Intelligent Context – Aware Middleware For Mobile Ad Hoc Networks

Apostolos Malatras and George Pavlou

Centre for Communications Systems Research, Department of Electronic Engineering, University of Surrey, UK

Abstract. With the advent of wireless devices Mobile Ad Hoc Networks (MANETs) are gaining great importance as an emerging networking solution. The lack of wired infrastructure and ease of deployment are significant advantages, although there are still significant issues to be considered as far as network stability and reliability are concerned. We believe that the dynamic nature of these networks acts as a hindering factor to their wide development and that this can be mitigated through proper mechanisms that can derive the node mobility for every mobile node forming a MANET. To that respect we view context awareness as a highly promising approach, since by collecting such information about a mobile node we are able to predict its future movements and thus have an understanding of the network's topology. In this paper we present the design challenges and the architectural blueprints of an Intelligent Context Aware Middleware specifically targeted to MANETs that addresses the above issues.

1 Introduction

The proliferation of mobile ad hoc networking solutions observed in the past few years and the high rates of user adoption regarding this technology combined with the continuously increasing number of mobile devices, leads us to consider that there is an established paradigm shift from traditional, infrastructure networking towards a mobile, operator – free and with no fixed infrastructure type of networking, the one based in MANETs [1]. MANET based networks together with other emerging networking technologies such as sensor networks will constitute the foundations for pervasive applications such as those already envisaged by visionary researchers. The major strengths of this technology lie in the fact that it is easy to be deployed and has a very low cost, while allowing for user creativity through the lack of central, authoritative management and control [2], [3].

MANETs undoubtedly are not a panacea for every networking problem and the emerging pervasive realm. Noteworthy drawbacks include their highly dynamic topology, since every node consisting a MANET is mobile and possibly very volatile. These constant topological variations will eventually lead to a continuous state of network instability, which in turn can extremely deteriorate the performance of applications and services on these networks. Another important issue is that typically MANET devices have limited resources as far as storage and processing capabilities are concerned. By adding the energy constraints in the set of limitations on MANETs, one can easily understand that the road to fully grasping the potentials of ad hoc networking is not going to be an easy one [2], [3].

It is obvious there is a need for techniques that mitigate these problems in order to be able to harness the vast range and diversity of enhancements and advantages that mobile ad hoc networking has to offer. In this respect we assert that contextual awareness can greatly assist in building and maintaining reliable, long – term MANETs. It is our belief that context awareness can assist in building more predictable and thus more reliable MANETs by providing higher and lower level information regarding the mobile node's context, which can aid in predicting its mobility patterns. When the mobility of each and every node is predicted the future topology of the network can be foreseen, allowing thus the deployment of services and facilitating scalable routing protocols. The term context is used to indicate all the information accessible by the mobile node and descriptive of its surroundings whether they refer to computational or physical properties. The driving force behind this concept is based on the observation that if one has an idea of what is happening or what is going to happen in the MANET in the near future they can act proactively and adjust the MANET in order to preserve a certain degree of network stability.

This paper proposes the deployment of Intelligent Context Aware Middleware (ICAM) to support the aforementioned functionality in every mobile node of a MANET. The middleware considers all the unique features of MANETs as described earlier in this section and lead to a more reliable MANET, without of course promising full reliability since exceptional conditions cannot be forecasted by use of context awareness or by any other means. The structure of this paper is the following. After this brief introduction to the problem domain and the motivation for our research Section 2 presents the requirements analysis for the proposed middleware and in Section 3 the system design is outlined based on these requirements. Section 4 describes the implications to the design process that need exceptional handling when building the proposed middleware, while conclusions are discussed in Section 5.

2 Requirements Analysis

The middleware platform that will enable and act upon context awareness in a MANET environment can be conceptually divided into two different views, context-related and middleware-related. The context aspect refers to all the processes related to context capturing, context processing and knowledge extraction from different sources of contextual information. The middleware aspects of the proposed information system include providing a platform that enables collaboration amongst mobile nodes in an ad hoc mobile network environment. The middleware should provide support for the dynamic and unstable nature of this type of networks. Several requirements should be met by the middleware platform so that specific issues of the problem under consideration are satisfied. In general no centralized protocols or services are to be used due to the inherent nature of a MANET.

