Impact of Dynamics on Situated and Global Aggregation Schemes

Rafik Makhloufi[1], Guillaume Doyen[1], Grégory Bonnet[2], and Dominique Gaïti[1]

[1] ICD/ERA, UMR 6279. Université de Technologie de Troyes,
[2] GREYC/MAD, UMR 6072. Université de Caen Basse-Normandie

Abstract. Recently, numerous management approaches have emerged in order to manage networks and services in a decentralized and autonomous way. Some of them propose to minimize their cost by using a situated view when collecting aggregates for the decision making process, while others propose to improve their accuracy by using a more conventional approach which is global view. So far, little attention is given to the evaluation of situated view while many studies propose to evaluate global approaches. As a consequence, there is no work in the literature that compares the performance of situated and global aggregation schemes. Being able to choose the suitable approach for a given context is still a real challenge. Mastering it will ensure the efficiency of the autonomous management system. In this paper, we present a comparative study of situated and global schemes deployed over large scale and dynamic networks. We consider two factors: network and information dynamics. We implement typical aggregation schemes from each category and then we compare them according to the accuracy of the estimated aggregates and the efficiency of the decision making process.

Key words: Autonomic Networking, Decentralized Aggregation, Situated View, Management Information.

1 Introduction

In the context of autonomous network management which is one of the most relevant class of management approches, the decentralized Autonomic Managers (AMs) are based on the control loop functions Monitor, Analyze, Plan, Execute (MAPE) [8]. Information used in the monitoring can come from both the local managed element itself or be the result of an aggregation of environmental informations provided by remote AMs. Two concurrent strategies are commonly deployed for this purpose. On one hand, it can be the result of a process involving all nodes in the network, thus providing a global view (GV), standard supporting algorithms that rely on tree and gossip schemes. These global schemes are expected to provide a good accuracy at the cost of a larger convergence time and overhead. On the other hand, the monitoring process can only consider a limited neighborhood, thus providing a situated view (SV). Approaches based on SV are expected to minimize their cost and to provide a high reactivity to AMs with a low accuracy.

To date, there is no evaluation in the literature that compares the cost and the performance of situated and global aggregation schemes since the former have appeared recently. Thus, we do not know the exact behavior and the performance of these schemes while it is essential to precisely identify the appropriate context in which each of the aggregation categories performs better to design and deploy efficient autonomous management frameworks.

In this paper, we provide a comparative performance analysis of situated and global aggregation schemes. Since results coming from the use of such schemes in a static context are given by the protocols complexity, we focus our work on the resilience of such schemes to both the network and aggregated information dynamics. To this end, we implement three typical aggregation schemes, one situated view based and two global view based, i.e., gossip and tree. Then, we compare them according to two important performance criteria related to autonomous monitoring and control which are respectively the accuracy of the estimated aggregates and the efficiency of the decision making process.

The remainder of the paper is organized as follows. Section 2 presents the related work on the evaluation of aggregation schemes. Section 3 describes the aggregation schemes that we implement from each category. In section 4, we propose an evaluation of the developed schemes. Finally, in Section 5 we conclude and we present our working perspectives.

2 Related work

Several research efforts propose to compare the performance of global aggregation schemes. The closest works to ours are given below.

In [1], the authors propose a set of aggregation schemes for estimating aggregates on a P2P network. They compare one gossip-based scheme Propagate2All to two other tree-based schemes: SingleTree and MultipleTree. This comparison shows that tree schemes outperform gossip ones in terms of convergence time, communication and computation costs, but they are less resilient to faults under a dynamic network. However, this study does not discuss the situated schemes.

A second work [17] proposes a comparison between GAP (Generic Aggregation Protocol), a basic tree-based aggregation protocol and G-GAP (Gossip-based GAP), a gossip-based protocol for continuous monitoring of aggregates. Surprisingly, in opposition to the first study, this evaluation shows that, for comparative overhead, the tree-based scheme GAP outperforms the gossip protocol G-GAP both in terms of accuracy and robustness even under a dynamic network.

Another study of the performance of aggregation schemes [2] discusses the strengths and limitations of gossiping. According to this characterization, gossip schemes present simplicity, bounded load on nodes, topology independence, ease of local information discovery and finally robustness to transient network disruptions. But, the small bounded message sizes and the relatively slow periodic exchanges limit the information carrying capacity of a gossip algorithm. Furthermore, gossip scales well in some dimensions but not for all. A node failure can also delay or even defeat the aggregation. This study provides only a qualitative analysis of the gossip's limitations and strengths without giving any quantitative result.

