

Towards a P2P-Based Deployment of Network Management Information

Rafik Makhoulfi, Grégory Bonnet, Guillaume Doyen, and Dominique Gaiti

ICD/ERA, UMR 6279, Université de Technologie de Troyes, 12 rue Marie Curie,
10010 Troyes Cedex, France
rafik.makhoulfi@utt.fr

Abstract. Standard static centralized network management approaches are unsuitable for managing large, dynamic and distributed networks. Some decentralized approaches based on the P2P model have emerged to overcome the limitations of these centralized approaches like lack of scalability and fault-tolerance. However, they do not address issues related to the deployment of management information which is crucial in a decentralized management context because of the heterogeneous nature of this kind of information. Furthermore, current decentralized approaches still faces difficulties to ensure security, persistence and consistency of management information. In this paper, we investigate the use of a DHT as a framework for the deployment of network management information. First, we feature management information from a deployment perspective. Then, we propose a basic deployment strategy. We evaluate this approach in the context of the monitoring service considering network scalability, propagation delay and information loss under churn.

Keywords: P2P-based network management, decentralized monitoring, CIM, DHT.

1 Introduction

Current networks are evolving in terms of complexity, distribution and dynamics, making standard, static and centralized network management approaches unsuitable to manage them. In order to overcome these problems, new decentralized management architectures such as those based on the P2P technologies have emerged. Indeed, the P2P model is seen as a promising way to improve centralized network management solutions and to avoid their limitations like the lack of scalability and fault-tolerance. Due to the decentralization of management approaches and to the distribution of management information, current management approaches have difficulties to ensure security, persistence and consistency of this kind of information. Furthermore, the management information has an heterogeneous nature. So, it is necessary to have an appropriate deployment strategy for this particular kind of information. However, in the literature the proposed P2P-based management approaches do not address issues related to the deployment of management information.

In this context, we focus our work on the design of a P2P-based approach for the deployment of management information on a P2P network. For this purpose, we rely on CIM (Common Information Model)¹ as target information model. In order to validate our deployment approach, we evaluate it in the context of monitoring which is a constraining service where information is distributed, dynamic and should be retrieved in real time. The performance of the monitoring service is evaluated through both experimental and simulation tests, considering the network scalability, the delivery delay and the loss of information under churn.

This paper is organized as follows. First, we give an overview of the P2P-based management paradigm in Section II. Subsequently, we present our approach for the deployment of network management information on a P2P system in Section III, and we depict our experiments and analyze results in Section IV. Finally, we present our conclusions and perspectives in Section V.

2 P2P-Based Network and Service Management

The evolution of current networks has introduced different decentralized network management approaches, coming to overcome the limitations of centralized network management approaches like lack of scalability and reliability, caused by the use of a central network manager as a single point of failure. Due to the characteristics of the P2P model, new promising network management architectures based on this paradigm have emerged [8,17]. The general perspective behind the P2P-based management is to incorporate the P2P model advantages into network and service management architectures. P2P systems are known for their characteristics such as decentralization, self-organization, scalability and fault-tolerance. They are used to implement applications like file sharing, distributed computing, collaborative work and recently network management.

In the latter, a node (peer) in the overlay network can be considered on a management plan as an agent and a manager at the same time. Thereby, one resource can be managed by several management entities and a manager can be interested in the management of several resources [10].

In addition to [17] that presents the general principles of a P2P-based management architecture, several works address particular uses of the P2P model in a management framework. Among them, we cite: (1) Astrolabe [13], an information management system, implemented using a P2P protocol and a restricted form of mobile code based on the SQL query language for aggregation; (2) MANP2P [8], a P2P-based management solution based on key management entities such as top-level managers (TLMs) and mid-level managers (MLMs). This model is used as basis for several studies on P2P-based management like [17] and for the development of P2P-based management and monitoring systems, such as [12,4].