Contextual information is to be gathered by various sensors that communicate or are controlled by a mobile device. The middleware should be extensible enough to allow for the interaction with various types of sensors and even sensors that are not currently supported. The actual devices that form the mobile nodes are expected to have limited resources in terms of energy, storage and processing power. These deficiencies should be ameliorated by the proposed middleware. Context representation is an important parameter, since a generic model must be adapted that enables information exchange on the one hand and a way to uniquely handle semantically diverse context types on the other hand. Support must be provided for interoperability amongst different contextual domains, with the use of ontologies as the most apparent solution.

A certain level of programmability as far as network elements and applications are concerned is desirable. To this respect the middleware should support relevant functionality. The middleware should be as lightweight as possible so that only few resources and energy of the mobile node. It should also cater for data and task distribution from a mobile node to a set of other nodes in cases where insufficient resources from a mobile node call for such an action. Robustness and reliability are two major concerns that the middleware should take care of, especially in this highly dynamic environment, while in parallel user transparency and interoperability are a necessity.

3 System Design

The ICAM platform is deployed on every mobile node comprising a given MANET and acts on top of the ad hoc networking layer. ICAM lies between the application and networking layers having the advanced functionality of being able to configure both of these layers when context information state that configuration changes are necessary. Figure 1 depicts the middleware platform architecture on a mobile device and the communication flow when two mobile nodes interact with each other. It is worth noticing that communication flows exist amongst all layers on a single node and when two or more nodes cooperate. Ad hoc communication is thus not influenced by the ICAM and so is inter-application communication, since the design requirements state that ICAM is to be as transparent as possible. ICAM lies on the background of the mobile node's operation and assumes an active role when the contextual information gathered denote the need for alterations in either the application or the networking layer. ICAM retains a continuous monitoring and managing interface with all the sensors available to a mobile device.

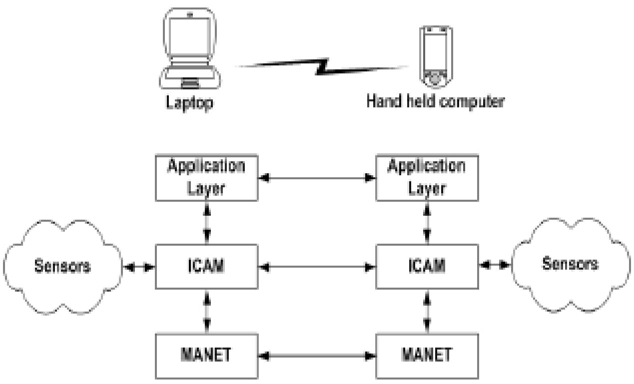

Figure 1 ICAM and its relation to other networking layers

Figure 2 presents the architecture of ICAM in its full deployment. As we have already made clear through the requirements analysis, not every mobile node needs to have the entire functionality encompassed by this architectural design. Each mobile node will have a single instance of the main middleware entity, the Context Manager, running at any time, while support for exception handling includes object re-initialization and restart in cases of system errors occurring. The Context Manager is basically an abstract wrapper for the 4 important modules of the architecture, namely the Sensor Communication Interface (SCI) Handler, the Context Handler, the Mobility and the MANET Handler.The Context Manager component represents the baseline of the whole middleware architecture and the MANET Handler that is present at any time enables a standardized, context-related communication protocol among all mobile nodes of a given MANET. The remaining and more useful functionality of the middleware is built on top of this baseline component in the sense of operational layers in hierarchical order: SCI handling, context handling and mobility handling. The purpose of ICAM is to predict a mobile node's future mobility patterns and disseminate this information across all mobile nodes forming the MANET in order to gain MANET stability in cases where topological changes occur, since the MANET will be adjusted proactively to tender for such changes.

