International Journal of Grid and Distributed Computing
Vol. 5, No. 4, December, 2012

An Context-aware Management and Control Mechanism in a Mobile Ad-Hoc Environment

Qi Liu[1], Nigel Linge[2], Jin Wang[1], Sai Ji[1] and Ting Huai Ma[1*]

[1]*Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, 210044, China*

[2]*Computer Networking and Telecommunications Research Centre, University of Salford, Salford, Greater Manchester, M5 4WT, UK*
qi.liu@nuist.edu.cn

Abstract

Complete integration and interaction of network objects can be achieved in a mobile ad-hoc environment. Yet, the dynamic nature of free movement and interaction among these objects also brings challenges on information representation, processing and communication. This paper presents a context-aware mechanism, which is able to implement efficient information exchange and management between nodes in a mobile ad-hoc network (MANET). Rejoining and information loop issues happened heavily in a mobile ad-hoc environment has been discussed in this paper. According to the simulation results, our approach to context-aware management and control makes the information loop problem well controlled.

Keywords: *Information Loop Control, Context Awareness, Rejoining Control, Mobile Ad-Hoc Networks*

1. Introduction

Recent research on context-aware middleware brings rapid growth in the number of mobile applications, where a device's context information has been explored including its location, type, technical capability, functions, nearby devices and more general information about the physical environment [1]. For ad-hoc networks there too is a requirement to be able to detect, share and respond to contextual information. However, the unpredictability and dynamically changing topology, especially when one introduces mobility, present particular challenges for the reliable and timely dissemination of context information.

This paper introduces a new method to manage and control context exchange within an ad-hoc network. The framework comprises two components, a context information database that provides context representation and storage, and a communication protocol called CiComm used for exchanging context information. In such environment, efficient management strategies are established in order to eliminates potential problems arisen by random node movement, e.g. information loops, rejoining, etc.

The remainder of this paper is organized as follows. Related work on context awareness and its application for ad-hoc networks is presented in Section 2. Section 3 introduces our framework with its features and functions. Simulation results are discussed in Section 4 and Section 5 concludes the paper and evaluates the framework.

2. Related Work

The concept of context awareness to a mobile ad hoc network is implemented and applied using two main categories. Some prefer context information as integrated components to provide internal application program interfaces for the invocation of upper and lower layers. In these applications, environmental information and relevant properties are regarded and recorded as context information. Alternatively, complete frameworks have been also considered and designed to provide a comprehensive context awareness based solution. In this section, the applications of context aware are firstly reviewed, followed by both categories mentioned above. After that, service discovery protocols are also discussed compared with information discovery in context-aware applications.

2.1. Context-Aware Applications

Context awareness can be applied into most domains of computer science and networking. For example, a context-aware routing protocol called Communication Inter Vehicle Intelligent Cooperative (CIVIC) was designed for the communication between vehicles [2]. The CIVIC was implemented in LiveNode sensors with LIMOS [3]. Eigner and Mair reported their work in [4], where a Vehicular Ad Hoc Network (VANET) was defined for ad hoc communication between vehicles at high speed.

Vertical soft handover for better performance was discussed in [5], which used context-aware parameters to achieve the goals. The context parameters in the work included user required bandwidth, user traffic cost, access network utilization, and user received Signal to Interface-plus-Noise Ratio (SINR). In [6], context information was used to represent user information in a mobile network including their behaviors, mobility, network usage, device type, etc.. This information was then converted into Keyhole Markup Language (KML) scripts and virtualized in the Google Earth software.

2.2. Context-aware Components

A de-facto definition of context and context awareness was given by Dey [7, 8], which identifies all relevant information that is able to describe the "situation of an entity" as its context. This definition has been well used in context-awware components, which identified context as explicit parameters, e.g. devices, users, locations, time, the network. Semantic languages such as XML, OWL, etc. were used to represent context information in these designs [9].

In [10], a cross-layer routing protocol was presented in MANETs, where context information is defined to include energy consumption for better route discovery, such as TX and RX power consumption, Angle of Arrival (AoA), etc. Better results were concluded for more context information had been collected and processed.

A schema calculating a trust value of nodes in MANET was designed in [11]. The considered parameters included previous interactions, observation of present behavior, recommendation from direct paired devices, etc. The collection can be recognized as context information to build general nodes trust.

