International Journal on Data Science and Technology
2016; 2(1): 9-14
Published online January 29, 2016 (http://www.sciencepublishinggroup.com/j/ijdst)
doi: 10.11648/j.ijdst.20160201.13

Heal Gossip: A Secure Overlay for Unstructured P2P Networks

Anubhava Srivastava, Dharmendra Kumar

Dept. of Computer Science and Engineering, United College of Engineering and Research, Allahabad, U.P., India

Email address:
anubhavacse@gmail.com (A. Srivastava), kumar.dharmendra@rediffmail.com (D. Kumar)

To cite this article:
Anubhava Srivastava, Dharmendra Kumar. Heal Gossip: A Secure Overlay for Unstructured P2P Networks. *International Journal on Data Science and Technology.* Vol. 2, No. 1, 2016, pp. 9-14. doi: 10.11648/j.ijdst.20160201.13

Abstract: Gossip-based protocols are an e cient mechanism for man-aging pure unstructured peer-to-peer (P2P) networks. Such protocols are Newscast, Cyclone, Lbpcast, etc. They have overcome from several di culties of such P2P random overlay connection. Such difficulties are randomness, high churn rate, very large unstructured distributed net-work, etc. But the performance of all gossip-based protocols have been completely vanished by presence of few malicious nodes. Since, non-detectable messages and behaviour of attackers are not leave them se-cure. These malicious nodes divide the overlay into several isolated clusters such as in Hub Attack or may be engaged non-malicious nodes in such a way that they are denying actual work such as in Denial of Ser-vice (DoS) Attacks. For securing unstructured P2P networks, there are some existing security protocols such as Secure Peer Sampling (SPSS), TooLate, S-Gossip etc. They are able to identify the malicious nodes and restrict them from gossiping. But restricting some malicious nodes on each node is not sufficient the security purpose of such epidemic over-lays. Especially in completely distributed networks, the malicious nodes may affect other non-malicious nodes although they have been already captured and restricted for gossip on others. In this regards, a new gossip mechanism is proposed, named HealGos-sip. It uses an additional property to inform captured malicious nodes on a node to all its neighbours. This process helps to identify and re-strict malicious nodes faster than other security mechanism. The propose mechanism relief the non-malicious nodes from the group of malicious nodes while performing detecting process. Hence, the propose protocol reduces the communication overhead as well as paralyses almost all malicious nodes within the network. For confusing among malicious and non-malicious nodes while detecting, a new variant of Hub attack is pro-posed and is called Hide and Seek (HnS) attack. It is able to miss-lead existing security protocols regarding the restriction of malicious nodes from gossip.

Keywords: Heal Gossip, Pear to Pear Network, Hide and Seek Attack Model, S-Gossip, Routing

1. Introduction

Now days, the Internet plays a great role in any type of digital communication. The continuously expansion of Internet lead to the discard of traditional client-server based system. Alternative way is to use of distributed architecture. Peer-to-Peer (P2P) network is a kind of distributed application architecture. In this, the system nodes or peers are equally privileged and collaborate with each other to carry out a service. It refers to network communication without servers and allows host to communicate directly with other peers. Basically P2P system architecture is based over the Internet architecture. It is implemented as an abstract overlay network built over application layer. It is used to handle various complex services such as le sharing system, data containing digital formats e. g, audio les, and real time data e.g, VoIP telephony traffic passed through P2P system. P2P networks are mainly categorized into two types: Structured and Unstructured based on how the peers in the overlay are connected to each other.

Structured P2P system has a predefined specific connection between nodes, and connection of the nodes is based on their assigned IDs. Structured P2P overlays also known as distributed hash tables (DHT). They provide a natural support to do such functionality. DHTs logically organize peers in a well-de ned structured. In P2P systems, DHTs are of great importance as they support to perform an exhaustive and exact search in very large-scale systems [1].

Unstructured P2P systems, peers are linked either randomly or probabilistically based on some proximity metric between the nodes. The overlay achieves random topology, which is allowing more flexibility in its structure. Regarding this, gossip-based protocols came as an efficient way to construct such unstructured overlays [2, 3].

