Scalable Policy Distribution for Ambient Networks

Alberto Gonzalez Prieto, Rolf Stadler
KTH, Royal Institute of Technology, Sweden
{gonzalez,stadler}@imit.kth.se

Abstract— The characteristics of policy-based management make it a promising candidate for managing Ambient Networks, which are characterized for being highly dynamic and heterogeneous. However, current policy-based approaches are not scalable, which is a must for such dynamic scenarios. A key aspect for developing scalable systems is policy distribution, the mechanism that provides the right policies at the right locations in the network when they are needed.

In this paper, we present a scalable framework for policy distribution for Ambient Networks. The framework is based on aggregating the addresses of the policies and applying multipoint communication techniques. The aggregation is based on grouping the managed elements by the role they play in the network and distributing policies that apply to all the elements in a group. We show the validity of the framework by applying it to a case study.

***Index Terms*—Ambient Networks, Policy-based Management.**

I. INTRODUCTION

Ambient Networks (AN) aim at allowing the establishment of inter-network agreements on-demand without the need for pre-configuration and offline negotiation between network operators [5]. The key concept of Ambient Networks is network composition. Networks compose and gain connectivity through instant establishment of inter-network agreements. In order to achieve this, Ambient Networks require the creation of robust and technology independent platforms for mobile communication networks. Instant composition makes AN highly dynamic, bringing new challenges to network management [6].

Policy-based Network Management (PBNM) is a management paradigm based on specifying management objectives, instead of the mechanisms that implement them. The characteristics of PBNM make it particularly appropriate for AN. First, PBNM provides **implementation independence**, a key aspect in heterogeneous scenarios like AN. This is obtained raising the level of abstraction of the interaction with the managed elements [9].

Second, under dynamic compositions, it is crucial to **automate the reaction** of the managed elements to network events. Policies are especially suited for this. They are guidelines for decision-making processes [7].

Third, the dynamic nature of AN makes costly maintaining a centralized list of the managed elements. PBNM permits us to avoid such a list by grouping the managed elements into **roles**. The use of roles permits operators to manage devices, even though the existence is not known in advance. A policy that applies to role X will apply to all elements playing that role, including those that will join the network after the policy was introduced.

In order to apply PBNM to AN, PBNM systems must be scalable [6]. A key step to achieve this goal is developing scalable policy distribution schemes. **Policy distribution** is the mechanism that provides the right policies at the right locations in the network, when they are needed. Current distribution schemes are centralized and therefore scale badly. This problem is worse for dynamic networks like AN.

The focus of this paper is to design a scalable –in terms of processing load and traffic- policy distribution schema for Ambient Networks. Our approach is engineering a scalable framework.

The work presented in this paper focuses on architectural aspects and its presentation is qualitative. Quantitative evaluations are scenario specific and are underway.

The paper is organized as follows. Section 2 presents an Ambient Networks study case we use to show how to apply our approach. Section 3 introduces the existing PBNM framework and motivates our work. Section 4 discusses the types of policies our framework distributes. Section 5 presents our architectural framework. Section 6 concludes the paper and presents future work

II. AN AMBIENT NETWORKS STUDY CASE

In this section, we present a scenario (figure 1) that we will use to show how to apply our framework. The scenario considers a number of network domains:

- a WLAN (domain W) located at a train station. It consist of access points and switches;
- moving trains (domains T1 and T2);
- an UMTS network (domain U);
- a number of mobile terminals carried by users.

These networks compose dynamically to gain end-to-end connectivity. Compositions happen at high rates. Some examples are:

- Trains arrive and leave the station: T1 and W compose and decompose.
- Passengers get into and off the trains: users join and leave domains T1 and T2.

This work has been supported in part by VINNOVA under the project Policy-Based Network Management and in part by the European Commission under the Ambient Networks Project.