A location-aware algorithm was presented to broadcast messages in a mobile ad hoc network [12]. Different zones were defined based on neighbor positions to allow or deny forwarding. The results indicate better performance than other broadcasting algorithms in low densities.

Other research work involves a self-policing reputation mechanism that values nodes' neighbors [13], a context-aware adaptive routing protocol (CAR) for best forwarding route considering a node's battery level, mobility and co-location etc. [14, 15], a context sensitive binding mechanism for better service migration [16], and a decision support engine based on a general Bayesian network approach [17].

2.3. Context-aware Frameworks

Context-aware architectures were designed to provide complete representation and sharing mechanism of context information. Compared to context-aware components above, the solutions below provide comprehensive context-aware strategies/approaches in a mobile ad hoc network.

Context-aware migratory services were offered based on a Smart Messages platform in a ubiquitous environment [18], where Ccontext information was interpreted (MonitoredCxt), stored and shared (Context Manager), and evaluated (Validator).

A CoBrA system (Context Broker Architecture) provided an agent based architecture featuring contextual information in pervasive spaces. A context model based on a context-aware message broker was designed to offer common interfaces to context-aware applications [19]. In this system, a context-aware conceptual model federated multiple context brokers was designed to facilitate context provision and usage efficiently [20].

In [21], a platform called ACORD-CS that supports full deployment of context-aware applications was presented, including a meta-data model identifying potential context information, and a middleware providing APIs to context-aware applications.

Other models and solutions include a Reference Model for a MANET environment [22], the LMSS system to provide relevant location information of mobile users [23], and a context-aware architecture for Service-Oriented Computing [24].

2.4. Context Awareness and Service Discovery Architectures

Service discovery protocols (SDPs) were proposed for the purposes of the automatic discovery of the presence of services, the determination of where and who to provide the services and the management of the retrieval and execution of these services. In terms of network services, Sun's Jini technology explains it as "an entity that can be used by a person, a program, or another service". Two examples were given as services, including a printing job and a translation process. In our paper, the exchange of context information is regarded as a service of an ad-hoc network; and hence, an SDP protocol can be used for context discovery and exchange.

In [25, 26] SDPs have independently proposed that they should be categorized as directory based and directory-less. Directory servers are used in Directory based SDPs to maintain the information of devices and their services. These SDPs can be further classified as either distributed or centralized based on their database types. Directory-less SDPs discover services by sending broadcast or multicast requests to neighboring devices. In general a directory based SDP, e.g. Jini and DSDP [27] fits large scale networks with low levels of device mobility whilst directory-less SDPs are more suited to small scale networks or networks with high levels of device mobility. Another example is that of the Service Location Protocol (SLP) which has been adapted for use in mobile ad-hoc networks, resulting in the creation of SLPManet [28, 29].

3. Strategies to Information Management in MANET

In mobile ad-hoc networks devices can exchange context if their behavior and attributes can be justified and decided upon by locally retrieved and remotely shared context information. To enable the exchange of such contextual information we propose a method comprising three parts: a representation model, a context information repository and a communication protocol. The model defines context information in an ad-hoc network environment. It introduces a scheme for context representation and establishes a hierarchical structure to categorize context parameters. The predefined context parameters can then be maintained and shared via the repository and communication protocol.

3.1. Context-aware Model

A complete representation of relevant information requires a generic method with common rules for abstraction of the information. To meet this requirement, an ontological solution is presented in this thesis. The ontology defines five basic interrogative dimensions; they are Who, When, Where, What and How.

Each dimension retrieves one aspect of relevant information of mobile devices within an ad hoc network. The "who, when, where, what and how" questions are raised to recognize and categorize potential information within an ad hoc network. These devices' information can correspondingly be abstracted into parameters, separated in each dimension. This structure allows for context parameters from other devices within the ad hoc network to be added without interference; thereby providing more flexibility in maintaining these parameters.

3.2. Context Information Base and Communication

A context-aware framework previously designed for the representation, storage and communication of contextual information [30] has been employed in this paper to exchange relevant information between mobile devices in a mobile ad-hoc network. Firstly, a hierarchical structure is maintained underneath each interrogative dimension. In addition, an index scheme is conducted to assign a set of reference numbers to corresponding parameters according to their position at the structure. Following the design of the context-aware framework, the predefined parameters are stored in an information repository called context information base (CiB). These parameters can then be shared among mobile device in a MANET via the CiComm protocol.

