

Consistency of States of Management Data in P2P-based Autonomic Network Management

Jéferson Campos Nobre, Lisandro Zambenedetti Granville

Institute of Informatics – Federal University of Rio Grande do Sul
Av. Bento Gonçalves, 9500 - Porto Alegre, RS - Brazil
{jcnobre, granville}@inf.ufrgs.br

Abstract. Autonomic network management is a vision that brings autonomic computing principles to network management. Besides, it is necessary some level of decentralization to enable broad autonomic capabilities. An interesting alternative of infrastructure for this union is the utilization of peer-to-peer (P2P) overlays. However, the consistency of states of management data among peers is an important challenge. Traditional mechanisms to maintain consistency of these states are supported by some centralization which wastes some desirable properties of P2P facilities. In contrast to these mechanisms, we propose a distributed, scalable and robust mechanism to maintain the consistency of states of management data. In this paper we introduce multi-agent truth maintenance features in P2P-based autonomic network management as a mechanism to bring consistency maintenance of these states. We developed a model of a P2P-based autonomic network management system on Peersim to perform simulation experiments. Besides, the utilization of P2P-based autonomic network management systems in access networks is presented as a case study.

1 Introduction

The increasing complexity of computer networks requires sophisticated solutions to manage the underlying communication infrastructure and help network human administrators in their daily tasks [1]. The application of autonomic computing (AC) principles in network management, normally refereed as *Autonomic Network Management* (ANM), has been proposed as a way to address some demands faced by traditional network management, such as controlling highly dynamic environments like ad-hoc and peer-to-peer (P2P) networks [2]. ANM systems increase the efficiency of network human administrators by decreasing the number of manual interventions. This efficiency improvement is done by ANM systems through automation and/or optimization of some operational details of management tasks, such as fault handling and performance management.

ANM systems can be deployed using different models of management distribution, from totally centralized models up to highly decentralized ones. Today, there are no clear evidences that could link the quality of autonomic actions of an ANM system with the distribution model adopted. Some authors, however, claim that some level of decentralization is required to achieve a more adequate ANM [3]. In this scenario, the

typical design of decentralized ANM systems is based on a set of *Autonomic Management Elements* (AME) that execute management tasks and interact with one another to form an *Autonomic Management Domain* (AMD). Multiples AMDs could be integrated to form an *Autonomic Management Environment*.

Different technologies can be employed to provide an infrastructure for decentralized ANM systems. An interesting possibility is using peer-to-peer (P2P) overlays, which incorporate characteristics of P2P networks into ANM systems, such as the support for collaborative management, robustness in connecting management entities, and load balancing of management tasks [4]. There are some initiatives investigating P2P-based ANM [5] [6] and, in these initiatives, peers have some properties found in AMEs. Besides, peer with similar properties (e.g., managing the same devices) can be organized into groups. These peer groups have some properties found in AMDs.

A managed element (e.g., a network router) can be controlled by one or multiple peers in P2P-based ANM system, for example, for the purpose of management robustness. In this case, the status of the managed element becomes a distributed, replicated information among the controlling peers. Inconsistencies in this information could lead to an improper operation of the ANM system. For example, the status of a router's link (e.g., ethernet interface) can be advertised differently by the controlling peers to external requesters (e.g., management station). This inconsistency can be caused by faults in network (e.g., losses in messages exchanged among peers) and computational resources (e.g., incorrect operation of peers). Besides, inconsistency in management data can occur even in the regular operation of P2P-based ANM system, due to lack of information synchronization among peers. This information synchronization is a challenge in unreliable asynchronous distributed systems, such as an unstructured P2P overlay.

Mechanisms to support consistency of management information in current P2P-based ANM systems are, in general, still supported by some centralization through, for instance, external repositories [7] or *super peers* [5] [6], missing opportunities of unstructured P2P facilities. This centralization complicates the achievement of good scalability and robustness features, thus, in spite of being P2P-based ANM systems, these examples show undesirable characteristics of client-server systems.

In this paper, we introduce *Multi-Agent Truth Maintenance* features [8] in the P2P-based ANM to improve the consistency of states of management data. The introduction of these features maintains desirable scalability and robustness characteristics of P2P-based ANM. It is also proposed a communication strategy for message exchange among peers to support this truth maintenance. This strategy uses biology-inspired processes (e.g., replication), which have well know scalability and robustness features [9].

