Middleware Issues and Approaches for Mobile Ad hoc Networks

Salem Hadim, Jameela Al-Jaroodi, Nader Mohamed
Dependable Systems Middleware (DeSyM) Research Lab.
Department of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, NJ 07030, USA
{shadim, jaljaroo, nmohamed}@stevens.edu

Abstract
Recently, there has been a growing interest on mobile ad hoc networks (MANET) that offer a broad spectrum of new applications. However, application development for mobile ad-hoc networks is not easy. Obviously, solving these issues for each application from scratch is not a feasible approach. Instead, developing middleware services to support the development of such applications is a novel approach that will offer many possibilities, and considerable flexibility for ad hoc networking. This survey shows the current state of the research, in this domain by presenting and discussing some representative middleware for wireless MANET. The selection of the studied methods tries to cover as many views of objective reflections as possible. We concentrate on discovering similarities and differences between the studied approaches by making comparisons, evaluation and appropriateness studies. Then we argue that most of the work is at fledging stage and there is still a long way to go for such middleware to fully meet MANET applications requirements.

1. Introduction
Our future living environments are likely to rely on information provided by various types of devices connected over different types of networks. The increasing popularity of wireless devices like personal digital assistants, mobile phones and handhelds are offering a new spectrum of applications in a highly dynamic society. It is foreseeable that the interaction between computing and communication will increase drastically and will make it possible for fixed devices to incorporate microprocessors with communications capabilities. Such devices may include household appliances, computing devices embedded in cars, etc. This will extend the field of communication to a fully pervasive computing environment.

Unlike a fixed wireless network, wireless ad-hoc or on-the-fly networks are characterized by the lack of infrastructure and centralized authority. Nodes in a mobile ad-hoc network are free to move and organize themselves in an arbitrary fashion. MANETs are very dynamic in terms of available communication partners and network resources, connectivity and bandwidth. Furthermore, end-user devices are very heterogeneous, ranging from high-end laptops to low-end pads and mobile phones. In addition, the resources are limited in terms of available memory, CPU speed and battery power. In addition, the available bandwidth is much smaller compared to wired networks.

There is a broad spectrum of potentially useful applications for MANET, but application development in this domain is not easy. Obviously, solving these issues in every new application from scratch is not feasible. Instead, middleware services that support development of applications for mobile ad-hoc networks is a novel approach that will offer information access and sharing, and considerable flexibility for MANET. Due to the unique characteristics of MANETs, traditional middleware solutions that assume a relatively fixed network infrastructure are not suitable. Most traditional paradigms adopt a synchronous models of communication and generally are not resource aware. In recent years, there has been a growing interest in designing a middleware layer that fully meets the needs of MANET applications. In this paper we investigate some representative middleware solutions for MANETs and provide a state of the art evaluation and comparison of the approaches. The selection of the studied approaches is based on how innovative they are in supplying new concepts and solutions to cope with ad hoc scenarios requirements.

The remainder of the paper is structured as follows. Section 2 surveys some related work carried out in the field then section 3 outlines the most relevant challenges faced in middleware design for MANETs. In Section 4 we describe the studied research projects and approaches undertaken towards this perspective. Section 5 presents an evaluation and comparison of the projects. In section 6 we discuss some open research issues, and then we conclude the paper in Section 7.

2. DISCUSSION AND RELATED WORK
Here we discuss some existing work and aspects related to our research in middleware for mobile ad-hoc networks. First it is important to mention that limited work has been done on surveying middleware for mobile distributed systems in general and MANETs in particular due to the complexity and the non obvious taxonomy of the available middleware solutions. The survey [13] constitutes a thorough study of middleware for distributed systems whether fixed, nomadic or mobile. However, at some extend, the paper failed in surveying many relevant approaches with proven results in an ad hoc environment. Our work is among the rare surveys that really focus on a pure ad hoc environment and provides a selection of well suited work in the field.

Several middleware solutions have been proposed for distributed systems, generally with heavy computational load often adapting synchronous communication style. These approaches are more suited for fixed distributed systems since devices are resources rich and high steady bandwidth is assured by the wired links. Examples of such approaches are: Object Oriented middleware such as CORBA [21], Microsoft
The transceivers should be automatically turned on and off. The middleware should provide low level programming models to meet the major challenge of bridging the gap between hardware’s raw potential and the needed activities. It should establish system mechanisms interfacing to the various types of hardware and network systems, only supported by basic distributed primitive operating system abstractions. This will enable covering a wide range of application.

