International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 10 (2015) pp. 25643-25652

http://www.ripublication.com

Towards An Alternative Middleware Approach For Mobile Ad Hoc Networks

[1]Nor Effendy Othman andRosilah Hassan

Network and Communication Technology Lab,
Research Center for Software Technology and Management,
Faculty of Information Science and Technology, UniversitiKebangsaan Malaysia, Selangor, Malaysia.
E-mail: [1]effendy@ukm.edu.my

ABSTRACT

One of the challenges in developing applications for Mobile Ad Hoc Networks (MANETs) is the mobility of nodes and the sharing of the communication medium in contrast to traditional networks. These characteristics lead to bad performance of traditional transport protocols and middleware solutions in this networkeven when the routing problem has been solved. A middleware solution for MANETs needs to take the differences into account and if possible exploit them. In this paper, we discuss currently available middleware solutions for MANETs as well as classifying them into programming paradigms. To evaluate these middleware approaches, we developed a twitter-like application, TwitHoc to investigate the suitability and reliability of these solutions for applications in MANETs. In contrast to Twitter, which is based on a centralised architecture with clients connecting to a coordinating server, TwitHoc cannot rely on a single central server. The development of this application with currently available middleware approaches will exhibit how these approaches handle the control of consistency and distribution of information among the devices of the users.

Keywords: MANET, middleware, programming approach, application development

1. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are wireless networksthat have no fixed infrastructure. The typical deployment ofsuch a network is in conditions where other types of networkare not possible, for instance, in emergency and disasterrecovery environments. A typical example of a mobile ad hocnetwork is illustrated in Figure 1.

The topology in an ad hoc network can change frequentlybecause of node mobility. Mobile nodes in a MANETmove freely and dynamically to form an arbitrary andtemporary network structure without specific configuration.The high mobility of nodes also contribute to volatile networkconnection where frequent and temporary disconnections arecommon rather than exceptional compared to traditionalnetworks.

In order to communicate, nodes in MANETs may functionas both an end system and a router. The sharing of the wirelesscommunication medium between these nodes will result insignal interference from each nodes. Additionally, dealing withmobile devices introduce constraint resource issues in manyways, for instance, the power and memory size are oftenlimited. In terms of transmission range, nodes in MANETshave shorter range and limited communication capabilities.

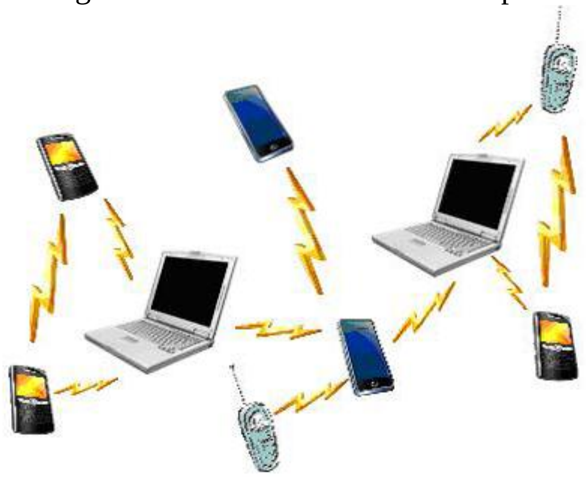

Figure1: A typical Mobile Ad Hoc Network

Although these unique characterizations introducechallenges in developing applications for MANETs, asignificant effort has been given particularly in this researcharea. However, most approaches involve solving the routingproblems with particular interest in designing transport layerrouting protocols. These solutions yet do not guaranteeto solve the problem arises due to the node mobility andcommunication medium sharing among nodes in MANETs.

In this paper, we discuss currently available middlewaresolutions for MANETs and classify them into types ofprogramming paradigm used throughout the developmentprocess. Further, for our investigation, we developed a sampleapplication, TwitHoc, to gain a better understanding andtest the suitability and reliability of these solutions for theapplications in MANETs.

