

A membership management protocol for mobile P2P networks

Mohamed Karim SBAI
Planete Project-team
INRIA, Sophia Antipolis, France
mksbai@sophia.inria.fr

Emna SALHI
Planete Project-team
INRIA, Sophia Antipolis, France
esalhi@sophia.inria.fr

Chadi BARAKAT
Planete Project-team
INRIA, Sophia Antipolis, France
cbarakat@sophia.inria.fr

ABSTRACT

MANETs are self-organizing networks composed of mobile wireless nodes with often scarce resources. Distributed applications based on the P2P paradigm are by nature good candidates to run over such networks. To profit from the service provided by a P2P overlay (e.g. file sharing using BitTorrent), a peer needs to be permanently informed about the other members of the overlay (e.g. other peers interested in the same file as currently provided by the BitTorrent central tracker). However, this P2P membership management is a costly and difficult task in a dynamic and resource limited environment as a MANET. We focus on this problem and we propose a robust, network friendly and decentralized membership management protocol allowing peer discovery and update. Compared to flooding, client-server or multicast based approaches, our protocol achieves significantly lower network overhead and less pollution of global view caused by peers who have left. Moreover, as network splits are very frequent in MANETs, our protocol is designed to be partition-aware. Namely, it allows separate overlays providing the same P2P service to efficiently merge together when direct communication opportunities occur. The efficiency of our solution is validated through extensive NS-2 simulations.

Categories and Subject Descriptors

D.2.2 [Network protocols]

General Terms

Algorithms, Management, Performance, Design, Experimentation.

Keywords

Membership management, P2P, mobile ad hoc networks

1. INTRODUCTION

The wide spread of mobile devices (Laptops, PDAs, Smartphones, etc) encourages users to connect directly to each other to build ad hoc communities. These devices, forming

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Mobility 2009, Sep 2-4, Nice, France Copyright © 2009
ACM 978-1-60558-536-9/00/0009.....\$5.00

spontaneous wireless multi-hop networks thanks to the use of ad hoc routing protocols, can support several services such as content sharing, multimedia streaming, instant messaging, chat conferencing, etc. The infrastructureless nature of mobile ad hoc networks (called MANETs for short) rules out the possibility of deploying services based on dedicated central servers. And even if a node volunteers to play the 'role of a central server, the global service will not scale and will suffer from bad performances due to interruptions caused by the mobility of nodes, network splits and bandwidth scarcity. Furthermore, MANET nodes are end users having, usually, modest resources. Hence, a single node cannot handle the global load of the service. A decentralized approach like the peer-to-peer one (P2P) is a good candidate solution to be adopted in such environments. Users of a P2P architecture, called peers, organize themselves in a collaborative overlay network by connecting to each other via logical links across the other nodes of the MANET. For example, a P2P content sharing application like BitTorrent 7 distributes the data-transfer load among all the peers interested in the same content. Peers who receive some content pieces are responsible of disseminating them to the rest of the P2P network.

Although P2P applications are designed to be completely decentralized, most of them rely on central servers in some of their functionalities. They generally use servers for discovering and updating the information on the members of their overlays. In fact, a P2P application needs to be permanently informed about the set of peers interested in the same service in order to adjust its overlay and account for the arrival and departure of peers. For instance, in BitTorrent 7, each peer is asked to contact periodically a central rendezvous server called Tracker to get up-to-date information about the members of the sharing session. Like this, the peer can choose the other peers with whom to exchange pieces of the content. In general, actual architectures like BitTorrent and instant messaging mainly focus on distributing the data plane but they keep the membership management plane centralized. By doing this, they keep the control on the service they provide while fully profiting from the advantages of the P2P semantic in distributing the load of the data plane.

The presence of a centralized component in some Internet P2P applications makes it difficult for them to run in MANETs. First, this raises the concern of the single point of failure of the service. In fact, unlike an Internet server, a MANET node has limited resources and cannot handle frequent solicitations. Subsequently, it becomes a bottleneck for the service and overwhelms the underlying network, known for its scarce and

¹ This work has been supported by the ITEA european project on Experience sharing in mobile P2P communities (Expeshare)

shared resources. Moreover, the mobility of nodes, the shadowing, the churn and the network splits can be major factors of interruption of the communication with the central server node. One can imagine the scenario where the network is partitioned into two completely disconnected parts. This means an interruption of the service in the part that does not contain the membership server.