Most of the explored P2P-based management approaches concern the monitoring service. Among these approaches, [4] propose a monitoring and notification service used in MANP2P, that relies on the publish/subscribe paradigm to

¹ <http://www.dmtf.org/standards/cim/>

diffuse monitoring information. This service is based on two main frameworks: JXTA, a generic framework to build P2P solutions and JXTA-SOAP, an implementation of SOAP to send notification messages. [10] propose a monitoring approach based on unstructured overlay networks, where components can self-organize with a given predefined neighbor degree, such that each component becomes responsible for monitoring its immediate neighbors. This architecture is based on their membership protocol HyParView that builds and maintains an unstructured overlay network for P2P network and service management. [2] propose a management framework that performs monitoring for fault and performance management, with local and distributed tests. This architecture employs a P2P structured overlay that consists in several distributed network agents (DNA). The Kademia DHT (Distributed Hash Table) is used in this architecture to allow each DNA searching for other DNAs. [1] propose P2PM (P2P Monitor), a P2P monitoring system which uses activeXML documents and alerters deployed on each peer to detect local events.

There are also some works on the management information models like [5], an extension of CIM for P2P network and service management, instantiated on the Chord DHT [15].

Although all these approaches discuss the operational and architectural aspects of P2P-based network management, they do not address the problematic of the deployment of network management information. The latter is stored locally on the agent node itself, causing a single point of failure. Indeed, if the question of information deployment is irrelevant in the context of a centralized management where information is collected and processed by a single centralized management entity, it becomes crucial in a decentralized management context because the management information is heterogeneous, dynamic, distributed. Furthermore, due to their decentralisation, current management approaches have difficulties to ensure security, persistence and consistency of the distributed management information. For this purpose, we propose a P2P-based approach for the deployment of network management information on a P2P network.

3 Deployment of Management Information on a P2P Network

In this section, we describe the investigated DHT-based deployment approach for network management information on a P2P network. We also present a case study on the utilization of this deployment approach in the context of monitoring.

3.1 Featuring Network Management Information

In order to propose a suitable P2P-based deployment strategy for network management information, we first need to feature the management information. Indeed, the network management information has some important deployment features. We propose to classify them as follow:

- Aggregation: a value of a managed object can be non-aggregated, a partial aggregate for a subset of managed elements or a global aggregate for the entire network;
- Persistence: determines the existence duration of an information and the importance of preserving information when the related hosting nodes leaves the network [16];
- Accuracy: defines the level of accuracy of a management information;
- Security: concerns access control, confidentiality and integrity of data;
- Dynamics: a value of the managed object can be static or dynamic.

Thereby, we propose to apply an appropriate deployment strategy according to the characteristics of the used management information. This can be mainly done using standard methods such as DHTs for deploying and retrieving information, and tree or gossip based algorithms for aggregating data [11]. For our case study, we focus on non-aggregated, persistent, accurate and dynamic management information. We do not address the security aspect in this paper.

3.2 Deployment Basis

The work we undertake is to propose a model for the decentralized deployment of network management information on a P2P system. We choose CIM as information model because it is a well-known standard, employed in most of available management technologies such as DEN-ng, WBEM and knowledge plane based approaches. For example, when using WBEM² technology, there is a central management entity called CIMOM (CIM Object Manager). The latter employs a CIM Repository which stores CIM class definitions and dynamic data. In the context of a decentralized management infrastructure, there is a need to have a decentralized entity like a CIMOM for managing information.

The deployment of management information can be performed through an unstructured overlay network or a structured one. In the first category, the deployment can be done through a diffusion algorithm like gossip or flooding [9]. As proposed in [10], an unstructured overlay can be reorganized with a fixed neighbor degree so that each component becomes responsible for managing its immediate neighbors in the overlay. In the second category, the deployment can be done through a DHT. The latter can be either used to index the management information by sending a reference to a managed object to another node, or to deploy a copy of this information on a remote node. A simple tree-based algorithm can be used to propagate the management information over the tree nodes.

3.3 DHT-Based Deployment Approach

Our approach consists in considering a method based on a P2P structured overlay network. It employs a DHT [15] for delivering and looking for network management information. Also, it uses an application-level multicast system for the

² <http://www.dmtf.org/standards/wbem/>

diffusion of information and by building and maintaining a publish-subscribe multicast tree. The information deployment process is divided into three tasks: instantiation, deployment and retrieval.

