

# Distributed Policy Management Protocol for Self-Configuring Mobile Ad Hoc Networks

Mouna AYARI, Farouk KAMOUN and Guy PUJOLLE

**Abstract** Mobile ad hoc networks (MANETs) are fundamentally different from wired networks. They are autonomously formed with a collection of mobile nodes without any preexisting infrastructure or administrative support. In the last few years, Policy-Based Network Management (PBNM) has gained a considerable interest in order to reach adaptive and automated management goals in MANETs. Previous work in this field either cluster-based or hierarchical architectures are highly dependent of the network topology. A distributed approach with a high degree of autonomy and self-management is still lacking. In this paper, we give an overview of existing policy-based management solutions and we focus on some of their limitations. We outline the need of incorporating self-configuration property in the design of MANETs. Then, we present a new protocol for distributing policies and high-level goals over all nodes in the network. Simulation-based performance evaluation results are described and analyzed.

## 1 Introduction

A mobile ad hoc network (MANET) consists of a collection of mobile wireless nodes that dynamically create a network without any existing infrastructure or ad-

---

Mouna AYARI

CRISTAL lab, National School of Computer Sciences, ENSI, 2010, Manouba, Tunisia and  
LIP6 University of Paris-VI, 104, Avenue Pd Kennedy, 75016, Paris, France e-mail:  
mouna.ayari@lip6.fr

Farouk KAMOUN

CRISTAL lab, National School of Computer Sciences, ENSI, 2010, Manouba e-mail:  
frk.kamoun@planet.tn

Guy PUJOLLE

LIP6 University of Paris-VI, 104, Avenue Pd Kennedy, 75016, Paris, France e-mail:  
guy.pujolle@lip6.fr

ministrative support. Some of the features that characterize ad hoc networks are the dynamic topology and the limitation of resources [1, 2]. Mobile nodes are also operating under severe constraints (limited battery power, variable link quality, limited storage capacity, etc.), which makes the network management process more difficult [3]. We aim in this work to improve self-configuration capabilities in MANETs management. We believe that it is very useful to incorporate this property of autonomic computing into the design of MANETs since these networks are self-creating and operate without a centralized control. The first initiative towards autonomic computing was proposed by IBM through the eLisa project [4]. After that, many researches have been carried out in this area [5, 6]. An autonomic network consists of autonomous network elements. These entities are able to adapt themselves to the changes that can affect their environment while respecting high-level policies. One of the defined properties of these networks is the self-configuration. The network will be able to configure and automatically reconfigure itself under varying conditions and changes in its environment. Human intervention will be limited to guiding the network behavior by defining high-level directives.

Policy-Based Network Management (PBNM) has been adopted by the IETF (Internet Engineering Task Force) mainly to provide automation in network configuration process [14, 15]. This approach allows the definition of high level objectives based on a set of policies that can be enforced in the network. Policies are defined as a set of rules to manage and control access to network resources [7]. An efficient network management system for MANETs can be realized through a reliable distributed policy-based management approach. However, traditional PBNM systems are originally conceived to be used in a centralized LAN-like network environment. We note that the idea to use a fully distributed policy-based management has been proposed the first time in [8]. This paper proposes a new protocol named DPMP for Distributed Policy Management Protocol enabling policies and high-level directives distribution in a fully distributed manner.

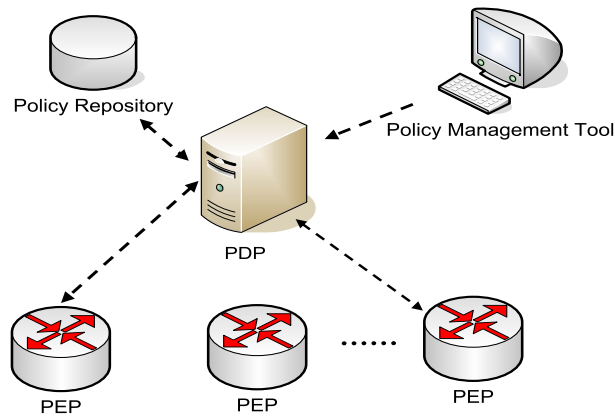
The remainder of this paper is organized as follows. Section 2 describes and reviews the related work in policy-based network management. In section 3, we present the basic idea of our proposed system, its components and an overview of a new protocol named DPMP for Distributed Policy Management Protocol. In particular, we provide a detailed description of the proposed policy distribution process. Simulation experiments and results are presented in section 4. Finally, section 5 concludes the paper and presents future work.