In this work, we study the membership management issue in MANET and propose a standalone distributed membership management protocol that emulates the central server and answers the needs of a variety of P2P services when they are deployed in these networks. It allows P2P applications designed for the Internet to migrate to MANETs without any significant changes in their service overlays. They can use our protocol to construct and share common knowledge about their overlay members equivalent to what is provided by central servers in the Internet.

Our protocol overcomes the limitations of the central-server solution and takes into account the constraints of a MANET. It is distributed by relying on a collaborative approach: peers organize themselves in a shared tree dedicated for disseminating membership information. Events like new arrivals or departures of peers are announced on the tree so that each peer can keep permanently an up-to-date list of the members of the P2P service and of any related information. Using ad hoc routing information, peers construct and adapt their logical links in the membership tree according to the current topology of the network. Their goal is to minimize the length of the tree (in terms of number of wireless hops) so as to reduce the membership traffic and the overhead on the underlying network. The membership tree can be seen as a distributed minimum spanning tree connecting peers of the service. We propose fully decentralized mechanisms that allow peers to adapt the membership tree structure to the frequent changes of the underlying network caused by nodes' mobility. Moreover, our protocol addresses the partitioning issue in MANETs. We achieve this by the help of a new and simple mechanism that allows peers to benefit from the information provided by the underlying routing protocol for discovering peers from separate overlays providing the same service. This makes two or more separate trees for membership management belonging to the same service (e.g. same content in BitTorrent) merge together to form a new covering tree.

In the literature, many works have been conducted to implement P2P applications in MANETs. We refer to the related work section for a brief description of these implementations. The majority of them do not study independently the membership management issue but rather focus on the data plane. In fact, the cost of the membership management is often ignored compared to the cost of the data traffic. More importantly, these works do not provide a solution for the network splits in MANETs that are caused by nodes' mobility and the finite range of the wireless. If not handled correctly, these splits may lead to an interruption of the service and an inefficient use of resources.

To validate the benefits of our protocol compared to classical solutions, we add a module to the NS-2 network simulator ⁷ and conduct extensive simulations. The performance of a membership management solution can be measured in terms of the volume of signalling traffic it generates and the level of freshness of the knowledge about the members of the P2P service it allows. As there is a tradeoff between increasing the

freshness of membership information and diminishing the cost of the management, we define appropriate metrics to measure the efficiency of the compared solutions in both regards. The comparison of our protocol to client-server, flooding and multicast-based solutions shows that it achieves indeed lower network overhead while ensuring a better membership information freshness.

The remainder of this paper is organized as follows. Section 2 overviews the related work. Section 4 explains the design of our protocol and includes a detailed presentation of its algorithms. Section 5 evaluates the performance of our protocol compared to other solutions. Section 6 summarizes the paper and gives some ideas on our future work.

2.RELATED WORK

In this section, we overview the body of the literature relevant to our membership management problem. First, we describe the efforts done to manage the membership of P2P systems in the Internet. Then, we present some P2P overlays implemented in MANET. Finally, we study P2P multicast overlays in MANET for the purpose of underlining the similarities and the differences that exist between membership management and multicast.

2.1 Membership Management in the Internet

Many membership management techniques have been proposed for the Internet. They can be subdivided into two categories: those decoupling the P2P data plane from the membership management and those coupling them together. One can mention here the client-server architecture used by BitTorrent to track peers as a solution that decouples the two functionalities. In BitTorrent, each peer contacts periodically a central rendezvous server named Tracker in order to update its list of peers. In parallel and to distribute the server functionalities, mechanisms based on Distributed Hash Tables (DHTs) have been also introduced to supply P2P applications with membership information without relying on one single server. For example, P2PSIP⁷ organizes nodes into a structured DHT-based overlay where ordinary peers having abundant capacities can become servers. Ordinary peers locate servers by DHT lookup functions. DHT-based solutions are efficient in the Internet since the graph of communication is totally meshed and the bandwidth is abundant. In a MANET, these properties do not hold. The network may split into separate clusters and nodes serving as DHT servers remain the bottleneck.