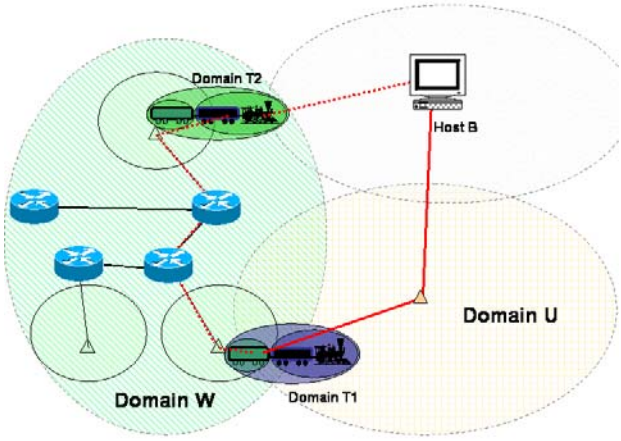

Figure 1: Ambient Network Study Case. Selecting a Path among Different Domains

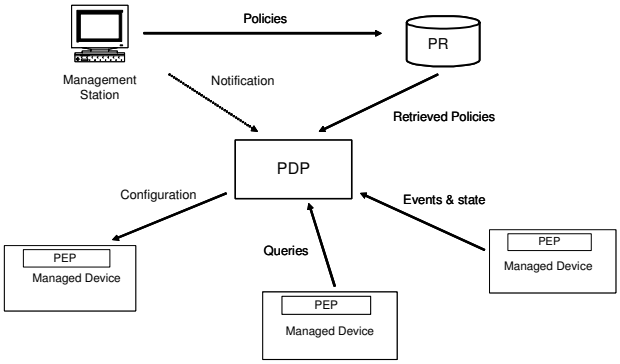

Figure 2: PBNM Architecture (de facto standard)

The policies we consider are **traffic management policies**. They indicate what the constraints for injecting traffic into the domain are: what traffic is permitted and what is not. For instance, assume that WLAN W is composed with domain T2 and that T2 does not allow peer-to-peer traffic (its gateway drops such traffic). Therefore, the WLAN has the policy "do not send P2P traffic along paths that traverse domain T2" (P-W1).

Consider a train passenger with a terminal having a WLAN and an UMTS interface. The passenger is running a P2P application that keeps a connection to host B. As the train travels between stations, he uses a low bandwidth access connection offered by the UMTS network. When the train arrives at the station, T1 and W compose. This gives the passenger an alternative access connection, which is faster. The management system has to decide whether the passenger switches to W. The decision is guided by the traffic management policies of W. Evaluating policy P-W1 involves evaluating whether the connection to B, through W, would cross T2. As it does, the connection is not established. Therefore, the passenger must keep using the UMTS access.

Note that in this scenario, P-W1 needs to be re-evaluated after network state changes. For instance, consider that a new domain (called T3) composes with W after T1 has composed with W. This changes W's topology, permitting the passenger to reach host B without crossing T2. A new evaluation of P-W1 leads to a reconfiguration of the passenger's terminal so that the connection to B is established through the WLAN.

The challenge for a policy-based management system in this dynamic scenario is to take the actions required to enforce the policies in a timely manner. In the next section, we argue why existing approaches do not meet this requisite.

III. POLICY-BASED MANAGEMENT

A. *De Facto Standard Architecture*

Virtually all proposed PBNM architectures to date follow the IETF framework [1] or can easily be mapped to it. This functional model is commonly mapped to the same physical model (figure 2). It consists of: (i) a management station running on a dedicated machine; (ii) a centralized policy repository (PR), which runs on a dedicated machine. Usually, it is based on LDAP; (iii) a single policy decision point (PDP) running on a dedicated machine; (iv) one policy enforcement point (PEP) running on each managed device. In the scenario above, the managed devices are the user terminals.

