Journal of Network and Systems Management (© 2006)
DOI: 10.1007/s10922-006-9052-7

Exploiting Context-Awareness for the Autonomic Management of Mobile Ad Hoc Networks

Apostolos Malatras,[1,2] Antonios M. Hadjiantonis,[1] and George Pavlou[1]

Mobile Ad Hoc Networks (MANETs) are characterized by a degree of dynamicity that can result in significant drawbacks regarding their useful deployment. The fact they are formed spontaneously, comprising possibly heterogeneous devices, hinders further their wide adoption. In this paper we present the design and implementation of a system that exploits context-awareness and couples it with policy-based management in order to enable the self-management of MANETs. The key idea is to support self-configuration by being adaptive to varying conditions modeled as context, with high-level management policies driving self-configuration towards particular goals. We propose the management of the MANET in a hierarchical but also distributed manner through a dynamically constructed set of manager nodes. We present and evaluate our work on context-awareness and context dissemination in MANETs through simulation and also by deploying the prototype system in our experimental MANET testbed for a proof-of-concept application scenario.

KEY WORDS: context-awareness; autonomic communications; mobile ad hoc networks; policy-based management.

1. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) have emerged as a new paradigm in communication networks, enabling pervasive computing and ubiquitous communication environments. Their main characteristic is the freedom that nodes exhibit in terms of movement, with the network topology changing potentially rapidly and unpredictably. Traditional wireless networks require some form of fixed network infrastructure and centralized administration for their operation. On the contrary, MANETs are spontaneously formed, with individual mobile nodes responsible for

[1]Centre for Communication Systems Research, Department of Electronic Engineering, University of Surrey, Guildford, Surrey GU2 7XH, UK.
[2]To whom correspondence should be addressed. E-mail: a.malatras@surrey.ac.uk, a.hadjiantonis@survey.ac.uk, g.paviou@survey.ac.uk.

dynamically discovering other nodes they can communicate with. The key benefits of MANETs such as the lack of centralized administration, tetherless computing capabilities and community-based short-term network establishment are hindered by the relevant drawbacks of management difficulty. There is an obvious need for frameworks that can support the MANET self-management according to predefined goals or policies.

We assert that such a highly dynamic environment can potentially benefit from context information that will drive its self-management, resulting in a degree of autonomy. High-level management rules expressed as policies can guide the MANET configuration, triggered by context information gathered by every node and disseminated across the MANET for network-wide understanding to be established. This closed-loop adaptive management can thus lead to self-configuration, self-optimization, and hence autonomy. This paper addresses the design of a context-aware policy-based framework to achieve this, focusing mainly on its context-aware aspects; it also presents the practical deployment of the proposed infrastructure in a small-scale real MANET, such as our experimental testbed.

The rest of this paper is structured as follows. After this brief introduction, Section 2 reviews related background work. Section 3 gives an overview of the proposed MANET management framework, along with a performance analysis of the proposed clustering algorithms. Section 4 presents the context model we propose and analyzes the efficiency of context dissemination through simulation. Section 5 presents the system's design and architecture, providing justification for our choices. Details on the implementation of the system and its deployment in our experimental MANET testbed for an example application scenario are the subject of Section 6. Finally, Section 7 concludes the paper and discusses future research directions.

2. BACKGROUND

Autonomic computing emerged as an initiative by IBM and has generated an active research stream bridging interdisciplinary domains. Autonomic computing refers to the self-managed operation of computing systems and networks, obviating as much as possible the need for human administrators. In such systems, high-level objectives drive the system's functionality to an optimal state in an adaptive self-managed manner. The IBM autonomic computing blueprint defines four distinct concepts behind autonomy, self-configuration, self-optimization, self-healing and self-protection [1, 2]. Most autonomic computing platforms are targeted to systems with sufficient resources that are relatively stable [3, 4]. The application of autonomic principles on MANETs has not been adequately researched. In [5] we presented our initial approach and results on self-configuring and optimizing MANETs. The exploitation of context in network management has been addressed before since the potential benefits can be tangible. Context can lead to adaptive systems that interact with the surrounding environment and function according to emerging conditions [6–8]. A key drawback of all these approaches, including our previous work [6, 9], is the static evaluation of context against predefined rules. The use of a context-aware system driven by policies can achieve adaptability of a more dynamic nature and can be easily tailored towards new high-level goals through policy modification. In the rest of this section we review related systems with context-aware properties that have been mainly applied to pervasive computing and communication environments.

The reconfigurable context-sensitive middleware (RCSM) [8] deals with context-awareness in mobile devices. It assumes reliable underlying ad hoc network transport protocols and proposes a CORBA-based middleware that uses context awareness to support application adaptation. Our approach exploits context information to provide more complex configuration adaptation, both in the network as well as in applications. In [10] a collaborative context determination approach well suited for MANETs is introduced, in which a mobile node gains its context from its neighboring peers. A key drawback of this work is the fact that the context definition is too narrow, effectively group context information regarding neighboring nodes. Another drawback is the absence of a concrete context model and the consumption of significant amount of bandwidth, which is a scarce resource in MANETs.

The middleware proposed in [12] addresses context-awareness in ubiquitous computing environments through a well-defined context model based on ontology-specified predicates. The use of mobile agents that handle all context-related tasks is also proposed. While this architecture seems well-suited to relatively static ubiquitous environments, its use in MANETs is questionable due to their potentially volatile topology that does not lend itself well to the centralized approach proposed. The Aura project [13] studied the provision of contextual information to pervasive computing applications based on a virtual database of context which the applications can access via a Contextual Information Service (CIS). An important aspect of Aura is that it associates accuracy and confidence values to context. Its major drawback is that relevant evaluation measurements indicated relatively high traffic overhead.

3. MANAGEMENT FRAMEWORK

In order to achieve our objective of creating an efficient framework for the autonomic management of MANETs, we adopt a modular design and realize an adaptive closed loop approach. The twofold architecture we propose combines context gathering, processing and dissemination with policy-based management (PBM). High-level objectives expressed through policies can guide the self-management of the MANET by providing guidelines as to what course of action should be followed when certain conditions are met. Through context monitoring, a real-time understanding of the network conditions and of the surrounding environment is gained and is used for policy conditions evaluation. Configuration changes may be subsequently deployed in order to achieve self-management by driving the network to a desired state, according to higher-level objectives specified as policies. The described process is repetitive, leading to an adaptive closed loop of control that results in autonomic management. The adaptation loop is initiated with the deployment of uniform high-level policies, which are dynamically translated into management logic and distributed to the MANET's nodes. Policies can drive context gathering, i.e. the monitored context depends of the types of policies deployed, and in turn the gathered context drives policy activation and execution, leading thus to autonomic decision making. In order to control potentially large MANETs, a scalable organizational model is necessary to support the distribution of the relevant management functionality. Such a model should support cooperative distributed management given the relevant characteristics of MANETs. In the following subsection we present such a model, with emphasis placed, as in the rest of the paper, on the context-aware aspects of our approach. Details on the policy-based aspects can be found in [14].

3.1. Distributed Hierarchical Model

We adopt a hybrid approach by proposing a distributed but also hierarchical model. We aim to provide an organizational model for MANET management by splitting the network into clusters and assigning management roles and responsibilities. Before analyzing the proposed model, we explain the differentiation between node "modules" and "roles" (Fig. 1(a)). The three proposed "roles" are MN (Manager Node), CH (Cluster Head) and CN (Cluster Node), supporting management responsibilities in a hierarchical manner.

A "module" is the preinstalled software of a node. In our approach every node possesses two modules, CM (Cluster Manager) and TN (Terminal Node).