3.1 System Components

SCI Handler

SCI Handler is the basic module for handling the various SCI that will be used by ICAM to interact with the sensors that are accessible to it. These interfaces correspond to a specific template and are generic enough to allow for the capturing and extraction of various context sources and their corresponding, diverse data types. One instance of this module should be instantiated at any given time. The functionality of the SCI Handler consists of managing the SCIs that are already loaded or stored locally on a given mobile node and locating and retrieving SCIs on demand and at runtime, in cases whether there is no appropriate SCI available or no storage capacity to store it locally. The SCI DB Handler component of the architecture deals with issues of accessing locally stored SCIs in the Local SCI Database and advertising these SCIs in the underlying P2P Network for other neighboring nodes to be able to locate an appropriate SCI when needed. The term Database is used only to denote a form of store and not imply the use of an actual database. We chose to use an XML-based indexing service to access and manage the locally stored SCIs. The SCI Loader component addresses the problem of loading SCIs so that context can be read from the available sensors. There are two different ways to load an SCI, locally or remotely. In the first case the SCI is already stored on the mobile node, while in the latter case the SCI Search component attempts to find an appropriate SCI by seeking in the P2P Network that the mobile nodes have formed for that purpose. JXTA [4] protocol services are used to employ the P2P characteristics of the aforementioned component. The choice of P2P searching is apparent in contrast to using hierarchical or centralized approaches due to the inherent nature of MANETs.

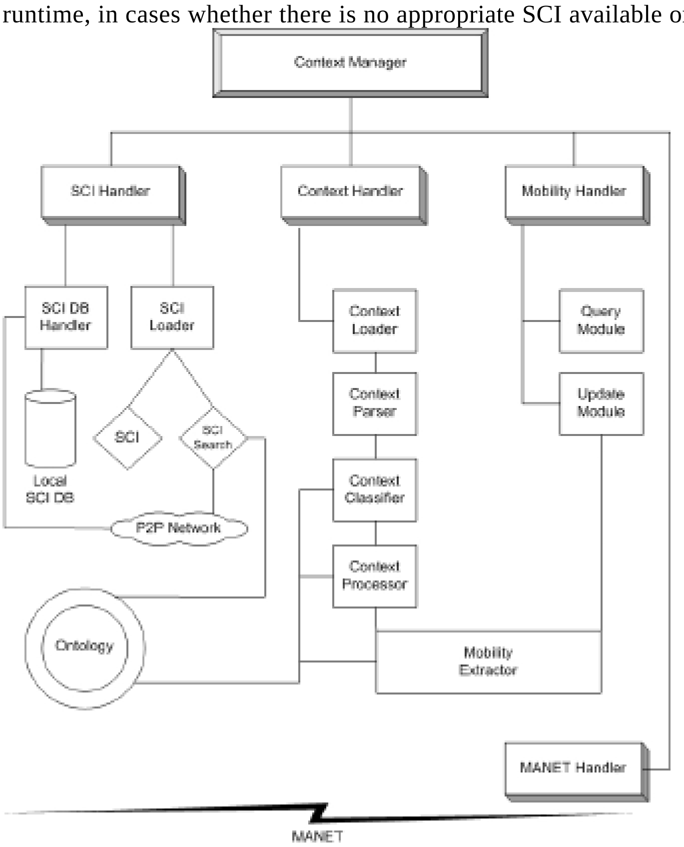

Figure 2 ICAM modular architecture

Context Handler

The Context Handler module of the architecture can have multiple instances loaded at any given time respectively to the number of the sensors that are used to extract contextual information from the mobile node's surroundings (node's location, computing resources etc.). The outcome of this component's operation is an understanding of a particular context domain i.e. an understanding of the node's current and future location. This module comprises several components that it exploits in a plugin-based manner, meaning that they are loaded, installed and used only when needed. The Context Loader component involves data loading from sensor collected datasets that have already been gathered and stored locally or are runtime observations. The Context Parser component is then activated and its responsibility lies on getting these data and parsing it to a widely accepted, pre-specified format. XML was chosen also because it is interoperable and extensible enough to allow for a generic representation of a variety of semantically different context domains. The abstract XML structure that maps the contextual information relies on the fact that all information can be subdivided into a number of atomic elements that are self-explanatory. The Context Classifier component is responsible for assigning semantics to the formatted context data so that meaningful knowledge can be gained through its processing. We have selected a rather simplistic approach that assigns semantic tags to context information on the basis of pre-specified rules for sensor use, meaning that we define beforehand what kind of information a sensor will monitor. We are looking into more elaborate yet lightweight mechanisms to ameliorate this service. The Context Processor deals with the management of contextual information. It serves as a management entity for all the contextual information collected from a mobile node and disseminates this information in other mobile nodes when it is requested.