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 describes our proposal and its implicit concepts. Section 4 shows a case study. Section 5 shows evaluations of our proposal. Conclusions and future work are finally provided in Section 6.

2 Background

In this section we first discuss about the current state-of-the-art on P2P-based autonomic network management. Afterwards, the main concepts behind truth maintenance

systems are presented. Finally, some important services related to the maintenance of consistency of shared information in distributed applications are discussed.

2.1 P2P-Based Autonomic Network Management

The utilization of P2P overlays is identified as an emerging approach for Autonomic Network Management (ANM). This can be specially observed in the increasing number of research work towards this utilization [10].

PBMAN [5] merge traditional PBNM with P2P overlays to autonomically manage Ambient Networks (AN). PBMAN enables scalable mechanisms for network composition inside the AN, as well as policies distribution and retrieval. Through this approach it is possible to establish policies to manage devices or services. PBMAN is structured using super peers, in a hierarchical architecture. These super peers are responsible for consistency of states of management data, among other functions, in each hierarchical level.

The Madeira platform [6] is an approach to ANM that uses the concept of Adaptive Management Components (AMC), which are containers that run on managed elements. AMCs can communicate with other AMCs running on other managed elements through P2P communication services. AMCs form management clusters with super peers acting as cluster heads. These super peers are responsible for consistency of states of management data, among other functions.

ManP2P [11] is a P2P-based network management system that is evolving to an autonomic conception through the implementation of autonomic modules in peers [7]. ManP2P is partially inspired by the Management by Delegation (MbD) model and based on a service-oriented approach. There is no internal mechanism for consistency of state of management data, thus, authors propose the utilization of external repositories.

Despite many improvements brought by the utilization of P2P-based ANM systems, there are still issues to be addressed. The consistency of state of management data among the peers is usually addressed with some centralization, using super peers [5] [6] or external repositories [7], which misses opportunities of P2P overlays (e.g. robustness).

2.2 Truth Maintenance Systems

Truth-Maintenance Systems (TMS) were proposed to keep the integrity of Knowledge Bases (KB). The origin of these systems was proposed in the 1970s, for resolutions in mono-agent systems [12]. A TMS keeps track of logical structure of the set of beliefs of agents. A belief is a member of the current set of beliefs if it has valid reasons.

TMSs provide considerable power using few computational resources [13]. Thus, although not being well known outside artificial intelligence community, TMSs are used in different contexts, such as policy systems [13] and network management systems [14].

TMSs have been extended for Multi-Agent System (MAS) versions, *Distributed Truth-Maintenance Systems* (DTMS) [8]. In a MAS, agents must be able to maintain the integrity of their KBs, despite message exchange with other agents. This maintenance of

integrity can be done by a multi-agent TMS. In an analogous way, during the operation of P2P-based ANM system, peers must be able to maintain the integrity of states of management data, despite message exchange with other peers. This similarity indicates the use of multi-agent TMS in P2P-based ANM as an interesting possibility [14].

2.3 Services for consistency of shared information in distributed systems

Services for consistency of shared information can be used as a basic building block for distributed applications. A P2P-based ANM system, as a distributed application, can appropriate some good characteristics found in these systems.

ZooKeeper [15] is a coordination service for distributed applications. It exposes a simple API that distributed applications can be built upon to implement higher level services for synchronization, data diffusion, and publish-subscribe schemes. *ZooKeeper* uses distributed server databases for read operations, however, write operations use a “leader” server (i.e., centralized database) to assure the consistency of the database.

Astrolabe [16] is a distributed information management service. It works locating and collecting the status of a set of servers and reporting summaries of this information. *Astrolabe* is implemented using a P2P overlay, where every peer run an *Astrolabe* agent (i.e., in a MAS fashion). However, *Astrolabe* was developed primarily using simple data models. Besides, its operation is aimed at read-oriented applications.

Scalable Distributed Information Management System (SDIMS) [17] is a service to aggregate information about large-scale network systems. The service is built using ideas from *Astrolabe* [16] and Distributed Hash Tables (DHT). However, as in most DHT approaches, consistency and replication issues are a known challenge.

The presented efforts show interesting characteristics for consistency of shared information in distributed systems. However, these efforts have vulnerabilities which make them not appropriate for P2P-based ANM systems, such as centralization [15], simple data models [16], and replication issues [17].