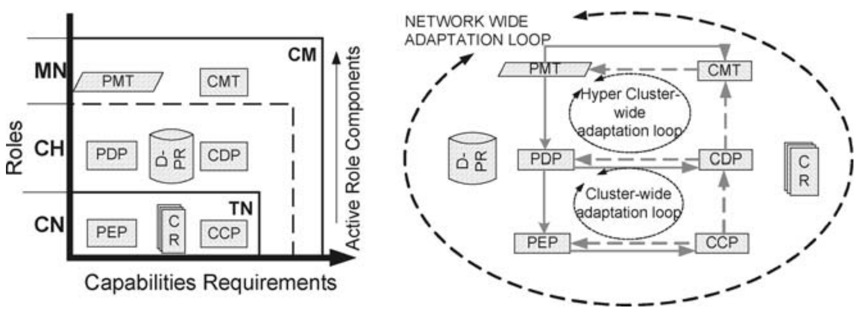

Fig. 1. (a) Node roles and modules and (b) closed loop adaptive management and information flow.

This separation was deemed necessary to accommodate a wider range of node capabilities in the MANET. TN is the simplest module and is lightweight enough to make it suitable for devices with limited capabilities. Thus TNs can only be assigned the cluster node role at the bottom of the management hierarchy. On the other hand, CM modules have full system functionality encompassing that of TN. CMs are thus collaboratively responsible for MANET management and can be assigned all three roles. The selection of the appropriate module for each node depends mainly on device capabilities. A set of minimum requirements (depending on the actual module implementation) offers a prescribed guideline, indicating whether a device can host the CM module. The node roles and modules are depicted in Fig. 1(a) encapsulating their respective components. The software components that realize PBM are depicted next to their context-aware counterparts, as there is an apparent and intentional 1-1 relationship. A device in CN role employs a Policy Enforcement Point (PEP), paired with a Context Collection Point (CCP) and a Context Repository (CR). In addition to these, a CH employs a Policy Decision Point (PDP) and a Context Decision Point (CDC), while the Distributed Policy Repository (DPR) stores policies in an LDAP server. Finally the devices in MN roles have the additional functionality of the Policy Management Tool (PMT) and Context Management Tool (CMT).

Based on the above node classification, we present our organizational model. The multi-manager paradigm and the hyper-cluster formation are introduced, aiming to offer a balance between the strictness of hierarchical models and the fully-fledged freedom of distributed ones. At the same time our model embraces both, as it can be deployed as either of these. Figure 2 presents our proposed approach in full deployment (all three roles with their respective components) including the standard IETF-proposed components of a policy-based management system. It is worth noticing the information flow between the various components (bottom-up for the context-related and top-down for the policy-related) as well as the interaction amongst the context and the policy components.

The essence of our framework is the realization of autonomic management through the aforementioned adaptive closed loop. This adaptation loop exists in different hierarchy levels in our design as depicted in Fig. 1b and can be identified in our overall design (Fig. 2). Cluster-wide adaptation is achieved with context collected within a cluster and local Policy Decision Points restrict further broadcasting of raw context to higher levels. In the same manner, "hyper-cluster"-wide adaptation concerns decisions influencing only the hyper-cluster nodes. Finally, network-wide adaptation can exploit MANET-wide context in order to achieve network-wide enforcement of management decisions, ultimately resulting in MANET autonomic management.

The idea behind the multi-manager paradigm lies in the nature of ad hoc networks and the purpose of their formation. Having more than one manager gives the flexibility to form networks between distinct trusted administrative authorities.

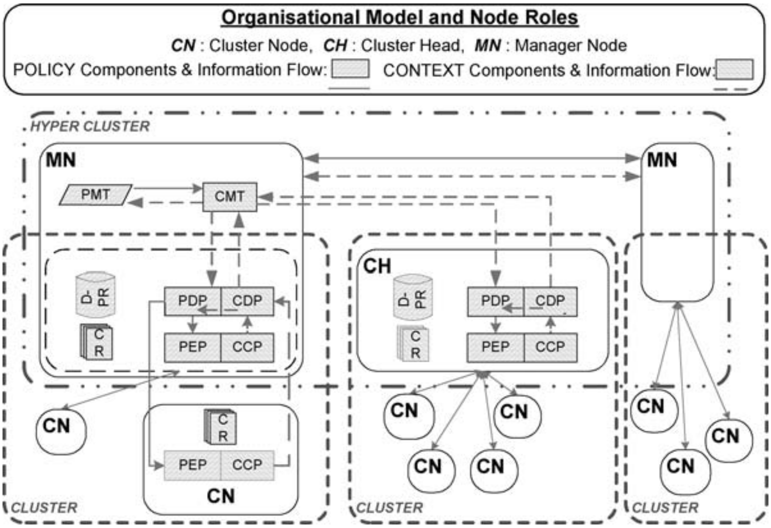

Fig. 2. Organizational model.

This is performed without any of these being forced to forfeit its management privileges. Instead managers cooperatively introduce policies that guide the overall network's behavior. For example, a MANET can be setup for a corporate meeting between two companies' representatives. The multi-manager paradigm treats the companies' managers as equals and allows both to affect network behavior by introducing policies. In addition, from a functional viewpoint, in large scale ad hoc networks scalability issues demand more than one manager in order to control and administer effectively the numerous cluster heads. The introduced policies are stored in the Distributed Policy Repository (DPR) components, where an automated replication mechanism ensures their consistency. The following section provides more details on the MANET roles and their assignment to nodes.

3.2. Hyper-Cluster Formation

We introduce the hyper-cluster notion, which is a set of Cluster Manager (CM) nodes that are assigned the MN or CH roles, utilizing available context information. Nodes that hold the MN role encapsulate the CH role as well. The assignment of the MN role to a node depends on the MANET formation purpose and its use. In the corporate meeting example, the two devices which the companies' managers use are explicitly assigned the MN role. In a military scenario, e.g. platoon leaders would become MNs. If there is no apparent specification for the MN assignment or in dynamic conditions where MN re-assignment is necessary, this occurs in an algorithmic fashion as described later; future work will consider a secure, reputation-based election procedure.

The Capability Function (CF) of a node is very important in our scheme and it denotes its current ability to host resource-consuming software modules. As the name suggests, it reflects the nodes' capabilities, namely static and dynamic computing properties and its mobility ratio (MR). The key concept is that if a node moves quite often and is responsible for link breaks with its neighbors, then it should not be considered capable. In our approach, the CF considers the following computing properties: memory (MEM), processing power (PP), battery power (BP) and computing load (CL). MEM, PP and BP have a proportional relationship to the CF while MR and CL an inversely proportional one. By assigning common significance weights to these variables, Eq. 1 is derived:

$$\mathrm{CF}(t) = \frac{w_1 \times \mathrm{MEM}(t) + w_2 \times \mathrm{PP}(t) + w_3 \times \mathrm{BP}(t)}{w_4 \times \mathrm{MR}(t) + w_5 \times \mathrm{CL}(t)}. \quad (1)$$

For the CF to be comparable, the variables are normalized to a value range of [0, 1] by dividing each one with its maximum counterpart (e.g. we define the maximum PP as 3 GHz). The sum of the weights w_1, w_2 and w_3 yields 1 while the same holds for w_4 and w_5. This ensures a bounded CF within the range (0, 1].