Relevant information can be exchanged between mobile devices in a MANET. The management of information communication can be separated into three steps including Preparation, Update and Upgrade. In the Preparation step, the mobile device will inspect the integrity of the local CiB structure, collect local context information for further exchange, and then initialize the neighbor list. After this, a SHARE packet is sent to new known neighbors. In the Update step, the CiData Table of the local CiB will be updated including assigning a neighbor index number, and then creating/updating neighbor entries in the CiData table. In the Upgrade step, the device will be registered into the receiver, which is achieved by upgrading the local CiSchema table. Two entries are created containing the neighbor's identity and the index number of the retrieved context parameter. In the case of entries having been created, the existing entries will be amended by updating the list of index numbers.

3.3. Information Management and Loop Control

A lease control is introduced to help maintain the relevant information from proximity and kept it up to date, as shown in Figure 1. By default, the lease timer is set as five times long as the heart-beating timer. Failure to receive periodically incoming heart-beating beacons for more than five times will cause the removal of the neighbor and the associated entries including all direct context from that neighbor in both CiData and CiSchema tables. The lease timeout value is deduced by one every time when the heart-beat occurs; whereas the value is reset/renown when a new Beacon_HB frame is received from the corresponding mobile device. The mechanism therefore actively controls neighbors' movements i.e. their departure and rejoining.

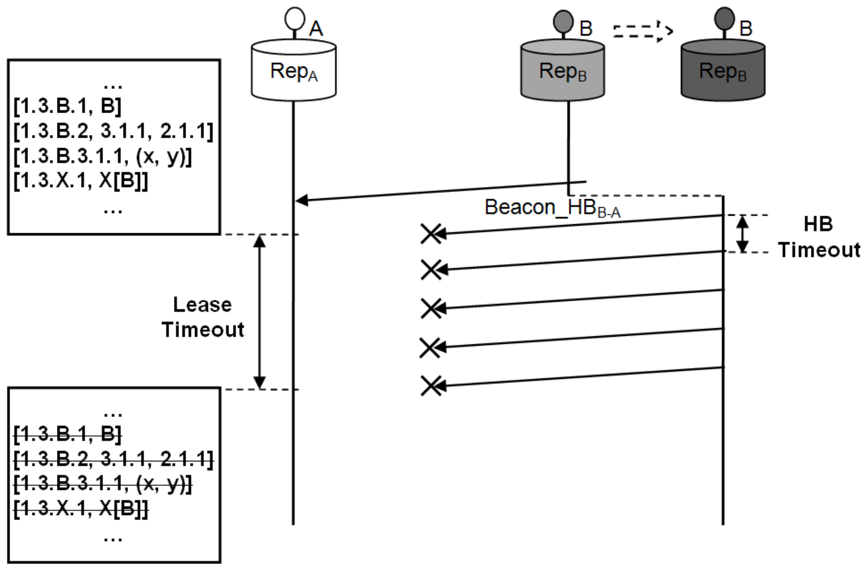


Figure 1. Lease Control within a Local Repository. When Node B moves out of the communication range of Node A, its beacon cannot reach Node A, and hence its relevant context information is removed from Node A

4. Simulation and Results

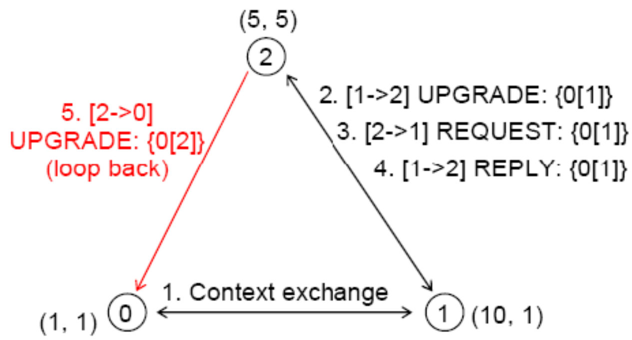
The context-aware framework has been implemented in NS2. Particular scenarios have been designed to evaluate the impact of loops and dead loops, and rejoining.

