

Policy Distribution Using the Publish/Subscribe Paradigm for Managing MANETs

Vasilis Sourlas, Paris Flegkas, and Leandros Tassiulas

Dept. of Computer and Communication Engineering,
University of Thessaly, Volos, Greece
{vsourlas,pflegkas,leandros}@inf.uth.gr

Abstract. Policy management has been an emerging management paradigm that has been extensively studied for the case of fixed networks and limited work has already been done for migrating it to mobile environments. Publish/subscribe has become an important architectural style for designing distributed systems and especially for mobile environments due to the loose coupling of the components involved. In this paper, we present a policy-based management system for Mobile Ad hoc Networks (MANETs) using the publish/subscribe paradigm for distributing policies to the managed nodes. Moreover, we enhance the publish/subscribe system with a novel request/response mechanism for tackling the problem of how newly joined nodes will retrieve previously introduced policies.

1 Introduction

Management of Mobile Ad hoc Networks (MANETs) has recently attracted a lot of attention due to the proliferation of mobile/pervasive devices and their ability and need to communicate. The nature and their inherent characteristics such as topology changes, mobility of nodes and limited terminal capabilities of such networks pose new challenges and requirements for their management. Traditional management approaches in fixed networks based on centralized approaches (SNMP) cannot be applied in ad hoc networks. MANETs are self-creating in the sense that mobile hosts need to dynamically discover other nodes and form spontaneously a network from a collection of mobile hosts that have the ability to route information. It is evident that the highly dynamic environment of MANETs can benefit from the self-management capabilities provided by Policy-based Management (PBM). PBM has been applied to fixed networks using also a centralized architecture where a Policy Decision Point (PDP) resides in a management server and is responsible for retrieving policy rules, defined by the operator in the Policy Management Tool (PMT), stored in the Policy Repository (PR) and translating them to management operations on the network elements i.e. Policy Enforcement Points (PEPs), realizing this way the business objectives of the operator [1]. In order to apply PBM in ad hoc networks, a hierarchically distributed management approach has to be followed, based on their characteristics and behavior as described by related work in the literature [2].

The publish/subscribe paradigm has become an important architectural style for designing distributed systems. Applications that exploit a publish/subscribe

communication paradigm are organized as a collection of autonomous components (clients), which interact by publishing event notifications and by subscribing to the classes of events they are interested in. The event dispatcher (broker), is responsible for collecting subscriptions and forwarding events to subscribers. The anonymity and dynamism of publish/subscribe allow the systems to adapt quickly to frequent connections and disconnections of mobile nodes.

Network management has always been one of the most popular applications using an event-based paradigm where managers subscribe to events published by the managed elements. In this paper, we present a policy-based management architecture for mobile ad hoc networks based on the publish/subscribe communication paradigm for distributing policy rules from the points they are introduced to all the responsible components for their enforcement. We believe that the dynamic environment of MANETs can benefit from such a communication paradigm. Policy distribution is a critical task in managing MANETs since there is a need to assure that all nodes comply with policies introduced by the operator leading the system to stability. Moreover, we enhance the publish/subscribe paradigm with a request/response mechanism so that nodes that join the network can retrieve previously introduced policies. This is important since nodes might not have joined the network when the operator defines a policy and current publish/subscribe systems do not support retrieval of previous or missing events.

The rest of the paper is organized as followed. Section 2 describes related work in the area of management of MANETs and publish/subscribe systems. Section 3 presents our proposed policy management architecture based on the publish/subscribe paradigm for distributing policies. Section 4 describes the enhanced request/response mechanism while section 5 presents our initial design and implementation. Finally, section 6 concludes this paper and gives pointers to our future work.