The policy data flow is divided in three stages. First, policies are introduced into the system using a management station and are stored in the PR. Second, the PDP is notified of the existence of new policies in the PR and it retrieves them. Third, the PDP evaluates the policies and sends the appropriate management information to each PEP. This evaluation can be triggered by a network element, the management station or a customer service request. The communication between the PDP and the PEPs can be based on CLI commands, COPS, SNMP, etc.

The IETF framework proposes two working modes: **outsourcing** and **provisioning**. In the first, the PDP performs all the policy evaluation and reports the PEP the actions to enforce. In the second mode, the PDP provisions policies to the managed devices. This includes (i) determining the policies that apply to each managed device and (ii) configuring each device with these policies. Then, the managed nodes evaluate the provisioned policies themselves to decide the actions to take.

B. *Limitations of Existing Approaches*

The presented physical architecture is not scalable due to its centralized nature. For instance, the processing load on the PDP increases linearly with the network size. For the provisioning mode, this load is lower since part of the policy evaluation is delegated to the PEPs. We only consider the provisioning mode in the rest of the paper.

[2] studies the performance of a PDP in provisioning mode. The experiments are based on COPS-PR, the IETF proposal for this mode. The obtained performance is below three requests per second. This clearly falls short for

dynamic scenarios. We illustrate this with three examples based on the scenarios presented in section II.

First, consider a train arriving at the station. At that moment, hundreds of terminals compose with W. All these terminals contact the centralized PDP to retrieve the appropriate policies. This poses an excessive load on the PDP and would result in long delays in the policy distribution.

A second example is the introduction of policy P-W1 when W and T2 compose. In existing distribution schemes, this would require the PDP to contact all the managed devices, posing a high load on the PDP. Moreover, it requires keeping a list of the managed devices, which is costly in dynamic scenarios.

A third example is a network state change, such as T3 composing with W. It may lead to the re-evaluation of the governing policies and the distribution of new provisioned ones. In existing frameworks, this task is identical to the distribution of a new policy. It requires the PDP to contact all the managed devices.

Next, we present another scenario to illustrate the lack of scalability of existing approaches. Consider that the railway company runs a domain that covers the entire railway network and supports Differentiated Services. Existing approaches address such scenarios by setting in each edge router a PIB entry for each flow that enters the network through that router. So, the policies sent to edge routers are of the type: "mark as EF those packets which IP Source= 130.237.15.25 and IP Destination = 130.237.212.136". When a train moves, the PDP must detect for each user that the incoming router has changed, provision the appropriate PIB entry to the new incoming router and remove it from the former one. Considering the reported performance of PDPs in provisioning mode, current approaches cannot cope with such a dynamic scenario.

One way to address the scalability problem is to deploy a number of dedicated PDPs. This option is not a valid approach for networks with dynamic topologies and loads, such as AN. The reason is that the optimal number and location of PDPs should be re-calculated after network state changes and this is a NP-complete problem for large networks [3].

IV. Role-level Policies for Ambient Networks

In this section, we present the types of policies that our framework distributes

A. Role-level Policies

The lack of scalability of existing frameworks comes from the fact that every single managed device is contacted individually in order to send it its individual management information. For instance, in the above scenario, the centralized PDP would provision the policies that apply to the joining terminal. First, the PDP determines whether the P2P connection can be established through W. In this case, it determines it cannot since it would cross T2. Then, the PDP sends the terminal the provisioned policy: “flows from this terminal to host B must not be routed through access point X (in domain W)”.

Our approach is based on avoiding contacting single entities. We aggregate the managed devices by the role they play in the system (e.g.: bronze customers). The policies we distribute are addressed to a whole group, not just one device. We call such policies role-level policies. P-W1 is a role-level policy for role “user terminal”. This permits us (i) to dispose of the provisioning PDP, the bottleneck of existing approaches and (ii) to create a scalable distribution scheme based on multipoint communication techniques. We discuss this in section V.