4.1. Scenario One: Copping with Information Loops and Dead Loops

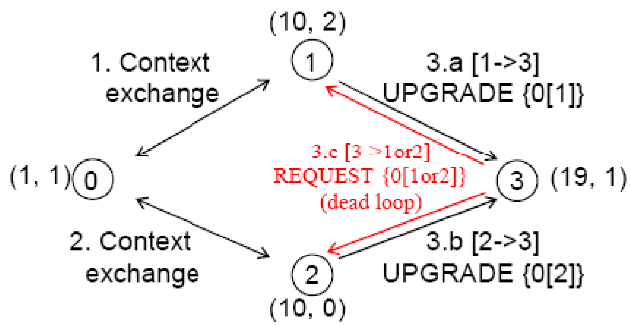
In this scenario, two simulations are designed to examine the potential loop back and dead loop issues. Here a loop means same information is shared back to the source device via one or more intermediate devices. A dead loop means two or more devices continuously send the same request and response packets to each other. The scenario settings are shown in Table 1, whereas the topology is shown in Figure 2.

Table 1. Simulation Settings for Scenario One

Simulation Settings	Simulation 1	Simulation 2
Simulation Area	10x10 m^2	20x20 m^2
Simulation Duration	30 seconds	30 seconds
Number of Devices	3	4
Bandwidth	11Mb	11Mb
Communication Range	10 meters	10 meters
Heart Beat Timer	1 second	1 second
Lease Timer	2 seconds	2 seconds
CiB Load and Release time	random	random



(a) Context information looping back via an intermediate node



(b) Context information continuous updating at Node 3

Figure 2. Topologies of Two Simulations. The upper topology (a) depicts how information of device 0 is looped back to itself. The bottom one (b) demonstrates information of device 0 is continuously updated at device 3.

The list of total UPGRADE packets generated in Simulation 1 is captured and shown in Figure 3. Four columns are created in the list; they are, from left to right, the receiving time, the source and destination CiB address, the source and destination MAC address, and the packet body.

Receiving Time	CiB Address	MAC Address	Packet Body
1.0132	CiB(0->2)	MAC[1->2]	‘{0}3.1.1”
1.0149	CiB(1->2)	MAC[0->2]	‘{1}3.1.1”
1.0166	CiB(2->0)	MAC[1->0]	‘{2}3.1.1”
1.0176	CiB(1->0)	MAC[2->0]	‘{1}3.1.1”
1.0204	CiB(2->1)	MAC[0->1]	‘{2}3.1.1”
1.0210	CiB(0->1)	MAC[2->1]	‘{0}3.1.1”

No Loop back happens

Figure 3. Results of Simulation 1. Six UPGRADE packets were captured. According to the fact that in each packet, the next-hop address tagged in the MAC Address field differs from the destination address recorded in the packet body, it can concluded that no looping back happened during the simulation

The reason that only UPGRADE packets are displayed is because context information is only looped back via this type of CiComm packets. Through the list, it can be seen that six UPGRADE packets were generated. The device identity shown within the curly brackets in each packet body is different from the corresponding destination address in the MAC Address field, which proves that no packets are looped back to original devices. The CiComm protocol in each device always checks the destination that it is sending a packet to. An UPGRADE packet is only transmitted to those devices that are not the same as those drawn within the curly brackets of the packet body.

Figure 4 summarizes the REQUEST packets generated in Simulation 2. According to Figure 2(b), dead loops in this scenario happen only if context information of an intermediate device is requested and there are two active routes to that device. In this figure, the four entries shown in red are REQUEST packets that acquire context information from indirect devices. This can be seen through the fact that their CiB destination addresses are different from their MAC destination addresses. The limited number of REQUEST packets generated for indirect context information proves that no dead loops happen within the simulation. In this scenario, there are two active routes between device 0 and 3.

It is also noticed in Figure 4 that before requesting context information from Device 3 and Device 0, Device 2 requested their information from other devices. In the figure, Device 2 sent a REQUEST packet to Device 1 containing Device 3's context information "3.1.1" at 1.5318 seconds. This happened again at 1.5403 seconds but containing Device 0's information via Device 3 at that time. After these events, Device 2 then sent REQUEST packets directly to Device 3 at 1.5395 seconds and Device 0 at 1.5952 seconds respectively.

Such situations happened due to actual reception order of Device 2. That is, before the reception of context information directly from the original device (in this case, Device 3 at 1.5318 seconds), the destination received forwarded packets via other indirect device(s) (i.e. Device 1).