Mascolo et al. [7] define a middleware as an abstraction tosimplify the programming task especially for the applicationlayer. The keys to efficient middleware services in an adhoc network are lightweight and able to support asynchronouscommunication between mobile nodes. These characteristicsare essential to deal with since mobile devices have limitedresources with significant small memory size, slow CPU speed and limited battery power. Furthermore, these mobile devicesconnect to the network intermittently thus, it is essential for a middleware system to be able to support asynchronouscommunication.

The goals of this research are as follow: 1) to identifyavailable middleware approaches for pure MANETs scenariosand presents the selected approaches and their characteristics.2) to classify these approaches and map according to MANETsrequirements. 3) to discuss appropriate broadcast techniques asone of the main services a middleware should provide. 4) tointroduce TwitHoc, a sample application developed to evaluateexisting middleware approaches for MANETs.

The remainder of this paper is organised as follows:Section II provides a summary on currently availablemiddleware approaches for MANETs and their advantagesand disadvantages. Section III lists the broadcastingmethods for MANETs as broadcast is one of the mainservices a middleware should provide to support effectivecommunication for mobile nodes. Section IV describes oursample application, TwitHoc which will be used to furtherinvestigate the suitability of these approaches. Section Vdiscusses the ideal characteristics a middleware designed forMANETs should possess. Later in section VI, we concludeour findings and further work in proposing an alternativemiddleware approach.

2. RELATED WORK

2.1 Data Sharing Approach

Data sharing approach focuses on sharing the data betweenmobile hosts and usually provides solutions on data structuresand mechanism to optimize the availability of the data amongmobile hosts.

Tuple space originally introduced in LINDA [13]coordination programming language as its form of sharedmemory. It introduces a method of exchanging messagesbetween two processes indirectly. Generally, the process whichhas data to share will add it to the tuple space and any processwho needs the data will acquire it from the tuple space. Thisinitial idea is then applied into mobile network scenarios inLIME (Linda in mobile environment) [1].

This middleware approach supports the development ofapplications exhibiting physical mobility of hosts, logicalmobility of agents, or both. It defines a tuple space foreach mobile host which is permanently associates with it.Additionally, these tuple spaces can merge with remotetuple spaces when they happen to be in the same coveragearea. This is a very efficient way of information sharingbetween connected mobile hosts. Besides, each mobile hostmaintains an interface tuple (ITS) that is attached to the hostpermanently. Every ITS has tuples in which the mobile hostwishes to share with other hosts.Due to the frequent disconnection properties in mobileenvironment, tuple space solutions offer opportunisticcommunication paradigm as it may exploit the connectivityamong mobile hosts whenever possible.

XMIDDLE [5] is a data sharing middleware that utilizes a robust tree data structure in each host, allowing the data to be accessed and modified by other nodes in the same communication range. In order to permit such communication, a private tree data structure at each host has access points. XMIDDLE attempts to be implemented in a complete ad hoc scenario with assumption of no wired network exists at all.

The process of sharing data in XMIDDLE consists of online collaboration, off line data manipulation and application dependent data reconciliation. Generally, when two hosts are connected to each other, they can perform the data sharing and modifying on the linked data trees. When the connection breaks or not available, the hosts preserved the replica of the shared tree before disconnection occur. At this point of time, the hosts can still manage to modify data on their own. Later when reconnection takes place between those nodes, the data tree is updated to the latest version. This way, the nodes only need to restore a small specific part of tree rather than to duplicate the whole data tree.

MESHMdl [6] is a middleware platform designed to suit ad hoc network environment. It is built based on the combination of two main technologies namely mobile agents and tuple space. Autonomous mobile agents introduce logical mobility and further represent the lower-level components in MESHMdl application. The main advantage of these mobile agents is that they possess their own control thread, which means they can possibly travel to other nodes based on their own decision. The tuple space paradigm is deployed for agent communication in MESHMdl. It allows applications to exchange data anonymously since tuples are addressed in an associative way by specifying their contents. Furthermore, tuple space presents a high degree of decoupling because of the entities wishing to communicate to each other do not have to be in the same location at the same time. The MESHMdl architecture consists of a generic connection layer, interaction layer, space, agent runtime called as Engine and agent application. Applications are defined as groups of mobile agents that collaborate among each other via spaces. The generic connection layer encapsulates networking technologies such as Bluetooth or WLAN and complements the upper layers with neighbor discovery mechanisms and communication method for interaction among them. By doing so, the middleware is independent from the underlying network layer.

