International Journal of Computer Applications (0975 – 8887)
Volume 2 – No.8, June 2010

Concept of Middleware Services in Mobile Ad-hoc Networks

Anil Kumar
Dept. of C.S.E
H.C.T.M. Kaithal (INDIA)

Dr. Parveen Gupta
Dept. of C.S.E
APIIT, Panipat (INDIA)

Pankaj K. Verma
Dept. of C.S.E
H.C.T.M. Kaithal (INDIA)

Dr. Vijay Lamba
Dept. of E.C.E
HCTM, Kaithal (INDIA)

ABSTRACT
The increasing diffusion of wireless devices suggests novel service deployment scenarios where there are no constraints on device mobility and distributed applications are the result of impromptu collaborations among wireless peers. Mobile adhoc network (MANET) makes it possible to dynamic changes in topology and in the availability of resources from different sources. Different sources can be of Laptops, PDAs, Desktops, mobile phones etc. A middleware generalize the concept of collaborating among these different kinds of devices. We focus on the point to develop middleware services that provide services for information sharing and retrieval with replication in mobile ad-hoc networks, because it is very critical to share information in mobile ad-hoc network infrastructures.

Categories and Subject Descriptors
[**Mobile Adhoc Networks**]: MANET Services and Applications – *Infrastructure less Communication, Variation in structure, Middleware Services, Replica Dissemination, Replica Retrieval, Replica Degree Maintenance.*

General Terms
MANET Configuration, Civilization Game, Replication Strategy, Middleware Concept, Resource Delegates, Resource Manager.

Keywords
MANET, Middleware, Civilization Game, Replica Degree Maintenance, DMC.

1. INTRODUCTION
Mobile ad-hoc networking has been continuous emerging research area from past few years. Typical application scenarios for mobile ad-hoc networks include emergency situations in which wireless devices are used to coordinate the efforts of rescue personal and different meetings in which different devices are connected in a wireless network to help in collaboration of the participants and increase the meeting's efficiency. This is for example reflected by the IETF working group MANET (Mobile Ad-Hoc Networks). There is no doubt that routing and service location problems have to be solved to provide a working infrastructure in mobile ad-hoc networks. Instead, we propose to develop a set of generic services, or middleware services, that support the development of applications and services for mobile ad-hoc networks. It is the goal of this position paper to (1) analyze the requirements of ad-hoc network services, (2) replication strategy in MANET (3) to outline our approach to develop middleware services for information sharing

2. MOBILE AD-HOC NETWORK SERVICES
Application domains for ad-hoc networks include E-learning [1], vehicular communication [2] collaborative electronic shopping [3] and game playing at a specific place. It is the purpose of this section to identify characteristics and requirements that are common to these and other ad-hoc network services with respect to information disseminate, sharing and retrieving.

2.1. Characteristics
There are three generic characteristics that ad-hoc network services disseminate and share: (1) information disseminates and sharing is very critical mission, (2) integration of three classes of information, and (3) maximum inter-cooperation is required, but not always desired. The simplest ad-hoc network services have no other purpose than sharing some resources, e.g., a printer. However, services such as those mentioned above are based on dissemination, sharing and retrieving of information. In E-business services, more, accurate information might lead to better deals. E-learning services are only possible if teaching material is accessible. Each device has some data; some of these may provide useful information for other device in the ad-hoc network. It is a major problem to access information because devices change their position dynamically. Mobile devices with new information may enter the ad-hoc network, while others may leave and leave some information inaccessible. Generally, it is very critical to predict which mobile device will be the part of a ad-hoc network. Information about physical location and user entry and exit is also a major problem.

2.2 Requirements
Information is stored and managed in the system as data. Based on the discussion above, we identify the following functional high-level requirements of application with respect to data access and sharing:

• Disseminates and access to all relevant data and resources: resources are the major elements which contained by all participants it is important to consider the highly dynamic nature of ad-hoc networks. Otherwise, a node that keeps a unique copy of mission critical data could leave the range of the ad-hoc network and the data is "lost."

• Collecting and giving structure to accessible data: when data is collected from different devices it may be in unstructured format so it might decrease the efficiency of the user. Thus, the accessible data has to be structured in such a way that the end-user is able to access quickly the important data. Since the data is collected from sources that might be only temporary part of the adhoc network it is important to find solutions for the dynamic nature of data availability.