## 2 Overview of Policy-Based Management solutions

Many researches have been carried out in the area of policies for the network management. In this section, we provide an overview of architectures and the commonly used protocols for policy-based management.

The IETF (Internet Engineering Task Force) has defined a policy framework to the admission control. As shown in Figure 1, the proposed architectural model for

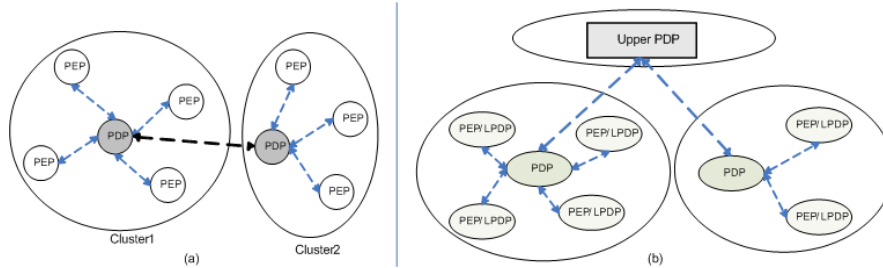
policy framework consists of the following four tiers: a policy management tool, a PDP or Policy Decision Point, a PEP or a Policy Enforcement Point and a Policy Repository. The policy management tool is an interface between a network manager and the PBNM system through which policies may be edited, modified and deleted. The PDP is a component responsible for high-level decisions making. A PEP represents the network element where decisions will be enforced. The PDP's decision is based on network level information collected from network devices and policies retrieved from the policy repository (a location where policies are stored in a structured way).



**Fig. 1** Policy-Based Management System

Based on information collection and communication strategy, we can distinguish in the literature two types of policy-based management architectures: centralized and cluster-based. The IETF framework [17] is based on a centralized approach. System components are operational within a single domain administrated and managed using a common set of policy rules. There is a single PDP that controls and manages the network. It is clear that this approach is not designed for dynamic and distributed environments, such as MANETs. On one hand, the PDP is a single point of failure that manages and controls entire network devices. The performance of the system deteriorates rapidly when the number of PEPs connected to the PDP increases [16]. On the other hand, ad hoc networks have a frequently varying-topology. So, the network may get partitioned periodically and nodes will become disconnected frequently. In this case, disconnected partitions will be left without any management control.

In order to extend policy-based management to MANETs, cluster-based architectures have been adopted. As shown in Figure 2, several PEPs are grouped into clusters. Each cluster is managed by a PDP [9]. The system management can be also hierarchical [18, 12, 13]. In this case, PDPs in turn are managed by an upper PDP. Clustering techniques transform the ad hoc network to a logical centralized system within each cluster. However, the process of forming and maintaining clusters may



**Fig. 2** Cluster-Based architectures

produce a significant additional message overhead. This is highly dependent on the network topology. Cluster-based techniques are adapted for static ad hoc networks with a low varying topology. But, in the case of a dynamic mobile ad hoc environment, as the topology of the network changes, the structure and the composition of clusters change. Similarly, nodes acting as clusterheads may change all the time. In addition, due to the unpredictable and varying nature of ad hoc networks, policy information may change at any time. Thus, new adequate decisions should be sent from the PDP to all PEPs belonging to its cluster. Each decision message issued due to a request or a policy update should be acknowledged by each PEP with a report message. This is a critical scenario since updating a group of PEPs (possibly large) at the same time may impact the bandwidth consumption. Moreover, contacting the PDP each time a local event or message invokes a PEP for a policy decision, would affect policy management response time. Degradation in management system performance is observed as the number of hops between a policy server and client increased [9]. Furthermore, a policy transfer between PDPs is required each time a PEP leaves its domain or cluster to another one. So, the problem of inter-domain policy transfer is further exacerbated by the high degrees of mobility in ad hoc networks.