Other P2P protocols do not consider quality of service criteria when constructing their overlays and so they use the same structure to do both peer discovery and data dissemination. Content-based routing P2P networks⁷ are examples of these techniques. In general, when quality of service is a concern, it is better to decouple the membership management from the data plane to allow for more efficient overlay construction.

Some other works (e.g. ⁷) address the scalability problem by deploying gossiping techniques. The solution proposed is to contact a random sub-set of peers and to exchange with them known information on other peers. This technique generates random graphs over which peers exchange their knowledge about the service overlay. It is an acceptable option when the knowledge of a sub-set of peers is sufficient for a good P2P service and when the communication between faraway nodes is not constrained by physical connectivity. Otherwise, the

overhead on the underlying network will be very important and the P2P application will suffer from bad performances.

2.2 P2P Overlays in MANET

In general, one can divide the design space of P2P overlays in MANETs into four subspaces: non-structured and layered design [7], non-structured and cross-layer design [7], structured and layered design [7] and structured and cross-layer design [7]. Cross-layer design approaches have been introduced because MANET nodes are both end-users and routers. This supposes that P2P applications operate jointly at the network and application layers. Structured approaches suppose that peers are organized following a structured virtual topology. For example, the structure can be a DHT allowing service lookup. Generally, the peers responsible of providing the service can be discovered through a routing in the DHT network. Unfortunately, these DHTs are difficult to adapt to the underlying MANET topology.

2.3 P2P Multicast Overlays in MANET

The problem of constructing a P2P multicast protocol for MANET has some common challenges with our problem. Indeed, multicast protocols require a membership management component to track the MANET nodes interested in the session. The problem is that multicast protocols often aim at optimizing the data transfer plane and neglect the signaling plane. The energy spent on signaling is compensated by the efficiency of the data plane itself. In our case, we only focus on the dissemination of the membership information itself, which could be seen as only having the control packets of a P2P multicast overlay without the data. Moreover, some existing multicast protocols are centralized or require global knowledge which we want to avoid [7]. We add to that the fact that there is no multicast-based solution for the problem of network splitting.

To optimize the data transfer plane, multicast protocols proposed for MANET were constructed following two approaches: protocols based on meshed overlays and protocols based on tree overlays. Meshed overlays are non-structured; they represent random graphs linking nodes of the network. This kind of overlays offers more connectivity and more robustness by maintaining redundant paths between nodes. Nevertheless, the meshed topology is not efficient in MANET due to the overhead caused by duplicated transmissions of packets on redundant paths. Unlike meshed overlays, the tree topologies are very efficient in the MANET environment as they result in low load on the network by avoiding path redundancies. But they are less robust and require specific mechanisms to adapt to the frequent changes in the physical network. Our protocol adapts a minimal-cost tree structure while making it adaptive and resilient to network splits. Here are two examples of overlay multicast protocols recently proposed for MANETs:

- **PAST-DM** stands for Progressively Adapted Sub-Tree in Dynamic Mesh [7]. Peers in PAST-DM first organize themselves in a mesh network and then each of them, knowing the topology of this mesh, computes in a centralized way a minimum spanning tree. Each peer discovers its neighbors in the meshed graph by broadcasting messages in a limited scope. This discovery is done periodically in order to adapt the mesh to the underlying topology. Neighbors in the mesh are linked through unicast tunnels in order to exchange link-state information allowing the computation of the spanning tree. When a peer leaves the overlay, the information on its departure propagates via the unicast tunnels until it reaches all the members of the overlay. This

periodic exchange of link-state information is costly and the computation method is suboptimal since it must be done periodically by each node.