• Use middleware to control the Access of resources and data: there is a middleware which play very important roll in resource manipulations like resource disseminations, retrievals among participating nodes. Middleware use the concept of replication of the available resources. When it is required utilize it.

3. REPLICATION STRATEGY IN MANET WITH AN EXAMPLE

To practically introduce the Replication middleware functions, let us consider travelers with Replication middleware and components (with the desired resource-associated replication degree) to a subset of devices in the waiting room at Airport. For instance, consider the well-known single-player Civilization strategy game [4] and let us illustrate how Replication Strategy can distribute replicas of Civilization game components, to support game availability in the dense MANET with no need of explicit and static installations, even if mobile nodes continuously enter/exit the waiting room. Consider the example of deployment scenario depicted in Figure 1. Suppose that a device enters the dense MANET by carrying new game components (the Civilization server engine, the Civilization lightweight client, and some playing sceneries). Replication Strategy classifies node responsibilities according to two main roles:

• *Resource delegates* are dense MANET nodes that host resource replicas. Resource delegates help in the dissemination of information. They dynamically change their position like Entry and Exit from the range of the network circle. They are not only resource giver but also retrieve the information as they required any time with the help of middleware.

• *Replica managers* are dynamically elected MANET nodes in charge of `determining and maintaining the proper replication degree for the associated resources.

Here we explain the working concept of Civilization Server for Replication Strategy, when the delegate D with the Civilization server engine enters the waiting room; the Replication Strategy middleware component running on D's device sends the descriptions of D's resources to the Replication Strategy replica manager M, running in another node of the dense MANET (message 1). The replica manager is in charge of enforcing the needed replication degree and of maintaining information about shared resources replicated in the dense region (Shared Resource Table - SRT). For any shared resource, the corresponding SRT entry includes the associated target replication degree and weakly consistent information about the nodes where the resource is currently replicated. Messages pass to node to node in a desirable and discrete mode in a consistent agreement among them. If M decides that D's resources should be replicated, it commands D to start the replication operations (message 2).

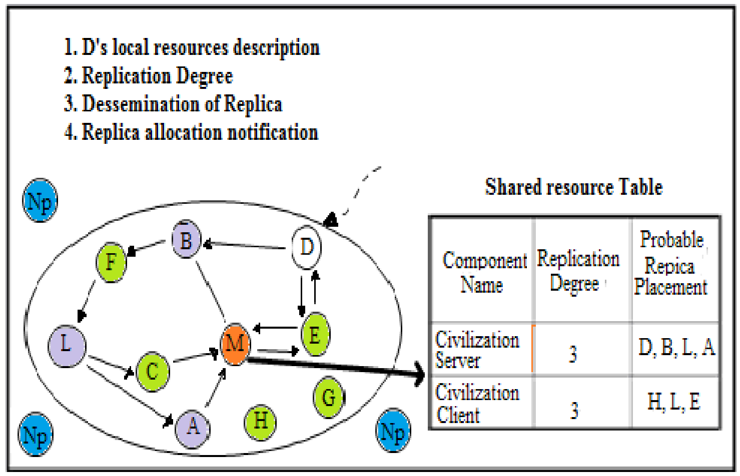

Figure 1: Replication Strategy for Civilization server

D reacts by forwarding a resource replica to a randomly chosen neighbor (message 3 to B), If the neighbor accepts to locally store the resource, it makes a local copy and recursively forwards the message, by decreasing the number of replicas yet to be instantiated. Otherwise, it only forwards the message. In the figure, node F refuses to host a server engine replica, while nodes B, L, and A accept to cooperate and notify their decisions to M (message 4).

There are two challenges for replication in MANETs:

• How to dynamically identify the nodes belonging to the dense region, given that the nodes continuously move.

• How to suitably choose the replica manager node, e.g., by taking into account its position in the dense MANET topology.