Instantiation. When instantiating a CIM object class, The CIM naming mechanism offers a way to uniquely identify instances of this class. This object name is composed of the namespace path, which locates a particular namespace of a CIM implementation, plus the model path, which is the concatenation of the class name and the properties of the class that are qualified with keys. Figure 1 shows an example of a CIM class `CIM_EthernetPortStatistics`, issued from the CIM device model with one possible instance of this class, which describes the statistics for the management of an Ethernet port. To better understand the deployment process, we employ the same example in the rest of the paper.

<code>class CIM_EthernetPortStatistics {</code>	<code>instance of CIM_EthernetPortStatistics {</code>
<code> [Key] string InstanceID;</code>	<code> InstanceID='0x3:23H87A8';</code>
<code> :</code>	<code> :</code>
<code> uint64 PacketsTransmitted;</code>	<code> PacketsTransmitted=15;</code>
<code> uint64 PacketsReceived;</code>	<code> PacketsReceived=27;</code>
<code> uint32 SymbolErrors;</code>	<code> SymbolErrors=2;</code>
<code> uint32 LateCollisions;</code>	<code> LateCollisions=0;</code>
<code> uint32 FrameTooLong;</code>	<code> FrameTooLong=2;</code>
<code>};</code>	<code>};</code>

Fig. 1. An example of a CIM object class with an instance of this class

In this CIM class written in MOF (Managed Object Format), there is one key `InstanceID` and five other attributes. An instance of this class is a set of property values, declared using the keyword sequence `instance of` followed by the class name `CIM_EthernetPortStatistics`. The model path for the this class is `CIM_EthernetPortStatistics.InstanceID=0x3:23H87A8`. Therefore, following the naming scheme of CIM, the object name of this class in the case of WBEM would be `http://CIMOMNamespacePath:CIM_EthernetPortStatistics.InstanceID=0x3:23H87A8`.

In the context of a P2P-based management, information is distributed over the peers and the same object can be owned by many peers. Thus, when using the CIM naming mechanism, a manager has no indication on the source node from the object name. Thereby, a good way to resolve this problem is to add to the object name a reference to the source node so that a manager can retrieve the management information of a particular node.

Deployment. This task consists in deploying the management object on a remote peer by using a DHT. We choose a DHT tree-based solution because of its benefits. The structured topology allows us to have control on the information distribution over the peers. Besides this advantage, DHT-based overlay networks

provide a content lookup service with load balance, fault-tolerance, scalability and a mastered routing performance. They are able to retrieve information from the DHT using $O(\log(N))$ hops in a network of size N [14]. Contrary to the classical case where the management information is stored only locally with a single point of failure, we use a DHT to store the information on remote nodes. Thereby, when the source node leaves the network, the management information is still available on a remote peer. This allows us to ensure information persistence.

To deploy the management information on a DHT, we need to build an appropriate **Id** which is the hash of the object name generated by the naming scheme. The class instance of the managed object is routed by the agent node over the overlay to the nearest node (root) to the hash of the managed object name. According to this push-based algorithm, the agent periodically sends a new class instance of the managed object in order to update the old value.

Retrieval. There are two possibilities to retrieve a management information. In case where only one manager is interested by this information, the manager sends a request message for this information, by using a lookup service like the one offered by a DHT. Then, the root sends the received object to the requesting manager. On the other hand, in case where several managers exist, we use a publish/subscribe ALM (Application-Level Multicast) system for the diffusion of the management information on all the subscribed managers.

3.4 Application to the Monitoring Service

In our study, we applied our deployment approach on the monitoring service which is very constraining: monitored information is dynamic, distributed and must be retrieved periodically and in real time. Furthermore, one resource can be monitored by several managers. According to the deployment approach described above, the architecture we use to deploy the monitoring information and to send notifications from the agent to the managers is done according to a push-push publish/subscribe communication mechanism as illustrated in Figure 2.

In this architecture, there are three main actors which interact in the system. Firstly, an agent node a , which acts as a server and publishes periodically the new value of a managed object. Secondly, the root node r , which is the nearest node to the hash of the managed object, receives the managed object sent by the agent node. Finally, the M managers m_i , where $0 < M < N$ and N the number of nodes in the network, which are interested in the monitored variable, subscribe to the communication group created by the ALM in order to periodically receive the new value of the managed object stored on the root.