Receiving Time	CiB Address	MAC Address	Packet Body
0.5379	CiB(1->3)	MAC[1->3]	"3.1.1"
0.5948	CiB(1->0)	MAC[1->0]	"3.1.1"
0.5972	CiB(3->0)	MAC[3->1]	"3.1.1"
0.6377	CiB(3->2)	MAC[3->2]	"3.1.1"
0.6417	CiB(0->2)	MAC[0->2]	"3.1.1"
0.6453	CiB(1->2)	MAC[1->2]	"3.1.1"
1.5297	CiB(2->1)	MAC[2->1]	"3.1.1"
1.5318	CiB(2->3)	MAC[2->1]	"3.1.1"
1.5353	CiB(3->1)	MAC[3->1]	"3.1.1"
1.5395	CiB(2->3)	MAC[2->3]	"3.1.1"
1.5403	CiB(2->0)	MAC[2->3]	"3.1.1"
1.5559	CiB(0->1)	MAC[0->1]	"3.1.1"
1.5589	CiB(0->3)	MAC[0->1]	"3.1.1"
1.5952	CiB(2->0)	MAC[2->0]	"3.1.1"

Figure 4. Results of Simulation 2. REQUEST packets were captured. No dead loop or looping back issues happened.

Figure 5 shows SHARE packets transmitted and received by Device 2 in Simulation 2. Through the figure, it can be seen that when Device 1 shared its context information with Device 2 at 1.5279 seconds, context information of Device 0 and Device 3 was also contained, which proves that Device 1 had retrieved context information from those devices. Meanwhile, due to no SHARE packets about these two devices being received by Device 2 yet, REQUEST packets were sent to Device 1 for these devices' Information at 1.5318 and 1.5403 seconds respectively, (see in Figure 4). After that, context was received from directly connected devices to Node 2 at 1.5384 and 1.5943 seconds, as shown in Figure 5. Corresponding REQUEST packets were therefore generated again at 1.5395 and 1.5952 seconds, as shown in Figure 4.

Receiving Time	CiB Address	MAC Address	Packet Body
0.6369	CiB(2->3)	MAC[2->3]	"4.2.2.1,3.1.1,2.1.1,1.1.1,"
0.6409	CiB(2->0)	MAC[2->0]	"4.2.2.1,3.1.1,2.1.1,1.1.1,"
0.6445	CiB(2->1)	MAC[2->1]	"4.2.2.1,3.1.1,2.1.1,1.1.1,"
1.5279	CiB(1->2)	MAC[1->2]	"4.2.2.3,3.1.1,2.1.1,1.1.1,{0}{3}3.1.1"
1.5384	CiB(3->2)	MAC[3->2]	"3.1.1,2.1.1,1.1.1,{0}3.1.1"
1.5943	CiB(0->2)	MAC[0->2]	"4.2.2.2,3.1.1,2.1.1,1.1.1,{3}{1}3.1.1"

Context received from the direct devices
is later than it from indirect one (Device 1)

Figure 5. Results of Simulation 2. SHARE packets were captured. No dead loop or looping back issues happened.

Through Figure 4 and 5, it can be seen that late reception of context information from directly connected devices can generate redundant CiComm packets, but this traffic does not result in dead loops. Having received SHARE packets from Device 3 (at 1.5384 seconds) and Device 0 (at 1.5943 seconds), Device 2 updated its local CiB with direct context information and stopped requesting context from the intermediate device, i.e. Device 1. In addition, Device 0 and 3 requested context from each other via Device 1 and no dead loop was created.

4.2. Scenario Two: Copping with Departure and Rejoining

This scenario aims to examine the rejoining behavior of devices in context driven ad hoc networks. Here rejoining is defined as the process by which a device's movement causes it to move out of communication range with its neighbors and subsequently travelling back into range, as shown in Figure 6.