- **Heterogeneity**: The middleware should provide low level programming models to meet the major challenge of bridging the gap between hardware’s raw potential and the needed activities. It should establish system mechanisms interfacing to the various types of hardware and network systems, only supported by basic distributed primitive operating system abstractions. This will enable covering a wide range of application.

- **Mobility and Network Topology**: Due to the dynamic nature of a MANET, it exhibits frequent and unpredictable topology changes. The mobile nodes dynamically establish routes among themselves as they move about; moreover a user in the MANET may not only operate within the ad-hoc network, but may also require access to a public fixed network. MANETs therefore should be able to adapt the traffic and propagation conditions to the mobility patterns of the nodes.

- **Scalability**: If an application gets bigger, the network should be flexible enough to allow the addition of more nodes anywhere any time without affecting the network performance. Efficient middleware services must be capable of maintaining acceptable levels of performance as the network grows larger.

- **Limited Resources**: Middleware should provide mechanisms for efficient use of processing, memory and communication resources, while maintaining low power consumption. A node should accomplish its basic operations [20] without resources exhaustion. As an example of energy aware middleware, most of the device’s components including the transceivers should be automatically turned on and off based on the application requirements.

- **Quality of Service**: An important and unique property of middleware for MANETS is dictated by the design principles of application knowledge [13]. However middleware has to include mechanisms for injecting application knowledge in the infrastructure of the network. This allows mapping application communication requirements to network parameters for fine-tuning the network monitoring process. Most ad hoc networks applications dictate minimum quality of service (QoS) requirements sustained over an extended period of time. Middleware should be able to support QoS and dynamically adjust to changes in QoS requirements.

- **Context awareness**: It is a general term used to encapsulate almost all characteristics of mobile ad hoc applications. Context means every aspect that can impact the behavior of an application; therefore the middleware should be context aware [14]. We can distinguish two types of awareness: device awareness and environment awareness. Device awareness relates to the internal resources of the device: battery power, processing power, and memory. Environment awareness relates to external resources around the device such as network connectivity, bandwidth, location, and other hosts in range.

- **Security**: Providing communication among hosts in a hostile environment is a primary concern. Unique characteristics of MANETs pose various challenges to the security design such as open peer-to-peer (P2P) network architecture, a shared wireless medium and a highly dynamic topology. These challenges raised the requirement of developing secure solutions that achieve wider protection, while maintaining desirable network performance. There is no standard security mechanism in a MANET from the security design perspective to address this issue.

4. **Middleware Approaches for MANETS**

In this section some design principles and research projects that have already been proposed will be presented. We will argue that middleware for MANETs are tightly coupled with applications and there is no single general middleware that solves all problems. The surveyed middleware approaches are purely and specifically designed for an ad hoc environment. We exclude those designed for fixed, nomadic or semi ad hoc infrastructures which is beyond the scope of this paper.

4.1 STEAM [16]: It introduces the concept of Event Based Middleware in a mobile Ad hoc environment. It addresses some specific constraints related to MANETs. One of the constraints is that some middleware components of the event services cannot be located on independent physical machines. In addition, such components may not be co-located with mobile entities and pose problems regarding availability, consistency, coverage and computational resources. STEAM utilizes the implicit publish/subscribe model thus does not require separate dedicated fixed cluster of event servers for to operate as the case in P2P and mediator-based models. Significantly the implicit publish subscribe model allows the consumers to subscribe to particular event types and the publishing entities to publish events. The entities are therefore fully anonymous. STEAM uses the proximity (geographical and functional) group communication model. The
geographical aspects specify the area where the information is valid and the mobile devices within the propagation range. The functional aspects represent the common interest of producers and consumers based on the type of information propagated among them. Using these two aspects the mobile devices discover each other and therefore communicate. STEAM supports three different types of event filters: Subject filters, Proximity filters and Content filters. The usage of content filters enables subscribing entities to express sophisticated queries, which enable fine grain filtering of events. The subject and proximity filters are utilized to address the scalability of the system. In STEAM subject and proximity filters are applied on the publisher’s side. Events are only routed to subscribers if both filters match. On the other hand content filters are deployed at the subscriber’s side and utilized when an instance of an event is received to determine whether or not to deliver the event to the application.