The space layer is responsible to define the way of how spaces are managed and defined. The principle of JavaSpaces is adopted by using an object oriented implementation of the tuple space paradigm. The role of the space as a shared communication medium is twofold, as an inter-agent communication and agent-Engine communication. MESHMdl is mobility aware as the node level spaces are not federatedwhen two nodes meet. They tend to appear as separatespaces to the application agents. An engagement protocol isimplemented for communication purposes when two nodesare in the same transmission range. MESHMdl also supportsspace based information diffusion by creating and registeringa Xector, a mechanism used to propagate information throughthe network under certain constraints.

2.2 Event-based Models

Event based communication model is typically beingapplied in distributed applications which require itscomponents to react in the presence of a state change[2]. Event based systems are particularly sufficient fordistributed environments with the lack of centralize control,in particular, it is very well-suited a network type such asMANET.

STEAM [3] is an event based middleware service that hasbeen designed specifically to adapt to the wireless ad hocnetwork environment. The main approach is to propagatingevents within a certain area surrounding producers which willlimit forwarding of event messages, and therefore reduces theusage of communication and computation resources, which aretypically scarce in mobile environments.

Out of three event models outlined in Meier et al. [2], STEAMchose to implement an implicit event model. The argumentnot to choose the other two types (peer to peer and mediator)is because both of those models need the entities to locateintermediate components, which is inappropriate for ad hocenvironments.

Three different types of event filter were employed inSTEAM. On the producer side, the subject and proximityfilters are imposed which allow consumer to express interestin a set of events based on their type and the geographical areain which they are propagated. Whereas, on the consumer side,the content filters are being exploited and when an instance ofan event is received, it will be evaluated to determine whetherthe event should be delivered to the application. The advantageof this novel combination is that it is easier to maintain thesubscriptions and content filters in a system possesses dynamicreconfiguration requirement due to the mobility of the entitiesin wireless ad hoc network.

STEAM tackles scalability issue in two ways. First, itincreases the ability of a system to grow by limiting thepropagation of events in space. Second, the use of combinationfilters so that events will be matched efficiently. An applicationis permitted to outline the scope in geographical space(proximity filter) within which some information (subjectfilter).

In terms of delivery time, STEAM allows applications todefine delivery deadlines and to assign them to specific events.A dispatcher then exploits these deadlines to determine thetime to deliver the corresponding events to the subscribedconsumers.

2.3 Message Oriented Middleware

One example of this middleware type is EMMA [4]which is based platform which adapts Java Message Service(JMS) and originally designed for partially connected adhoc scenarios. It applies JMS specifications in dealing withMANET characteristics but with some minor modificationsin order to suit the mobile scenarios. EMMA uses epidemicrouting as the medium to send messages to mobile hosts. Fromthe application perspective, EMMA is designed to supportboth point to point (P2P) and publish-subscribe model. TheP2P model is based on the queue concept which enablesasynchronous communication between the message producerand potential consumers. The advertisement of the queue isdone periodically to mobile hosts reside in the transmissionrange. Asynchronous epidemic routing protocol is used todeliver messages to hosts out of range.

As for the publish subscribe model, the topics are advertisedthe same way as advertising queue in P2P. Any mobile hostsinterested to subscribe a particular topic has to register withthe host who hold the topic. Topic holders will then spread themessage to all subscribers by using synchronous protocol ifpossible or else, epidemic protocol is being used. EMMA alsoprovides a mechanism for failure notification if acknowledgepacket is not being received within a certain timeout.