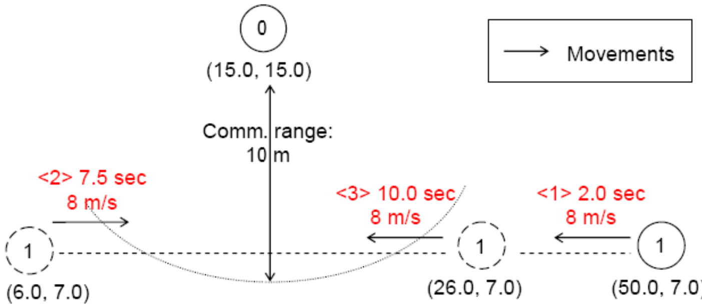


Figure 6. Topologies of Two Simulations. The movement of Device 1 is configured in three phases. The first one starts at 2.0 seconds, when Device 1 moves from position (50.0, 7.0) to (0.0, 7.0) at 8 meters per second. At 7.5 seconds, it changes its direction to its original position (50.0, 7.0) at the same speed. Finally at 10.0 seconds, Device 1 moves back to position (0.0, 7.0) again at 8 m/s and stops there until the simulation is over.

According to the movement of Device 1, it enters and leaves communication range with Device 0 three times, as shown in Table 2. Because the speed of Device 1 is constant (8 metres per second), the duration that it stays in the communication range of Device 0 is always 1.5 seconds.

Table 2. Entering and Leaving Device 0"s Communication Range

	Joining Time	Leaving Time
1	5.625	7.125
2	7.875	9.375
3	10.625	12.125

In this scenario, three different lease timers are configured to demonstrate how the lease timer controls potential redundant contextual traffic generated because of frequent rejoining events. In Figure 7, SHARE packets transmitted between two devices are listed. These packets can be separated into three groups according to the time stamps; these are transmitted at about 6.5, 8.5 and 11.5 seconds respectively. The lease timer was set to 1 second, so every time when the devices lost contact with each other, their context information was removed from the local CiBs. As a result, when Device 1 joined into Device 0's communication range for the second and third time, more contextual packets had to be exchanged for their registration in each other again.

Receiving Time	CiB Address	MAC Address	Packet Body
6.5023	CiB(0->1)	MAC[0->1]	"4.2.1.3,3.1.1,2.1.1,1.1.1,"
6.5091	CiB(1->0)	MAC[1->0]	"5.1.1,3.1.1,2.1.1,1.1.1,"
8.5025	CiB(0->1)	MAC[0->1]	"4.2.1.3,3.1.1,2.1.1,1.1.1,"
8.5090	CiB(1->0)	MAC[1->0]	"5.1.1,3.1.1,2.1.1,1.1.1,"
11.5021	CiB(0->1)	MAC[0->1]	"4.2.1.3,3.1.1,2.1.1,1.1.1,"
11.5093	CiB(1->0)	MAC[1->0]	"5.1.1,3.1.1,2.1.1,1.1.1,"

Figure 7. Results of Scenario 2. Six SHARE packets were captured. The lease timer was set to 1 second.

When the lease timer was set to 2 seconds, the first leaving period (from 7.125 to 7.875 seconds in Figure 8) was ignored by Device 0 and Device 1. This is because during this period, the devices only checked their lease timer once (at 7.5 seconds), so that when Device 1 rejoined at 7.875, both devices still had neighbors' context information. Therefore, there are no SHARE packets generated during the first and third joining periods.

Receiving Time	CiB Address	MAC Address	Packet Body
6.5023	CiB(0->1)	MAC[0->1]	"4.2.1.3,3.1.1,2.1.1,1.1.1,"
6.5091	CiB(1->0)	MAC[1->0]	"5.1.1,3.1.1,2.1.1,1.1.1,"
11.5025	CiB(0->1)	MAC[0->1]	"4.2.1.3,3.1.1,2.1.1,1.1.1,"
11.5091	CiB(1->0)	MAC[1->0]	"5.1.1,3.1.1,2.1.1,1.1.1,"

Figure 8. Results of Scenario 2. Four SHARE packets were captured. The lease timer was set to 2 seconds.

When the lease timer is set to 4 seconds, according to Table 5.9, the lease timer is longer than both leaving periods, i.e. from 7.125 to 7.875 seconds and from 9.375 to 10.625 seconds. Therefore, there is only one pair of SHARE packets being transmitted, which is when the two devices communicate with each other at 6.5023 seconds.

Receiving Time	CiB Address	MAC Address	Packet Body
6.5023	CiB(0->1)	MAC[0->1]	"4.2.2.1,3.1.1,2.1.1,1.1.1,"
6.5091	CiB(1->0)	MAC[1->0]	"3.1.1,2.1.1,1.1.1,"

Figure 9. Results of Scenario 2. Only two SHARE packets were captured. The lease timer was set to 4 seconds.