Different protocols have been used to distribute policies over MANETs. In [9], COPS-PR (COPS for Provisioning) protocol [10] was used to exchange policy information and decisions between PDPs and PEPs. COPS-PR was originally conceived as an extension to COPS (Common Open Policy Service) protocol [11] to support differentiated service policies over wireline fixed Internet. So, extensions to the COPS-PR protocol [9] were proposed in order to allow delegation and redirection capabilities in MANETs. However, the use of COPS-PR may affect the limited ad hoc network resources. This protocol was designed for a centralized policy management system where bandwidth is not a critical resource. It is a heavyweight connection-oriented protocol.

In [12], several protocols (YAP (Yelp Announcement Protocol), AMPS (Ad hoc Mobility Protocol Suite), DRCP/DCDP (Dynamic and Rapid Configuration Protocol/Dynamic Configuration Distribution Protocol) were used jointly in order to distribute, control and collect policy management information. However, this may limit the adoption of such a solution. Besides, DRCP/DCDP protocols have been designed for IP-address configuration and assignment in MANETs.

## 3 Distributed Policy Management Protocol

### 3.1 *Basic idea*

The basic idea is to distribute the functionalities of the manager PDP among ad hoc nodes as LPDP. However, the use of a fully-distributed approach within a policy based network management system can encounter some difficulties. In fact, if multiple autonomous management entities (LPDP) are making independent decisions, a lack of consistency between them may occur. In order to regulate and coordinate LPDP's decision making, we propose to use predefined proactive policies. Initially, the human network manager defines policies and high level directives to manage the network and introduces them into at least one node. Policies are expressed as event-condition-action rules. A specific action can be taken at the observation of an event or/and when a particular condition occurs. These policies will be automatically distributed and replicated in the policy repository of each node. They regulate the access to resources and govern dynamic reconfigurations in function of network condition changes.

Each mobile ad hoc node will contain LPDP, PEP, local policy repository and a monitor. The MANET will be formed with a set of autonomous elements that know how to operate in a cooperative way. The LPDP makes local decisions to be enforced by the PEP. The monitor is responsible for collecting monitoring information and changes in environment conditions.

### 3.2 *Protocol Description*

Based on requirements expressed in previous sections, we have proposed a distributed protocol that we called: DPMP (Distributed Policy Management Protocol).

#### 3.2.1 DPMP Messages

Firstly, we present different DPMP messages.

- *DPMP\_DISCOVER*: a one-hop broadcast message sent by a non-configured LPDP to neighboring nodes in the network to request proactive management policies.
- *DPMP\_OFFER*: a unicast message sent by a configured LPDP in response to a DISCOVER message. A policy requester can receive more than one OFFER message. An OFFER message does include neither policies nor decisions. It simply indicates a service availability.
- *DPMP\_DECISION*: this message includes policy decisions. It is a unicast message sent by a LPDP to another LPDP. It is used also as a local message sent by the LPDP to its local PEP.

- *DPMP\_ACCEPT*: a unicast message sent by a LPDP in order to accept an OFFER or DECISION message from another LPDP.
- *DPMP\_REJECT*: a unicast message sent by a LPDP in order to refuse or reject an OFFER or DECISION from another LPDP. It is also used by the PEP to communicate to the LPDP its failure in carrying out the LPDP's decision.
- *DPMP\_RETRIEVE\_STATUS*: a unicast message including a request for monitoring information. It is sent from the monitor to either the local PEP or a specific monitor.
- *DPMP\_REPORT\_STATUS*: a unicast message containing reported monitoring information. It can be a response for a DPMP\_RETRIEVE\_STATUS message or triggered by a specific event.