4 Experimental Framework

We perform semi-real experiments for evaluating the performance of the proposed DHT-based management information deployment approach, in the context of the constraining monitoring service of a P2P network.

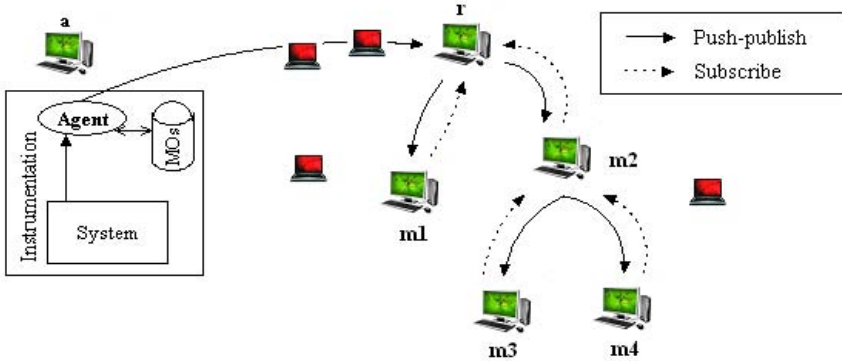


Fig. 2. System architecture

4.1 Test Environment

In order to implement the deployment approach described above, we rely on Pastry [14] to create an overlay network and on Scribe [3] to build and maintain a publish-subscribe multicast tree. We conducted our experiments on FreePastry³, a Java implementation of the Pastry DHT.

To provide results under semi-real experimental conditions, we carry out our experiments on a LAN environment consisting of 12 computers. We run multiple Pastry nodes within each JVM (Java Virtual Machine). In order to coordinate the 12 computers, we use DELTA [6], a generic environment dedicated to the test and measurement of Java-based distributed applications.

4.2 Experimental Setup

We create N pastry nodes on the overlay and we select uniformly at random one source node that will act as an agent a with M randomly chosen nodes acting as managers, where $M < N$. The agent node is in charge of periodically publishing the new values of the managed object. We create a Scribe topic whose groupId is the hash of the managed object name described in section 3.3. Scribe sets the Pastry node with the nodeId that is numerically closest to the groupId as a root of the multicast tree. All the managers send a subscribe request to join the created communication group. When the agent node sends a new value of the managed object in a publish message, this message is routed to the root before being broadcasted over the multicast tree on all the subscribed managers.

The management information that we handle in our experiments is dynamic, distributed and non-aggregated. Thus, the agent node publishes every 5 seconds a new value of the managed object. This object contains a counter, for example, the attribute `PacketsTransmitted`, depicted by Figure 1. The use of a counter

³ <http://freepastry.org>

as a monitored variable allows us to easily compare the values emitted by the source node with those received by the subscribed managers.

4.3 Results

In order to validate our DHT-based deployment approach on a P2P network, we used criteria close to those provided in [7]. Thereby, we measured the information loss in the presence of churn, the propagation delay and finally the scalability of this approach while having a large number of managers.

Information loss under churn. We evaluate the capacity of the system to handle churn (arrivals and departures of network nodes) through the information loss rate which occurs on the network in the presence of such a phenomenon. Information loss is the number of times the agent publishes a value of the managed object without being received by a manager.

To implement a realistic user behavior, we use as a reference model in our experiments the Video on Demand (VoD) over IP analysis study given in [18]. Nowadays, P2P VoD systems take a large percentage of the traffic over the Internet. It is important to ensure quality of service in this kind of systems. They are well-known for the dynamics of users, where the user behavior modeling is available in some works. In our experiments, we generate random session durations (i.e. online time) in lognormal form with parameters ($\mu = 2.2$, $\sigma = 1.5$). Similarly, the nodes arrivals into the system follow the revised Poisson process [18], where the number of arrivals usually ranges from 0 to 5 users per second. The nodes are created as long as the maximum number of nodes on the hosting machine is not reached.