We recently proposed in [14] an evaluation study of typical global and situated aggregation schemes that we implemented in the context of a static environment. This work offers quantitative results on the costs and the performance of aggregation. However, this study does not compare these schemes in the context of a dynamic environment.

Globally, these studies address the global aggregation schemes but never compare them to the situated ones while the latter is massively used in autonomous management frameworks. Thus, we do not know the exact behavior and the performance of the situated schemes in comparison to the global ones. Situated schemes are expected to provide a high reactivity to AMs by minimizing the amount of maintained information by each AM. As for global schemes, they are expected to provide a good accuracy at the cost of a larger convergence time. Through this, we clearly observe that it is essential to precisely identify the appropriate context in which each category performs better in order to be able to design efficient management frameworks.

3 Developed aggregation schemes

In order to show under which conditions an aggregation approach outperforms the others, we implement and compare state of the art typical schemes that can be used for the computation of any aggregate function. In this section, we give a brief overview of them. The notations used in these algorithms are summarized in Figure 1.a.

3.1 Global view

In global aggregation schemes, each node collects information from the entire network. According to the global data structure, these schemes are either tree or gossip-based.

Tree We implemented a simple push tree-based aggregation scheme that uses the typical aggregation process of GAP under the DHT-based deployment topology proposed in [13]. The latter is a structured P2P overlay where nodes communicate their local aggregates to a single root node that computes an overall aggregate and spreads it on all the interested nodes through a publish-subscribe mechanism. As shown in Algorithm 1, which describes the developed tree-based scheme, one active and one passive threads are executed on each node. The first one initiates the aggregation process by sending the value of each node to its parent. The second one waits for messages sent by an initiator to process them. Initially, each node i selects its parent p with the help of the getParent() method and sends it a couple $< X_i, 1 >$ on the form $< aggregate, weight >$ as given in lines a.1 and a.2. A node i that receives a message from a child node j computes a new partial aggregate according to a given aggregate function, updates its local state with the new values and then forwards them to its parent p as shown in lines b.2 to b.5. If node i is the root then it waits until it receives all its children's aggregates and it diffuses the computed global aggregate X_i over a publish-subscribe system on all the interested nodes as given in lines b.6 to b.9. Thus, each node that receives X_i from the root updates its partial aggregate with the global one as shown in lines a.3 and a.4.

Algorithm 1 Push tree-based algorithm executed on a node i

```
(a) Active thread
1: $p$←getParent()
2: send ($< X_i, 1 >, p$)
3: receive($X_j, j$)
4: $state_i$←update($X_j$)

(b) Passive thread
1: **loop**
2:     receive ($< X_j, w_j >, j$)
3:     $state_i$←update($X_j, w_j$)
4:     $p$←getParent()
5:     send ($< X_i, w_i >, p$)
6:     **if** (getParent()=null) **then**
7:         wait until receive all aggregates
8:         diffuse ($X_i$)
9:     **end if**
10: **end loop**
```

Gossip Unlike tree-based techniques where nodes are organized into a tree, gossip-based schemes do not require a particular structure to perform aggregation. In each round of a gossip, a node contacts one or more of its neighbors usually chosen randomly and exchanges information with them [2, 5, 7]. Initially, each node in the network has only its own raw management information. When messages are exchanged, nodes compute new partial aggregate and then decrease the variance over the set of all aggregates in the system. The gossip aggregation algorithm converges when the global aggregate is computed and available across all the network nodes.

In this category, we implemented a push-pull gossip scheme based on a typical gossip scheme [6]. As illustrated in Algorithm 2, with the help of the getNeighbors(1) method that selects uniformly at random one node j from a list of direct neighbors $\mathbb{D}_i$, each node i sends to j a message containing its partial aggregate X_i and waits for a response as shown in lines a.2 and a.3. When i receives a response (X_j, j) from j, it updates its local state through the update() method that computes a new partial aggregate according to the selected aggregate function as given in lines a.4 and a.5. The node i waits for a certain time and repeats the same process (line a.6). Similarly, when the passive thread of a node i receives an exchange request message, it replies with its local aggregate X_i and then it updates its local state as shown in lines b.2 to b.4.