- **MOST** (Multicast Overlay Spanning Tree Protocol) [7] is an overlay multicast protocol based on the construction of a minimum spanning tree. It requires imperatively the use of the OLSR routing protocol. In fact, each peer uses the topology information provided by OLSR in order to compute an optimal spanning tree. This tree is recomputed periodically to adapt to the changes in the topology and in the members of the overlay. Peers flood periodically JOIN messages including the addresses of the multicast groups they belong to. Each peer then maintains a list of peers per multicast group. If it does not receive any JOIN message from one of the peers during a specific period of time, it deletes it from the list of peers of its multicast groups. Here also the cost of flooding JOIN messages periodically is very important and the solution requires the use of OLSR as the underlying routing protocol.

3. GRAPH THEORY CONCEPTS

In this section, we present some graph theory concepts that we use in the design of our membership management protocol. Let $G(V,E)$ be a graph. V and E are respectively the set of vertices and the set of edges of the graph.

- A *cycle* is a subset of edges that forms a path such that the first node of the path corresponds to the last one.
- A *cut* is a partition of the vertices of the graph into two disjoint sets S and T . Any edge $e(u, v)$ in E with u in S and v in T is a cut edge.
- A tree is a graph in which any two vertices are connected by exactly one path.
- Given a connected undirected graph, a *spanning tree* of that graph is a subgraph which is a tree and which connects all the vertices together. A single graph can have many different spanning trees.
- One can assign a weight to each edge of a graph. The weight of a spanning tree can be computed as the sum of the weights of the edges in that spanning tree. A *minimum spanning tree* is then a spanning tree whose weight is less or equal than the weight of every other spanning tree. More generally, any undirected graph (not necessarily connected) has a minimum spanning forest, which is a union of the minimum spanning trees for its connected components.

The following properties of a minimum spanning tree have been profitable for the development of our protocol:

- **Cycle property:** For any cycle C in the graph, if the weight of an edge e of C is larger than the weights of other edges of C , then this edge e cannot belong to a minimum spanning tree.
- **Cut property:** For any cut C in the graph, if the weight of an edge e of C is smaller than the weights of other edges of C , then this edge e belongs to all minimum spanning trees of the graph.

4. THE MEMBERSHIP MANAGEMENT PROTOCOL

In this section, we describe our protocol for P2P membership management in MANET. Our protocol constructs a spanning

tree to be used for the exchange of membership information among peers in the P2P network. We want this tree to match the topology of the underlying network in order to minimize the cost of the dissemination of membership information among peers and to ensure the freshness of the lists of members maintained by each peer. As optimality is needed, we propose to construct a minimum spanning tree in terms of number of hops, covering all peers of the underlying routing graph (some MANET nodes might not be P2P members). This guarantees a minimum cost for the membership information dissemination in terms of number of hops, transmissions and power. We introduce efficient and distributed mechanisms to track the intermittent connections and disconnections of the MANET nodes. Moreover, because nodes are continuously moving and so tree weights are subject to frequent changes, our protocol is able to restructure the tree when needed to maintain its optimality property. Centralized algorithms, as the well known Kruskal algorithm [7], are not appropriate in this context because they require global knowledge as each peer calculates its own spanning tree. Since the minimum spanning tree is not unique, peers might then calculate different spanning trees which disconnects the service overlay. Our protocol is based on a completely distributed approach which guarantees the uniqueness of the tree by making all decisions locally. Optimality is ensured by satisfying the cycle and cut properties in Section 3.

Another important problem that we consider in the construction of our adaptive tree is network partitioning. We add to our protocol a specific technique to merge separate trees belonging to the same P2P network when communication opportunities occur. The following paragraphs describe our protocol.

Table 1. Packet fields description

Field name	Field description
SRC_ID	Identifier of the peer sending the message
DEST_ID	Identifier of the peer that will handle the message
TYPE	Type of the message (e.g. HELLO)
OVERLAY_ID	Identifier of the service overlay
SEQ_NUMBER	Sequence number of the message.
PEER_TREE	A string representing the tree
CROSSED_NODES	Nodes tagged for partitioning awareness.
NEW	Identifier of the node joining the overlay.
ADD	List of logical links to be added to the tree.
DEL	List of logical links to be removed from the tree.
COST	Weight of the most costly logical link in a cycle.