Mobility Handler

The Mobility Extractor component is one of the most important components of the architecture since its purpose is to perform the task that ICAM is mostly concerned about: predicting a mobile node's future mobility patterns and making that information available across the MANET in order to gain perspective on the possible changes in the MANET topology. The detailed techniques used to extract this knowledge from contextual information lies outside the scope of this paper. In short we consider three diverse categories of context regarding upcoming mobility: mobility declarations, limitations on mobility and validating statements, and the mobility prediction is based on a combination of assertions from these categories. The Update Module handles all mobility information for a single node and disseminates it to all other nodes in the MANET, while the Query Model responds to mobility-related requests from other mobile nodes.

MANET Handler

The MANET Handler module is used to control all the ad hoc networking issues. This module retains information on the neighboring nodes of a mobile node in order to enable communication channels with them. All the other modules of the architecture interact with the MANET Handler when communication with another mobile node is required. ICAM at its most lightweight version consists of the Context Manager and the MANET Handler. It is also responsible for enforcing configuration changes to mobile nodes when needed as this is derived from contextual information gathered form sensors. The programmability aspects of ICAM as these were defined during the requirements analysis are implemented in this module. To present the programmability perspective of ICAM lies outside the context of this paper. In a nutshell, new protocols or changes to existing ones on a mobile node are employed using a plugin distribution procedure. Plugins are Java objects that are distributed to mobile nodes and are executed inside a Java Virtual Machine that we consider is running on every mobile node.

4 Design Issues

Throughout the design process of ICAM, several issues were raised concerning specific aspects of the architecture. Implementation technologies had to be thoroughly selected bearing in mind all considerations requiring interoperability, lightweight operation and being efficient in terms of processing capabilities due to the nature of the devices used. The Java programming language was the obvious choice since it serves the purposes of platform independence and interoperability. Nevertheless, the native Java platform cannot be regarded, as it is highly demanding in terms of memory and CPU consumption. For these reasons J2ME [6] was selected and ICAM is developed using it, grasping the functionality and benefits of the Java language embraced with the enhanced and mobile-device oriented features of the J2ME.

The storage of contextual information is based on XML for two reasons. The extensibility of XML structures assists in describing a variety of diverse contextual sources with the use of one XML Schema. XML is becoming a standard selection when interoperability and performance are a necessity. The alternative option of using a database was rejected because of the significant resources needed for its deployment, while simple binary storage is not viewed as a probable solution since it is extremely difficult to search in these files and process them. XML is moreover best suited for our purposes because of the semantic information it attaches to the contextual data. This operation is very helpful to us when we classify contextual data and want to extract the mobile node mobility from these. As far as the XML parser used is concerned we came to the decision of using the open source kXML2 one [5], because it performs well under the J2ME platform and is aimed at small devices like the ones we are dealing with.

5 Conclusions – Future Work

In this paper we presented ICAM that serves its goal as a lightweight, transparent middleware layer that enables context awareness in MANET environments and acting upon the information collected to proactively configure the network so that it retains a certain degree of stability. Application adaptation based on the gathered contextual information is also supported. The architecture is targeted to be implemented in MANET environments taking under consideration the specific requirements and limitations that are posed by these types of networks. ICAM collects information from various context sources including location identification sensors, computing resources sensors, network information sensors and so forth, while its extensible context capturing mechanism allows for the integration of virtually every context source. We are in the process of implementing, testing and validating this architecture in our testbed that comprises Linux-capable laptops and PDAs.

References

[1]. Perkins, C. E., Ad Hoc Networking, 2001 Addison Wesley Longman Inc.

[2]. Corson, S. & Macker, J., Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, IETF RFC 2501

[3]. Chakrabarti, S. & Mishra, A., *QoS Issues in Ad Hoc Wireless Networks,* IEEE Communications Magazine, pp. 142-148, February 2001

[4]. Project JXTA Homepage, http://www.jxta.org/ , visited June 2004

[5]. Kobjects, kXML2 Parser Homepage, http://kobjects.dyndns.org/kobjects/auto?self=$c0a80001000000f5ad6a6fb3, accessed June 2004

[6]. Java 2 Platform, Micro Edition Homepage, http://java.sun.com/j2me/index.jsp, accessed June 2004