3.2 Situated view

In opposition to the global schemes where global aggregates are computed over the entire network, situated view approaches like [9, 12, 3] propose to reduce

Algorithm 2 Push-pull gossip-based algorithm executed on a node i

(a) Active thread

```
1: loop
2:     $j$←getNeighbors(1)
3:     send ($X_i, j$)
4:     receive($X_j, j$)
5:     $state_i$←update($X_j$)
6:     wait(round duration)
7: end loop
```

(b) Passive thread

```
1: loop
2:     receive($X_j, j$)
3:     send ($X_i, j$)
4:     $state_i$←update($X_j$)
5: end loop
```

the costs of aggregation by limiting the knowledge of a node to a bounded neighborhood. Thus, apart its own local information each node collects data from its direct neighbors or a subset of the network nodes to compute a partial aggregate representing its view.

In this category, we implemented a typical situated scheme inspired from the membership protocol HyParView (Hyper Partial View) [9] where each node maintains two views. In our implementation, each node executes Algorithm 3 to collect information from its view and then to compute a partial aggregate. As shown in Algorithm 3, each node i sends a query message containing the maximum number of hops h to all its direct neighbors obtained through the getNeighbors(all) method as given in lines a.1 and a.2. A node i that receives a request from j verifies if it does not previously answer to the same request (line b.2). If so, i answers by sending a response (X_{raw_i}, j) directly to the requesting node j (line b.3). The node i decrements the value h and tests if the maximum number of hops is not reached and then forwards the received request to all its direct neighbors as given is lines b.4 to b.8. When a requesting node i receives an answer (X_{raw_j}, j) from a neighbor j, it computes a new partial aggregate and updates its own state as shown in lines a.3 and a.4.

Algorithm 3 Pull situated view algorithm executed on a node i

(a) Active thread

```
1: $\mathbb{D}_i$←getNeighbors(all)
2: send ($h, \mathbb{D}_i$)
3: receive($X_{raw_j}, j$)
4: $state_i$←update($X_{raw_j}$)
```

(b) Passive thread

```
1: loop
2:     receive($h, j$)
3:     send ($X_{raw_i}, j$)
4:     $h$←$h - 1$
5:     if ($h > 0$) then
6:         $\mathbb{D}_i$←getNeighbors(all)-{j}
7:         send($h, \mathbb{D}_i$)
8:     end if
9: end loop
```

4 Evaluation study

In this section, we present the evaluation study that we carried out to compare the performance of the previously described aggregation schemes in terms of the accuracy of the estimated aggregated and the efficiency of the decision making process in a dynamic environment.

4.1 Dynamics factors

We identify two important factors that can affect the performance of an aggregation process: network and information dynamics. For this, we propose to use the following models.

Network dynamics The network dynamics, or churn, consists in the nodes arrivals in the system and their departures that can be planned or due to a sudden failure. Based on the analysis of the traces on different systems like mobile ad-hoc networks (MANETs) and P2P ones, many prior studies assumed a Poisson process of parameter λ for the node arrivals and an exponential distribution of parameter μ for the departures [10, 11, 15]. Thus, we adopt the same models in our experiments and we choose the values of arrival and departure rates so that they accurately represent existing networks. The values of λ and μ are chosen so that the average number of nodes, given by N=λ/μ, will always be equal to 500. Thus, in our study λ varies in [0.0041;0.4166] and μ is ranging in [8.33×10^{-6};8.33×10^{-4}], corresponding respectively to average inter-arrivals ($1/\lambda$) ranging in [2.4;240] seconds and to average lifetime durations ($1/\mu$) in the range [20;2000] minutes.

Information dynamics It determines the evolution degree of a management information over the time. It can be characterized through two criteria: (1) the changing frequency of the values on managed elements, for example a value X_{rawi} that changes every 1sec; (2) the changing degree of these values that is defined by the distance between two consecutive values. We generate realistic local values according to a triangular signal that could stand for a gauge associated, for example, to the quantity of traffic in a network. Thus, X_{rawi} is periodically incremented by α until reaching 100 and it is then decremented by the same α until 0. This allows us to control the information dynamics level. The local values are ranging in [0;100] for an α that we vary in [0.02;5].