Gossip protocols can connect overlays even in the presence of high churn where nodes joining and leaving the system at any time. Such connected random overlays provide an efficient support for range and keyword-based queries. Gossip-based protocols [4, 5] have come as popular paradigm for pure unstructured P2P systems. Interestingly these gossip protocols have been designed pre-serve to connectivity even if 70% of nodes at a time become dead or leave from existing overlay [6, 8]. In these protocols, each node maintains a set of links to a small set of neighbours, constituting a partial view of the network. The resulting connected graph forms an unstructured overlay [9, 10].

Thus, the major drawback of gossip protocols is that a few malicious nodes can easily create a hub and partition the network. SPSS [11], TooLate [12] and S-Gossip [13] are some known protocols are able to secure unstructured P2P networks against Hub attacks. Some other papers such as Uniform and Ergodic Sampling [14] and Brahms [15] are also mentioned about the effects of malicious nodes.

SPSS approach is intuition based. A node having large in-degree assumed to possess potentials to become a hub, so malicious. There is a central authority keeps a watch on every node and determines threshold for in-degree. It maintains two lists, namely, (i) blacklist, and (ii) whitelist. Nodes having in-degree higher than the threshold are stored in blacklist. Every node first checks with central authority to determine if the selected peer for gossip belongs to whitelist. If so, then gossip is carried out.

Since gossiping process is quite fast, before a node determines whether the selected node is in blacklist or whitelist, there is high probability that the malicious nodes can pollute the cache or view. Also central authority may fail or can be hacked. Therefore, SPSS cannot provide integrated security features one would expect from a secured gossip protocol. To address this issue, namely, the problem of centralized blacklist as stated above, TooLate was proposed. TooLate is a completely decentralized protocol for secured gossip which tightly couples maintenance of blacklist with the base gossip protocol.

Recently, S-Gossip is proposed to reduce multiple instances of Toolate. In this protocol, each node maintains three tables to ban malicious nodes. These tables are referred as three level of filtering for the overlay nodes. Third level of filtered nodes are marked as malicious one and restricted from the gossip.

Further, the work is organized to start with describing the basic of S-Gossip in section II. Section III explains the propose Hide and Seek attack model. The security mechanism of HealGossip is discussed in section IV. Simulation results are analyzed in the section V. Finally, we conclude our research work with future direction in section VI.

2. Basic of S-Gossip

S-Gossip has been proposed to reduce communication overhead while protecting with malicious nodes. Unlike Toolate security protocol, it uses only one instance of protocol. There is no central authority as in SPSS. For capturing malicious nodes, it uses three tables on each node. The tables are Genuine Table (GT), Suspicious Table (ST) and Malicious Table (MT). The tables are updated on each gossip. All tables are maintained with different descriptors for node entries. GT has Node Identification (NID), Suspicious Count (SCount) and Time-to-Live (TTL) values of a node while ST maintains Malicious Count (MCount) instead of SCount. On the other hand, MT has only NID and TTL values. All tables is maintained a unique NID and is forced to shift when the de ned threshold will be crossed. The threshold of GT is defined on SCount. The average or mean value of SCount is computed through equation 1 and standard deviation through equation 2.

$$\mu SCount = \frac{1}{|N|} \sum_{i-0}^{(N-1)} node_{i.(SCount)} \quad (1)$$

and

$$\sigma SCount = \sqrt{\sum_{i=0}^{(N-1)} \left(node_{i.(SCount)} - (\mu SCount) \right)^2} \quad (2)$$

Here, N is the size of overlay. While updating the table entries, the nodes of GT are shifted to ST when the node's SCount will be more than the addition of mean and standard deviation of entire SCount of the table. The shifting of nodes from ST in to MT is happened whenever MCount value of the node will be exceeded the MCount threshold. The value is prede ned which is equal to the View Table (ViTab) length of the S-Gossip. ViTab refers as view of a node in the traditional gossip mechanism.

The mechanism assumed that the entry of nodes inside malicious table will be declared as malicious. This table maintains only TTL value for the captured nodes and the value is reinitialized whenever the node reappear while gossip. From this mechanism of S-Gossip, it ensures that the captured actual malicious nodes inside Malicious table will be stuck in a loop. The value of TTL is initialized with the predefined ViTab length size. The nodes are free from tables as soon the TTL value becomes zero.