4.1 Joining the Membership Tree

We suppose that each P2P service has a unique identifier. Knowing this identifier, a node that becomes interested in the service initiates a join procedure. This procedure can be divided into two phases: discovering the nearest peer and disseminating the new arrival information to all other peers. This can trigger the update of the membership tree to maintain its optimality.

4.1.1 Discovering the nearest peer:

In order to discover a first attachment point to the membership tree, we propose to use a simple flooding technique with controlled scope. The new member floods a HELLO message in its one hop neighborhood (TTL=1) and then waits for a HELLO REPLY message, sent in unicast, from any member of the P2P network located at one hop. In case there is no answer, the new member increments exponentially the value of the TTL of the HELLO message and waits again for at least one HELLO REPLY. If the maximum TTL is reached and no answer is received, the

node considers that the service is not provided by the network and that it is up to it to construct a new membership tree. If an answer is received, the new member gets in the HELLO REPLY message a copy of the current tree with the list of members and other useful information as the canonical names and descriptions. We underline that in our method a peer does not need to know the cost of the edges of the tree, it only knows which peer is connected to which other peer. The format of HELLO and HELLO REPLY is depicted in Table 1 and Figure 1.

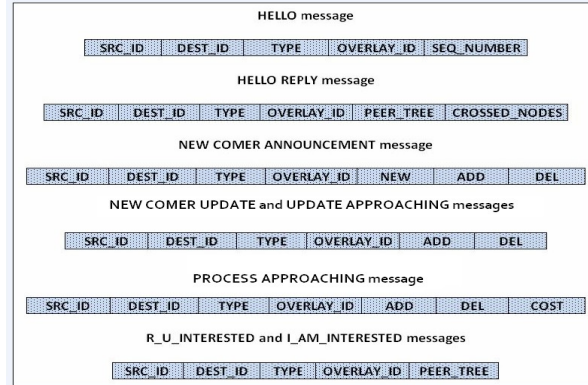


Figure 1. Packet format

Using its current routing table, the new arriving peer compares the costs in number of hops to other peers in the tree. This comparison allows it to identify the closest peer to it. The new peer should then connect to the spanning tree by attaching itself to this closest peer as required by the cut property described in Section 3. This property requires that the connection to add is the one having the lowest cost in the cut formed on one side by the old tree and on the other side by the new peer. In practice, after identifying this closest peer, the new arriving peer sends in unicast a simple CONNECT ME message to it. Receiving this message, the nearest peer triggers a new arrival information dissemination phase on the old tree. This phase is coupled with a phase of adaptation of the tree, to be described next.

4.1.2 New arrival information dissemination and tree adaptation

When a member of the tree receives a CONNECT ME message from a newly joining peer, it adds this peer as a child node. This modification of the tree is then disseminated to the other peers to trigger any necessary modification that keeps the tree optimal. The new parent sends to all its neighbors in the tree, except the newly arriving one, a NEW COMER ANNOUNCEMENT message containing the identifier of the new peer and the modifications it has made on the tree. We refer to Figure 1 and Table I for details on this message. Every peer that receives the NEW COMER ANNOUNCEMENT message updates its knowledge about the tree and verifies whether it can modify some of its logical links to improve its connectivity to the tree next to this new arrival. This modification is described later in this section. After making these local decisions, it informs its neighbors in the tree, except the peer which has sent the NEW COMER ANNOUNCEMENT message. It does that by sending a new version of the NEW COMER ANNOUNCEMENT message adding, eventually, its own changes. Upstream peers that have already seen the NEW COMER ANNOUNCEMENT message are informed by the modifications through a NEW COMER UPDATE message, which contains the logical links that the peer has added or removed from the tree. The NEW COMER UPDATE message