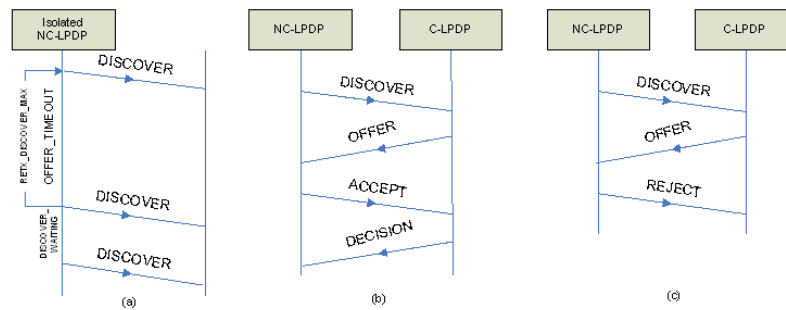
Each DPMP message consists of a header followed by a number of typed body objects. The header includes a common header and a variable optional part specific for some particular messages. We note that DISCOVER and OFFER messages don't contain neither options nor body. We note also that any transport protocol can be used with DPMP. We have defined appropriate timers and confirmation messages in order to handle message losses. In particular, we propose to use UDP since it is a lightweight protocol.

### 3.2.2 Policy Distribution

In the scope of this paper, we describe in details and evaluate the policy distribution mechanism. The main objective of this mechanism is to distribute proactive policies to non-configured nodes with a low communication overhead. We need to reduce the number of messages transmitted, received and processed at each node.

We call a configured node a node that has implemented proactive policies. Initially, all nodes are not configured. The network manager has to introduce a set of proactive policies through a policy management tool in at least one node.

Nodes can join or leave the ad hoc network at any time. Each new node starts a policy discovery procedure. The mechanism is simple. A non-configured LPDP (NC-LPDP) requests for proactive policies.



**Fig. 3** Policy Discovery Process

As shown in Figure 3.a, a NC-LPDP first broadcasts a DISCOVER message to its neighbouring nodes and waits for the receipt of an OFFER message until the DISCOVER\_TIMEOUT timer expires. If it doesn't receive any response, it re-sends the DISCOVER message. This process can be repeated up to RETX\_DISCOVER\_MAX retransmissions. When the allowed retrials are exhausted, the node concludes that there is no configured node in the neighbourhood. It is due to an isolation state or a network partition event. We call this period of isolation "isolation period". The node waits for a DISCOVER\_WAITING period and then repeats the policy discovery procedure.

As depicted in Figure 3.b, if a configured LPDP (C-LPDP) receives a DISCOVER message, it responds with an OFFER message. Upon receiving an OFFER message, the NC-LPDP checks the message. If the offer is refused, a REJECT message is sent back to the C-LPDP with the corresponding error code and the NC-LPDP waits for the receipt of another OFFER message. Otherwise, the NC-LPDP responds with an ACCEPT message. Upon receiving an ACCEPT message, the C-LPDP sends a DECISION message back to the NC-LPDP. This message embeds related proactive policies. DECISION messages may be lost. The NC-LPDP waits for a DECISION\_WAIT\_MAX. If the timer expires without receiving the DECISION message, it re-sends another ACCEPT message and waits for a response. If the retrial fails, the NC-LPDP re-broadcasts a new DISCOVER message to its neighborhood. If a NC-LPDP receives a DECISION message, it processes policy objects, installs policies and updates its state to a configured LPDP. Then, it continues to listen to DPMP messages.

It is important to note that the NC-LPDP can receive multiple OFFER messages from its neighbors (if it has more than one configured neighbor). The NC-LPDP chooses the first one. Thus, the OFFER message doesn't contain policy information in order to reduce the signaling overhead. In fact, it is evident that the size of an OFFER message (consisting of the common header), is smaller than the DECISION size (that depends on the number of policies). For example, the size of the LPDP common header is 24 bytes while the size of policies may reach a few kilobytes.

## 4 simulations and Results

In this section, we evaluate the performance of the proposed policy distribution mechanism.

### 4.1 Simulation Environment

We implemented the described policy distribution mechanism and evaluated its performance using the network simulator NS-2. We have considered different scenarios. In all these simulation scenarios, we have used the random-waypoint mobility

model. The minimum node speed parameter was set to around 95% of the maximum speed. The pause time was fixed to 10 seconds. We have used AODV as routing protocol, although any other routing protocol can be deployed. Different chosen constants and timers values were set as follows: 1 second to DISCOVER\_TIMEOUT, 5 seconds to DISCOVER\_WAITING and 10 to RETX\_DISCOVER\_MAX. Simulation started with one configured node randomly chosen. The arrival of the nodes is assumed to be a poisson-process. The arriving node could appear anywhere within the simulation area. We run each simulation scenario until all nodes were configured. we experimented with a large number of topologies. 30 runs of each simulation scenario were performed with varying random simulation seeds. Each scenario represents a random initial placement of arriving nodes. The 95% confidence interval is computed for all simulation results.