4.2 Simulation setup and evaluation scenarios

In order to evaluate the performance of the investigated aggregation schemes under realistic conditions, we carry out our experiments on the FreePastry simulator, an implementation of the Pastry DHT [16]. All simulations are done under the Euclidean network topology model, where nodes are randomly placed within a 2-dimensional plane in a uniform way. Concerning the tree-based scheme, we also rely on Scribe [4] to build and maintain a multicast tree that allows the diffusion of the root's global aggregates on all nodes that subscribed to the same diffusion group concerning the monitored variable.

Based on the realistic parameters summarized in Figure 1.b, we create a network where each node executes the previously described aggregation schemes to compute an average aggregate of the local values on nodes. We execute the situated scheme with a view limited to the direct neighbors (SV1) and also with a neighborhood limited to 2 hops (SV2). The gossip process is executed with rounds of duration 600ms that corresponds to the maximum time for to exchange and process an information between two nodes. The neighborhood degree in each scheme is limited to 8 neighbors per node. It is on the form 2×2^b of Pastry where b=2. The maintenance frequency of this neighborhood is fixed to 60 seconds like the one used by FreePastry to maintain the leaf sets. Concerning the local raw values, we assume that each 1sec a new value is generated on each node. In all our simulations, a 95% confidence interval is computed for each average value represented in the curves. These intervals are plotted as error bars.

Symbol	Signification
X_i	local aggregate of node i
X_{raw_i}	raw (non-aggregated) value of node i
w_i	weight of X_i
$state_i$	local state on i
h	number of hops
$\mathbb{D}_i$	set of direct neighbors of node i

(a)

	Parameter	Value
Constants	**Aggregate function**	Average
	Network topology model	Euclidean
	Topology maintenance	60sec
	Neighborhood degree	8
	Max. number of nodes	1000
	Value changing freq.	1sec
	Decision making freq.	30sec
	Gossip round duration	600ms
Variables	**Node arrivals**	$\lambda \in [0.0041; 0.41]$
	Lifetime durations	$\mu \in [8.33 \times 10^{-6}; 8.33 \times 10^{-4}]$
	α	[0.02; 5]
	Number of hops in SV	h∈[1;2]

(b)

Fig. 1. Notations (a) and simulation parameters (b)

4.3 Evaluation results

We propose here to compare the developed aggregation schemes according to their costs and according to two other important criteria, the accuracy of estimated aggregates and the efficiency of the decision making process.

Cost of situated and global schemes Considering the cost of the aggregation schemes in a static context allows us to have a first idea on their expected performance in a realistic dynamic environnement. To this end, we compare the previously described schemes in the context of a static environment according to convergence time and communication cost. We notice in Figure 2.a that the convergence time is proportional to the number of nodes in the network. Under a network of 1000 nodes, gossip's convergence time is about 6 times higher than the one of the tree and about 23 times the one of SV2. This high delay is caused by the blind communication used to exchange messages at each round of gossip. As expected, the situated scheme, with a comparable delay between SV1 and SV2, is more scalable and converges more quickly than the global ones, because in a situated view only partial aggregates are computed by nodes. However, we see in Figure 2.b that SV2 involves more messages than the global schemes because it is based on a broadcast algorithm where each node floods its request message on all its h-hops neighbors. Numerically, under a network of 1000 nodes, the communication cost of SV2 is almost 3 times higher than the one obtained in the case of gossip and about 42 times the one of the tree. The tree causes the lowest communication cost, which corresponds to the messages sent in a bottom-up fashion to the root and those used by Scribe to spread the global aggregate.