These are the two major issues for Replication Strategy middleware which directly relate to the actual distribution and retrieval of resource replicas. The addressed deployment scenarios require considering highly decentralized and lightweight solutions for dense MANET configuration and resource replication, capable of scaling well in execution environments with several hundreds of mobile wireless peers. Therefore, it is necessary to design and implement original mechanisms specialized for the peculiar characteristics of dense MANETs. In addition, to maintain the replication degree unchanged after first replica distribution, the Replication Strategy replica manager should react to possible exits of Civilization delegates from the dense MANET:

it should be notified of such kind of events, should understand the replicas of which resources are leaving, should identify other delegates for those leaving resources still available in the dense region (via the corresponding SRT entries), and should command new replicas. Note that there is not the need to guarantee that, for any replicated resource, at any moment, the desired replication degree is exactly enforced; it is sufficient to try to maintain the replication degree in a lazy consistent way, by avoiding that all the replicas of a resource leave the dense region before performing a further distribution of resource copies. When a new user enters the waiting hall, he can decide to play Civilization even if he has not yet installed the needed game components on her PDA. Replication Strategy supports his distributed discovery to understand which games are currently available in the dense MANET and which nodes host the desired resource replicas.

Then, Replication Strategy is in charge of downloading both the Civilization client and the playing scenery to her device. Once downloaded them locally and once identified via Replication Strategy a suitable node with a Civilization server engine replica, the client can interact with the remote engine by adopting usual MANET communication protocols, by passing the Replication Strategy middleware intermediation. The assumption of MANET guarantees that network partitioning will not occur during service provisioning. In the case that the used engine delegate leaves the waiting hall, the client can ask the Replication Strategy middleware for retrieving another engine replica in the dense MANET; then, the client can restart her gaming session by connecting to the newly retrieved replica. Let us remark that the above scenario only represents a possible Replication Strategy use case. Any other service where resources to replicate are read-only data (the list of train departures/arrivals, of movies on in town, of utility phone numbers, and of pictures taken at a rock concert that fans would like to share with other people among the public) is a simplification of the sketched case study, with no need of dynamic composition and distributed coordination of service components.

4. THE REPLICATION STRATEGY MIDDLEWARE

According to the above guidelines, we have designed and implemented the Replication Strategy application-level middleware [5, 6]. Replication Strategy disseminates replicas of common interest resources and maintains their desired replication degree, independently of unexpected node exits from the dense MANET. More formally, a dense MANET is defined as the set of MANET nodes DM(n) = {d0, . . . , d_{N-1}}, where i) $\forall$ j $\in$ [0, N − 1] d_j has at least n neighbors at single-hop distance, and ii) the spatial node density in the area where DM(n) nodes are is almost constant with regards to time. Given a resource with a desired replication degree k, Replication Strategy is in charge of instantiating and distributing k replicas of it, and of maintaining the k replication degree notwithstanding the changes in the composition of the DM(n) set. Entertainment service developers can simply benefit from Replication Strategy replica management facilities via very little modifications on their application code: they only need to include Resource Description Framework-based [7] descriptors for resources to share, to specify the suggested replication degree, and to invoke methods of the Replication Strategy API to command service-transparent resource replication and retrieval. Figure 2 depicts the Replication Strategy Middleware architecture organized in two layers of facilities. This architecture will help to implement the Replication Strategy.

MANET Configuration includes mechanisms for participant identification and manager election; on top of the configuration, Replication Strategy provides facilities for Replica Distribution (RD), Replica Retrieval (RR), and Replica Degree Maintenance (RDM). The Dense MANET Configuration (DMC**)** facility is in charge of determining the participants of the dense MANET DM(n), i.e., the subset of nodes that have more than n nodes at single-hop distance. In addition, DMC optimistically identifies when nodes enter/exit the dense MANET and triggers the dynamic election of replica managers by trying to minimize the number of hops required for their messages to reach any dense MANET participant.