The consistency of states of management data is still an issue to be addressed in P2P-based ANM systems. It is necessary to maintain this consistency keeping scalability and robustness features of P2P overlays. The maintenance of integrity of KBs among agents brought by multi-agent TMS seems to be a valid direction to introduce this consistency maintenance [14]. Besides, it is interesting to bring characteristics of services for consistency of shared information.

3 Proposal

In P2P-based Autonomic Network Management (ANM) systems, peers must share management data. In this work, *management datum* is defined as a management information described in a defined form (i.e., using a specific language). Besides, in these systems, management data must allow their use in automation and/or optimization procedures. It is also expected that sources of management data (e.g., highly dynamic environments) impose challenges to the ANM system. Despite these requirements, it is necessary to avoid potential inconsistencies in state of management data among peers.

Our proposal is aimed at meeting requirements of consistency of states of management data in a P2P-based ANM system. The proposed mechanism introduces *multi-agent truth maintenance* [8] features through a *consistency maintenance module* that runs in each peer. As far as we are aware of, the only study that incorporates multi-agent truth maintenance features in a network management system was carried out by *Nobre and Granville* [14], and only some results were published so far.

Multi-agent truth maintenance is a multi-agent extension to Truth-Maintenance Systems (TMS). The goal of TMS utilization is to keep the integrity of knowledge bases. In this work, a *knowledge base* is a group of integrated management data. In a multi-agent TMS, there are multiple agents and each one has its own TMS. TMSs keep integrity performing belief revision and exchange in a set of beliefs. A belief is a member of the current set of beliefs if it has valid reasons.

The exchange of beliefs about management data is done asynchronously and we do not consider the message exchange to be reliable. Unfortunately, it is well known that the utilization of asynchronous unreliable distributed systems imposes challenges to achieve consistency in shared data. Thus, the consistency model used is non-deterministic, in other words, it uses a “weak” notion of consistency. This model is adopted for scalability, robustness, and update dissemination issues. Given a belief X that depends on some other belief Y, when an update is made to Y, it is eventually reflected in X. Some authors call this notion as “eventual consistency” [16]

3.1 Justifications for Management Data

Multi-agent TMS is a kind of *justification-based* TMS. In a justification-based TMS, a datum is believed when it has valid justifications (i.e., valid reasons). This belief can be shared by different peers (which play the agent role) in a P2P-based ANM system (which aggregates some MAS characteristics). Thus, justifications improve the alignment of peers with system-wide objectives (i.e., objectives of the P2P-based ANM system). The datum and its list of possible justifications must be provided by network human operators or expert systems for the P2P-based ANM system.

The associated states of a datum are “in” (believed) or “out” (disbelieved), according to its justifications. These justifications can be generated by processes inside the peer or received through the P2P overlay communication services. Thus, the “in” state can assume two additional states: “internal”, where the datum has only valid internal justifications, and “external”, where the datum has some valid external justification (provided by other peer). A datum is labeled “out” when it lacks, at least, one of its associated justifications [8].

For instance, the activation (belief) of a QoS policy (datum) can be justified by a network human administrator command (justification) and an asynchronous signal from a managed device (justification). The code below shows a possible internal representation of this datum and its justifications. In the example, each datum or justification (“just” in the code) is represented with 2 fields, “name” and “desc” (description). The last line of the code defines that if the justifications “adm_cmd” and “async_sig” are present, the datum “qos_pol” is believed.

```
datum: name qos_pol
```

```

desc QoS policy
just:  name adm_cmd
       desc network human administrator command
just:  name async_sig
       desc asynchronous signal from a managed device
tms:   qos_pol (adm_cmd async_sig)

```

3.2 Architecture of Peers

Peers are commonly viewed as composed by one or more *management service modules*. *Management service modules* perform regular management tasks (e.g., collecting statistics) in each peer, and, in these tasks, modules produce management data, building *management knowledge bases*.

We introduce the *consistency maintenance module* to registrate the set of belief about management data in each peer. This module works associating management data and their respective justifications. When there is a belief change (i.e., justification change), the consistency maintenance module uses the P2P overlay communication services to spread the change. Figure 1 shows the relation between the consistency maintenance module, management service modules, and P2P overlay communication services.