B. Implications of Distributing Role-level Policies

Once the role-level policies are distributed, the managed devices auto-manage themselves after them. This involves interpreting high-level policies. In our study case, this means that the user terminal must (i) determine that a connection through W would cross T2, and (ii) configure the terminal so that the connection is not established through W.

Role-level polices are evaluated by a local PDP (L-PDP) running on the managed device (figure 3). The L-PDP must (i) **evaluate** the policies to find the actions it has to enforce**,** and (ii) **refine** (map) the role-level policies into enforceable management actions**.**

The network operator is responsible for providing the managed devices the appropriate logic to perform these two functions. To do so, it distributes the L-PDP software to the devices. We address this similarly to policy distribution. In this case, however, we aggregate the managed devices according to their role and device type. The reason is that mapping and evaluation are device-specific tasks. For instance, we can aggregate terminals into “bronze users using Linux OS”, “gold users using Palm OS”.

Note that both evaluation and mapping are common tasks in PBNM architectures. Our framework can benefit from all the work done in these areas, which is complementary to ours.

V. An Architectural Framework for Policy Distribution

In this section, we address the distribution of role-level policies. Figure 3 shows our functional architecture for managed elements.

Note that we comply with the IEFT functional model. Our approach does not modify the functional model, but the physical model commonly used.

In our framework, there are two situations when policies are distributed. The first is when the operator introduces a new policy that must be distributed to the managed devices. The second situation is when a node joining a network is sent the domain governing policies. Next, we discuss both.

A. Distributing a New Policy

The introduction of a new policy can be triggered by a (de)composition. The composition of W with T2 leads to the introduction of policy P-W1.

Aggregating the managed devices into roles permits us addressing the distribution of role-level policies as a multipoint communication problem. The traditional

approach to scalable multipoint communication is multicast.

We keep one multicast group for each role in the network (e.g.: gold users, silver users). The members of the group are those nodes playing the corresponding role, the management station and the policy repository. The multicast trees are composed and maintained by the group members and the nodes that connect them. These tasks are performed by the "Mcast Tree Creator" block (Figure 3).

There is no universal approach for multicast [8]. The exact approach to use has to be tailored considering the characteristics and requirements of the application.

The characteristics of the distribution are:

- The network administrator is the only entity allowed to introduce new policies. Therefore, the management station is the only sender in the multicast group.
- Policy distribution tolerates out-of-sequence delivery.
- The number of multicast groups used to manage a network is small. It is limited to the number of considered roles, which we assume to be bounded to 10-20.
- The density of the group members is generally high. Consider the passengers in a train.

The requirements of role-level policies distribution in large networks are:

- processing load, state and traffic **must scale** with the network size;
- The distribution must be **reliable** in the sense that all devices in a network must have a replica of the governing policies that apply to the role it plays. This includes nodes that join the network after a policy has been introduced in the system.

1) Scalability

Scalability is the main requirement for the distribution schema. Therefore, we require a multicast algorithm that does not demand global knowledge of the network topology and group membership. Reverse Path Multicast (RPM) provides this.

RPM is a distributed algorithm based on Reverse Path Forwarding [10], which computes a spanning tree. RPM requires each node to know how to route data to the source. This information is provided by unicast protocols. Therefore, RPM scales with the unicast routing algorithm deployed in the network. RPM keeps state information per source and group. This cost is affordable for our policy distribution framework: (i) we have only one source, and (ii) the number of groups (roles) is expected to be small.

In our architecture, each managed device runs RPM. This is the functionality of the "Mcast tree creator" block (see Figure 3).

New policies are distributed along the created tree by the "Mcast Forwarder" block of the Policy Distributor (figure 4).

We have chosen to create the distribution multicast tree on demand, since the cost of maintaining a tree is very expensive in dynamic networks.