2 Related Work

Limited work has been done in the area of management of ad hoc networks where most approaches follow a hierarchical organization model. The first attempt was made by [3], proposing an Ad hoc Network Management Protocol (ANMP) based on hierarchical clustering of nodes in a three level architecture. The “Guerilla” architecture [4] adopts an agent-based two-tier distributed approach where at the higher level “nomadic managers” make decisions and mobile agents exploit a utility function to decide their migration and probe deployment to fulfill management objectives. Finally in [2], a hybrid hierarchical and distributed approach is adopted, presenting a system architecture that uses the policy-based management paradigm together with context awareness for managing MANETs. While all the above approaches were designed for solving relevant problems in the area of ad hoc network management, none of them has dealt with the critical issue of policy distribution in MANETs. The use of a publish/subscribe architecture for distributing policies in MANETs has not been considered in the past and comprises one of the most innovative aspects of our approach.

There are several research efforts concerned with the development of an event notification service such as Siena [5] and REDs [6] which implement the publish/subscribe architecture. REDs defines a protocol to organize the nodes of a mobile

ad hoc network in a single, self-repairing tree that efficiently supports content-based routing. In achieving this goal REDs implement COMAN (Content-based routing for Mobile Ad hoc Networks) [7], which was designed to tolerate the dynamics of the underlying physical network, characteristic of MANETs.

3 System Architecture

We adopt a 2-tier hierarchical and distributed approach for our management organization model where nodes of the network are categorized based on their management functionality. The simplest nodes are the ones only with the PEP functionality, offering only a management interface to nodes with enhanced management intelligence denoted as cluster managers in [4]. Depending on the degree of distribution policy repositories (PR) can be hosted in several nodes in the network avoiding this way the retrieval of all policies from the PDPs from a single point in the network. Finally, in the highest level of our hierarchy are the nodes with the PMT functionality, enabling the administrator of the network to introduce policies that should be stored in the repositories and enforced by all the nodes with the PDP functionality.

In order to incorporate a robust and efficient policy distribution mechanism all the policy management components described above communicate using the publish/subscribe paradigm. In order to achieve that, several nodes of the network take the role of the event broker i.e. a publish/subscribe (pub/sub) router as it is shown in Figure 1 forming an overlay publish/subscribe network handling the distribution of policies from the point they are introduced to the repositories and the PDPs responsible to enforce them. All the policy components are the clients of the pub/sub network namely the publishers and the subscribers. Our system architecture is depicted in Figure 1 where all nodes with PDP functionality subscribe to the pub/sub router they are connected to waiting for a relevant policy to be introduced/published by the administrator (PMT).

In our work, we assume the following well adopted representation of a policy as defined by IETF [8] and used in our previous work on policy management [9]:

$$\{Roles\} [TimePeriod] \text{ if } \{conditions\} \text{ then } \{actions\}$$

The concept of roles is critical in our architecture since it defines the scope of the policy i.e. to which nodes it applies and comprises the subscription filter in the pub/sub network. Nodes with only the PEP functionality first supply their role to their cluster manager node that creates an aggregated subscription with all the roles of the managed nodes in order to receive all the relevant policies. Nodes with the PDP functionality present and not in a Cluster manager role subscribe only to policies related to their own operation.

The operator of the network defines the policies in a node with the PMT functionality which publishes the policy to the pub/sub router it is attached to (or physically co-located) and the pub/sub network is responsible for delivering this policy to all the relevant components based on the Roles attribute and the subscription tables of the pub/sub routers that were created by the client subscriptions. With the recent advances in the pub/sub systems and their enhancements for mobile environments the pub/sub overlay network takes all the reconfigurations actions needed to assure a minimum

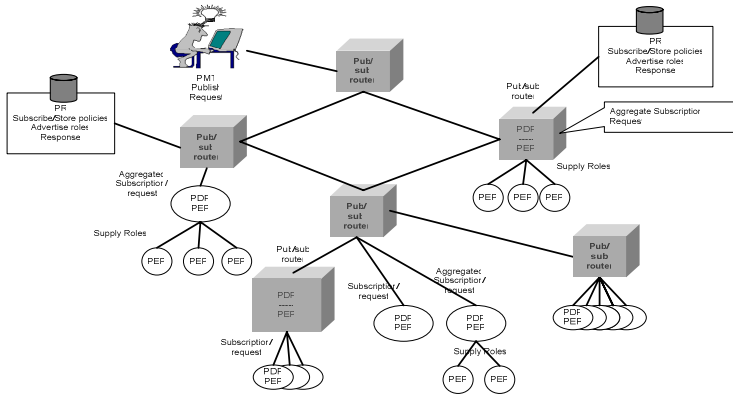


Fig. 1. System Architecture

amount of lost messages. Finally, nodes with the Policy Repository component also subscribe to receive policies and are responsible to store them for later retrieval by new subscribers or for validation and conflict detection checking before a new policy is introduced.

