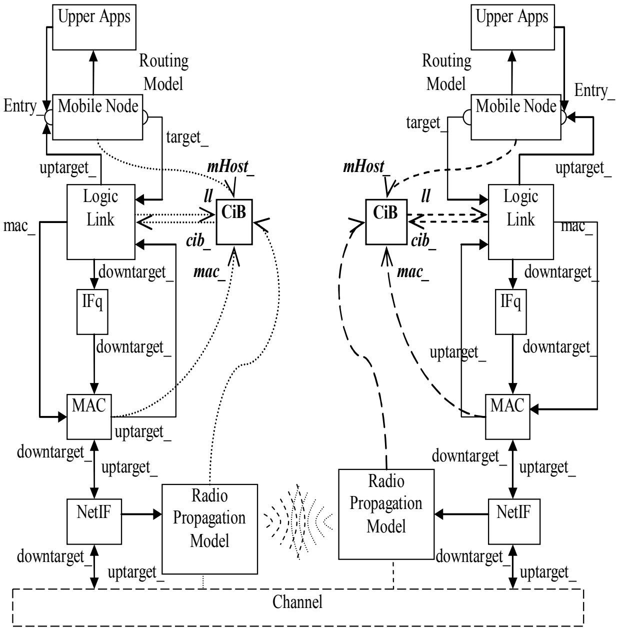


Figure 2: Simulation of CiB and CiComm in NS2

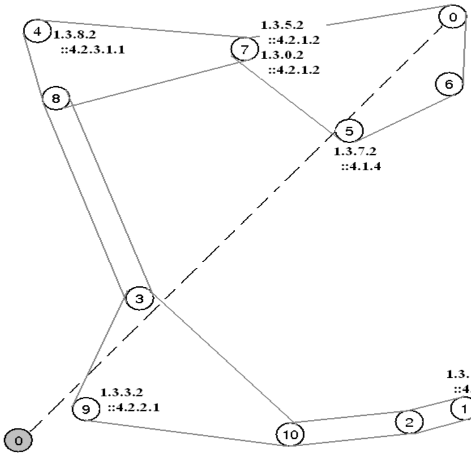


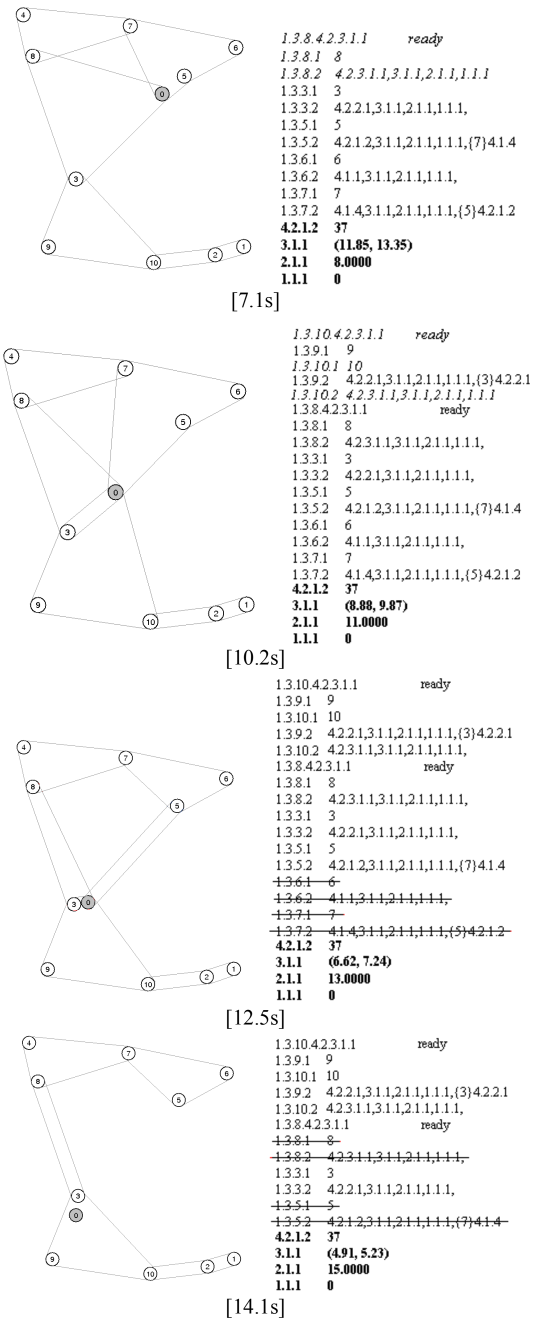
Figure 3: initial topology and CiSchema of designated nodes