Figure 3.a shows the cumulative distribution of the information loss rate. For example, we have obtained a loss rate of less than 5% in 40% of the cases. This high loss is caused by the multiple departures of the root nodes. In order to better understand this distribution, the Figure 3.b shows a typical example of the variation of the average information loss during experiments, with 120 managers. When the system is not under churn, there is no loss of information. However, under churn, we observe that there is a reasonable average loss of 0.75%, except the two peaks, where the loss reaches up to 100%. It corresponds to the case, where the root node leaves the network. So, none of the nodes receives the correct new value during a certain time interval, corresponding to the tree maintenance time. It is generally in the range of 20 – 40 seconds. A good way to avoid the information loss when the root node leaves the network is to replicate the information on another node.

Propagation delay. It is the necessary time between publishing a value of the managed object by the agent node and retrieving the same value by a subscribed manager node. Experimental results obtained when measuring the propagation delay on 120 managers without churn and under churn are given respectively in Figure 4.a and Figure 4.b. Thereby, we calculate the propagation delay for each received monitored value.

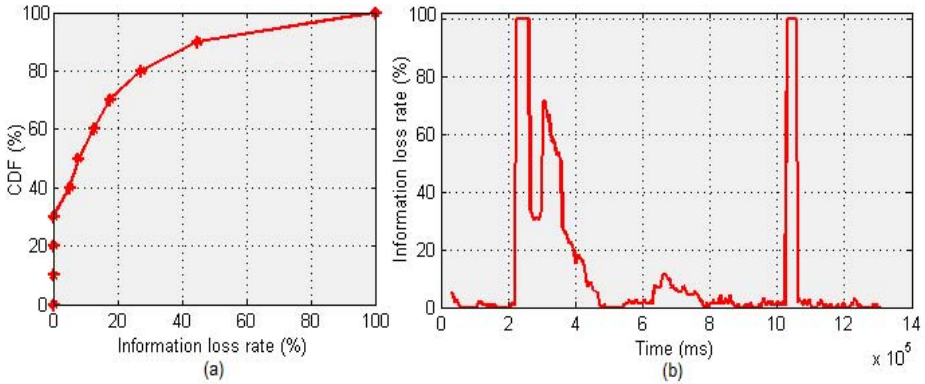


Fig. 3. Information loss: (a) CDF; (b) Variation in function of time

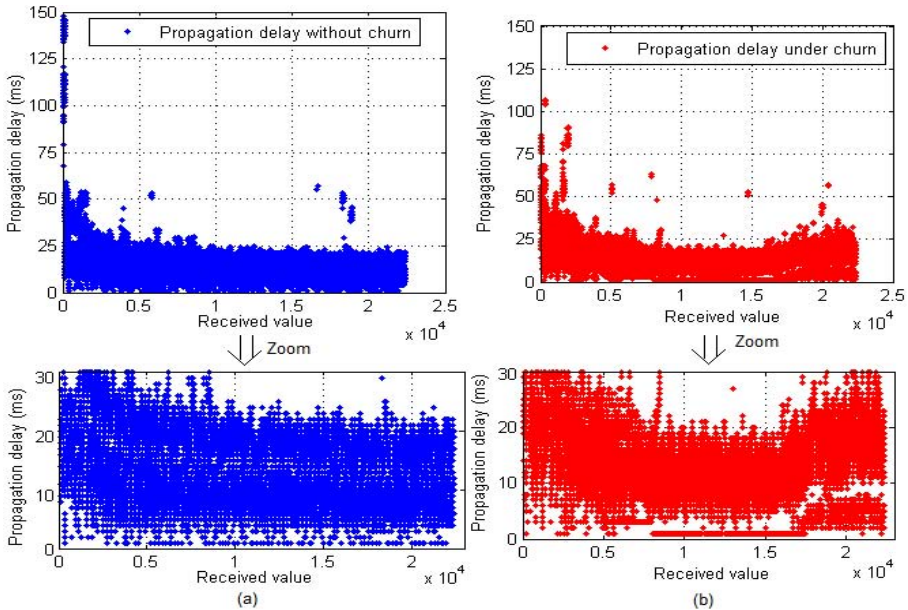


Fig. 4. Propagation delay: (a) Without churn; (b) Under churn

We notice two levels of delay in both figures. The top level, where the propagation delay is more important, corresponds to the bottom of the Scribe multicast tree and the second one is the level just below the root node. We notice a propagation delay in the range of 5-30 ms. We do not observe an influence of churn on the propagation delay because we only measure this delay at a manager when it holds the same monitoring value as the one sent by the agent.