The battery power is obtained from the system profile as the product of the percentage of remaining power and the maximum capacity. The current load refers to the current computational load on a node, not instantaneously but using a moving average over a period of time. For comparability, the product of the percentage of CPU consumption and the PP value is used. In order to calculate the MR, we cannot simply consider the frequency of node movements, since this is not indicative of topology changes, e.g. all nodes might be moving in parallel towards the same destination. In addition, a static node may not be deemed as capable since all other nodes might be moving away from it. Taking these considerations into account, we associate the MR of a node with the frequency of link breaks with its neighbors over a period of time. This information is obtained from the network layer that monitors one-hop neighbors for routing purposes. In the future we plan to use context-driven mobility predictions to enhance the MR proactively. Equation 2 gives the MR for past time period T until the current time t, with the actual value being in the range of [0, 1] as the number of link breaks cannot exceed that of the neighbors. The MR value is also smoothed-out through a moving average algorithm but uses a number of previous observations to produce the actual value at time t.

$$\mathrm{MR}(t) = \frac{\mathrm{link\_breaks}([t - T, t])}{\mathrm{neighbors}([t - T, t])}. \quad (2)$$

The proposed hyper-cluster formation algorithm (Fig. 3) receives input regarding the module installed on every node, their capabilities and the network

Hyper Cluster Formation Algorithm

Assumptions

- N is the set of nodes in the MANET
- Nb(x) is the neighboring set of a node x
- MN, CH, CN boolean flags denote node role
- CM, TN boolean flags denote node module functionality

Construction Rules

1. $\forall x \in N, if\ TN(x) = 1 \rightarrow CN(x) = 1$
2. $\forall x \in N, calculate\ CF(x)$
3. $\exists x \in N : MN(x) = 1\ by\ default$
4. $\neg \exists y \in Nb(x) : CM(y) = 1\ AND\ CM(x) = 1 \rightarrow CH(x) = 1$
5. $\forall y \in Nb(x) : CM(x) = CM(y) = 1 \rightarrow (x \xrightarrow{Nb(x)} y) \wedge (y \xrightarrow{Nb(y)} x)$
6. $\exists y, w \in Nb(x) : CM(x) = CM(y) = CM(w) = 1,$ $y \notin Nb(w)\ AND\ w \notin Nb(y) \rightarrow CH(x) = 1$

Optimization Rules

1. $x, y \in N, Nb(y) \subset Nb(x), CF(x) \geq CF(y),$ $CH(x) = CH(y) = 1 \rightarrow CH(y) = 0 \wedge CN(y) = 1$
2. $x, y, w \in N, Nb(w) \subset Nb(x) \cup Nb(y),$ $\min\{CF(x), CF(y), CF(w)\} = CF(w),$ $CH(x) = CH(y) = CH(w) = 1 \rightarrow CH(w) = 0 \wedge CN(w) = 1$

Fig. 3. Hyper-cluster formation algorithm.

topology and yields the assignment of roles to nodes. When the network is setup, every node calculates the CF and periodically tracks it. Some of the nodes with CM capability will be marked as MNs by default, i.e. are pre-selected privileged nodes. After the initial phase, a selection process commences to establish the role of the nodes. All nodes having the TN module acquire the CN role. The hyper-cluster will consist of nodes that form the dominating set (DS) of the graph of nodes with CM modules, thus ensuring one-hop accessibility for the remaining nodes with the CM module, which are assigned the CN role. The idea is borrowed from backbone overlay networks used for routing in MANETs. Based on the extensive work on the area [15–17] we propose a distributed algorithm for DS formation using CF as the optimization heuristic.

Effectively, the CHs together with the MNs form the hyper-cluster and collectively manage the MANET. Every plain CN registers with its CH neighbor with the highest CF value, while those that do not have such neighbors acquire a route to one of them through their CN neighbors. Depending on the application use, the MNs are either dynamically introduced (the DS algorithm is executed once again upon the set of CHs to deduce the set of MNs) or statically configured upon the

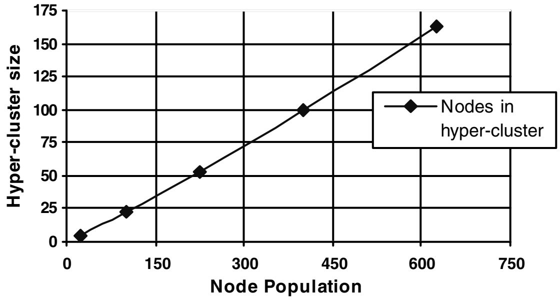

Fig. 4. Hyper-cluster size vs. MANET size.

initial construction of the MANET by assigning explicitly these nodes to the MN role and thus the hyper-cluster. The result is a clustered MANET with nodes in all three roles. Space limitations avert us from further describing the algorithm used to maintain the hyper-cluster when node movements affect topology.

We used the ns2 simulator to assess the scalability of our proposed hyper-cluster formation algorithm. The transmission range of each node is set to 100 m and the link capacity to 2 Mbps (worst-case scenario). The simulations were performed for a stationary MANET. In order to assess the effect of increasing network size on our clustering scheme, the terrain-area is accordingly increased, so that the average node-density is kept constant during simulations. The number of nodes in this case is varied from 25, 100, 225, 400 to 625 and the terrain-area size from 200×200 m^2, 400×400 m^2, 600×600 m^2, 800×800 m^2 to 1000×1000 m^2 respectively. Figure 4 shows the average hyper-cluster size as a function of node population. As it can be deduced, the increase in hyper-cluster size is almost linear to the increase in MANET size, which confirms the scalability of our approach.

We then examine the time required for our distributed algorithm to lead to a stable hyper-cluster. Figure 5 presents the average time required for nodes to acquire roles in the clustering process. Although the increase appears to be almost exponential, the time required for each node is in the range of 1465 msec for 625 nodes which is deemed acceptable. The sharp increase in time is attributed to the increased node connectivity when the node population increases. This leads to delays when neighborhood information is exchanged among nodes.

4. CONTEXT MODELING

Before describing our context-aware platform, it is essential to study the context model we have designed. Context modeling is significant as it affects the

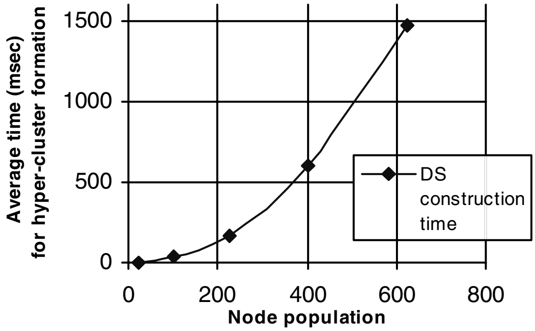

Fig. 5. Average hyper-cluster formation time vs. MANET size.

performance and scalability of the proposed system. It is the key building block of the context-aware platform and, as such, inherently linked to its implementation. After briefly reviewing related work in the area, we illustrate our proposed context model and use simulation experiments to evaluate the performance of our context gathering scheme, based on the previously described organizational model.

The subject of context modeling for wireless environments has been addressed by the research community but not in a systematic and holistic manner. The problem lies in the fact that every effort towards a context-aware system for wireless networks assumes a different context model to serve its own purpose, so relevant solutions lack commonality and are obviously non-interoperable. In order to achieve a unified approach, there is an obvious need for generic context information representation through a well-defined model. An important effort towards this goal is the W3C CC/PP standard that defines a generic model to express mobile device characteristics. [21] uses UML to model context information in pervasive environments. This approach is promising in terms of the model capabilities but imposes a significant degree of complexity. We distinguish ourselves by providing a less resource-consuming model. The theory of context spaces is described in [22] where context can cater for both numerical and non-numerical values and the set of operators defined for the various contexts enables the inference of complex, higher-level contexts. Context spaces are a significant step towards context modeling in pervasive realms, since it formulates a systematic approach defining a formal grammar and rules to handle context data. In [19] on the other hand, context information is represented in a predefined ontology targeted to the needs of a sensor-based wireless environment, while XML schemas are used to represent context in [20].

4.1. Proposed Context Model

Context information collected from all the nodes forming the MANET refers to their computational and physical environment and is tightly coupled with the policy-based management system since it is this information being monitored that may trigger the execution of a certain policy. Every node collects its own context information from its sensors. The term sensor here is generic and can, for example, include a battery monitor or a GPS receiver. The establishment of a MANET-oriented concrete context model that will cater for interoperability, extensibility and efficiency in terms of both processing and storage is of foremost importance. Our proposed context model takes into account the specific characteristics of the MANET environment, as these are mapped onto design requirements. Relevant requirements include *extensibility* to allow for diverse data types to be represented, *limited memory* requirements to store the collected context information, *lightweight processing* so as to cater for the resource-constrained MANET environment, *interoperability* amongst different context domains and, finally the model should provide support to establish a *degree of accuracy* of the collected context data.