numbers of nodes, i.e. 20, 30 and 40 are filled into the simulation area. In each simulation, the maximum speed of each node is set to 0, 4, 8, 16 and 30 metres per second correspondingly. Because some initial settings i.e. the topology control and the CiB loading and releasing time are decided following the randomness scheme, three tests are carried out in each situation. Therefore, 45 simulations generate their own set of experimental results.

Two criteria are used to evaluate the performance of the platform. The first one is the average overhead generated in the deployment phase; it measures the traffic load with regard to the amount of CiComm packets during the phase, as shown in Figure 5(a). It can be seen in the figure that in the simulation results of 20 and 30 nodes, the average overhead gradually grows with the increase of the speed. In the simulation results of 40 nodes, the overhead goes up when the speed steps up until 8 m/s and then drops down conversely. This is mainly because of congestion and collision generated by frequent movement in an overcrowding space. Also, rapid joining and leaving cannot activate a deployment process and hence reduces the total amount of packets.

The second metric for the platform performance is “Learning Time”, i.e. the time duration in a simulation from the interval when a Beacon_HB packet is sent to an appropriate SHARE packet is received by the same node. The average Learning Time for each simulation is shown in Figure 5(b), which reveals that the Learning Time keeps steady or even reduces a little when nodes raise their speed to 8 metres per second. After that, with the acceleration of nodes, the Learning Time in each simulation varies according to the number of nodes and their density.

The measurement of performance on these simulations depicts that the overhead at the speed of 8



1.3.8.4.2.3.1.1	*ready*
1.3.8.1	*8*
1.3.8.2	*4.2.3.1.1,3.1.1,2.1.1,1.1.1*
1.3.3.1	3
1.3.3.2	4.2.2.1,3.1.1,2.1.1,1.1.1,
1.3.5.1	5
1.3.5.2	4.2.1.2,3.1.1,2.1.1,1.1.1,{7}4.1.4
1.3.6.1	6
1.3.6.2	4.1.1,3.1.1,2.1.1,1.1.1,
1.3.7.1	7
1.3.7.2	4.1.4,3.1.1,2.1.1,1.1.1,{5}4.2.1.2
4.2.1.2	**37**
3.1.1	**(11.85, 13.35)**
2.1.1	**8.0000**
1.1.1	**0**

[7.1s]

1.3.10.4.2.3.1.1	*ready*
1.3.9.1	9
1.3.10.1	*10*
1.3.9.2	4.2.2.1,3.1.1,2.1.1,1.1.1,{3}4.2.2.1
1.3.10.2	*4.2.3.1.1,3.1.1,2.1.1,1.1.1*
1.3.8.4.2.3.1.1	ready
1.3.8.1	8
1.3.8.2	4.2.3.1.1,3.1.1,2.1.1,1.1.1,
1.3.3.1	3
1.3.3.2	4.2.2.1,3.1.1,2.1.1,1.1.1,
1.3.5.1	5
1.3.5.2	4.2.1.2,3.1.1,2.1.1,1.1.1,{7}4.1.4
1.3.6.1	6
1.3.6.2	4.1.1,3.1.1,2.1.1,1.1.1,
1.3.7.1	7
1.3.7.2	4.1.4,3.1.1,2.1.1,1.1.1,{5}4.2.1.2
4.2.1.2	**37**
3.1.1	**(8.88, 9.87)**
2.1.1	**11.0000**
1.1.1	**0**

[10.2s]