In order to evaluate the performance of our DPMP distribution mechanism, we have considered the following metrics:

- *Latency*: It is the time taken by the protocol DPMP from the instant a non-configured node enters the network and sends a DISCOVER message until it receives a DECISION message. So, for a node that joins the network at time  $t_1$  and obtains policy decisions at time  $t_2$ , its related latency is  $(t_2-t_1)$ . The DPMP latency is averaged over all nodes in the network.
- *Message overhead*: it represents different types of DPMP messages required for the policy distribution process, namely DISCOVER, OFFER, ACCEPT and DECISION messages. We consider the average amount of bytes generated by these messages.

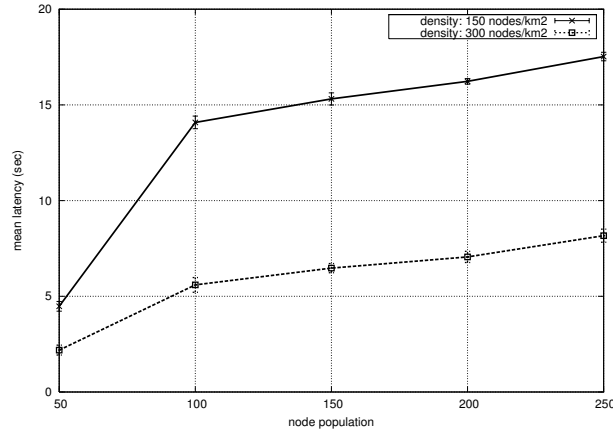
## 4.2 Simulation Results

In the first set of simulations, we examine the impact of varying the network size, the mean node arrival rate and node speed on latency. The transmission range of each node is set to 100m. The nodes arrival is assumed to be a poisson-process.

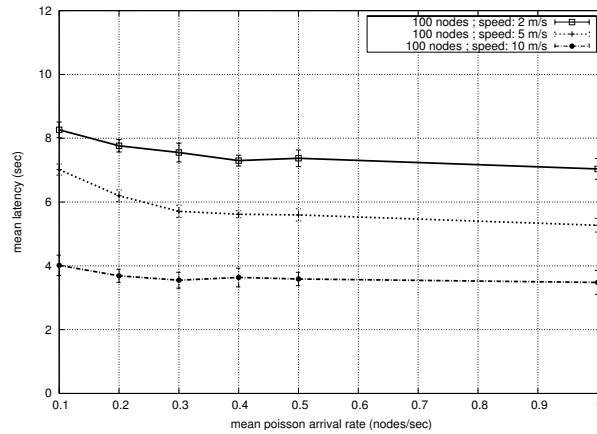
In order to assess in more details the impact of varying network topology in the mean latency, we have varied node population from 50 to 250 nodes and we considered two network density 150 and 300 nodes/km<sup>2</sup>. Figure 4 plots the mean latency as a function of node population. The node speed was fixed to 5 m/s. The mean poisson arrival nodes was set to 0.2 node/s. As shown, starting from 100 nodes, the mean latency increases slightly when the network size increases. Moreover, we can observe that the mean latency decreases when the network density increases. The more the network density increases, the more the probability to have rapidly a configured node in the neighborhood increases.

Then, we varied the mean poisson arrival rate from 0.1 to 1 node arrivals/s. We considered three node speeds: 2m/s, 5m/s and 10m/s. Figure 5 plots the mean latency as a function of the mean poisson arrival of nodes for 100 node-system population and 300 nodes/km<sup>2</sup> network density.