Scalability. Scalability is the capacity of a network to operate correctly and to ensure accuracy despite of the large number of nodes in the network. A good

way to evaluate the scalability of the management service is therefore to look at the end-to-end communication delay under a large number of nodes. This delay is measured between the agent node and the managers.

Because of a small number of available machines on our test bed, we rely on the FreePastry simulator in order to consolidate the evaluation of the scalability of our approach. We use the provided Euclidean network topology model to reproduce the latency of a real P2P network topology. Furthermore, we produced 12 simulation runs for each experiment, and each value represented in the curves is the average of 3.6×10^5 values. In order to evaluate the scalability of our approach, we measure its capacity to handle a large number of managers for each managed object. The curves of Figure 5.a and Figure 5.b show the simulation and the experimental results, respectively.

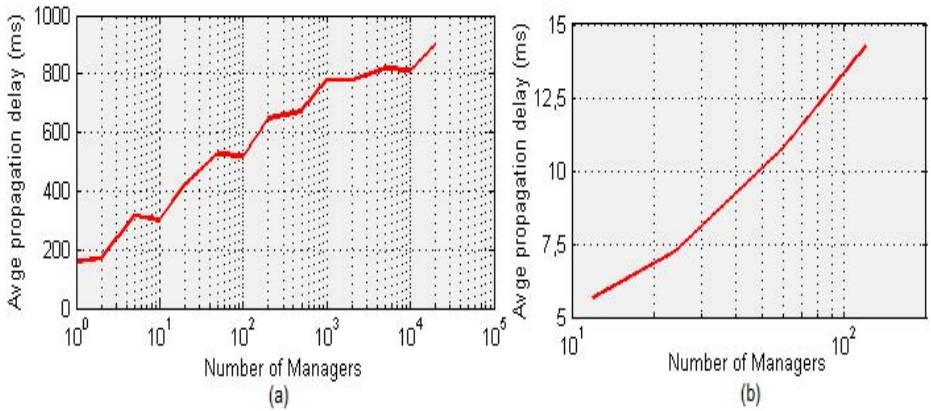


Fig. 5. Impact of the number managers on the propagation delay: (a) Simulation results; (b) Experimental results

We notice in this figure that both curves of the propagation delay in our approach follow a logarithmic distribution $O(\log_{16}(M))$, where M is the number of managers. We observe a maximum average propagation delay in the order of 950ms, obtained when using 2×10^4 managers. Our objective is not to evaluate the Pastry DHT performances, but those of our deployment approach. This is why we measure the delay between the agent and the managers, and not only the search time for a node as measured in [2]. nonetheless, our simulation and experimental results show that by exploiting the properties of the DHT, the propagation delay in our approach does indeed scale. Furthermore, the propagation delay follows the same distribution in both simulation and experimentation.

5 Discussion and Conclusion

In this paper we have introduced a DHT-based infrastructure for the deployment of network and service management information on a P2P network. We first

propose a classification for the management information features according to their aggregation, persistence, accuracy, security and dynamics, before applying our deployment approach for one category of information (non-aggregated, persistent, accurate and dynamic). Thus, in order to validate this P2P-based approach, we applied it to the monitoring service. The performance of the monitoring service is evaluated with both experimental and simulation tests, considering the network scalability, the delivery delay and the loss of information under churn.

The experimental and simulation results show that the DHT-based approach is a satisfying solution. First, under churn, there is a low information loss, except the case where the root node leaves the network. Moreover, the propagation delay is reasonable, and we do not observe an influence of churn on this delay. Furthermore, with the introduction of additional managers, the propagation delay in our approach follow a logarithmic distribution. In addition, the results show that our approach is scalable regarding the propagation delay.

For future work, we plan to enhance the classification of network management information and to consolidate our evaluation by reproducing the actual behavior of some managed objects. Finally, we consider that the deployment problematic is the same as the one addressed in database management with some common aspects such as replication, access control and complex queries. Thereby, on a long term, we will explore approaches used in the data management community and study the possibility to employ them in the context of decentralized network management information deployment.