4.2. Expeerience [4]: Among the key objectives of an ad hoc environment is information sharing, access and communication. On one hand this makes P2P communication an adequate approach to tackle many issues since it shares some key characteristics with MANETs. On the other hand they also have some key differences such as the dynamic topology and the lack of guaranteed connectivity. In addition, scalability in P2P is limited in terms of data rates, whereas in MANETs it is conditioned by low bandwidth and low processing power. From this stems the design of Expeerience. It takes advantage of services provided by a P2P environment and adds various modules needed for MANET. The P2P framework chosen is JXTA [4] which offers key advantages such as interoperability, platform independence and ubiquity. JXTA tries to create a common platform for developing distributed P2P services and applications. Experience enhances some of the services in JXTA and adds another software layer that meets the requirement of MANET that is not met by JXTA. Thus introducing new features like: management of the intermittent connections and multiple interfaces, more efficient resource discovery mechanisms and code mobility. It is important to mention another concept used by Experience which is code mobility. This new paradigm allows the download and installation of new services dynamically. This feature allows the middleware to dynamically adapt to situations that were not considered during the design and that only take place at run-time.

4.3. EMMA [18]: As discussed in earlier sections, one solution for designing middleware for MANETs is to find a way to adapt a well known middleware technique used in traditional systems. This has a clear advantage in allowing application developers to adopt the same standards on mobile and dynamic devices. Also it allows interoperability between the wired and the ad hoc infrastructures. Therefore using paradigms based on asynchronous mechanisms constitute an adequate solution for an ad hoc environment where frequent disconnection and bandwidth fluctuation are the norm. Message Oriented Middleware (MOM) is a popular paradigm. EMMA is a MOM based middleware that exploits the Java Message Service (JMS) originally designed for semi mobile distributed systems. EMMA is an attempt to adapt the JMS to fit MANETs by providing a slight modification of the message passing used in JMS and adding an epidemic routing mechanism [18] that facilitate delivery of messages in a MANET environment. As in JMS, EMMA applications can use the point to point or the publish-subscribe communications style. In point to point, applications use queues for asynchronous message exchange between the producer and possible consumers. The optimal location of the queues is determined by a negotiation process that is application dependant, which makes the middleware context aware. To allow the hosts that are not within range to receive messages, the asynchronous epidemic routing protocol is used. Each host maintains a buffer of messages created and messages received and messages are dropped if the buffer overflows. As a result the reliability of this protocol a best effort one and it does not guarantee that all messages are delivered. In the publish-subscribe model some hosts contain topics and subscriptions of hosts interested in the topic. Topics are exchanged through subscribed group members using a synchronous protocol if possible or an epidemic protocol. This model also provides mechanisms for maintaining subscription and unsubscription messages.

4.4. LIME [17]: is information sharing middleware. It adopts the tuple space based approach that stems from Linda [9], which provides tuple space data structure for fixed distributed systems. The tuple space systems have been demonstrated to provide many useful features for wireless environments. LIME extends the model adopted in Linda and makes it suitable for highly dynamic mobile environments. LIME defines a tuple space for each mobile host and permanently associates it with it. When a mobile host connects to other hosts, rules for transient sharing of the individual tuple spaces are defined. This is effective for a fast and accurate information exchange between mobile hosts when a connection is established. Each mobile host maintains an interface tuple (ITS) that is permanently and exclusively attached to that unit and transferred with it when movement occurs (like in the data tree of Xmiddle). Each ITS contains tuples that the unit wishes to share with others and it represents the only context accessible to the unit when it is alone. However, the content of the ITS is dynamically recomputed so it looks like the result of the merging of the ITSs of other mobile units currently connected. Upon the arrival of a new mobile unit, the content perceived by each mobile unit through its ITS is recomputed taking the content of the new mobile unit into account. This operation is called engagement of tuple spaces; the opposite operation, performed on departure of a mobile unit, is called disengagement. Information about the system configuration is made available through a read-only transiently shared tuple space called Lime System, containing details about the mobile components present in the community and their relationship. Moreover, reactions can be set on the tuple space, to enable actions to be taken in response to a change in the configuration of the system. An important aspect of Lime is tuple access and movement; events are used to notify users when a new tuple is available. It is worth noting that the tuple space approach is widely used by other middleware to support mobility such as TSpaces [29] of IBM and L2imbo [6] that focused on the quality of service.