differs from the NEW COMER ANNOUNCEMENT message by the fact that it does not trigger any modification of the tree. A peer receiving this former message just updates its knowledge about the tree. In this way, all peers are aware of the new arrival and the tree is restructured in parallel. The decision that a peer must make when it receives a NEW COMER ANNOUNCEMENT message is based on a simple verification of the cycle property described in Section 3. The cycle to consider is the one formed by the logical links on the path of the current tree starting from the intermediate peer making the decision to the newly joining peer, and by adding the direct logical link between both peers. If any optimization is possible, it will result in cutting the logical link through which the peer receives the NEW COMER ANNOUNCEMENT message and adding the logical link to the newly joining peer. This way the cost of the tree is always kept minimal. One can notice that all the decisions are made locally and in a distributed manner without compromising global optimality.

4.2 Adapting the membership tree to mobility

Due to the mobility of MANET nodes, the distances between the peers of the membership tree vary in time. If the spanning tree is not adapted to these movements, it will quickly lose its optimality property. One can distinguish four possible movements of peers:

- Two peers that are neighbors in the tree can get closer to each other. In this case, the cost of the spanning tree becomes smaller but it remains a minimum spanning tree. This movement has no impact on the structure of the tree.
- Two peers that are not neighbors in the spanning tree get farther from each other. In this case, the cost of the tree does not change and there is no decision to be taken.
- Two peers that are neighbors in the spanning tree get farther from each other. The cost of the current tree increases which means there might exist a better tree to be identified.
- Two peers that are not neighbors in the spanning tree get closer to each other. In this case, the cost of the current spanning tree does not change but there might be another spanning tree with a smaller weight. This movement requires, eventually, an adaptation of the spanning tree seeking for the existence of an optimal one.

In the following, we describe how we adapt the membership tree in response to the two latter movements impacting its optimality.

4.2.1 Two neighbors in the tree get farther

If two peers that are neighbors in the spanning tree get farther from each other due to mobility, this leads to two possible situations. The first situation is that one of these peers or maybe both will get closer to other peers of the tree. Here, the tree adaptation can be done by applying the approaching adaptation procedure which we describe in 4.2.2. The second situation is that no one of these two peers gets nearer to other peers, in this case no better spanning tree can be found and no adaptation of the tree is needed. Hence, the problem raised by neighbors getting farther from each other can be transformed into a simple approaching problem and solved by the solution we come up for the latter one.

4.2.2 Two peers that are not neighbors get closer

Let $P1$ and $P2$ be two peers that are not neighbors in the spanning tree. Suppose that the cost of the physical direct path between these two peers becomes smaller due to the mobility of nodes. Take the cycle formed by the tree path from $P1$ to $P2$, and by adding the network link that connects directly $P1$ to $P2$. If there is a logical link L in this cycle such that $cost(L) > cost(P1,P2)$, the cycle property described in Section 3 indicates that the logical link $(P1,P2)$ must belong to the minimum spanning tree. So the actual tree should be adapted in such a way to replace the logical link L by the logical link $(P1,P2)$. Each peer in the tree tracks continuously other peers and verifies if another peer, who is not its neighbor in the current tree, becomes closer than its farthest logical neighbor. Here, it forms the cycle between it and this peer and initiates a procedure of identification of the most costly link in the cycle. Between the two peers, the one having the lowest identifier initiates the procedure. This is done by circulating a PROCESS APPROACHING message on all logical links of the cycle. Each peer in the cycle adds its logical link to the next peer in the cycle as being the link to be removed (DEL field in the message) if this link is more costly than the link it finds in the message. It also updates the field COST in the message because peers only know the costs of their own logical links. This procedure is repeated until the message returns to its original sender which then decides on the link to be removed and the one to be added. This modification is then disseminated to all peers. The peer sends an UPDATE APPROACHING message to its neighbors in the tree which forward it to their neighbors and so on. All the above messages are described in Figure 1 and Table 1.