4 Enhancing Publish/Subscribe with Request/Response

In our pub/sub system, we use the subscription forwarding routing strategy [5] where the routing paths for policies are set by subscriptions, which are propagated throughout the network so as to form a tree that connects the subscribers to all the brokers in the network. This scheme is optimized to avoid forwarding the same subscriptions in the same direction by exploiting “coverage” relations among filters. This means that a subscription is forwarded to a neighboring broker only if it is not being covered by a subscription already forwarded to the same neighbor. Particularly we say that a subscription S_1 covers another subscription S_2 , denoted by $S_1 \geq S_2$, iff any event matching S_2 also matches S_1 [10]. When a client (PMT) publishes a message (policy) that matches a subscription, the message is routed towards the subscriber following the reverse path put in place by the subscription.

The subscription forwarding routing algorithm and all its known implementations ([5] and [6]) does not provide the capability of retrieving a published message (policy) at a time later than the time of its publication. In order to achieve this, we enhance the pub/sub system with advertisement messages. Each broker maintains a set ST “Subscription Table” containing the identifiers of the brokers to which the broker is connected and the subscriptions that those neighbors had sent to the broker. In our case, we add to each broker another set AT “Advertisement Table” which contains the identifiers of the brokers to which the broker is connected and the advertisements that those neighbors had sent to the broker.

Advertisements are messages sent by the PR nodes containing the Roles to which the stored policies are referred to and are treated similarly to subscription messages so as to form a tree that connects the PRs to all the brokers in the network. Coverage also

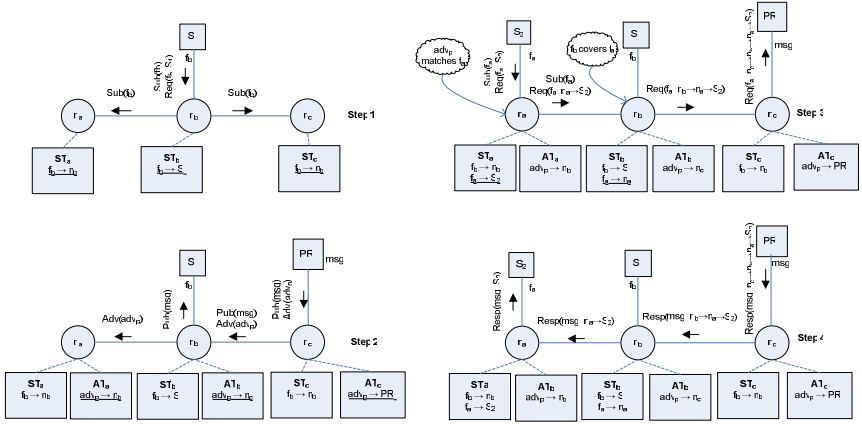


Fig. 2. Enhanced publish/subscribe paradigm. Step 3 and step 4 illustrates the novel introduced Request() and Response() messages.

occurs with advertisements and as in subscriptions is used to avoid forwarding the same advertisements to the same direction.

We also add to the system two additional types of messages. We call them Request() and Response(). As shown in figure 2, when a client node s_2 (PDP) subscribes to a broker node n_a (step 3), sending a `Subscribe(f_a)` message also sends a `Request(f_a, s_2)` message. The `Request()` message is similar to the `Subscribe()` message but apart from the filter it also carries the sequence of the nodes that passes from. Node n_a upon receiving the `Request()` message checks in AT_{n_a} for advertisements matching filter f_a . If such an advertisement occurs (adv_a), the broker forwards the `Request()` message to the identifier for which the adv_a was in the table otherwise the `Request()` message is dropped.