The mechanism tried to eliminate false + ve and false - ve from the tables. S-Gossip proved through simulation that it can provide secure from Hub attack. But, the mechanism gets in trouble if the malicious nodes will stop gossip for some interval. This will reduce the SCount and MCount of the malicious nodes. The side effect of this it that the malicious nodes may come out from ST or MT tables. This effect will be reduce to get knowledge of malicious nodes from neighbours. But in S-Gossip, there is no provision to inform a captured malicious entries among neighbours.

3. Hide and Seek Attack Model

We propose an attack model especially for those networks who care about frequency of hits of the malicious nodes while updating their routing tables. As discussed that some security protocols are very effectively captured the malicious nodes

while gossiping with others such as in Toolate, S-Gossip, etc. They used to measure the frequency of continuously gossiping nodes and mark as malicious if they crossed the predefined threshold.

The basic mechanism of the proposed HnS attack model is that it rst attacks the overlay with malicious peers and then hide from the overlay. In case of hide, it performs indirect attack from sending the affected non-malicious nodes. These affected non-malicious nodes are managed in such way that all non-malicious nodes are confused to mark the malicious nodes. For implementing the attack, malicious nodes have two tables rst for own malicious peers, called Malicious Peer Table (MPeerTab) and second for affected non-malicious, called Affected Table (AffecTab). The size of MPeerTab is predefined and it is equal to total number of malicious nodes. On the other side, AffecTab can grow up to network size, but normally it is allow to grow up to size of ViTab. The AffecTab is refreshed by TTL value for avoiding dead links. The percentage of entries in AffecTab trigger the frequency of indirect attack. Less than 50%, out of the size of ViTab, entry indicates no need of indirect attack. It means only direct attack is triggered. More than 50% entries indicate alternate while 100% stops the direct attack. The summary of direct and indirect attacks are illustrated in the Table 1.

Table 1. *Trigger Points of Direct and Indirect attacks.*

Size(%)	Direct Attack	Indirect Attack	Overlay Connection
< 50	Continuous	No	1-Hop
> 50	Alternate	Alternate	Partially
= 100	No	Continuous	2-hops

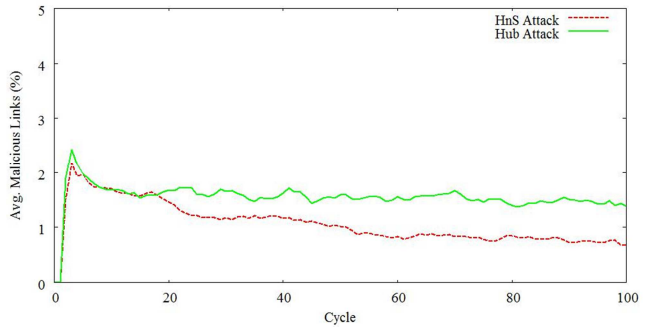

Fig. 1. *Attack on S-Gossip with 20 malicious nodes. Network size is 1000 nodes.*

The propose HnS attack mechanism insures, in worst situation, that each non-malicious node is maximum two hops away from malicious peers. After leaving all malicious nodes they divide the overlay in several partitions. The best case of the attack is to make complete partition of the overlay. In such situation, the non-malicious nodes can be forced to connect only one hop away from malicious nodes. After leaving all malicious nodes, the attack insures the complete partition of the overlay. The complete partition occurs only in the case of 100% direct attack of malicious nodes, which is equivalent to Hub attack. The HnS attack is able to protect malicious peer to being paralysed, which is shown in the Figure 1. Here, S-Gossip captures only few malicious nodes in HnS while more in Hub. It indicates that the propose HnS attack can miss-lead non-malicious nodes regarding malicious nodes.

4. Heal Gossip

HealGossip assists non-malicious nodes to identify and prevent malicious nodes from being involved in gossip with non-malicious node in the network. The in-formation about suspicious nodes get disseminated in normal course of gossip. This process heals the overlay before affected from malicious nodes.

Having described the main idea behind HealGossip in the following sub-sections, first we discuss the sharing and the dissemination information among nodes in the network through gossip. Next we introduce the concept to update of different node tables. Finally, we discuss the rationale behind the rules used for table updates.

4.1. Gossip Modes

Unlike other gossip-based protocol, HealGossip uses two modes for gossip with neighbours. One mode is called General Gossip and the second is called Protect Gossip. While gossiping, HealGossip assumes that the General Gossip contains malicious or non-malicious nodes, and the Protect Gossip contains malicious nodes to inform about them to neighbours.