4.5. LIMONE [8]: An enhancement of LIME middleware is investigated and implemented in Limone. One limitation of LIME is that it assumes that the network is not highly dynamic
limited bandwidth and the large energy draw from network
new programming paradigms to overcome constraints such as
tackle the different challenges of WSN, which carry
various systems and Micro Browsers.
seamless integration of Xmiddle applications with the
representing mobile data structures in XML enables
manipulated using technologies such as Document
branches and references in the XML documents are then
can be semantically associated to trees. Nodes, address
is implemented using XML [14] and related
to update the whole tree at all times. The tree data structure
be a critical issue since not all devices have sufficient resources
specific part of the tree rather than the whole tree. This turns to
reconnect, the reconciliation is accomplished using the replicas
the information on each other's linked data trees. When
disconnections occur, the disconnected hosts retain replicas of
the trees they were sharing while connected, and continue to be
able to access and modify the data. When the two hosts
reconnect, the reconciliation is accomplished using the replicas
of the tree previously stored. This allows restoring only a small
mind. Finally, no mechanisms have been provided to handle
scalability problems can arise when it comes to
distributed applications using entities which are not in close
proximity. This significantly limits the usefulness of STEAM in
terms of application heterogeneity. Indeed, STEAM has
been designed with the traffic management application in
mind. Finally, no mechanisms have been provided to handle
disruptions in ad hoc environments. Experience moves a step
forward by containing efficient features to cope with ad hoc
scenarios. It supports code mobility and service migration,
including support for mobile agent systems allowing
scalability. The new enhanced resource discovery service
component allows better disconnection management and the
discovery of distributed agents hence the mobility is addressed,
This also enables power and hardware resources awareness,
such as the life time of mobile nodes is increased. Experience does not, however, address the component models issue with JXTA nor protocol exchangeability. In its present state Expeirience uses some libraries written in J2SE and in C, a light weight version using J2ME or lighter version of the JXTA middleware would be needed to support heterogeneity and to include devices such as PDAs.

EMMA uses an adapted Java Message Service (JMS) armed with two communication styles: point-to-point and publish-subscribe to cope with ad hoc scenarios such as intermittent connectivity and partial mobility. An epidemic routing mechanism is added to support message delivery to mobile hosts that are not in reach. However, the poorly performing epidemic algorithm in terms of the number of replicas that are spread across the network dictates that a tradeoff between application level routing and resource such as power and bandwidth usage should be investigated. This also poses scalability issues since the middleware doesn’t provide other resource discovery mechanisms beside the epidemic routing protocol. Another limitation is that the reliability offered is a best effort one which results in the loss of some messages. A prototype of EMMA middleware has been implemented using J2ME - a virtual machine implementation - which is actually suitable for heterogeneous scarce resources devices such as HP IPac PDAs and laptops.

LIME introduces a new approach in designing middleware services for ad hoc environments which is data sharing based on tuples spaces. However, LIME mobile hosts are connected only when the distance between them allows direct communication using events notification. Mobile agents are connected when they are co-located on the same host, or they reside on hosts that are connected. This turns to be a serious limitation for ad hoc applications where efficient multi hop communication mechanisms should be provided to support high mobility. That is, in its present state LIME is only a middleware service to support partial physical mobility between host and logical mobility between agents. It doesn’t constitute a full middleware package designed to meet all MANETs requirements.