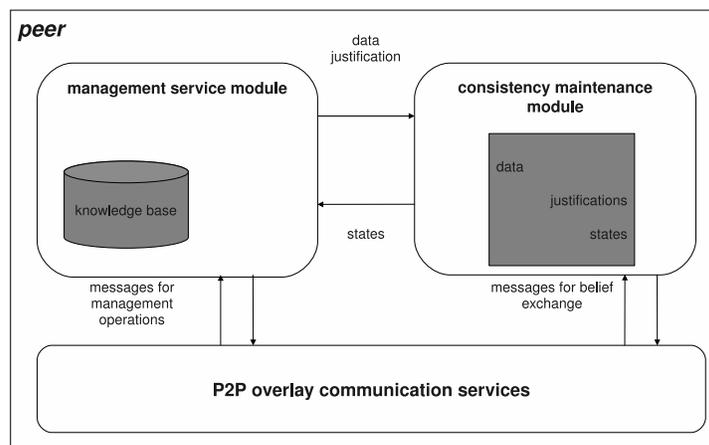


Fig. 1. Peer architecture

The management service modules should inform consistency maintenance module about their internal beliefs of management data. The management service modules are also responsible for querying and requiring services from the consistency maintenance module, possibly updating their internal beliefs. These beliefs are exchanged through

justification using a simple interface that management service modules and consistency maintenance module must use.

When the presence of a justification is modified, the consistency maintenance module performs the following steps: unlabels management datum, includes (or removes) the presence of justification and labels datum again according to new restrictions. The P2P overlay communication services are used to spread changes, which can change beliefs of other peers.

It is important to stress that there is only one consistency maintenance module inside a peer, thus, it is not specific of a management service module. Therefore, every management service module in a peer interacts with the same consistency maintenance module. This fact can be explored for the integration of different management services. For instance, a policy processing module, a fault handling module, and a configuration management module (possibly using different languages for representing management data) could be integrated by the consistency maintenance module through justifications.

3.3 Communication within a peer group

The consistency maintenance module handles the message exchange through P2P overlay communication services. In this process, requests are adapted in messages to be spread among peers and vice-versa. The P2P overlay is modelled as an unstructured overlay networks, thus there is no relation between the information stored at a peer and its position in the overlay topology.

We use the premise that there is support for group organization (i.e., peer groups) through management services modules. Thus, peers that have a specific management service module are organized into a group (without human intervention) and peers can participate of several groups accordingly to modules that they have.

The methods used for message exchange inside the peer group are modeled using concepts from biology-inspired distributed computing models [9]. Among these models, proliferation-based ones are an interesting choice for communication requirements of our proposal. All peers in the peer group run exactly the same communication algorithm, which can be initiated from any peer in the peer group.

We have chosen *replication* as the initial proliferation mechanism in the peer group. This mechanism can support a number of different strategies [9]. In our proposal, peers spread messages to replicate changes in justifications among the participating entities (i.e., peers of a specific peer group). This unbridled replication is restricted to peer group, fulfilling the criterion of robustness and controlling the number of messages within the P2P overlay.

4 Case Study

The case study presented is an illustration of the collaborative fault management of links in access networks through failure notification sent by devices and human knowledge about these notifications. The integration of these information (failure notification in addition to human knowledge) produces a management datum, which can assume

different states. This datum can be used against a Service-Level Agreements (SLAs) to support or clarify service level claims.

Among access network technologies in metropolitan networks, Ethernet is one of most interesting and promising choice, thus, we choose it to build our case study. In this context, an access network link is an *Ethernet Virtual Connection* (EVC) [18]. Fault management in this link is done through *Alarm Indication Signal* (AIS) messages [18]. These messages are triggered when a failure between two nodes occurs. Thus, AIS messages provide asynchronous notification to other elements in the network that there is a fault in the Ethernet network. The efforts to manage layer 2 Ethernet service must consider an overlaid IP infrastructure [19].

A management service module collects AIS messages and another module collects information from human administrators. The consistency maintenance module is responsible to integrate the information from both management service modules and maintain the consistency of the state of management datum in the peer group.

The *link failure detected* (datum) is justified by human administrator commands from both domains, *network operator detection* and *network consumer detection*, and a *device notification received* (AIS message). The code below shows the representation of this datum and its justifications. These justifications are provided by management service modules and kept inside the peer group that offers this management service.

```
datum:  name link_fail_det
        desc link failure detection
just:   name net_opt_det
        desc network operator detection
just:   name net_con_det
        desc network consumer detection
just:   name ais_msg
        desc ais message
tms:    link_fail_det (net_opt_det net_con_det ais_msg)
```

Initially, the datum has the “out” state, since the justifications are not present. This state can be changed after the verification of new beliefs (i.e., justification changes) by consistency maintenance module. These beliefs are informed by management service modules or received as a message in the peer group. If every justification of the datum is present, the state changes to “in”. This “in” state assumes two options according the sources of justifications: “internal” (every justification was generated internally) and “external” (some valid external justification).