4.2. Table Update

In terms of HealGossip protocol, the common of nodes represent those nodes which appear in any of the three tables of the gossiping pair and the respective neighbours received at the two ends. The occurrence of a common node is con-sidered as a hit. The value of hits of the nodes in any table determines whether the node is suspicious, malicious or reliable.

Suppose, the initiator of gossip is denoted by A, the node selected for gossip is denoted by B, and C_i denotes a node received through the gossip. The descriptors are updated in tables while maintaining view. Maintain view is described in the Algorithm 1.

```
Algorithm 1: maintain view(ReceivedNodeList)
1 gossipMode = abstract from ReceivedNodeList
2 forever (gossipMode == General) do
3       if (node ∈ Tables then
4             update in tables;
5       end
6       else
7Insert in ViTab/S-Tab;
8       end
9 end
10 forever (gossipMode == Protect) do
11      if (node ∈ Tables then
12            update in tables;
13      end
14      else
15            Insert in S-Tab;
16      end
```

17 end

There are major four criterion which are explained below.

- If the gossiping node B belongs to G-Tab then our gossip protocol believes that information of gossiping node is reliable.

(i) If B reports that C_i non-malicious and C_i does not belong to any of the table then A includes C_i in its ViTab.

(ii) If B reports that C_i is malicious and C_i neither in A's S-Tab nor in M-Tab then A inserts C_i in its S-Tab.

- If the gossiping node B belongs to S-Tab, then it is viewed as a suspicious node. So, B's information about Ci is treated with suspicious.

(i) If B reports that is C_i non-malicious then A places C_i in its S-Tab

(ii) If B reports that C_i is malicious then A places C_i in its S-Tab.

When a hit occurs for the first time, the corresponding common node can only be in ViTab. In case, the number of hits is increased and then the node moves from one table to another depending whether the hit crosses the thresholds set for movement of the corresponding move. The thresholds are calculated at each gossiping cycles. The update process is mentioned in the Algorithm 2.

On the basis of previous knowledge of long stayed nodes, we can take an action on related behaviour newly arrived nodes within few cycles. Hence the identification time will be reduced for the new nodes and take action on group of nodes, instead of single node. It also helps to identify dead and live nodes, which results to reduce false +ve and false -ve. This concept is able to scale the cope to strengthening security in P2P.

Algorithm 2: update tables()

1 Decrement TTL of nodes in G-Tab, S-Tab and M-Tab

2 Calculate Mean and Standard Deviation of Hits and TTL for the tables

3 $\mu^{s}_{hits} \leftarrow \left(\mu^{cs}_{hits} + \mu^{ps}_{hits}\right)/2$

4 // Moving G-Tab Nodes towards S-Tab

5 if $(node.GTab.TTL < 0)$ then

6 delete GTab(node);

7 end

8 else

9 if $\left((node.GTab.Hits) > \left(\mu^{g}_{hits} + \sigma^{g}_{hits}\right)\right)$then

10 $node.GTab.Hits = \mu^{s}_{hits}$

11 $node.GTab.TTL = VL$

12 Insert STab(node);

13 delete GTab(node);

14 end

15 end

16 // Moving STab Nodes towards MTab

17 if (node:STab:TTL < 0) then

18 delete STab(node);

19 end

20 else

21 if $\left((node.STab.Hits) > \left(\mu^{s}_{hits} + \sigma^{s}_{hits}\right)\right)$ then

22 $node.STab.Hits = \mu^{s}_{hits}$

23 $node.STab.TTL = VL$

24 if $(node.STab \in ViTab)$ then

25 delete ViTab(node);

26 end

27 Insert MTab(node);

28 delete STab(node);

29 end

30 end

31 // Exiting nodes from MTab.

32 if (node.MTab.TTL < 0) then

33 delete MTab(node);

34 end

4.3. Storing Objects

The objects are hashed by respective source node to get an object ids (objIDs). Each object is assign a version id (verID) whenever they will be updated. The latest verId helps to update the old version of the object. The HealGossip ensures that the replication of an object can be done and they are up to dated by source node only. For this reason, a hopCount value increments by one after each gossip. The increment will be stop after reaching at source. The final value of hopCount considers as the maximum replication path of the object. The objIDs can be updated within its own hopCount only. For each object, the hopCount value will be di erent. The objID, verID and hopCount refer as object descriptors. These descriptors are used to manage objects in the overlay by each node.