References

1. Abiteboul, S., Mariniou, B., Bourhis, P.: Distributed monitoring of peer-to-peer systems. In: Proceedings of the 24th IEEE International Conference on Data Engineering, pp. 1572–1575 (2008)
2. Binzenhfer, A., Tutschku, K., Graben, B., Fiedler, M., Arlos, P.: A P2P-Based framework for distributed network management. In: Proceedings of the 2nd International Workshop of the EURO-NGI-Networks-of-Excellence, pp. 198–210 (2006)
3. Castro, M., Druschel, P., Kermarrecand, A.-M., Rowstron, A.: Scribe: a large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications* 20(8), 1489–1499 (2002)
4. dos Santos, C.R.P., Santa, L.F.D., Marquezan, C.C., Cechin, S.L., Salvador, E.M., Granville, L.Z., Almeida, M.J.B., Tarouco, L.M.R.: On the design and performance evaluation of notification support for P2P-based network management. In: Proceedings of the 2008 ACM symposium on Applied computing, pp. 2057–2062. ACM, New York (2008)
5. Doyen, G., Festor, O., Nataf, E.: A CIM extension for peer-to-peer network and service management. In: de Souza, J.N., Dini, P., Lorenz, P. (eds.) ICT 2004. LNCS, vol. 3124, pp. 801–810. Springer, Heidelberg (2004)
6. Doyen, G., Ploix, A., Lemercier, M., Khatoun, R.: Towards a generic environment for the large-scale evaluation of peer-to-peer protocols. In: Proceedings of the 2008 Networking and Electronic Commerce Research Conference (2008)

7. Ehrig, M., Schmitz, C., Staab, S., Tane, J., Tempich, C.: Towards evaluation of peer-to-peer-based distributed information management systems. In: Proceedings of the AAAI Spring Symposium on Agent-Mediated Knowledge Management, pp. 73–88 (2003)
8. Granville, L., da Rosa, D., Panisson, A., Melchior, C., Almeida, M., Tarouco, L.: Managing computer networks using peer-to-peer technologies. *IEEE Communications Magazine* 43(10), 62–68 (2005)
9. Jelasity, M., Montresor, A., Babaoglu, O.: Gossip-based aggregation in large dynamic networks. *ACM Trans. Comput. Syst.* 23(3), 219–252 (2005)
10. Leitao, J., Pereira, J., Rodrigues, L.: Large-scale peer-to-peer autonomic monitoring. In: Proceedings of the 3rd IEEE Workshop on Distributed Autonomous Network Management Systems, in Conjunction with GLOBECOM, pp. 1–5 (2008)
11. Makhoulfi, R., Bonnet, G., Doyen, G., Gati, D.: Decentralized aggregation protocols in peer-to-peer networks: a survey. In: Proceedings of the 4th IEEE International Workshop on Modelling Autonomic Communications, pp. 111–116 (2009)
12. Panisson, A., da Rosa, D.M., Melchior, C., Granville, L.Z., Almeida, M.J.B., Tarouco, L.M.R.: Designing the architecture of P2P-Based network management systems. In: Proceedings of the 11th IEEE Symposium on Computers and Communications, pp. 69–75. IEEE Computer Society, Los Alamitos (2006)
13. Renesse, R.V., Birman, K.P., Vogels, W.: Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Trans. Comput. Syst.* 21(2), 164–206 (2003)
14. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Guerraoui, R. (ed.) *Middleware 2001*. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
15. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 149–160. ACM, New York (2001)
16. Tout, R.N., Ghodous, P., Ouksel, A., Tanasoiu, M.: Data persistence in p2p backup systems. In: Proceedings of the 16th ISPE International Conference on Concurrent Engineering, vol. 1, pp. 149–156 (2009)
17. Xu, H., Xiao, D.: Towards P2P-based computer network management. *International Journal of Future Generation Communication and Networking* 2(1), 25–32 (2009)
18. Yu, H., Zheng, D., Zhao, B.Y., Zheng, W.: Understanding user behavior in large-scale video-on-demand systems. *SIGOPS Oper. Syst. Rev.* 40(4), 333–344 (2006)