Based on these requirements we propose exploiting Unified Modeling Language (UML) design principles for our context model (Fig. 6). The general context of a node consists of higher level contexts that have been deduced from simpler ones. Every context can be partitioned into a number of atomic attributes that have the ability to fully describe the initial context and not be composed of any simpler attributes. In this respect, context is composed of self-explanatory atomic attributes and perhaps other contexts, leading to more complex context structures. Relationships that could be simple inference rules or even mathematical functions are used to derive the higher level contexts from their components. The model incorporates semantic information regarding the context, the sensors and their relationships. For every sensor, context and attribute we store metadata information that describe their functionality, operation or meaning accordingly.

Fig. 6 presents a generic UML representation of our context model. Specific contexts can be modeled by extrapolating and using this model, as it will be shown in Section 6. The model is expressive and extensible, since custom, user-defined types of context with relationships and semantics can be introduced. Accuracy values for collected contexts are also supported. For every type of context information the system monitors, an individual context model is provided, which can be as simple as containing a single attribute, e.g. time that cannot be broken into simpler attributes, or very complex comprising many high-level contexts, attributes and inference relationships. The context model we propose is generic and allows the human user/administrator to express any context s/he wishes by incorporating available semantic information. Upon initialization of the

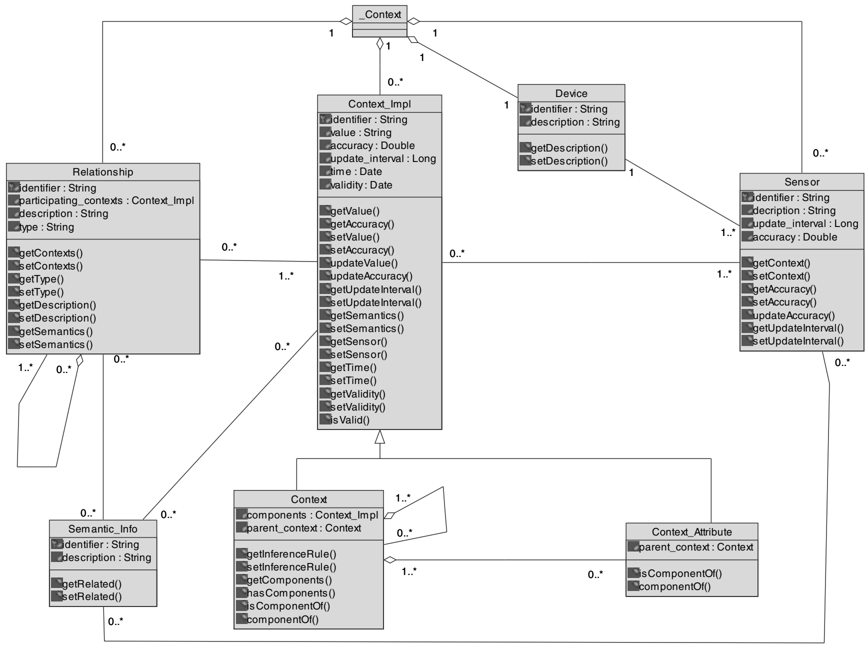

Fig. 6. UML diagram of proposed context model.

proposed system, hard-coded models of specific context information are loaded and are ready to be used.

The UML-inspired model is inherently associated with the data representing the collected context. As such, it can be easily mapped to an XML document for efficient and interoperable storage, while an XML Schema is used for the validation of any context modeled using the principles employed by our context model. The use of the XML Schema is binding for the uniform representation of all contexts. We understand the potential burden imposed by the XML formatting; the next section though shows that our context gathering and dissemination scheme performs relatively despite XML-based textual encoding. This justifies the requirements regarding efficient processing and limited memory requirements that we placed on our context model.

4.2. Context Dissemination

Context dissemination refers to the overhead imposed on network traffic by context information being collected by Cluster Nodes, transported to Cluster Heads and later on to Manager Nodes. It is extremely important to retain this

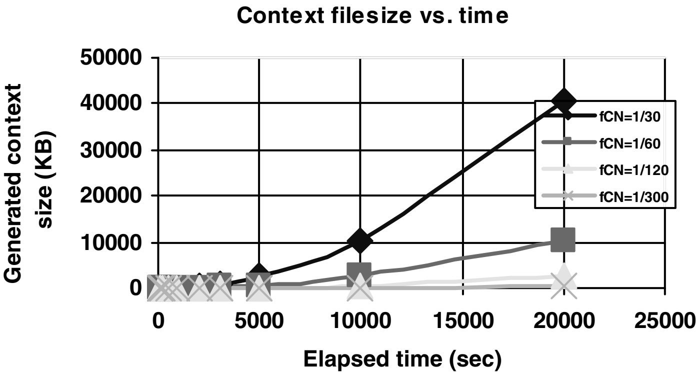

Fig. 7. Collected context at Cluster Nodes vs. frequency of collection.

overhead at an acceptable level, since the benefits one can gain by using context information to achieve proactivity and autonomy in MANETs can be diminished by poor deployment that requires a lot of data to be shipped around. We analyze next the effect of our proposed management framework and context model on the traffic at all 3 levels of the management hierarchy through simulation.

The experiments were performed using the XML Schema of our proposed context model, by modeling an example higher-level context comprising two simpler contexts, each with two context attributes and equal number of sensors and relationships. This type of context is indicative of many types of specific context information modeled using our proposed representation. As will be shown in Section 6, the relative velocity context results in identical parameters. As such, the example context we used is indicative and the purpose of the performed experiments is to provide suggestions and indications on the expected efficiency of our approach. Using these results, general conclusions can be extrapolated. The time variant values for context updates are generated using random data at intervals specified by the frequency of context collection. f_{CN} refers to the frequency of collecting context from a sensor, f_{CH} to that of passing contexts from the CNs to the CH and f_{MN} to that of aggregated context being passed to the MN.

Based on the management framework described in Section 3 and the ns2 simulator, we were able to simulate scenarios for the dissemination of context throughout the MANET. We experimented with various node populations and terrain sizes, keeping their respective ratio constant. We also varied the context monitoring frequency and the frequency of transmitting context from CNs to their corresponding CH and then to the MNs. As illustrated in Fig. 7 the amount of context collected from the CNs is inversely proportional to the collection frequency. For approximately the first 3 h (10000 s), the amount of context is at an acceptable level of a few MBytes even for very high frequencies. After a point,

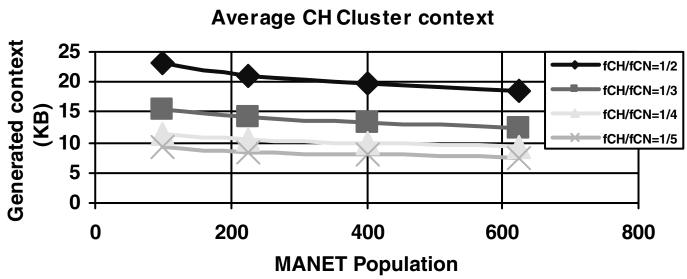

Fig. 8. Average context dissemination cost per CH per second as a function of context collection frequency.

we observe a sharp increase in the size of collected data. Not all collected context though is stored locally but is used and after serving its purpose it is discarded. The results shown in Fig. 7 bundled with the device memory specifications can be used as a set of guidelines for system designers as to when “stale” context should be archived or discarded.

Figure 8 presents the context information passed from the CNs to the CHs for various network sizes and topologies. Another parameter is the frequency through which CHs collect context from the CNs they manage, in relation to the context collection frequency itself. It is evident that the amount of context passed around in the MANET as a whole is significant, but this amount is actually much less per CH justifying that the clustering approach we have adopted allows for scalable context management. In both cases, the frequency by which CHs collect context from CNs greatly influences the overall transferred context. One should note that Fig. 8 involves monitoring of 5 contexts and the fact that the amount of context presented has been averaged for the whole duration of the measurement.