According to the simulation results in the second scenario, the longer the lease timer, the less contextual packets are generated. When the lease timer is set as 4 seconds, no rejoining communication is needed since the context information of each device is kept until both devices meet again.

Raising the lease timer in the CiComm protocol can reduce the number of redundant contextual packets that result from frequent leaving and rejoining events. However, a trade-off is required to set the lease timer since a large lease timer can also bring other issues. For example, if the lease timer is extended, the updating of context information will be delayed, which hence renders it out-of-date. In addition, it also causes more CiB resources to be wasted through a reliance on out-of-date context. In the remaining simulations the lease timer is set to be twice as long as the HB timer.

5. Conclusion

In this paper, a new approach to information management of mobile devices in a MANET is discussed. The approach uses a context-aware framework to represent, store and exchange relevant information of mobile devices, where dead loops and rejoining issues are effectively eliminated. This is mainly achieved by setting up appropriate heart-beat timer and lease timer. However, a trade-off is required to set the lease timer since a large lease timer can also bring other issues. The update of context information can be delayed with the extension of the lease timer, which hence renders it out-of-date.

Acknowledgements

This work was supported by A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions. It was also funded by The Project Sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, Nanjing.

References

[1] A. Saeed and T. Waheed, "An Extensive Survey of Context-aware Middleware Architectures", 2010 IEEE International Conference on Electro/Information Technology (EIT), **(2010)**, pp. 1-6.
[2] X. X. Diao, "Cooperative Inter-vehicle Communication Protocol Dedicated to Intelligent Transport Systems", in New Technologies, Mobile and Security, **(2008)**, NTMS"08.
[3] K. M. Hou, "LiveNode: LIMOS versatile embedded wireless sensor node", Journal of Harbin Institute of technology, vol. 32, **(2007)**, pp. 139-144.
[4] R. Eigner and C. Mair, "Using Context Ontologies for Addressing and Routing in Mobile Ad Hoc Networks", in International Conference on Networking, **(2009)**.
[5] K. Yang, "Context Aware Vertical Soft Handoff Algorithm For Heterogeneous Wireless Networks", in Vehicular Technology Conference, **(2008)**, VTC 2008-Fall, IEEE 68th.
[6] S. Moon, "Visualization and Representation of Mobile Network Users, in Sensor, Mesh and Ad Hoc Communications and Networks", 5th Annual IEEE Communications Society Conference on SECON"08, **(2008)**.
[7] A. Dey, "Understanding and Using Context", Personal Ubiquitous Comput., vol. 5, Issue 1, **(2001)**, pp. 4-7.
[8] M. Perttunen, J. Riekki and O. Lassila, "Context Representation and Reasoning in Pervasive Computing: a Review", International Journal of Multimedia and Ubiquitous Engineering, vol. 4, no. 4, **(2009)**.
[9] M. Baldauf, S. Dustdar and F. Rosenberg, "A survey on context-aware systems", International Journal of Ad Hoc and Ubiquitous Computing, vol. 2, no. 4, **(2007)**, pp. 263-277.
[10] M. D'Angelo, "A Layered Approach to Modelling and Design of Cross-Layer Protocols in Ad-Hoc Wireless networks", in Wireless Communications and Networking Conference, **(2008)**, WCNC 2008, IEEE.
[11] A. A. Bakar, "Forming Trust in Mobile Ad Hoc Network", in Communications and Mobile Computing, **(2009)**, WRI International Conference on CMC"09.
[12] B. Garbinato, "Six-Shot Broadcast: a context-aware algorithm for efficient message diffusion in MANETs", Lecture Notes in Computer Science, **(2008)**.
[13] S. Buchegger and J. -Y. Le Boudee, "Self-policing mobile ad hoc networks by reputation systems", Communications Magazine, IEEE, vol. 43, no. 7, **(2005)**, pp. 101-107.
[14] M. Musolesi, "Adaptive routing for intermittently connected mobile ad hoc networks", in World of Wireless Mobile and Multimedia Networks, **(2005)**, Sixth IEEE International Symposium on WoWMoM 2005.
[15] M. Musolesi and C. Mascolo, "Evaluating Context Information Predictability for Autonomic Communication", in World of Wireless, Mobile and Multimedia Networks, **(2006)**, International Symposium on WoWMoM 2006.
[16] R. Handorean, "Context aware session management for services in ad hoc networks" IEEE International Conference on Services Computing, **(2005)**.
[17] K. C. Lee and H. Cho, "Designing a Ubiquitous Decision Support Engine for Context Prediction: General Bayesian Network Approach", International Journal of u- and e- Service, Science and Technology, vol. 3, no. 3, **(2010)**, SERSC.
[18] O. Riva, "Context-Aware Migratory Services in Ad Hoc Networks", IEEE Transactions on Mobile Computing, vol. 6, no. 12, **(2007)**, pp. 1313-1328.
[19] J. Yamamoto, H. Nakagawa, K. Nakayama, Y. Tahara and A. Ohsuga, "A Context Sharing Message Broker Architecture to Enhance Interoperability in Changeable Environments", In: Mobile Ubiquitous Computing, Systems, Services and Technologies, **(2009)**, pp. 31-39.
[20] S. L. Kiani, M. Knappmeyery, N. Baker and B. Moltchanov, "A Federated Broker Architecture for Large Scale Context Dissemination", In: Computer and Information Technology (CIT), **(2010)**, pp. 2964-2969.
[21] R. P. Pinto, "A Component Framework for Context-Awareness", in Wireless Communications and Mobile Computing Conference, **(2008)**.
[22] D. Dudkowski, "Design and Implementation of a Reference Modelfor Context Management in Mobile Ad Hoc Networks", in Proceedings ofthe 22nd International Conference on Advanced Information Networkingand Applications, **(2008)**.
[23] Y. Chen, "LMSS A location-based multimedia service systemusing wireless networks", in 8th International Conference of IntelligentSystems Design and Application, **(2008)**.