For instance, human administrator commands (network operator detection and network consumer detection) can be generated internally and the presence of an AIS message can be received as a justification change message. In this example, the “in” state assumes the option “external”. The code below shows the answer from the consistency maintenance module in this situation.

```
tms:    link_fail_det:external (net_opt_det:mod net_con_det:mod ais_msg:msg)
```

This management operation is traditionally performed through separate centralized systems, a network management system (collecting notifications from devices), and a trouble ticket system (collecting information from human administrators). The traditional procedure brings concerns in scalability and robustness and imposes difficulties

in the integration of the information. Besides, justifications can be used to provide explanations to the user [13], improving the understanding of management data.

5 Evaluation

The evaluation of our proposal can be performed in different ways. To enable a fully controlled environment for the evaluation, we chose to develop some simulation experiments. In these experiments, we present simulation results that support our scalability and robustness claims. Scalability and robustness are some of the most important motivations for using decentralization in the infrastructure of different systems [20] [9], such as P2P facilities.

As previously stated, we expect that the introduction of multi-agent TMS features keeps decentralization properties of a P2P-based ANM system, maintaining each peer as an independent and self-sustainable entity. As many systems have demonstrated, a system that does not share resources can scale almost infinitely simply by adding constitutive elements (e.g. peers in a P2P-based ANM system). Besides, maintaining the independence of each peer, single points of failure are eliminated.

The simulation experiments were implemented in Java using *PeerSim* [21], an open source event-based simulator of P2P systems. The preliminary system version used has the ability to simulate failures in peers and message exchange, and the overlay is built randomly. The experiments use a simple model of transport layer that can emulate some characteristics, such as loss and delay probabilities. All peers in peer groups run exactly the same algorithm.

In the experiments, we varied the number of peers of the peer group from 4 to 14 (we do not expect large peer groups in P2P-based ANM systems). Besides, these peer group sizes seem reasonable for the case study provided in Section 4. In addition, a peer is chosen randomly as the primary source of changes to not affect measurements and message delay is controlled. Each experiment was conducted at least 10 times. In the experiments, the variance observed was low.

In the first experiment, it is measured the number of messages exchanged to spread justification changes in the peer group. This number must be considered as an important cost of the peer group operation, thus, it is important for scalability analysis. Besides, we consider the number of transmitted messages as indicative of network load. In this experiment there were no faults in peers or in message exchange. We show the results in Figure 2.

Our proposal shows acceptable scalability characteristics on number of exchanged messages, since this operation (message exchange) is restricted to each peer group. The experiment shows that our system behaves like we expected, without stability and convergence problems. Network load grows linearly with the number of participating peers, thus we can infer the behavior trend of peer groups with larger number of participating peers. Of course, an efficient operation of large peer groups needs modifications in communication strategies, such as the utilization of gossip-based protocols.

In the second experiment, we determined the influence of message loss on the dissemination of a justification change. In this experiment, we varied the message loss probability with following values: 25%, 50%, and 75% (respectively, 0.25, 0.5, and 0.75