Figure 9 presents the context transferred at the second level of the hierarchy, where context has been collected by CHs at the cluster level and is then shifted towards the corresponding MNs. These entities have been identified using the dominating set algorithm proposed in Section 3 for various topologies in ns2. It is obvious that the overall amount of context data at this level of the hierarchy is high, but as with the CHs one should consider that this is distributed among MNs and is also distributed over time. Bearing in mind the worse-case conditions for wireless link capacity in MANETs based on the IEEE 802.11 standard, which is in the region of 2 Mbps, we argue that the overhead cost from context dissemination does not impose unbearable conditions on the MANET. As in the previous case, the results presented in Fig. 9 have been averaged for the whole duration of the measurement.

Even under these conditions there exist options to minimize the amount of context information transferred throughout the MANET. The major requirement

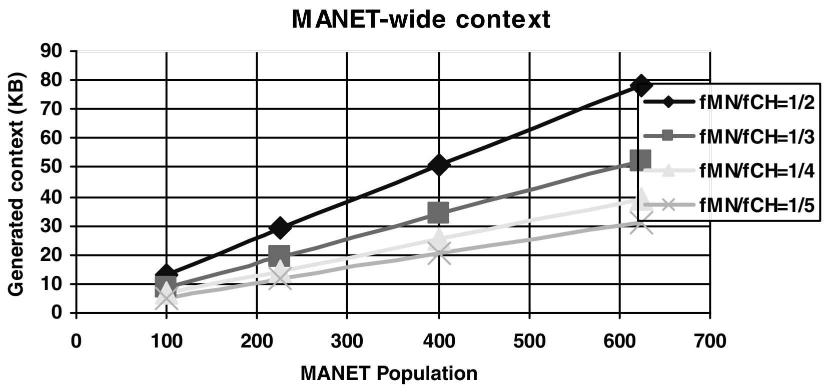

Fig. 9. Average context dissemination cost per MN per second as a function of context collection frequency.

for every potential solution is to be lightweight and consume minimal device resources as the majority of devices participating in MANETs have typically limited resource capabilities. We experimented with compression techniques that are specifically targeted for XML documents, since the tagging and individual characteristics of XML can be exploited to reduce the size of generated documents. The results of our custom benchmarking tool are encouraging; we opted though against using such techniques since they require considerable processing time and subsequently have a detrimental effect on node batter power. We propose instead to exploit common characteristics of context in order to achieve a degree of optimization, such as:

- Context aggregation: Context information is periodically aggregated and average values sampled over time are actually transmitted, not every single change in monitored context (see Section 6).
- Normalization of context values: When it is possible and without loss of precision, context values are normalized in certain ranges, allowing for smaller data to be transmitted.
- Threshold criteria: Criteria associated with specific contexts may result in context transmission only when certain thresholds regarding context changes have been exceeded.

It is not feasible to generically evaluate the performance of these techniques since their efficiency is dependent on specific context characteristics; the achieved data size reduction is though evident, as will be elaborated in Section 6.

5. CONTEXT-AWARE PLATFORM

Autonomic management of MANETs is the goal of our proposed framework. We have proposed an organizational framework for the efficient, robust and scalable management of MANETs based on a hybrid, both hierarchical and distributed, approach. The autonomic operation of the MANET is achieved through a closed control and information loop that is performed through use of high-level objectives expressed as policies that are triggered through context-awareness. The policy related part of our system has been the focus of our previous work [14], this paper though focuses on the context aspects of our overall system. In this respect we described the context model that formulates the foundation of our context-aware platform and by evaluating the performance of its gathering and dissemination we validated its scalability and suitability for the MANET domain. In order for context to be collected, processed, and disseminated, we have designed and implemented a context-aware platform to provide the desired functionality. Our design was guided by the requirements placed by the organizational framework, the context modeling analysis and the objective of providing autonomic management of MANETs.

The platform we propose cooperates and works in parallel with the policy-based framework for MANETs that was briefly introduced in Section 3. Context information is gathered locally from every node in the MANET and after basic processing it is passed to their corresponding CH that is responsible for its aggregation and processing to higher level contexts. Cluster-wide decisions based on this context can be imposed by the CHs, provided that certain conditions as specified by policies are met. At regular intervals, aggregated context from CHs is send to the MNs in order to establish if MANET-wide configuration changes are necessary based on that context and defined policies.

The platform's design is modular (Fig. 10) so as to cater for the diverse capabilities of mobile nodes. This is essential since the role and thus the functionality of any node can change over time as explained. Our modular design is context-aware in itself, since it allows for dynamic loading and offloading of modules according to their actual and prospective use. Based on our organizational model, we identify 4 main entities that form the basis for our context aware platform, namely the Context Collection Point (CCP), Context Decision Point (CDP), Context Repository (CR) and Context Management Tool (CMT).

5.1. Context Collection Point

The Context Collection Point (CCP) is the foundation of the Terminal Node (TN) module and is thus deployed on every node of the MANET. The CCP is responsible for communicating with the sensors available to the device i.e. GPS, storage media, processing unit, battery, and extracting periodically and in

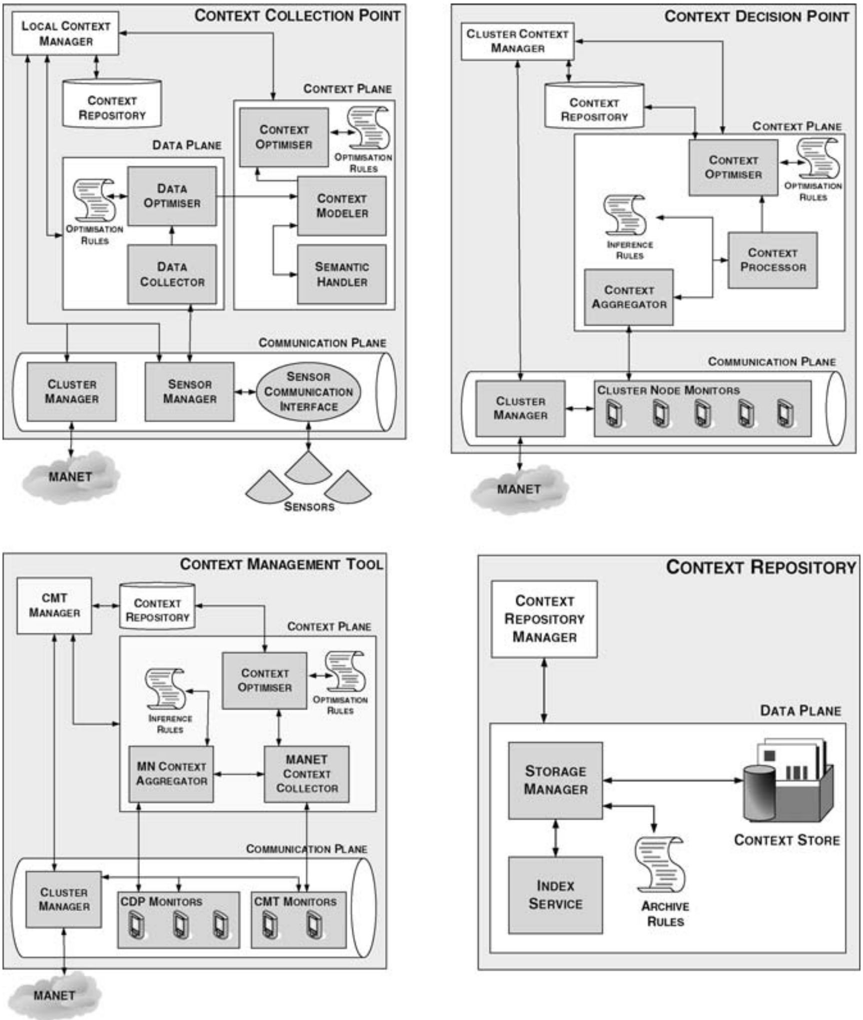

Fig. 10. Components of the Context Aware Middleware.

real-time values from these sensors. The diversity of the sensors necessitates a generic way in their handling by our system. This is the reason why we chose to separate the communication with the sensors from the actual system implementation with the use of interfaces to interact with the various sensors. The Sensor Manager entity interacts with the Sensor Communication Interfaces and presents the data and events collected from the sensor in a uniform way to the CCP.