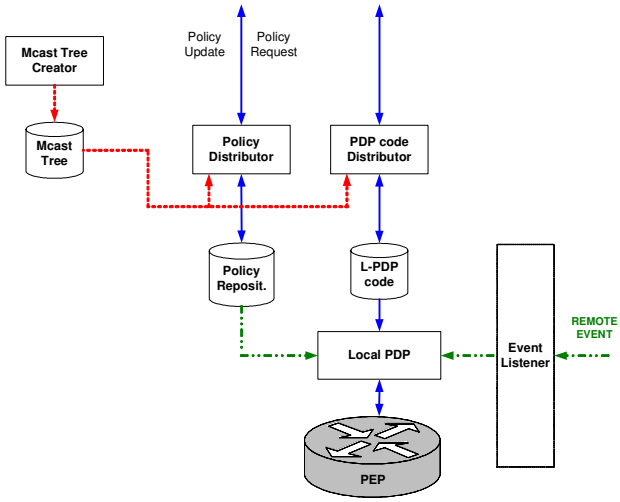

Figure 3: Policy Distribution Framework. Functional Architecture for AN Managed Nodes

2) Reliability

There are two threats to reliability. The first is temporary packet losses. The second is the existence of nodes that join the network after a policy has been introduced in the system. We will analyze the second threat in the next section.

The detection of packet losses and request for retransmission is responsibility of the receiver (the user terminal) in our framework. Losses are detected by finding missing packets in a sequence. This function is performed by the block labeled "Loss Control" in figure 4.

The algorithm deployed for requesting retransmissions must avoid the implosion of requests. Such an implosion happens when a large number of nodes detect a loss and all of them request for retransmission at the same time. Techniques like slotting and damping address this issue. SRM [4] makes use of these techniques, presenting a good performance in dense scenarios, like our study case. The "Loss Control" block implements SRM. In SRM, any node that has the appropriate policy can answer a retransmission request. In our case, that includes those nodes that –playing the same role as the requester- have received that policy successfully.

SRM presents a trade-off between duplicated messages and recovery time. That trade-off is controlled by the timeout values that control the algorithm. The delay sensitivity of policy distribution is scenario specific. Said this, delays above a few seconds might have consequences noticeable by end users, which is undesirable. For instance, the passenger may loose his P2P connection.

B. Distributing Policies after Domain Composition

When a node joins a network, we must check the validity of the policies it holds –if any. If its policies are not updated, we must distribute the governing ones to it. In our study case, this happens when a train reaches the station.

In our framework, the joining node is the responsible for requesting the appropriate policies. This is the function of the "policy requester" block. In order to do so, it must know

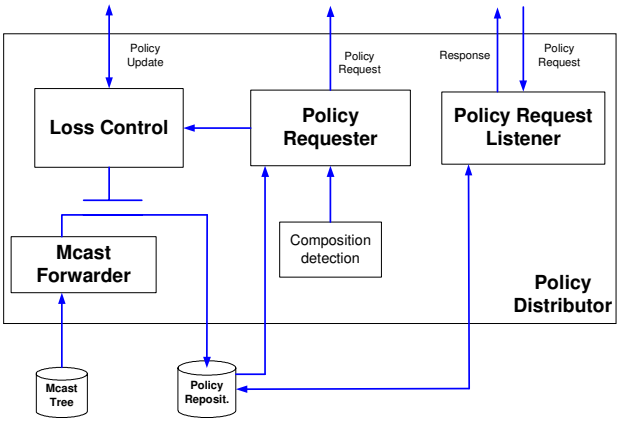

Figure 4: "Policy Distributor" Functional Block of an AN Managed Node (detail)

its role in the network. It must also detect it has joined the network. This is the responsibility of the functional blocks in the AN architecture that support composition, which is out of the scope of this paper.

We consider two algorithms to implement the "policy requester". The first consists in sending a request towards the PR. If a node in the path to the PR is able to answer the request, it does not forward the request to the PR and answers itself.

The second mechanism is expanded ring search (ERS).

The one to choose is scenario-specific. For instance, ERS is more appropriate in our study case, where a new passenger can obtain the policies from other passengers.