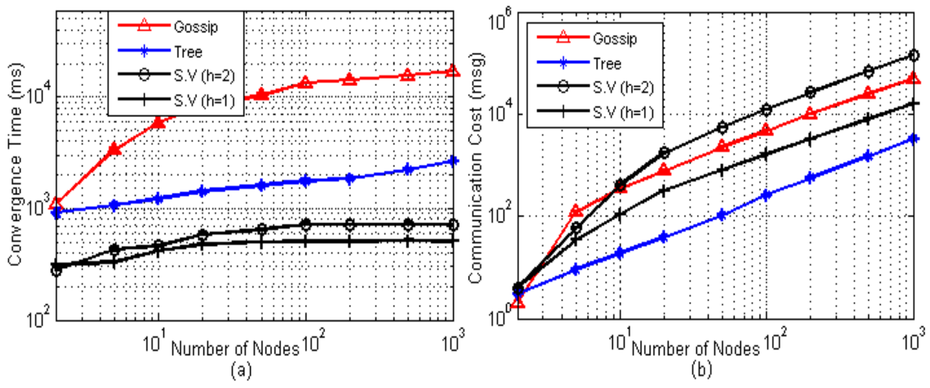


Fig. 2. Cost of SV and GV: (a) Convergence time; (b) Communication cost

Accuracy of estimated aggregates We define it as the precision level of the aggregation scheme when computing global or situated aggregates. In order to evaluate the impact of network and information dynamics on this criterion, we look at the distribution of estimates over all the network nodes and we measure the distance between estimated aggregates and the actual global aggregate that should be computed. Thus, we show how far these estimates lie from the real average value by computing the average deviation $D_X = \frac{1}{N}\sum_{i=1}^{N}|X_i - X|$.

By varying the information dynamics level within a static network of 1000 nodes, we obtained the results shown in Figure 3.a. We see that, with a deviation that reaches 2 when α=5, tree is about 5 times more accurate than SV1, about 3 times more than SV2 and almost twice more than gossip. Thus, it is the most accurate scheme under dynamic information. The situated scheme is less accurate than the global ones. This is caused by the computation of partial aggregates.

Indeed, the deviation never reaches 0 like in the case of global schemes even under a static network and information. As predicted in [2], gossip is more sensitive to the information dynamics than the tree. But we do not clearly see here the evolution of the accuracy in function of the information dynamics level.

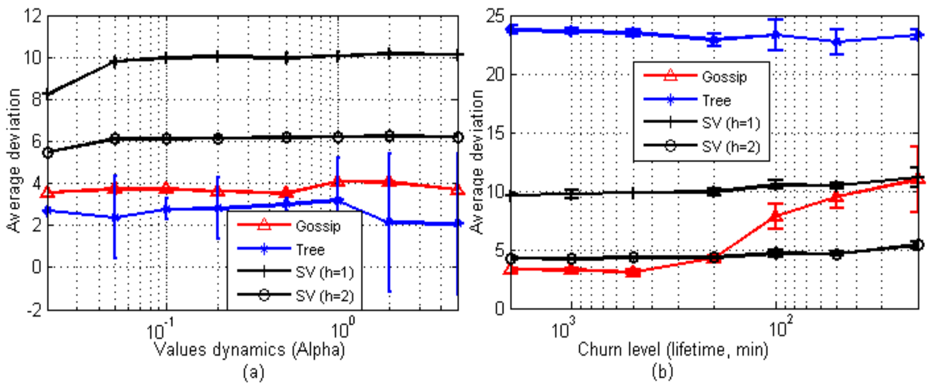
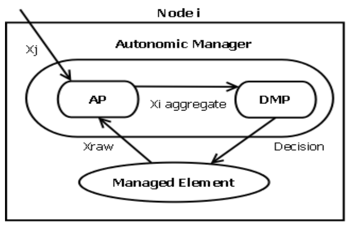


Fig. 3. Accuracy of estimated aggregates: (a) Information dynamics; (b) Churn level

When we consider random fixed values on nodes (i.e. α=0) and we vary the network churn level, we obtain the deviation measures illustrated in Figure 3.b. We notice that in the case of the tree-based scheme, the network dynamics dramatically affects the accuracy of estimated aggregates with a deviation that reaches 24. This sensibility is caused by the departures and arrivals of nodes causing the continuously maintenance and reconstruction of the tree, whatever the churn level. The situated scheme is more resilient to churn than gossip and tree since we register the same deviation for it in the case of static and dynamics environments. Thus, SV2 is about 4 times more accurate than the tree and about twice more than SV1. This is explained by the fact that, in the situated scheme, a departure or an arrival of a node does not necessarily affect other nodes if it is not in their view. For the gossip, the deviation is proportional to the churn level because it is a community process where nodes are dependent upon the correct behavior of all other nodes. However, it is still the most accurate scheme under a low churn level where the average lifetime and inter-arrival are respectively higher than 200min and 24sec. In the other interval, SV2 is the most accurate.