4.3 Leaving the Membership Tree

Adapting the membership tree after the departure of peers is very important for the efficiency and the uniqueness of the tree and for the freshness of the membership information. We ask every peer that decides to leave the P2P service to inform its logical neighbors in the spanning tree by sending to them an I AM LEAVING message to them. Except for leaves, this departure will result in the decomposition of the tree into two separated sub-trees. The first sub-tree represents the children of the leaving peer and the second one its parents. To reconnect the tree, the child of the leaving peer having the highest identifier connects to its parent and becomes the parent for the remaining children. This way, a new spanning tree is formed. The problem is that this new tree may not be optimal. The optimal is reached by having the peers apply the normal approaching adaptation procedure described earlier in paragraph 4.2.2. Note how this procedure is important for our solution to always rewire the tree in a way to ensure its optimality. All modifications made on the old tree are disseminated to all peers of the tree together with the identifier of the departing peer. Sometimes a peer can leave the service overlay improperly; in this case, it is the duty of the first neighbor detecting this departure to trigger the tree adaptation procedure.

4.4 Network Split Awareness

Due to the high dynamicity and mobility of MANETs, the network can split into different disconnected clusters. These clusters can merge again into one or more larger clusters. So one can imagine the scenario where one P2P network is split into two or more networks because of the underlying network splitting. Another scenario is that two or more membership trees constructed separately in different clusters but belonging to the same P2P network meet together. At the first merging

opportunity, it is very important to connect together the different partitions of a membership tree in one large and efficient tree. That is why; we add to our membership protocol a mechanism allowing peers to discover other peers coming from other partitions when they get close to each other.

After connecting to the current minimum spanning tree, each peer observes its routing table and tags network nodes that are not interested in the same service. Then, using its routing table, it tracks continuously the appearance of non tagged nodes in its neighborhood. We choose the neighborhood of a node in the P2P network to be equal to the maximum number of hops to one of its direct neighbors on the spanning tree. A new node not tagged and not belonging to the same membership tree is a good candidate to be asked whether it belongs to the same service but comes from another cluster. The peer sends to this newly detected node a message R U INTERESTED in order to ask whether it is interested in the service. If it receives no answer from that node then it tags it as a not interested node. The tagging information is disseminated to all the peers in order to reduce the number of R U INTERESTED messages. When a node receives an R U INTERESTED message and if it is interested in the service, it answers the source by sending an I AM INTERESTED message. In this case, the two trees maintained by the two peers need to be merged together. To ensure efficient merging, the peers of the smallest tree apply the join procedure described earlier in order to connect to the biggest tree.

5.PERFORMANCE EVALUATION

In this section, we study the performance of our protocol by comparing it to classical membership management approaches. The validation is based on extensive simulations run with our implementation of the different membership management methods in the NS-2 network simulator 7.

5.1Performance Metrics

Let P_t be the set of peers interested in the service at instant t and let P_t be the cardinality of this set of peers. When a peer is running a membership management approach, it maintains at instant t a set of peers N_t that corresponds to its view of the members of the P2P service. Let n_t be the cardinality of this set. Among these n_t peers, there are t_t peers belonging to P_t and f_t peers not belonging to it (e.g. due to peers which leave). During a specific measurement time (namely the simulation time for us here), peers exchange messages between them in order to discover the interested peers and update their knowledge about them. Let C be the cost in number of hops over paths crossed by the exchanged messages during a fixed period of time. The importance of this cost varies with the method used for membership management. However, this cost does not consider the freshness of information maintained by the peers. Thus, it is not enough to decide whether a method is appropriate or not. In fact, one can spend a very low cost and have a lot of pollution in its knowledge about the peers. That is why we propose another cost metric C_f that accounts for the freshness of information. This cost is also a global metric computed during a fixed time. After each T_s seconds, one takes a snapshot of the P2P network and measures the value of p_t and computes \hat{n}_t , \hat{t}_t and \hat{f}_t , the average values of n_t , t_t and f_t over peers interested in the P2P application. At the end of the simulation time, one computes \tilde{p} , \tilde{n} , \tilde{t} and the average values of p_t , \hat{n}_t , \hat{t}_t and \hat{f}_t over all measured samples. The cost corrected by freshness of information C_f is