[24] T. Ma, J. Huang and G. Wu, "An Architecture and Supporting Environment of Service-Oriented Computing Based-on Context Awareness", International Journal of Grid and Distributed Computing, vol. 2, no. 4, **(2009)**.
[25] A. N. Mian, R. Baldoni and R. Beraldi, "A Survey of Service Discovery Protocols in Multihop Mobile Ad Hoc Networks", Pervasive Computing, IEEE, vol. 8, Issue 1, **(2009)**, pp. 66-74.
[26] C. Cho and D. Lee, "Survey of Service Discovery Architectures for Mobile Ad Hoc Networks", Term Paper, Department of Computer and Information Science and Engineering (CICE), University of Florida, **(2005)**.
[27] U. C. Kozat and L. Tassiulas, "Service Discovery in Mobile Ad Hoc Networks: an Overall Perspective on Architectural Choices and Network Layer Support Issues", Ad Hoc Networks, vol. 2, issue 1, **(2004)**, pp. 23-44.
[28] A. El Saoud, M. T. Kunz and S. Mahmoud, "SLPManet: Service Location Protocol for MANET", in Proceedings of the International Wireless Communications and Mobile Computing Conference IWCMC 2006, **(2006)**, pp. 701 -706.
[29] M. Abou, "BENCHManet: An Evaluation Framework for Service Discovery Protocols in MANET", in Sensor and Ad Hoc Communications and Networks, **(2006)**, SECON 2006, pp. 860-865.
[30] Q. Liu, N. Linge and F. Gao, "A Context-aware Protocol for Location Discovery Exchange in a Ubiquitous Environment", International Journal of Future Generation Communication and Networking, vol. 7, no. 2, **(2012)**, pp. 177-182.

Authors

Qi Liu

BSc (2002), MSc (2006), PhD (2010) is an appointed Professor at the Nanjing University of Information Science and Technology. His research interests include context-awareness, data communication in MANET and WSN, smart home and energy efficiency. He also devotes time to WSN solutions on intelligent agriculture in protected field. He is a member of IEEE and ACM.

Nigel Linge

BSC (1983), PhD (1987) is a Professor of Telecommunications at the University of Salford. He is an electronic engineer by profession and has research interests in network design, context aware information systems, communication protocols, mobile systems and practical applications of networking technology in areas such as energy and building monitoring. He also devotes time to the preservation and dissemination of the history of telecommunications and provides external advice to organization on their heritage strategies. He is a Fellow of the IET and BCS.