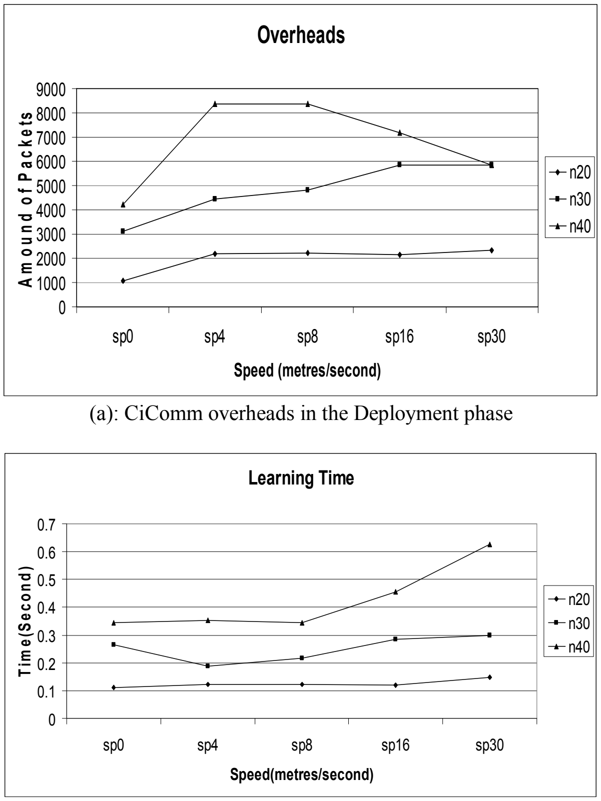
1.3.10.4.2.3.1.1	ready
1.3.9.1	9
1.3.10.1	10
1.3.9.2	4.2.2.1,3.1.1,2.1.1,1.1.1,{3}4.2.2.1
1.3.10.2	4.2.3.1.1,3.1.1,2.1.1,1.1.1,
1.3.8.4.2.3.1.1	ready
1.3.8.1	8
1.3.8.2	4.2.3.1.1,3.1.1,2.1.1,1.1.1,
1.3.3.1	3
1.3.3.2	4.2.2.1,3.1.1,2.1.1,1.1.1,
1.3.5.1	5
1.3.5.2	4.2.1.2,3.1.1,2.1.1,1.1.1,{7}4.1.4
~~1.3.6.1~~	~~6~~
~~1.3.6.2~~	~~4.1.1,3.1.1,2.1.1,1.1.1,~~
~~1.3.7.1~~	~~7~~
~~1.3.7.2~~	~~4.1.4,3.1.1,2.1.1,1.1.1,{5}4.2.1.2~~
4.2.1.2	**37**
3.1.1	**(6.62, 7.24)**
2.1.1	**13.0000**
1.1.1	**0**

[12.5s]

1.3.10.4.2.3.1.1	ready
1.3.9.1	9
1.3.10.1	10
1.3.9.2	4.2.2.1,3.1.1,2.1.1,1.1.1,{3}4.2.2.1
1.3.10.2	4.2.3.1.1,3.1.1,2.1.1,1.1.1,
1.3.8.4.2.3.1.1	ready
~~1.3.8.1~~	~~8~~
~~1.3.8.2~~	~~4.2.3.1.1,3.1.1,2.1.1,1.1.1,~~
1.3.3.1	3
1.3.3.2	4.2.2.1,3.1.1,2.1.1,1.1.1,
~~1.3.5.1~~	~~5~~
~~1.3.5.2~~	~~4.2.1.2,3.1.1,2.1.1,1.1.1,{7}4.1.4~~
4.2.1.2	**37**
3.1.1	**(4.91, 5.23)**
2.1.1	**15.0000**
1.1.1	**0**

[14.1s]

Figure 4: changes of CiData of Node 0 at critical intervals



(a): CiComm overheads in the Deployment phase

(b): Learning Time in the Deployment phase

Figure 5: protocol performance

m/s in the 40 nodes simulation considerably rises compared to the 20 and 30 nodes, but its overall network utility is still lower than 1%. Also, in terms of Learning Time, which contains multiple transmissions (e.g. beacons, SHARE packets, looking them up and updating, etc), the maximum is around 0.6 second. Beyond these results, more features of the platform are recognised through these simulations, for instance the parameter speed has less effect in an open area than a crowded one.