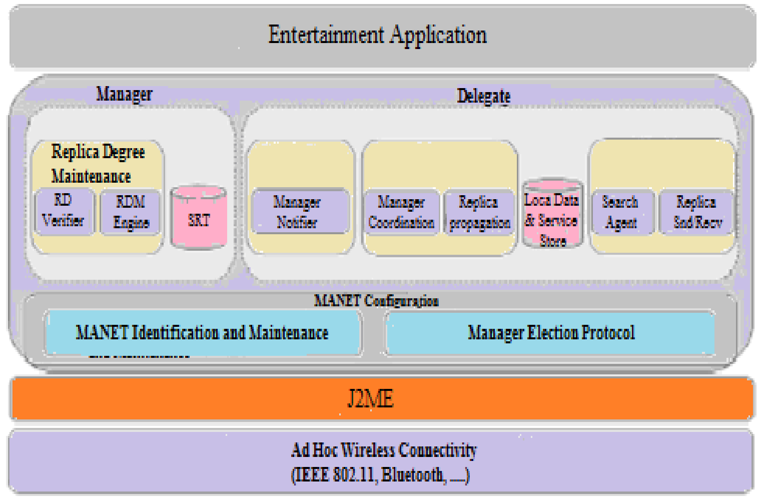

Figure 2: Replication Strategy Modular Architecture

The **RD** facility operates to transparently distribute resource replicas in the dense MANET. When a delegate enters a dense region, it communicates the metadata of its shared resources to the replica manager, which decides the replication degree to enforce, creates new SRT entries for newly arrived resources, and commands the delegate to start the replication operations. Replication Strategy distributes resource replicas to distant nodes located at the endpoints of straight paths. The **RR** facility has the goal of effectively retrieving resource replicas, by exploiting lightweight distributed search protocols. RR provides simple retrieval solutions, whose effectiveness mainly depends on the chosen diffusion strategy.

The **RDM** facility works to maintain unchanged the replication degree enforced for each shared resource, without strictly guaranteeing consistency: it is possible to have time intervals when the requested replication degree differs from the actual number of replicas in the dense MANET. RDM mainly works by reacting to resource delegates leaving the dense region. To face also delegate abrupt and non-detected failures/exits, at large time intervals RDM checks the number of available replicas and possibly commands the manager to coordinate the distribution of additional copies. There should be a good strategy to handle these facilities because these are big challenges. It is important to not that how would be implemented and designed RR, RD and RDM in detailed technical description followed by Replication Strategy. So, layered MANET configuration help in all these processes.

5 ACKNOWLEDGMENTS

We would like to thanks Dr. Vikram Singh, Chairman of Computer Department, CDLU Sirsa, INDIA and Dr. Dilbag Singh, Reader in Computer Department, CDLU, Sirsa INDIA for their gentle guidance and also thankful to our friends they prompt us to do research work. Internet and Communication Laboratory of HCTM, Kaithal provide us all facilities and environment to development and complete the concept.

6. CONCLUSION

In this short paper, we have briefly discussed the requirements of ad-hoc network, given an overview on related work and outlined how we approach the development of middleware services that fulfill the application requirements. In the full version of this paper we will give a more detailed discussion and analysis of requirements and related work, and especially elaborate more in detail on our approach towards middleware services for information sharing in ad-hoc networks.

7. REFERENCES

[1] B. König-Ries, M. Klein, "Information Services to Support E-Learning in Ad Hoc Networks", *1st International Workshop on Wireless Information Systems (WIS 2002)*, April, 2002, Ciudad Real, Spain.

[2] Franz, W.; Eberhardt, R.; Luckenbach, T., "FleetNet - Internet on the Road", *8th WorldCongress on Intelligent Transportation Systems (ITS 2001)*, Sydney, Australia, Oct. 2001

[3] L. Capra, W. Emmerich and C. Mascolo. "Middleware for Mobile Computing: Awareness vs. Transparency (Position Summary)", *8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, Schloss Elmau, Germany, May 2001.

[4] Sid Meier's Civilization. http://www.civ3.com/.

[5] P. Bellavista, A. Corradi, and E. Magistretti. Lightweight autonomic dissemination of entertainment services in wide-scale wireless environments. *IEEE Communications Magazine*, 1(6), June 2005.

[6] P. Bellavista, A. Corradi, and E. Magistretti. Redman: a decentralized middleware solution for cooperative replication in dense manets. In *IEEE PerWare Workshop*, Mar. 2005.

[7] S. Decker, P. Mitra, and S. Melnik. Framework for the semantic web: an rdf tutorial. *IEEE Internet Computing*, 4(6), 2000.