3. BROADCASTING IN MANETS

Broadcasting is the most common way for nodes in aMANET to communicate with each other. Thus, it is oneof the main services a middleware should provide to ensureeffective communication between mobile nodes. Because ofthe rapid topology change behavior, the easiest solution totransmit messages to the neighborhood nodes is to use themessage broadcast approaches. In general, broadcasting can besimply described as a process in which one node will send apacket message to all other nodes in the entire network withinits transmission range [8].

3.1 Flooding Technique

Flooding technique is the most fundamental and straightforwardapproach to broadcast message in a MANET.Typically, a source node will transmit a message packet toits neighbor nodes and afterwards, each of these nodes willretransmit the packet to their neighbors. This process willcontinue until all nodes in the network receive the particularpacket. Because of the properties of each nodes who willnaturally rebroadcast the message packet, flooding method canbe totally topology independence.

Nevertheless, basic flooding approach may lead to a seriouscongestion problem with many redundant messages beingtransmitted, which Ni et al. [11] define it as broadcast stormproblem. Among the main causes in contributing to the stormproblem are redundant transmission range of each nodes andthe location of the rebroadcasting nodes might be close toeach other. This will result in many superfluous rebroadcast messages collide in the network and furthermore, contributingin unnecessary usage of limited bandwidth available in thiskind of network.

3.2 Probabilistic Broadcast

Probabilistic broadcast is an intuitive way to reduce rebroadcast problem that occurs in simple flooding techniques. It applies a similar concepts to flooding but in contrast to it improves in the sense of calculating a predetermined probability in order for a node to rebroadcast message packets. However, the main issue in designing probabilistic broadcast algorithm is to set the probability value itself. Common solution is to apply a fixed threshold value. Nevertheless, this value does not represent the global optimum probability for all nodes in the network. In a denser area where nodes are located close to each other, the possibility to produce redundant rebroadcast packets are quite high due to the overlap coverage area. Thus, in such area, the rebroadcast probability should be set to a lower value.

Contradictorily, in a sparser area where the distance between nodes are much higher, the probability value should be higher to ensure the transmitted package can reach all nodes in the entire network. To facilitate this requirement, Zhang et al. [12] proposed a dynamic probabilistic broadcasting where the probability to rebroadcast at every mobile nodes is adjusted dynamically based on the node's current location. This technique applies the density estimation at each node in order to set the rebroadcast probability value.

3.3 Epidemic Broadcast

Epidemic broadcast strategies are inspired from the biological dynamic epidemic phenomena in human populations. It has successfully gain researchers' attention in developing a new primitive for information dissemination in MANET. Musolesi et al. [10] proposed a middleware for ad hoc network which applies epidemic based information diffusion. The main goal of the developed algorithm is to achieve an accurate tuning of the message diffusion process with the specific network structure so that the message will reach the desired percentage of mobile nodes.

Due to its properties as a store and forward approach, this epidemic based strategies can work in the presence of disconnections, which is one of the main features of a MANET. Generally, a mobile host is categorized as infected if it accepts the intended message packet that is being sent in the network, whereas otherwise, a node is classified as susceptible, which means that it could be infected.

Among the important parameters in the developed protocol is the infectivity rate. The proposed middleware will have to calculate the value of the infectivity rate by evaluating connectivity among nodes and the removal rate of messages from the buffer at each node. On top of that, the fundamental concept in epidemiology is the basic reproductive numberwhich defines the number of secondary infected nodesgenerated by one primary infective node. It is crucial to notethat in order for the epidemic broadcast to take place, the reproductive number has to be larger than one.

4. TWITHOC ARCHITECTURE DESIGN

To further understand and evaluate the above existing middleware approaches, we developed a twitter-like application, named TwitHoc. TwitHoc is used to test the suitability of these approaches in supporting the applications built specifically for MANETs. The idea of this simple application is to be a tool for sending and receiving short message services in an ad hoc scenario. One possible domain we foreseen is in the disaster rescue operations where the network infrastructure is totally destroyed. Rescuers will be using mobile devices to form an ad hoc network and start communicate with each other via TwitHoc over MANET where possible.