Efficiency of the decision making process In order to evaluate the efficiency of the decision making mechanism when using each of the implemented aggregation schemes, we implemented a simple threshold-based decision mechanism standing for a realistic case of control operations where each node makes decisions according to its last computed aggregate. The local values of managed elements are periodically retrieved by the AM to be used by the aggregation process (AP) and then by the decision process (DMP), as shown in Figure 4. According to Algorithm 4, each node i consults its local aggregate and makes a decision according to the defined threshold 50 (lines 2 to 7). Node i waits a certain time before renewing the same operation (line 8).



Algorithm 4 Decision making algorithm

```
1: loop
2:     if X_i >50 then
3:         decision←d1
4:     else
5:         decision←d2
6:     end if
7:     applyDecision(Decision)
8:     wait(round_duration)
9: end loop
```

Fig. 4. Decision making process

At the end of the aggregation process, we compare the resulting decisions on each node to the one that should be made in the case where we use the current actual global aggregate. Thus, we count the number of decentralized decisions that do not match with the actual one (i.e. the number of faulty decisions). Under the same test conditions as in the evaluation of the aggregation accuracy, we obtain the results presented in Figure 5.

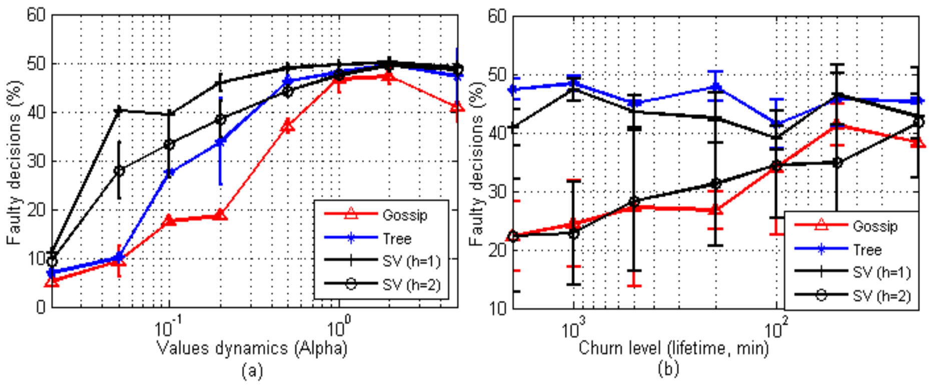


Fig. 5. Efficiency of the decision making: (a) Information dynamics; (b) Churn level

We observe in figure 5.a that the number of faulty decisions increases according to the information dynamics level. The information dynamics affects the aggregation process and then the decision making one. One can note that for all the aggregation schemes, the number of faulty decisions is ranging in [5;12]% when α=0.02, but when α=1 or 2 the number of faulty decisions reaches the interval [45;50]% where the decision making process is unable to take the right decisions. Globally, this error percentage is larger in the situated view while gossip offers the best quality of the decision making. Under churn, Figure 5.b shows that, in the case of gossip and SV2, the number of faulty decisions is comparable and lower than the one registered when using a tree or SV1. Moreover, it is proportional to the churn level. We also observe that churn hugely affects the quality of the decision making process on the tree where the average number of faulty decisions is about 45%. Thus, it is more relevant to use gossip or SV2 on a dynamic network.

To conclude, as expected, the situated view provides a higher reactivity to AMs than the global ones at the cost of a higher communication. SV1's convergence time is about 5 times lower than the one of tree and 33 times lower than the one of gossip. Concerning the communication cost, the one of SV2 can be almost 3 times higher than the one obtained in the case of gossip and about 42 times the one of tree. As for the global schemes, under dynamic information, they offer more accuracy in the computation of aggregates and in the decision making. With a deviation that reaches 2 when α=5, the tree-based scheme provides the best accuracy. It is about 5 times more accurate than SV1, about 3 times more than SV2 and almost twice more than gossip. Concerning the efficiency of the decision making, gossip provides the best performance. For all the aggregation schemes, the number of faulty decisions is ranging in [5;12]% when α=0.02, but when α=1 or 2 the number of faulty decisions reaches the interval [45;50]%. In the presence of an acceptable churn, gossip performs better than the other schemes in terms of the aggregation accuracy and the efficiency of the decision making mechanism, followed by the situated view and then the tree. When the average lifetime and inter-arrival are respectively higher than 200min and 24sec, gossip is about 4 times more accurate than the tree and about twice more than SV1, with a comparable accuracy with SV2. However, when the average lifetime and inter-arrival are respectively lower than 200min and 24sec, gossip has a comparable accuracy with the one of SV1, and SV2 becomes the most accurate scheme. Thus, SV2 has a comparable performance of the decision making mechanism to the one of gossip. As for tree, it is dramatically affected by churn since the average number of faulty decisions is about 45%.