Next, we discuss the concept of **extended membership**. As we have said, a high membership density helps in providing reliability. Therefore, we consider increasing the density of the multicast groups. We do so by extending the membership of a group to nodes that play a role different to the one associated to the group. We call this extended membership. The idea is that a node may store policies that it will not use for its own management, but may distribute to other nodes.

Deciding how to extend group membership involves a trade-off between the costs of distributing policies to nodes that will not use them for their own management and the cost of distributing it to those that will. The cost of distributing policies includes the traffic generated and the memory required to store them. Using extended membership requires a larger storage capacity in the nodes. The storage needed depends on each specific scenario. However, we expect it to be in the order of a few kilobytes. To the best of our knowledge, there are no studies on storage requirements for deployed systems.

Currently, we extend group membership role-wise. That is, we extend role Y membership to all the nodes playing role X. An interesting approach is extending replicas applying cache techniques.

For the scenario presented in section II, extended membership could be applied distributing the policies for the "user terminal" role to members of the "access point" role as well.

The quantitative gain –in terms of traffic and processing- obtained by extending membership is scenario-specific. It depends on the relative location of the nodes playing each role and the dynamics of the network.

VI. Discussion and Future Work

Policy-based management has a strong potential for the management of Ambient Networks. However, a number of challenges must be addressed before we can apply it efficiently. In this paper, we have focused on one of them, namely, policy distribution. First, we have shown that current approaches cannot cope with dynamic scenarios like AN. Then, we have presented a scalable framework for policy distribution for Ambient Networks. Our proposal scales with the unicast routing algorithm deployed in the network. The validity of the approach has been shown applying it to a realistic scenario.

We are currently working on the implementation of the framework in order to perform a formal analysis of the scalability aspects and provide a proof-of-concept for the framework.

One aspect that needs to be addressed is providing security to the policy distribution process. The need stems from the importance of assuring the integrity of the policies.

The work presented in this paper is consistent with the IETF framework and orthogonal to research in other areas of policy-based management, such as specification, analysis, conflict resolution, etc.

References

[1] A. Westerinen et al., "RFC 3198: Terminology for Policy-Based Management", Nov. 2001

[2] A. Corrente et al., "Policy Provisioning Performance Evaluation using COPS-PR in a Policy Based Network", 8th IFIP/IEEE International Symposium on Integrated Network Management (IM 2003), Colorado Springs, USA, 24-28 March, 2003

[3] A. Liotta, G. Pavlou, G. Knight, "Exploiting Agent Mobility for Large Scale Network Monitoring", IEEE Network, special issue on Applicability of Mobile Agents to Telecommunications, Vol. 16, No. 3, May/June 2002

[4] S. Floyd et al., "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", IEEE\ACM Transactions on Networking, Vol. 5, No. 6, 1997

[5] Niebert, N. et al. "Ambient Networks – Research for Communication Networks Beyond 3G"- 13th IST Mobile and Wireless Communications- Summit 2004, 27-30 June 2004, Lyon, France

[6] M. Brunner et al. "Ambient Networks Management Challenges and Approaches", First International Workshop on Mobility Aware Technologies and Applications (MATA 2004), Florianpolis, Brazil, October 20-22, 2004.

[7] J.D. Moffett and M.S. Sloman, "Policy Hierarchies for Distributed Systems Management", IEEE Journal on Selected Areas in Communications (J-SAC), Vol. 11, No. 3, December 1993

[8] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint Communications: A Survey of Protocols, Functions, and Mechanisms," IEEE JSAC, Apr 1997, pp. 277–90.

[9] M.J. Maullo and S.B. Calo, "Policy Management: an Architecture and Approach", First IEEE International Workshop on Systems Management, Los Angeles, USA, Apr 1993

[10] Y. K. Dalal, R. M. Metcalfe, "Reverse path forwarding of broadcast packets", Communications of the ACM, Vol. 21, Issue 12, December 1978