XMIDDLE also provides an enhancement for information sharing mechanisms between mobile hosts. Using an enhanced data structure based on trees implemented in XML and other technologies. It moves a step forward in addressing mobile computing issues such as scarce device resources and frequent disconnection. Mobility is supported using a tree like data structure allowing better information sharing between two connected hosts. Frequent disconnection are handled using different protocols such as link and reconciliation protocols. The replication protocol enables keeping a copy of the communication data structure when disconnections occur so that no updates are needed when the connection is reestablished. This will enable considerable energy savings and fast synchronization. XMIDDLE is implemented in Java and relies on the virtual machine which makes platform independent and easily supports heterogeneity. However, XMIDDLE suffers some limitations that require further investigation: the communication paradigm (i.e., sharing of trees) is basic and needs to be enhanced to model more complex ad hoc mobile interactions. Also a key limitation of XMIDDLE like LIME is that multi-hop scenarios are not considered where routing through mobile nodes is required. Also resource discovery mechanisms should be provided to meet requirements such as scalability.

MATE with its virtual machine approach supports scalability by the use of active messages to update the network protocols and parameters by injecting new capsules. This makes the network dynamic, flexible and easily reconfigurable. MATE gives a user–land supplemented by the VM, hence supports heterogeneity and provides efficient network and sensor access. Mobility is addressed by using various ad hoc routing protocols and protocol updates. However, In terms of energy -power awareness-, MATE is only suitable for sleepy applications that are in low duty cycle most of the time, for complex applications, it is wasteful because of the interpretation overhead of its instructions. Also in its current state, MATE is only architecture and byte codes, making it not easy to use; a higher-level language and programming model for application development are needed.

6. DISCUSSION AND OPEN ISSUES

The middleware approaches and projects surveyed in this paper all provide different mechanisms and techniques to tackle different challenges and impediments of the design and development of a middleware for MANETs. The different approaches are: event based, message oriented, P2P combined with code mobility, information sharing oriented and distributed tuples. Each is suitable for some of the various ad hoc scenarios such as frequent disconnection and resource management, low bandwidth and dynamic network topology. However, a close examination based on our evaluation in the previous section reveals that the approaches are tightly coupled with specific applications and none of them fully meets the challenges presented in section 4, more specifically context awareness, QoS, heterogeneity and efficient resource discovery.

We believe that a complete and effective middleware should combine more than one approach and mechanism to cover a wide range of ad hoc requirements. The virtual machine approach combined with mobile code techniques adopting an asynchronous model of interaction between hosts such as the message oriented style would offer many potential solutions and drastically enhance middleware possibilities in ad hoc environments. In addition, keeping the middleware light weight and taking the context awareness as a functional requirement throughout the design and development of the middleware is also essential for success. Indeed, mobile code techniques allow creating new services and migrating services at run time and deal with issues that could not be predicted in the design phase. Furthermore, efficient adaptable resource discovery mechanisms specifically adequate for a mobile ad hoc environment should be provided to give the middleware more robustness and flexibility in handling the frequent changes in the network components and the topology.

Security requirements pose another major challenge to address in MANETs. The security mechanisms used and proven in traditional networks are not suitable for Manet’s [1]. The absence of centralized authority, dynamic network topology
and mobility cause serious problems. In addition, the limited resources and device independence dictate the need for new sophisticated solutions that should be incorporated in the design of middleware for MANETs. These techniques must be capable of functioning efficiently on the independent devices, while keeping resource consumption as low as possible.

7. CONCLUSION

In this paper, we surveyed different issues and middleware approaches specifically adopted for wireless mobile ad hoc networks. We tried to clarify some of the ambiguity on middleware definitions. Then we identified the major challenges that the design and development of middleware for MANETs faces. Furthermore, we investigated many of the relevant existing projects carried out towards this perspective. We provided a thorough comparison and classification by concentrating on similarities and differences between the approaches. In the end, we tried to provide an overview of the positive features and advantages with the shortcomings and disadvantages of the approaches studied. We were able to identify the following distinct approaches: event-based, message-oriented (MOM), P2P and information sharing.

Some of the approaches are based on a virtual machine and mobile code techniques. Furthermore, and based on the results of our comparison, we discussed and proposed potential enhancements and new research possibilities in the field. At the end it is important to mention that designing and implementing the middleware that fully meets all the requirements and challenges of a mobile ad hoc environment is to some extent not a realistic venture. The more realistic approach is to incorporate various techniques and methodologies that will proved as much of the required functionality as possible, while maintaining flexibility, efficiency, and scalability.

REFERENCES