5 Conclusion and future work

In the context of autonomous network management, two concurrent approaches are commonly used to achieve the aggregation of management information which are situated and global views. However, we do not know the exact behavior and the performance of these different schemes. Moreover, it is crucial to precisely identify the appropriate context in which each of the aggregation categories performs better. To this end, we provide in this paper a comparative performance analysis of typical situated and global aggregation schemes. This work represents a first step in the evaluation of aggregation schemes. In an effort to enhance it, we are looking for ways to consolidate the obtained results by considering additional impacting parameters (e.g. the changing frequency of local values or various neighborhood degrees). On a long term, we will pursue this work by designing an adaptive system that combines the use of situated and global aggregation schemes. Such an autonomous management system would adapt itself to the supporting network operational behavior, bringing it self-optimization capabilities. A decentralized process will continuously analyze its environment to discover the current context of the management information and its environment and then selects the appropriate aggregation scheme that gives the best performance, in order to build efficient autonomous management frameworks.

References

1. Bawa, M., Garcia-Molina, H., Gionis, A., and Motwani, R. Estimating aggregates on a Peer-to-Peer network. Tech. rep., 2003.
2. Birman, K. The promise, and limitations, of gossip protocols. *SIGOPS Oper. Syst. Rev. vol.41*, 5 (2007), 8–13.
3. Bullot, T., Khatoun, R., Hugues, L., Gaïti, D., and Merghem-Boulahia, L. A situatedness-based knowledge plane for autonomic networking. *Int. J. Netw. Manag. 18*, 2 (2008), 171–193.
4. Castro, M., Druschel, P., Kermarrec, A.-M., and Rowstron, A. Scribe: a large-scale and decentralized application-level multicast infrastructure. *JSAC vol.20*, 8 (2002), 1489–1499.
5. Dietzfelbinger, M. Gossiping and broadcasting versus computing functions in networks. *IWACOIN vol.137*, 2 (2004), 127–153.
6. Jelasity, M., Montresor, A., and Babaoglu, O. Gossip-based aggregation in large dynamic networks. *TOCS vol.23*, 3 (2005), 219–252.
7. Kempe, D., Dobra, A., and Gehrke, J. Gossip-based computation of aggregate information. In *FOCS* (2003).
8. Kephart, J. O., and Chess, D. M. The Vision of Autonomic Computing. *Computer vol.36*, 1 (2003), 41–50.
9. Leitao, J., Pereira, J., and Rodrigues, L. Large-Scale Peer-to-Peer Autonomic Monitoring. In *DANMS* (2008), pp. 1–5.
10. Leonard, D., Rai, V., and Loguinov, D. On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks. *SIGMETRICS vol.33* (2005), 26–37.
11. Li, J., Stribling, J., Morris, R., Kaashoek, M. F., and Gil, T. M. A performance vs. cost framework for evaluating DHT design tradeoffs under churn. In *INFOCOM* (2005), pp. 225–236.
12. Macedo, D. F., dos Santos, A. L., Pujolle, G., and Nogueira, J. M. S. MANKOP: A Knowledge Plane for wireless ad hoc networks. In *NOMS* (2008), pp. 706–709.
13. Makhloufi, R., Bonnet, G., Doyen, G., and Gaïti, D. Towards a P2P-based Deployment of Network Management Information. In *AIMS* (2010), pp. 26–37.
14. Makhloufi, R., Doyen, G., Bonnet, G., and Gaïti, D. Situated vs. global aggregation schemes for autonomous management systems. In *DANMS* (2011).
15. Rhea, S., Geels, D., Roscoe, T., and Kubiatowicz, J. Handling churn in a dht. In *ATC* (2004), p. 1127140.
16. Rowstron, A. I. T., and Druschel, P. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Middleware* (2001), pp. 329–350.
17. Wuhib, F., Dam, M., Stadler, R., and Clem, A. Robust monitoring of network-wide aggregates through gossiping. *TNSM vol.6*, 2 (2009), 95–109.