The main entity of the CCP is the Local Context Manager. It is responsible for managing all context information locally and for the communication with other nodes of the MANET. Context collected by the CCP is sent to the CDP via this entity. It also keeps track of all the entities participating in the CCP and their operation. If an entity has performed its goal and is not used, it is the Local Context Manager that decides to offload it and perhaps later to dynamically load it again. The Cluster Manager is the entity that performs all the activities related to the hyper-cluster formation and maintenance as described previously. Should a node role transition be identified, the Local Context Manager is informed and acts accordingly to change the functionality of the node by deploying new software modules and ceasing operation of others. The Cluster Manager keeps track of organization information such as the CDP to which is a CCP is associated to at any time).

The Data Collector gathers the diversely formatted data as received from the Sensor Manager and passes them to the Data Optimizer. Optimization Rules guide the operation of the Data Optimizer whose responsibilities include pruning the collected set of data from non valid values and transforming the data to appropriate formats and with the expected precision for further processing (i.e. timestamp values from system clock are transformed to time and date). The data is then passed to the Context Modeler that converts the data to context modeled using our proposed model. The Semantic Handler feeds the Context Modeler with information regarding the type of data to be converted and the way this should be performed according to predefined context inference descriptions (i.e. the mobility prediction context as described in Fig. 6). After the data from the sensors has been transformed to useful context, this is passed to the Context Optimizer that based on Optimization Rules prunes the collected context and limits its size. Finally context information is passed to the Local Context Manager that stores it in the Context Repository.

5.2. Context Decision Point

The Cluster Context Manager is the main entity of the Context Decision Point (CDP) installed on every CH. It is responsible for monitoring and interacting with the CCP modules of the nodes that are associated with this CH and also with the corresponding MN. The Cluster Manager, as with the CCP, is responsible for monitoring and informing the Cluster Context Manager regarding changes in the clustering process. Through this monitoring the CDP is aware which CNs it is responsible for collecting their context and thus employs the respective Cluster Node Monitors. A Cluster Node Monitor retains a communication link with a CCP via which context information is transferred to the CDP. The Cluster Node Monitors periodically pass the collected context to the Context Aggregator. The Context Aggregator after having gathered the context from all managed CCPs produces average values and thus reduces the amount of data available to the CDP. Predefined, hard-coded Inference Rules combined with the aggregated context are used by the Context Processor to deduce higher-level contexts that have cluster-wide applicability. For example, mobility patterns from CNs collected by the CH can yield a cluster-wide view of the volatility of the whole cluster. The Context Optimizer together with Optimization Rules is used to reduce the size of the generated context for efficiency, while the cluster context is passed for storage to Context Repository.

The context collected at the cluster level can be used for cluster-wide adaptation when certain conditions are met. To accommodate this, the CDP communicates with the Policy Decision Point (PDP) also located at the CH and evaluates context against the monitored objects specified at the Distributed Policy Repository (DPR) to establish the need for cluster-wide configuration changes.

5.3. Context Management Tool

The functionality of third level of the role hierarchy, the Manager Node (MN), is realized by the Context Management Tool (CMT). The latter runs in MNs and allows the human manager to see the collected aggregate context through a graphical user interface; this may result in the human manager triggering a management decision either directly, or indirectly by modifying policies through the PMT, hence the CMT/PMT analogy and relationship. The main entity of the CMT is the CMT Manager whose responsibilities include communicating with the CMTs of other MNs and exchanging information regarding the context of the CHs each manages. This way ensures that all MNs have a uniform understanding of the context of the whole MANET in a distributed and efficient manner. The CMT Manager also interacts with the Policy Management Tool (PMT) and the PDP available at the MN in order to establish the need for MANET-wide configuration changes, by matching monitored context against monitored objects as specified in the policies stored in the distributed policy repository.

The Cluster Manager keeps track of the clustering process and notifies the CMT Manager for any changes, while retaining CDP Monitors for every CH it manages. At the same time it retains CMT Monitors for other MNs, if any. The CDP Monitors receive context from CDPs and the CMT Monitors exchange MN-wide context. The CMT functionality apart from that is essentially equivalent to the CDP one, with the distinctive difference of referring to MN-wide context and thus aggregation and processing occurs at a higher level with different inference rules.

5.4. Context Repository

The Context Repository exists at every node of the MANET and stores diverse types of context, according to the role of each node (CN, CH or MN). The Storage

Manager is contacted to either store new context or to retrieve stored context. An Indexing Service in the Context Repository optimizes resource access by exploiting standard data store indexing techniques. The Context Store is the actual collection of XML documents holding the context information, while Archive Rules guide the process of archiving out-of-date context information by storing it in compact, compressed files or discarding them completely. The Context Repository Manager is the main entity that provides the interface exposed by the Context Repository to other software modules that interact with it.

6. AUTONOMIC MANAGEMENT OF MANETS

The overall framework we have proposed for the autonomic management of MANETs can be used for various scenarios, depending on the monitored context and the defined policies. The only requirement placed on the system designer is to define the policies and model the monitored context to establish the need for policy activation. Once these are established, the system is capable of managing the MANET in an autonomic fashion. In the following subsection we describe in detail an application scenario concerning an adaptive routing strategy for MANETs, based on its implementation in our experimental testbed. We will also describe various other application scenarios for our system, with which we plan to experiment in the future.

6.1. Implementation Details

Our platform can be conceptually divided into two parts, the context-aware and the policy-based infrastructure. This paper examines the context-aware aspect of our platform in detail, while implementation details and performance measurements for the policy-based subsystem can be found in [14]. Our context-aware platform as described in the previous section has been implemented and deployed, although there are still ongoing modifications and optimizations. Using the overall system we have performed experiments to validate its efficiency, but also more importantly to validate it in terms of providing the desired functionality. In this respect we deployed our system in a real environment, specifically in our experimental MANET (based on IEEE 802.11b) testbed that comprises 4 laptops and 4 PDAs (see Table I for configuration details).

The platform is implemented using the Java 2 Micro Edition (J2ME). This version requires a much smaller memory footprint than the standard or enterprise edition, while at the same time it is optimized for the processing power and I/O capabilities of small mobile devices. We chose to use Java because of its ubiquity, platform independence and the fact that it allows for dynamic code loading and offloading (even for dynamic code generation). The use of Java requires nodes to have the Java Runtime Environment (JRE) installed.

Table I. Testbed hardware configuration

Platform	Configuration Attribute	Description
PDA	Processor	400 MHz Intel XScale
	Memory	48 MB ROM, 128 MB RAM
	Operating System	Familiar Linux 2.4.19
Laptop	Processor	1,7 GHz Intel Centrino
	Memory	512 MB RAM
	Operating System	Debian Linux 2.6.3

Although this is relatively memory-hungry, our hands-on experience confirms that even the resource-poor PDAs can comfortably support the execution of the JRE. XML handling necessitates a lightweight XML API and in that respect we used the kXML2 parser (http://kxml.sourceforge.net/), especially targeted for J2ME. The communication between nodes uses the lightweight XML-RPC protocol (http://www.xmlrpc.com/). It allows software running on different operating systems and architectures to communicate through remote procedure calls (RPCs). We chose an XML-based approach because we also use XML to represent context collected by nodes. Given our performance evaluation of XML and other management approaches [11], we argue that XML-RPC provides a useful blend of functionality and performance.