5. Conclusion

In this paper, a new context driven platform is presented in a ubiquitous environment for context representation, discovery, maintenance and exchange. The platform provides a generic support for upper application of context. As the position of the platform in the NS2 implies, potential upper layer applications and protocols can retrieve low-level context information through interfaces to this platform. The CiComm protocol is proved to be efficient and flexible through the simulations

Moreover, similar functionalities as in the CiComm are provided in Jini, a service discovery protocol. Both protocols focus on information processing; they also adapt to changes within a dynamic networking environment. However, Jini has an infrastructure-based architecture depending on a lookup service; the CiComm on the other hand is infrastructure-less and purely distributed. Additionally Jini is Java language dependent, whereas the CiComm is based on the ontological model and therefore independent to any programming languages. Therefore we believe that our platform offers a more open and flexible solution to the provision of context representation, discovery and maintenance.

References

[1] Yau, S.S. and F. Karim. Context-sensitive distributed software development for ubiquitous computing environments. in Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International. 2001.

[2] Dey A., Understanding and Using Context, *Personal Ubiquitous Comput.* Vol 5, Issue 1, pp 4-7, 2001.

[3] Want, R., et. al., The Active Badge Location System, *ACM Transactions on Information Systems*, Vol 10, Issue 1, pp 91-102, 1992.

[4] Abowd, G., et. al., Cyberguide: A mobile context-aware tool guide, *ACM Wireless Networks,* Vol 3, Issue 5, pp421-433, 1997.

[5] Pascoe, J., Adding Generic Contextual Capabilities to Wearable Computers, *In proceedings of the 2nd International Symposium on Wearable Computers,* pp 92-99, 1998.

[6] Rhodes, B., The wearable remembrance agent: A system for argumented memory, *In proceedings of the 1st International Symposium on Wearable Computers,* pp 123 – 128, 1997.

[7] Chen, G. and Kotz, D., a survey of context-aware mobile computing research, Technical Report, Department of Computer Science, Dartmouth College, 2000.

[8] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In *Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99*, pages 89-101, Karlsruhe, Germany, September 1999. Springer Verlag.

[9] Brown, P.J. The Stick-e Document: a Framework for Creating Context-Aware Applications. Electronic Publishing '96 (1996) 259-272

[10] Franklin, D., Flaschbart, J. All Gadget and No Representation Makes Jack a Dull Environment. AAAI 1998 Spring Symposium on Intelligent Environments, Technical Report SS-98-02, pp155-160, 1998.

[11] Hull, R., Neaves, P., Bedford-Roberts, J. Towards Situated Computing. 1st International Symposium on Wearable Computers, pp146-153, 1997.

[12] Anind K. Dey, Masayasu Futakawa, Daniel Salber, and Gregory D. Abowd. The Conference Assistant: Combining Context-Awareness with Wearable Computing. In *Proceedings of the 3rd International Symposium on Wearable Computers (ISWC '99)*, pages 21-28, San Francisco, CA, October 1999.

[13] Kozat, U. C. and Tassiulas, L., Services Discovery in Mobile Ad Hoc networks: an Overall Perspective on Architectural choices and Network Layer Support Issues, Ad Hoc Networks, Vol 2, Issue 1, pp 23-44, 2004.

[14] Cho, C. and Lee, D., *Survey of Service Discovery Architectures for Mobile Ad hoc Networks,* Term paper, Mobile Computing, CEN 5531, Department of Computer and Information Science and Engineering (CICE), University of Florida, Fall, 2005, from http://folk.uio.no/paalee/referencing_publications/ref-sd-cho-cice05.pdf, last visit on 10 May 2008

[15] Universal Plug and Play, Microsoft Corporation, Available from www.upnp.org, last visit on 10 May 2008

[16] Jini Network Technology, http://www.sun.com/software/jini/, last access: 20 April 2008.

[17] Battazzi, D., et al, Context Aware Middleware Solutions for Anytime and Anywhere Emergency assistance to elderly people. *Communications Magazine, IEEE,*Vol 44*,* Issue 4, pp 82-90, 2006.

[18] Liu, Q. and Prof. Linge, N., Context-Driven Ad-Hoc Networks, PGNET 2007, 8th Postgraduate Symposium on the Convergence of Telecommunication, Networking and Broadcasting, June 2007