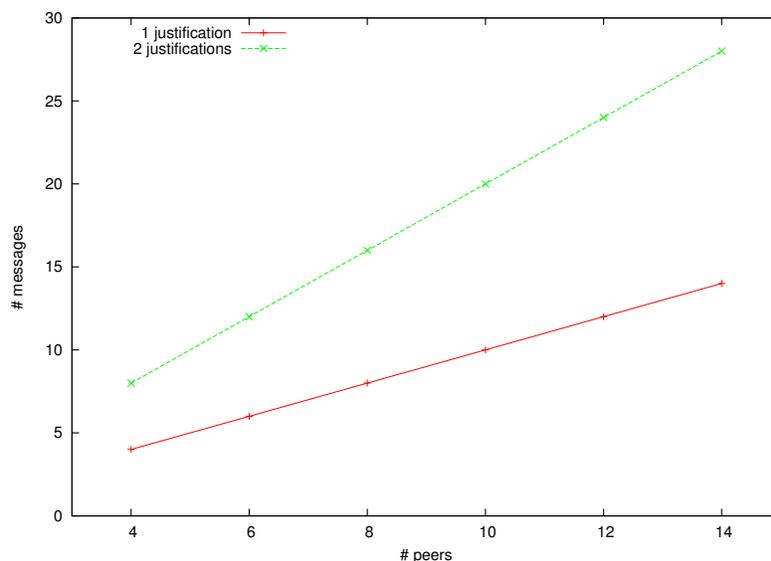


Fig. 2. Message exchange due to justification changes

as indicated in Figure 3). Using our case study, we would probably observe such message loss (specially 75%) due to faulty or overloaded network equipment (e.g., ethernet interfaces) and/or network links (e.g., ethernet lines). Since our case study is aimed at fault management (considering an overlaid IP infrastructure), our system must behave acceptably even in bad network conditions. In Figure 3, we show the average percentage of coherent (and correct) peers after message exchange to cease.

The experiment shows the influence of message loss in the replication process. As can be seen from the results in Figure 3, high loss probabilities do lead to less consistency in peer group, but, even with a few participating peers, the percentage of coherent peers is substantial. Besides, more participating peers in peer group decrease the influence of loss probability.

The results show some fault-tolerance features, since the peer group operation is not highly sensitive to peer crashes and message losses. But an increase in number of peers also leads to an increase in the number of exchanged messages, so the robustness advantages come at some cost. However, messages are exchanged only within the peer group, and a high number of peers in a peer group is not expected. Thus, the number of exchanged messages does not impose a issue in scalability.

6 Conclusions and Future Work

The support of new demands faced by traditional network management is a key research issue in network management area. One of these demands is the support for collaborative management tasks over access network links. In order to enable efficiently these tasks, it is necessary some decentralization and automation features. These features can

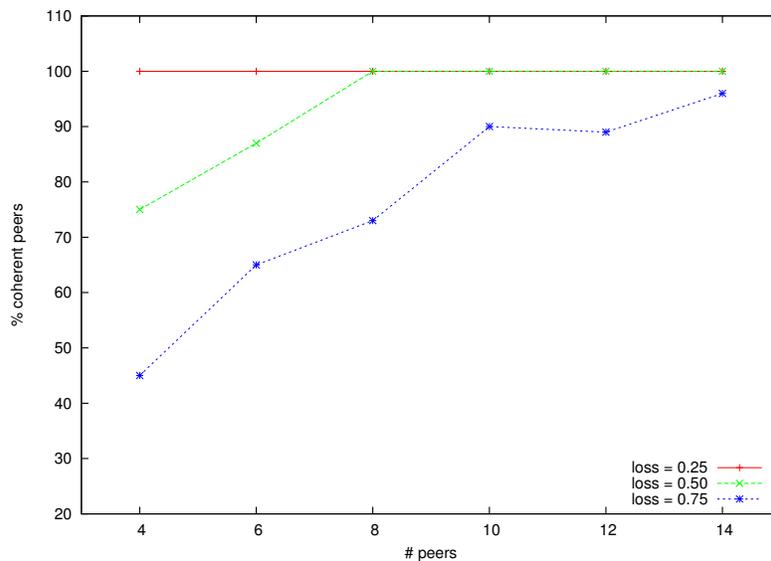


Fig. 3. Coherent peers after a justification change

be supported by P2P-based autonomic network management (ANM) systems. However, the consistency of states of management data imposes challenges for these systems.

In this paper we have introduced multi-agent TMS features to improve the maintenance of consistency of states of management data in P2P-based ANM systems. Our proposal aims at the integration of data used by the entities that form these systems (i.e., peers), through the utilization of justifications. We have also presented evaluations of this proposal through simulation experiments. In addition, we have described a case study of fault management in access networks to show the possibilities of our proposal.

Although the proposal shows good results in evaluations performed until the present moment, it is necessary to evaluate more complicated cases, in number of peers and peer groups, and in the participation of a peer in different peer groups. We are also looking at additional settings that could lead to important effects, such as network partitions. Thus, we are currently pursuing new experiments with *PeerSim*.