**Fig. 4** Mean latency with varying node population and density; node speed = 5m/s

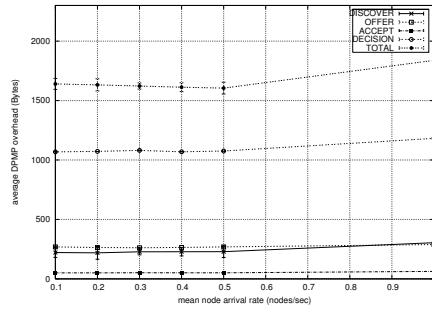


**Fig. 5** Mean latency with varying mean node arrival rate

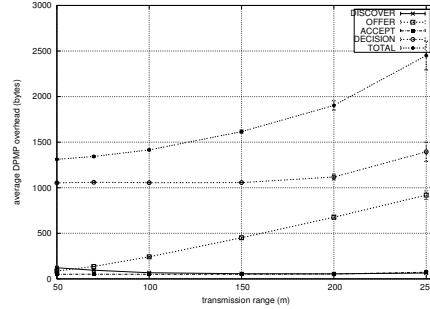
As depicted in Figure 5, the mean latency decreases slightly when the mean poisson arrival rate increases. Results show that the mean latency doesn't exceed 8.5 seconds. Moreover, we observe a decrease of the mean latency with the node speed increase.

In the second set of simulations, we evaluate the overhead generated by the DPMP protocol during distributing policies to all ad hoc nodes. We have fixed the network density to 150 nodes/km<sup>2</sup> for a population of 100 nodes. Nodes move with a speed of 5m/s and we have fixed policies size to 1024 Bytes. Considering the LPDP common header and the option of each message, the sizes of DPMP messages were set as follows: 49 Bytes to DISCOVER and OFFER messages, 51 Bytes to ACCEPT message and 1076 Bytes to DECISION message. The effect of increasing the mean poisson arrival rate from 0.1 to 1 nodes/s on message overhead is depicted in Figure 6. Results show that the distribution policy process of the DPMP protocol

does not produce high overhead. The overall size of OFFER and DISCOVER messages does not exceed 300 Bytes per node. We observe also that the overall size of a single type message is almost stable in function of node arrival rates. We observe a little increase of the amount of DISCOVER messages in the case of the mean poisson arrival 1 node/s. This is predictable since as the arrival rate of non-configured nodes increases, a node may stay sending DISCOVER messages until at least one of its neighbors becomes configured. This does not have a great impact since DISCOVER messages have a small size. Moreover, the little increase of the amount of DECISION messages generated per node for the mean arrival rate of 1 node/s is due to message loss.



**Fig. 6** Mean overhead with varying mean node arrival rate



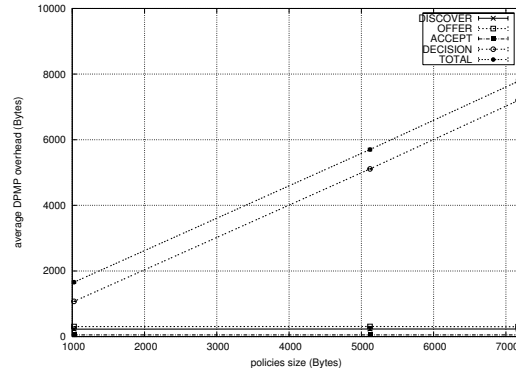
**Fig. 7** Average overhead with transmission range

Figure 7 plots the variation of the average DPMP message overhead in function of node transmission range for the same scenario described above. The mean poisson arrival rate of nodes is fixed to 0.2 node/s. We have varied the transmission range of nodes from 50 to 250 m. Results show a linear increase of the amount of OFFER messages with the transmission range increase. In fact, as transmission range increases, as the connectivity of nodes increases, the probability to have more configured nodes in the neighborhood increases. The amount of additional DPMP messages is varied from 180 and 1300 bytes per node. In brief, we can conclude that the DPMP policy distribution mechanism generates a limited additional overhead per node. In the majority of cases, policies information are exchanged one time per node as confirmed in Figure 8.

We have measured the average DPMP overhead when we varied policies sizes from 1024 to 7168 bytes. We have fixed the transmission range to 100 m. Results show a linear increase of the amount of DECISION messages generated per node with the increase of policies size. Amounts of DISCOVER, OFFER and ACCEPT messages generated per node are practically stable.

Finally, we can conclude that the increase of the DPMP overhead generated per node depends on policies sizes rather than on network topology. Moreover, the average number of DECISION messages exchanged per node is around 1 to 1.3 mes-

sages in all the undertaken simulations. We note also that in a real scenario, policies size may reach tens of KBytes.