When a broker n_b receives a `Request()` checks in AT_{n_b} for matching advertisements and according to the findings, it forwards the `Request()` appending its identifier. In this case, there will be an entrance in a brokers' AT (AT_{n_c}) which points in a PR node. Upon receiving the `Request()` the PR client checks for the matching messages and sends back a `Response()` message (step 4) for each matching message. The `Response()` message is similar to the `Publish()` one but apart from the message it also carries the sequence of nodes carried by the `Request()`. When a broker receives a `Response()` message pops off its identifier from the sequence and forwards it to the first broker of the remaining sequence. In the end, client s_2 will receive the message. With the above procedure every new subscriber and only that one will receive every old message (policy) matching its role.

5 Design and Implementation

Two of the most popular publish/subscribe systems, where the subscription forwarding routing algorithm is implemented, are Siena [5] and REDS [6]. We used REDS since it was designed to tolerate dynamic reconfigurations of the dispatching infrastructure. We mainly modified the Routing layer of the REDS systems in order to

implement the newly introduced messages and support the storage of the advertisements (Advertisement Table). We have also altered the Overlay layer in such a way that can transfer the new type of messages. Of course, minor changes had been made to other modules in order to make the system compatible.

6 Conclusion and Future Work

In this paper, we described an architecture for policy-based management of mobile ad hoc networks focusing on the critical problem of policy distribution. We showed how we can incorporate a publish/subscribe communication paradigm, providing a robust and efficient policy distribution mechanism in such a volatile network environment.

Initial tests showed very promising results and our future work focuses on performing experiments in different mobility scenarios, evaluating our system both in terms of message (policy) delivery as well as in terms of overhead.

Acknowledgement

This paper is part of the 04EP106 research project implemented within the framework of ENTER-2004 and co-financed by National and Community funds (25% from the Greek Ministry of Development – General Secretariat of Research and Technology and 75% from E.U.- European Social Fund).

References

1. Strassner, J.: Policy-Based Network Management, Solutions for the next generation. Morgan Kaufmann, San Francisco (2003)
2. Hadjiantonis, M., Malatras, A., Pavlou, G.: A Context-aware Policy-based Framework for the Management of MANETs. In: 7th IEEE International Workshop on Policies for Distributed Systems and Networks, Canada, pp. 23–32 (2006)
3. Chen, W., Jain, N., Singh, S.: ANMP Ad hoc network management protocol. IEEE Journal on Selected Areas in Communications 17 (1999)
4. Shen, C., Srisathapornphat, C., Jaikaeo, C.: An adaptive management architecture for ad hoc networks. IEEE Communication Magazine 41 (2003)
5. Carzaniga, A., Rosenblum, D., Wolf, A.: Design and evaluation of a wide-area event notification service. ACM Transaction On Computer Systems 19, 332–383 (2001)
6. Cugola, G., Picco, G.: REDS, A Reconfigurable Dispatching System. In: 6th International workshop on Software Engineering and Middleware, Oregon, pp. 9–16 (2006)
7. Mottola, L., Cugola, G., Picco, G.: A Self-repairing Tree Topology Enabling Content-Based Routing in Mobile Ad Hoc Networks. IEEE Transaction on Mobile Computing 7, 946–960 (2008)
8. Moore, B., Ellesson, E., Strassner, J., Westerinen, A.: Policy Core Information Model. RFC 3060, IETF (2001)
9. Flegkas, P., Trimintzios, P., Pavlou, G., Liotta, A.: Design and Implementation of a Policy-based Resource Management Architecture. In: IEEE/IFIP Integrated Management Symposium, Colorado, pp. 215–229 (2003)
10. Chand, R., Felber, A.: A scalable protocol for content-based routing in overlay networks. In: 2nd IEEE International Symposium on Network Computing and Applications, pp. 123–130 (2003)