6.2. Applicability Scenarios

6.2.1. Real-Time Adaptive Routing Strategy

In order to validate the feasibility of our framework we have implemented a real-time adaptive routing strategy that increases network performance while requiring minimal human intervention. For the sake of simplicity we present here an uncomplicated example that involves the collection and aggregation of four simple contexts (longitude, latitude, speed and angle) into one higher-level context (Relative Mobility), which will determine the actual routing protocol to be deployed in the MANET.

A generic classification of MANET routing protocols can distinguish them into proactive and reactive, regarding the strategy used to establish routes. Performance measurements show that in high mobility scenarios where link breaks are quite frequent, a reactive protocol (e.g. AODV – RFC 3561) performs better since route establishment is dynamic and on demand. On the contrary, proactive protocols (e.g. OLSR – RFC3626) establish a full set of routes periodically and perform better in relatively static MANETs. This observation has motivated the implementation of the described applicability scenario which undoubtedly increases network performance, since at any given time the most appropriate routing protocol is actually used. Our experimental testbed was used to test and evaluate the implemented system in a real ad hoc environment. The implemented scenario was that of the dynamic routing protocol change, according to the changes in node relative velocity. The two routing protocols we used, reactive AODV-UU and proactive OLSR, are realized as user space daemons, so any of the two can be easily deployed. Practical problems encountered during the experiments included wireless link interference given that the wireless interfaces were deployed in a confined space. In addition, since testing various network topologies was necessary and to cater for full mobility scenarios, we used a MAC address filter tool to emulate broken links or unreachable destinations due to mobility. Our emulator is integrated in our platform and makes mobile nodes appear as moving around, with links breaking and being re-established. The node movements follow the widely used random waypoint mobility model, bearing in mind the inherent limitations of this model [18].

Every MANET node is equipped with sensors to monitor its surroundings. In this particular case-study, the context all nodes monitor is their mobility in terms of current velocity and location. Mobility estimation assumes that every node is equipped with an accelerometer and a GPS receiver that can provide the node's exact velocity and location. Given that we emulated mobility, we also used emulated values for the mobility context that conform to the output of the relevant sensors. We model this context as shown in Fig. 11 and map it onto an XML document. Note that mobility as a context is mapped onto the same number and types of entities as the example context model for the context dissemination evaluation presented in Section 4. A node's mobility can be broken down into two simpler contexts, its location and velocity derived respectively from the GPS receiver and accelerometer. The location context is further broken down into two context attributes, longitude and latitude, and the velocity into speed and angle. Inference relationships guide the way contexts are deduced from their respective attributes; in this case both rules are simple boolean conjunctions, implying an aggregation relationship, in terms of UML modeling.

The mobility context (Fig. 11) is passed from the CNs' CCPs to the CHs' CDPs, where it is aggregated into a higher-level context, i.e. the Relative Mobility context. This is the relative velocity of the CNs comprising the cluster, which is obtained as the absolute value of the vector-based subtraction of the velocities of the CNs in pairs, as this is stored in the respective Current_Mobility context. The inference rule to deduce this higher-level context using pseudo-code, is the following:

RM = 0
for each I in the set of CNs
for each J in the set of CNs
RM = RM + |*Current_Mobility.Velocity (I) − Current_Mobility.Velocity (J)*|;
Return RM;

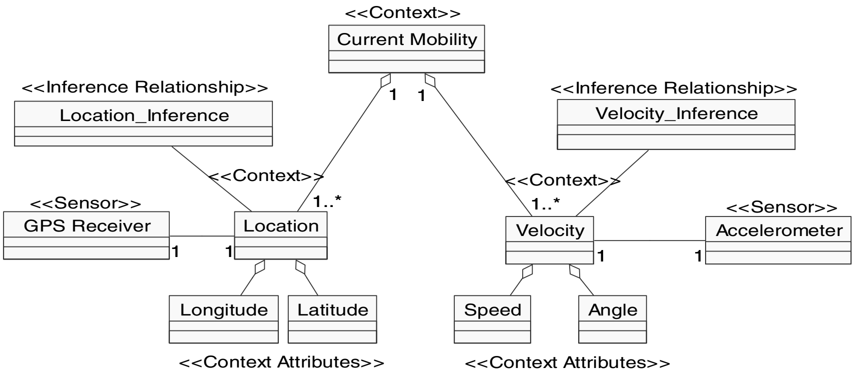

Fig. 11. Mobility context represented using our generic context model.

The overall Relative Mobility (RM) context is exploited by two policy rules to achieve network-wide adaptation of the routing strategy:

$$\{MN\}\ if\ (RM < rm_{thresh})\ then\ (RoutProt := OLSR)$$
$$\{MN\}\ if\ (rm_{thres} \leq RM)\ then\ (RoutProt := AODV)$$

In our organizational model, Cluster Heads (CH) have local access to a global synchronized Policy Repository (DPR), therefore can easily identify which higher context the Manager Nodes (MN) need for network-wide decisions. In this example, relative mobility context needs to be forwarded to the CMT of the Manager Nodes, as indicated by these policies' assigned role, namely $\{MN\}$. This tactic significantly reduces the dissemination of excessive context data between CH and MN. When MNs have collected the cluster-wide RM context from their controlled CHs nodes and exchanged these among themselves, they can effectively calculate a network-wide Relative Mobility value. It is this RM value which will be used from the Policy Decision Points to evaluate the conditions of these policies and ultimately decide on which routing protocol to use in the MANET. As soon as the policy rule is triggered, a reverse information flow will propagate the changes and provision all MANET nodes with enforcement decisions. For further details and performance analysis of the adaptive routing strategy we refer to our previous work in [6], while for a description of how the actual node configuration occurs please refer to [9].

In Table II we provide indicative results from our experimental evaluation (space limitations avert us from presenting full scale results). In our experiments, we emulated mobility based on random waypoint mobility model scenarios, varying the terrain size from 100 × 100 m to 200 × 200 m since we only had 8 nodes and larger terrains would lead to continuous breaks of communication links. The relative node velocity was varied from 5 m/s to 20 m/s. The first

Table II. Experimental results for routing protocol switch

	Value		
Metric node relative velocity	5 m/s	10 m/s	20 m/s
(Terrain size: 100 × 100)			
Time convergence (sec)	2.57	3.14	3.81
Routing traffic (bytes)	7523	7645	7701
Management traffic (bytes)	21092	21816	22902
(Terrain size: 200 × 200)			
Time convergence (sec)	6	6.43	8.07
Routing traffic (bytes)	10209	12788	13342
Management traffic (bytes)	25512	26813	28259

observation is that in these cases the time required for all nodes to efficiently activate the appropriately routing protocol is relatively small, in the range of 3 s on average. The reason for this is that due to the limited terrain size and the fact that the range of wireless transmissions is 100 m, there is increased connectivity between the nodes. Link and connectivity changes and disruptions caused by increasing node velocity yield a relative increase in traffic measurements and convergence time. The routing traffic is considerably bigger in the case of the larger terrain size since more link breaks cause increased number of requests for route reconstruction. Our platform is robust enough to cater for link breaks by assuming a threshold value for the number of nodes that need to be present in order to complete the operation. If more nodes than this threshold have left the MANET, then the process of routing switching is re-instantiated. Management traffic refers to the messages exchanged between the various platform entities as described in Section 5. Context dissemination results were found to be in accordance to the simulation results presented in Section 4 since we used the same node mobility context in both the simulations and the practical experiments, so we do not present them again here.

From our experience in designing and implementing the above case study for an adaptive routing strategy, we argue that tangible improvement in MANET performance can be achieved. There is great potential for autonomic management, if we consider more complex policy conditions and actions combined with sophisticated context information. It should be obvious that the described adaptation process does not require any human intervention beyond the initial definition of the policies and modeling of the monitored context, achieving a degree of autonomy.

6.2.2. Other Applicability Scenarios

The scenario presented above of routing protocol switching based on overall degree of mobility is a scenario we dealt with extensively as the first approach towards self-optimization in MANETs. We plan to study and experiment with other scenarios in the future as there are various possibilities to exploit context for MANET self-management.