TwitHoc clients at each mobile nodes can send out two type of messages; public or private messages. Public messages is intended to be delivered to all other discovered nodes while private messages are delivered to a certain receiver. As this is being implemented in a MANET, we have to bear in mind that disconnection and link failure may occur randomly while these nodes are sending or receiving messages. Since LIME is implementing tuple space technology, it can handle the disconnection quite well. However, the question of message guarantee becomes crucial, especially in emergency situations.

In contrast to twitter, which is based on a centralised architecture with clients connecting to a coordinating server, TwitHoc cannot rely on a single central server.

The development of this application with currently available middleware approaches [1], [3] will exhibit how these approaches handle the control of consistency and distribution of information among the devices of the users. Figure 2 shows the simplified architecture of TwitHoc on top of selected existing middleware approaches. The mapping layer is wrote to map the services provided by each solutions to TwitHoc API.

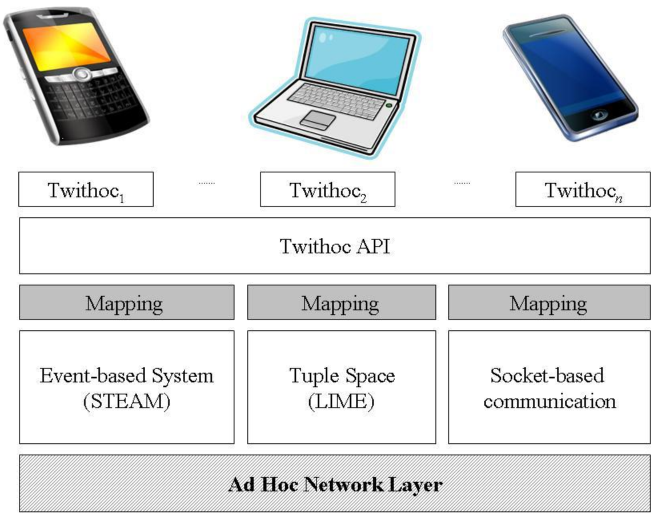

Figure 2: General architecture of Twithoc, a twitter-like application implemented in MANETs

5. DISCUSSION OF AN IDEAL MIDDLEWARE

In this section, we discuss three main characteristics thata middleware for MANETS has to inherit to ensure efficientapplication support.

- *Scalability*. Scalability is an important characteristic to beimplemented in designing efficient middleware service forMANETs. It has to be able to maintain certain level of desiredperformance as the application system gets bigger.
- *Mobility*. MANET demonstrates a dynamic nature ofnetwork type, thus the topology in it is typically unpredictableand may change frequently.Therefore, a developed solution forMANET should be able to adapt to nodes movement patternand topology changes.
- *Availability and Frequent Disconnection Handling*. Asdiscussed before, nodes in a MANET are highly mobileand either temporary or permanent disconnections occurvery frequently. Several identified causes include wirelesschannel variation, the mobility of destination or intermediatenodes or even mobile hosts dying out of power resources.Sometimes, disconnections may extend to unknown timeperiod, thus application layers have to deal with all thesepossible scenarios.

Table 1 presents the selected middleware approachesmapped to the identified characteristics.

Table 1: Middleware projects for MANETSand their characteristics

Approach	Programming Abstraction	Scalability	Mobility	Availability
STEAM	Event based	Yes	Yes	Yes
EMMA	Message Oriented	Yes	Yes	Partial
LIME	Tuple Space	Yes	Yes	Partial
MESHMdl	Tuple Space	Yes	Yes	Yes
XMIDDLE	Data Sharing	Partial	Yes	Partial

6. CONCLUSION

In general, several existing middleware approaches havebeen listed and classified into specific programming paradigmused in developing these solutions. These approaches isevaluated by developing a sample application, TwitHoc. Themain idea of TwitHoc is to be a tool of sending and receivingshort message services in an ad hoc scenarios. As discussedearlier, the purpose of this step is to see how each ofthese approaches handle the control of message distributionbetween the participate nodes in a MANET. This paperalso lists the broadcasting methods for MANETs as one ofthe main services a successful middleware should handle isthe broadcasting and message distributing among the mobilenodes. Future work will be based on the observation andfeedback obtained from the testing and experiments conductedof which from this ground, appropriate middleware serviceswill be designed to cater MANET specific requirements.