4.4. Routing Process

The routing process starts whenever a request will arrive on an active node. The requested object id (objID) is checked in ViTab and response after finding the id. In absence of the key or objID, the node forwards to an random neighbour from its ViTab. A sample routing process where an object key, say 101, to be searched on a node, say N0. In this scenario, the node N0 does not contain the key 101. Hence, the request message will be forwarded to a random, say node N3, which selects from its ViTab. The process is stopped at a node, say N8, after nding the key. The node N8 reply the response message to the requested node. The path of request and response message are di erent with high probability.

The routing process is depends on the network connectivity. Healthy connected overlay can route efficiently. Hence, the work focus on the connectivity of the overlay instead of routing. The issue of connectivity based on the ViTab nodes, which refers to the neighbours of a node. The nodes are polluted when non-malicious nodes of ViTab will be replaced by malicious nodes. It is called ViTab or cache pollution. More percentage of cache pollution indicates more separation from good nodes. In presence of malicious nodes, the propose security mechanism tries to reduce percentage of cache pollution to make overlay connectivity as closer to healthy overlay. The comparative analysis is performed in the following section.

5. Results and Analysis

Results of the proposed HealGossip protocol are compared and analyzed with existing security protocol S-Gossip. For healthy comparison, both security protocols are run on Cyclon [9] gossip mechanism. Simulations have been performed with the help of Peersim simulator.

Analysis of Hub Attack Figure 2 describes the Hub attack affects on the security protocols. The affect is comparatively less in the proposed HealGossip due to additional acknowledgement property of malicious nodes. The average malicious nodes inside ViTab or cache are reduced. The percentage is also reduced in presence of 2% churn rate and the result shows in the Figure 3. In these simulations, there are 1,000 nodes in the network and 2% malicious nodes are taken.

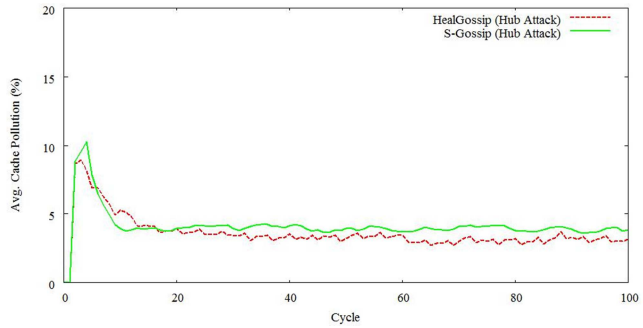

Fig. 2. *The Hub Attack with 2% malicious nodes where network size is 1000.*

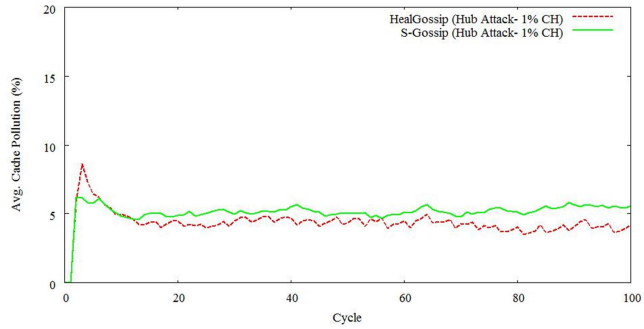

Fig. 3. *The Hub Attack with 2% malicious nodes and churn rate is 1%, where network size is 1000.*

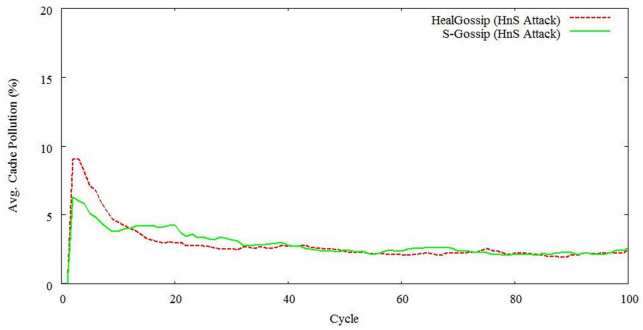

Fig. 4. *The hNs Attack with 20 malicious nodes where network size is 1000.*

Analysis of Hide and Seek Attack The effects on the cache shows in the Figure 4. Initial hype of HealGossip shows the affect of HnS indirect attack. Since there is no room for indirect attach, S-Gossip does not have such hype. But, the proposed gossip mechanism able to reduce the pollution after spreading the malicious messages among neighbours.