A particularly scarce resource in MANETs is the battery power. In fact, if the battery is below a certain threshold, relevant nodes should still be able to send and receive packets but not to be used to relay packets if possible. In this case the high-level policy rule is *"a node should not forward packets if the energy level is below X%"* and the context information communicated is the remaining energy level of MANET nodes. In this case, this information could be exploited by power-aware routing protocols that will be configured to avoid using particular nodes as relays. Another aspect we are particularly interested is the identification of main streams of information in the MANET from and to particular nodes. Again this information could be used for alternative routing by QoS-aware routing protocols so as to keep the network load-balanced (self-optimization). Another use of this information would be to try and identify and subsequently isolate malicious intruders that send bogus data streams in denial-of-service attacks (self-protection). Context may also be gathered regarding access to important servers and these might be relocated or replicated if possible in order to provide a good level of service. Finally, many other uses of context information may be eventually possible.

7. CONCLUSIONS AND FUTURE WORK

We presented a framework and key design principles of a context-aware platform that enables the adaptive self-configuration of MANETs by using policies triggered from context information. We proposed a three-tier organizational model, with context gathered at every node, and passed to local cluster heads, which in turn aggregate it and pass it to a "dominating set" of manager nodes. We evaluated the performance of both the organizational model and of context dissemination through simulation and we showed they are acceptable. We also implemented and successfully deployed the proposed platform in our experimental testbed, getting encouraging initial results. In this paper we focused on the context-aware aspects of the platform, with the policy-based aspects presented in [14]. Our future work will focus on security issues while we intend to also consider more elaborate management policies that will stress the capabilities of the context management system. We have also started work towards assessing the system in terms of software metrics from a software engineering perspective. We also plan to test the system performance, scalability and its effect on MANET optimization using further simulation experiments and complementing these MANET simulations with real-world practical experiments, as suggested in [23].

ACKNOWLEDGMENTS

The work presented in this paper was carried out in the context of the EPSRC Programmable Ad hoc Networks (PAN) project – GR/S02129/01. Work towards this paper was also partly supported by the EU EMANICS Network of Excellence project (IST-026854).

REFERENCES

1. R. Haas, P. Droz, and B. Stiller, Autonomic service deployment in networks, *IBM Systems Journal*, Vol. 42, No. 1, pp. 150–165, 2003.
2. J. O. Kephart and D. M. Chess, The vision of autonomic computing, *IEEE Computer*, Vol. 36, No. 1, pp. 41–50, Jan. 2003.
3. C. H. Crawford and A. Dan, eModel: Addressing the need or a flexible modeling framework in autonomic computing, *10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS02)*, October 2002.
4. X. Dong, S. Hariri, L. Xue, H. Chen, M. Zhang, S. Pavuluri, and S. Rao, AUTONOMIA: An autonomic computing environment, *IEEE International Conference on Performance, Computing and Communications*, 2003.
5. A. Malatras, G. Pavlou, S. Gouveris, S. Sivavakeesar, and V. Karakoidas, Self configuring and optimizing mobile ad hoc networks, poster, *2nd IEEE International Conference on Autonomic Computing (ICAC 2005)*, 2005.
6. A. Malatras and G. Pavlou, Context-driven self-configuration of mobile ad hoc networks, *IFIP International Workshop on Autonomic Communications (WAC 2005)*, October 2005.
7. P. Bellavista, A. Corradi, R. Montanari, and C. Stefanelli, Context-aware middleware for resource management in the wireless Internet, *IEEE Transactions on Software Engineering*, Vol. 29, No. 12, 2003.
8. S. S. Yau, F. Karim, Y. Wang, B. Wang, and S. K. S. Gupta, Reconfigurable context-sensitive middleware for pervasive computing, *IEEE Pervasive Computing*, Vol. 1, No. 3, pp. 33–40, July–September 2002.
9. S. Gouveris, S. Sivavakeesar, G. Pavlou, and A. Malatras, Programmable Middleware for the Dynamic Deployment of Services and Protocols in Ad Hoc Networks, *IEEE/IFIP Integrated Management Symposium (IM 2005)*, May 2005.
10. J. Mäntyjärvi, P. Huuskonen, and J. Himber, Collaborative context determination to support mobile terminal applications, *IEEE Wireless Communications*, Vol. 9, No. 5, pp. 39–45, October 2002.
11. G. Pavlou, P. Flegkas, S. Gouveris, and A. Liotta, On management technologies and the potential of web services, *IEEE Communications*, Vol. 42, No. 7, pp. 58–66, July 2004.
12. A. Ranganathan and R. H. Campbell, A middleware for context-aware agents in ubiquitous computing environments, *ACM/I/FIP/USENIX International Conference on Middleware*, 2003.
13. G. Judd and P. Steenkiste, Providing contextual information to pervasive computing applications, *IEEE International Conference on Pervasive Computing (PERCOM 03)*, March 2003.
14. A. Hadjiantonis, A. Malatras, and G. Pavlou, A context-aware, policy-based framework for the management of MANETs, *7th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 06)*, 2006.
15. P.-J. Wan, K. M. Alzoubi, and O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, *IEEE Infocom 2002*, 2002.

16. R. Friedman, M. Gradinariu, and G. Simon, Locating cache proxies in MANETs, *ACM MobiHoc04*, 2004.
17. U. Kozat and L. Tassiulas, Network layer support for service discovery in mobile ad hoc networks, *IEEE Infocom 03*, 2003.
18. J. Y. Le Boudec and M. Vojnovic, Perfect simulation nd stationarity of a class of mobility models, *IEEE Infocom05*, 2005.
19. P. Korpipää and J. Mäntyjärvi, An ontology for mobile device sensor-based context awareness, *Proceedings Context 03, LNAI no. 2680*, Springer-Verlag, pp. 451–459, 2003.
20. H. G. Hegering, A. Küpper, C. Linnhoff-Popien, and H. Reiser, Management challenges of context-aware services in ubiquitous environments, *14th IFIP/IEEE Workshop on Distributed Systems: Operations & Management (DSOM03)*, 2003.
21. K. Henricksen and J. Indulska, A software engineering framework for context-aware pervasive computing, *2nd IEEE International Conference on Pervasive Computing and Communications (PERCOM)*, 2004.
22. A. Padovitz, S. Wai Loke, and A. Zaslavsky, Towards a theory of context spaces, *2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW '04)*, 2004.
23. C. Tschudin, P. Gunningber, H. Lundgren, and E. Nordstrom, Lessons from experimental MANET research, *Ad Hoc Networks*, Vol. 3, Issue 2, pp. 221–233, March 2005, Elsevier, 2005.

Apostolos Malatras is a researcher and PhD candidate at the Center for Communication Systems Research, Department of Electronic Engineering, University of Surrey, United Kingdom, where he is an active member of the Networks Research Group. He holds a Diploma in computer science from the University of Piraeus, Greece and a M.Sc. degree in Information Systems from the Athens University of Economics and Business, Greece. His research interests focus on software engineering, context awareness, network management, networking, programmable networks, and communications middleware.

Antonios M. Hadjiantonis is a researcher and PhD candidate at the Center for Communication Systems Research, Department of Electronic Engineering, University of Surrey, United Kingdom, where he is an active member of the Networks Research Group. He holds a Diploma in Electrical and Computer Engineering from the National Technical University of Athens, Greece. His research interests focus on policy-based management, ad hoc networks, management technologies for wireless networks and ubiquitous computing.

George Pavlou is a professor of communication and information systems at the Center for Communication Systems Research, Department of Electronic Engineering, University of Surrey, United Kingdom, where he leads the activities of the Networks Research Group. He holds a Diploma in engineering from the National Technical University of Athens, Greece, and M.Sc. and Ph.D. degrees in computer science from University College London, United Kingdom. His research interests focus on network management, networking, and service engineering, covering aspects such protocol performance evaluation, traffic engineering, quality of service management, policy-based systems, multimedia service control, programmable networks, and communications middleware.