ACKNOWLEDGMENT:

This research was funded by the UniversitiKebangsaan Malaysia under Grant No. GGPM-2014-045.

REFERENCES:

[1] Murphy, A. L., Picco, G. P., Roman, G. C., 2006, LIME: A CoordinationModel and Middleware Supporting Mobility of Hosts and Agents. *ACMTransactions on Software Engineering and Methodology*, July 2006,15(3), pp.279-328.

[2] Meier, R. and Cahill, V., 2005, Taxonomy of Distributed Event-Based Programming Systems. *The Computer Journal*, June 2005, 48(5),pp.602-626.

[3] Meier, R. and Cahill, V., 2002, STEAM: Event-Based Middlewarefor Wireless Ad Hoc Networks. *Proceedings of the InternationalWorkshop on Distributed Event-Based Systems (ICDCS/DEBS'02)*,Vienna, Austria: IEEE Computer Society, pp.639-644.

[4] Musolesi, M., Mascolo, C., Hailes, S., 2005, EMMA: EpidemicMessaging

Middleware for Ad hoc networks. *Personal and UbiquitousComputing*, Springer, December 2005, 10(1), pp.28-36.
[5] Mascolo, C., Capra, L., Zachariadis, S., 2002, XMIDDLE: A DatasharingMiddleware for Mobile Computing. *Journal on Personal andWireless Communications*, April 2002, 21(1), pp.77-103.
[6] Herrmann, K., 2003, MESHMdl A Middleware for Self-Organizationin Ad Hoc Networks. *23rd International Conference on DistributedComputing Systems Workshops (ICDCSW'03)*, May 2003, 21(1), pp.446-451.
[7] Mascolo, C., Capra, L., Emmerich, W., 2002, Middleware for MobileComputing. Advanced Lectures on Networking, *Lecture Notes inComputer Science, Springer*, Berlin Heidelberg New York, 2497, pp.20-58.
[8] Coulouris, G., Dollimore, J., Kindberg, T., 2005. *Distributed Systems:Concepts and Design*. 4th ed. Harlow: Addison-Wesley.
[9] Murthy, C. S. R., Manoj, B. S., 2004. *Ad Hoc Wireless Networks:Architectures and Protocols*. Upper Saddle River, N.J.;[Great Britain]:Prentice Hall.
[10] Tanenbaum, A. S., Steen, M. V, 2007. *Distributed Systems: Principlesand Paradigms*. 2nd ed. Harlow: Prentice Hall.
[11] Musolesi, M., Mascolo, C., 2006, Controlled Epidemic-styleDissemination Middleware for Mobile Ad Hoc Networks. *Proceedingsof Int. Conference on Mobile and Ubiquitous Systems: Networks andServices (MOBIQUITOUS 2006)*, July 2006, pp.1-9.
[12] Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P., 2006, The BroadcastStorm Problem in a Mobile Ad Hoc Network. *Proceedings of the FifthAnnual ACM/IEEE International Conference on Mobile Computing andNetworking*, Aug 1999, pp.151-162.
[13] Zhang, Q., Agrawal, D. P., 2005, Dynamic Probabilistic Broadcastingin MANETs. *Journal of Parallel and Distributed Computing*, Feb 2005,65(2), pp.220-233.
[14] Carriero, N., Gelernter, D., 1989, Linda in Context. *Communications ofthe ACM*, April 1989, 32(4), pp.444-458.
[15] Murphy, A. L., Picco, G. P., 2006, Using LIME to Support Replicationfor Availability in Mobile Ad Hoc networks Proceedings of the the8th International Conference on Coordination Models and Languages(COORD06), June 2006, *Lecture Notes in Computer Science*, 4038.