6. Conclusion

The proposed HealGossip enhances the gossiping mechanism for gossip-based protocols in unstructured P2P network. Each node has capability to capture or detect malicious nodes independently. They also share their malicious entries which helps them to detect more malicious nodes. Spreading the information of malicious node is very unique feature which is introduced first time in such type of networks. These features make HealGossip protocol very scalable in respect of providing security if the malicious nodes will change their behaviours in future.

References

[1] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan: Chord- A Scalable Peer-to-Peer Lookup Service for Internet Applications. IEEE/ACM Transactions on Networking, vol. 11, no. 1, pp. 17-32, (2003).

[2] A. Rowstron and P. Druschel: Pastry- Scalable, decentralized object location and routing for large-scale peer-to-peer systems. in Proc. of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pp. 329-350, (2001).

[3] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz: Handling churn in a DHT. in Proc. of the USENIX Annual Technical Conference. Berkeley, CA, USA: USENIX Association, pp. 10-23, (2004).

[4] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry: Epidemic algorithms for replicated database mainte-nance. in Proc. of the 6th ACM Symposium on Principles of Distributing Computing (PODC87), pp. 1-12, (1987).

[5] P. T. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massouli: Epidemic infor-mation dissemination in distributed systems. IEEE Computer, vol. 37, no. 5, pp. 60-67, (2004).

[6] M. Jelasity, A. Montresor, and O. Babaoglu: A modular paradigm for building self-organizing peer- to-peer applications. in Proc. of Engineering Self- Organising Systems. Springer, pp. 265-282, (2004).

[7] Mark Jelasity, Alberto Montresor, and Ozalp Babaoglu: The bootstrapping service. in Proc. Of the 26th IEEE International Conference Workshops on Distributed Computing Systems (IDCSW06). IEEE Computer Society, pp. 11-16, (2006).

[8] S. Voulgaris and M. van Steen, Epidemic-style management of semantic overlays for content-based searching. in Proc. of Euro-Par 2005 Parallel Processing, pp. 1143-1152, (2005).

[9] S. Voulgaris, D. Gavidia, and M. van Steen: Cyclon- Inexpensive membership man-agement for unstructured P2P overlays. Journal of Network and Systems Manage-ment, vol. 13, no. 2, pp. 197-217, (2005).

[10] Marin Bertier, Francois Bonnet, Anne-Marie Kermarrec, Vincent Leroy, Sathya Peri, Michel Raynal: D2HT- The Best of Both Worlds, Integrating RPS and DHT. European Dependable Computing Conference, pp. 135 (144, (2010).

[11] G. P. Jesi, A. Montresor and M. van Steen: A Secure Peer Sampling., Elsevier Journal, 54, pp. 2086-2098, (2010).

[12] G. P. Jesi, D. Hales, and M. van Steen: Identifying Malicious Peers Before its TooLate: A Decentralized Secure Peer Sampling Service. IEEE SASO, Boston, MA(USA), (2007).

[13] Sumit Kumar Tetarave, SomanathTripathy, SathyaPeri. S-Gossip: Security En-hanced Gossip Protocol for Unstructured P2P Networks, 11th International Con-ference on Distributed Computing and Internet Technology, Springer, Volume 8956, pp 288-298, (2015).

[14] Anceaume, Emmanuelle and Busnel, Yann and Gambs, Sebastien: Uniform and Ergodic Sampling in Unstructured Peer-to-Peer Systems with Malicious Nodes. Springer, ISBN: 978-3-642-17652-4, Tozeur, Tunisie, (2010).

[15] Bortnikov, Edward and Gurevich, Maxim and Keidar, Idit and Kliot, Gabriel and Shraer, Alexander: Brahms: byzantine resilient random membership sampling. Pro-ceedings of the twenty-seventh ACM symposium on Principles of distributed com-puting, Toronto, Canada, (2008).

[16] A. Montresor and M. Jelasity: PeerSim: A scalable P2P simulator., IEEE Ninth International Conference, pp. 99-100 (2009).