References

1. Kind, A., Dimitropoulos, X., Denazis, S., Claise, B.: Advanced network monitoring brings life to the awareness plane. *IEEE Communications Magazine* **46**(10) (2008) 140–146
2. Pras, A., Schoenwaelder, J., Burgess, M., Festor, O., Perez, G.M., Stadler, R., Stiller, B.: Key research challenges in network management. *IEEE communications magazine* **45**(10) (October 2007) 104–110
3. Jennings, B., Van Der Meer, S., Balasubramaniam, S., Botvich, D., Foghlu, M.O., Donnelly, W., Strassner, J.: Towards autonomic management of communications networks. *IEEE Communications Magazine* **45**(10) (2007) 112–121

4. Granville, L.Z., da Rosa, D.M., Panisson, A., Melchior, C., Almeida, M.J.B., Tarouco, L.M.R.: Managing computer networks using peer-to-peer technologies. *IEEE Communications Magazine* **43**(10) (2005) 62–68
5. Kamienski, C., Fidalgo, J., Sadok, D., Lima, J., Pereira, L., Ohlman, B.: PBMAN: A Policy-based Management Framework for Ambient Networks. In: *Policies for Distributed Systems and Networks, 2006. POLICY 2006. IEEE Workshop on.* (2006) 79–83
6. Fallon, L., Parker, D., Zach, M., Leitner, M., Collins, S.: Self-forming Network Management Topologies in the Madeira Management System. *Lecture Notes in Computer Science* **4543** (2007) 61
7. Marquezan, C.C., dos Santos, C.R.P., Nobre, J.C., Almeida, M.J.B., Tarouco, L.M.R., Granville, L.Z.: Self-managed services over a p2p-based network management overlay. In: *Proceedings. 2nd Latin American Autonomic Computing Symposium (LAACS 2007).* (2007)
8. Huhns, M.N., Bridgeland, D.M.: Multiagent truth maintenance. *Systems, Man and Cybernetics, IEEE Transactions on* **21**(6) (1991) 1437–1445
9. Babaoglu, O., Canright, G., Deutsch, A., Di Caro, G., Ducatelle, F., Gambardella, L., Ganguly, N., Jelasity, M., Montemanni, R., Montresor, A., et al.: Design patterns from biology for distributed computing. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* **1**(1) (2006) 26–66
10. EMANICS: Activities - Work Package 9 - Autonomic Management. <http://emanics.org/content/view/63/109/> (2009)
11. Panisson, A., da Rosa, D.M., Melchior, C., Granville, L.Z., Maria, Liane: Designing the Architecture of P2P-Based Network Management Systems. In: *ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications, IEEE Computer Society* (2006) 69–75
12. Doyle, J.: A truth maintenance system. *Computation & intelligence: collected readings* (1979) 529–554
13. Kagal, L., Hanson, C., Weitzner, D.: Using Dependency Tracking to Provide Explanations for Policy Management. In: *Policies for Distributed Systems and Networks, 2008. POLICY 2008. IEEE Workshop on.* (2008) 54–61
14. Nobre, J.C., Granville, L.Z.: Towards consistency of policy states in decentralized autonomic network management. In: *Policies for Distributed Systems and Networks, 2009. POLICY 2009. IEEE Workshop on* (to appear). (2009)
15. Hunt, P.: ZooKeeper: A Distributed Coordination Service for Distributed Applications. <http://wiki.apache.org/hadoop/ZooKeeper> (2008)
16. Van Renesse, R., Birman, K., Vogels, W.: Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Transactions on Computer Systems (TOCS)* **21**(2) (2003) 164–206
17. Yalagandula, P., Dahlin, M.: A scalable distributed information management system. *ACM SIGCOMM Computer Communication Review* **34**(4) (2004) 379–390
18. McFarland, M., Salam, S., Checker, R.: Ethernet oam: key enabler for carrier class metro ethernet services. *IEEE Communications Magazine* **43**(11) (2005) 152–157
19. Ryoo, J., Song, J., Park, J., Joo, B.: Oam and its performance monitoring mechanisms for carrier ethernet transport networks. *Communications Magazine, IEEE* **46**(3) (2008) 97–103
20. Mccann, J.A., Huebscher, M.C.: Evaluation Issues in Autonomic Computing. In: *Grid and Cooperative Computing – GCC 2004. Volume 3252 of Lecture Notes in Computer Science., Heidelberg, Springer-Berlin* (2004) 597–608
21. Jelasity, M. and Montresor, A. and Jesi, G. and Voulgaris, S.: The Peersim Simulator. <http://peersim.sf.net> (2008)