**Fig. 8** Average overhead with varying policies size

## 5 Conclusions and Future work

In this paper, we present a fully-distributed PBNM system for MANETs. Policy information is distributed without central control. Our solution provides autonomy, self-configuration and limited human intervention. We have proposed and described a new protocol for policy distribution that addresses requirements for minimizing overhead and providing a reliable policy distribution. We have analyzed protocol performance through several simulation scenarios. Simulation results show that the increase in the mean of poisson-arrival node rates hasn't got a great effect neither on latency nor on DPMP message overhead. We verified also that varying node speed, network density and transmission range hasn't got a great impact on system performance. This clearly demonstrates the benefits of our distributed approach. In our future research activities, we will complete the implementation and validation of our designed DPMP protocol. We will also compare the performance of our fully-distributed policy-based management system to other policy-based solutions conceived for MANETs.

## References

1. Perkins and Hughes: Factors Affecting the Performance of Ad Hoc Networks. (2002). In: Proceedings of the IEEE International Conference on Communications, Vol. 4, April 2002, pp. 2048- 2052.

2. Chakrabarti and Mishra: QoS issues in ad hoc wireless networks, *IEEE Communications Magazine*, February 2001.
3. Mirhakkak, Schult and D. Thomson: Dynamic Quality-of-Service for Mobile Ad Hoc Networks. (2000). In: IEEE MobiHoc, Boston, Massachusetts, USA, August 2000.
4. Horn. Autonomic Computing:IBMs perspective on the State of Information Technology. (2001). IBM Corporation
5. Schmid, Sifalakis and Hutchison: Towards Autonomic Networks. (2006). In: proceedings of 3rd Annual Conference on Autonomic Networking, Autonomic Communication Workshop (IFIP AN/WAC), Paris, France, September 25-29.
6. Parashar and Hariri: Autonomic computing: An overview. UPP 2004. (2005). In: Springer Verlag, 3566:247259, January 2005.
7. Westerinen and al.: Terminology for Policy-Based Management. (2001). In: IETF RFC 3198, November 2001.
8. Munaretto, Agoulmine, M. Fonseca: Policy-based Management of Ad Hoc Enterprise Networks. (2002). In: HP Openview University Association 9th Annual Workshop. 2002.
9. Phanse and DaSilva: Protocol Support for Policy-Based Management of Mobile Ad Hoc Networks, Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP, Vol. 1, 2004, pp. 3-16.
10. Chan and al: COPS usage for Policy Provisioning (COPS-PR). (2001). In: IETF REF 3048, March 2001.
11. Durham and al.: The COPS (Common Open Policy Service) Protocol. (2000). In: IETF RFC 2748, January 2000.
12. Chadha, Cheng, Cheng, Chiang, Ghetie, Levin and Tanna: Policy-Based Mobile Ad Hoc Network Management for Drama, *MILCOM Journal*, Vol 3, 2004, pp. 1317-1323
13. Chadha, Cheng, Chiang, Levin, Li, and Poylisher. DRAMA: A Distributed Policy-based Management System. (2005). In: Proceedings of the Third International Conference on Mobile Systems, Applications, and Services, June 6-8, 2005, Seattle, WA.
14. Ponnappan, Yang, Pillai.R: A Policy Based QoS Management System for the IntServ/DiffServ Based Internet. (2002). In: Third International Workshop on Policies for Distributed Systems and Networks (POLICY.02), 2002, pp. 159-169.
15. Verma, Calo and Amiri: Policy Based Management of Content Distribution Networks, *IEEE Network Magazine*, vol.16, March 2002, pp. 34-39.
16. Eddie Law: Scalable Design of a Policy-Based Management System and its Performance, *IEEE Communications Magazine*, vol.3, Issue 6, June 2003, pp. 72- 79.
17. Yavatkar, Pendarakis, Guerin: A Framework for Policy Based Admission Control. (2000). In: IETF RFC 2753, January 2000.
18. Hadjiantonis, Malatras, Pavlou: A context-aware policy-based framework for the management of MANETs. (2006). In: Proceedings of the seventh IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'06), June 2006, pp. 23 - 34.