the ordinary cost to which we add two terms. The first term accounts for the cost that the P2P network should pay to discover the $\tilde{p} - \tilde{t}$ missing members. This term can be easily calculated considering that the members of the P2P application have paid $\frac{C}{\tilde{n}}$ to discover a peer. Hence, the term of lack of information cost will be equal to $\frac{C(\tilde{p} - \tilde{t})}{\tilde{n}}$. The second term to be added to the ordinary cost is a term accounting for the pollution existing in the knowledge of the peers. We consider that one pays the same cost to discover an interested peer or to remove an idle one. That is why we take this term equal to $\frac{C \cdot \tilde{f}}{\tilde{n}}$. The following formula computes the cost metric corrected by the freshness of information:

$$C_f = C(1 + \frac{\tilde{p} - \tilde{t}}{\tilde{n}} + \frac{\tilde{f}}{\tilde{n}})$$

5.2Scenario Description

To conduct our simulations, we consider a MANET composed of 50 nodes moving inside a bounded area of width 100m and length 500m following the Random Waypoint mobility model. The speed and the pause time of each node are taken equal to 2m/s and 30s respectively. The nodes connect to each other using the 802.11 MAC layer with the RTS/CTS-Data/ACK mechanism enabled. The range of transmission is fixed to 50m and the data rate is set to 1 Mb/s. For ad hoc routing, we use the proactive OLSR protocol 7. To simulate a dynamic membership, we suppose that a node has two states: the first state is an idle state where it is not interested in the P2P service; the second state is the one where it becomes interested in the P2P service. The membership of a node follows then an ON/OFF process until the end of the simulation. The durations of the ON and OFF states follow an exponential distribution of parameter λ and μ respectively. We define the density of the P2P overlay as the number of nodes interested in the P2P network divided by the total number of MANET nodes. One can easily show that this density is on average $1/(1 + \lambda/\mu)$. When not stated; the density is taken equal to 50 % by assuming that both λ and μ are equal to 500s. The simulation duration is set to 3600s. The sampling period T_s used to compute the corrected cost is chosen equal to 10s.

5.3Comparative Study

We compare, through extensive simulations, our protocol to four classical methods for membership management: a client/server method, a flooding-based method, a multicast-based method and a non-adaptive tree method.

- **Client/server method:** The classical client/server method supposes that peers contact periodically a server to update their knowledge about the members of the P2P application. In our simulations, a random node plays the role of the server. Figure 2. Real and corrected cost Vs. Period plots the real cost in number of hops-messages as a function of the period at which the peers contact the server. It shows that this cost is proportional to the inverse of the contact period. The same figure also plots the cost corrected by the freshness of the membership information. One can notice that this cost is higher than the ordinary cost. Contrary to the real cost, it does not continuously diminish when the period of contacting

the server increases. In fact, the freshness of information decreases with the increase in the contact period, which is accounted for in our corrected cost metric. In the following simulations, the contact period is set to 400s which according to the figure yields the best performances for client/server method.

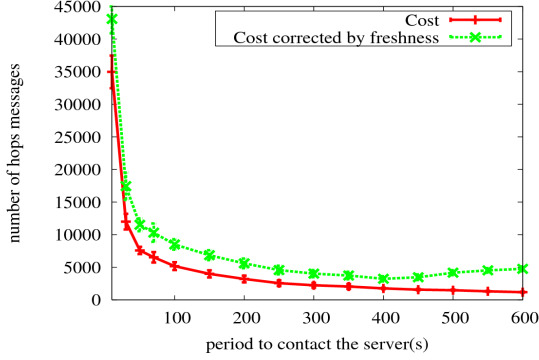


Figure 2. Real and corrected cost Vs. Period

- Flooding-based method:** Peers advertise their arrivals and their departures to other interested peers by physically flooding the network. Two types of sequenced messages are used for this purpose: peer-joining and peer-leaving messages. When receiving such a message, each node in the MANET forwards it to its physical neighbors if it is seen for the first time, otherwise it is discarded. During the broadcast, if a MANET node is interested in the P2P service, it updates its membership information.
- Multicast-based method (PAST-DM):** To compare with multicast, we use the PAST-DM protocol known for its efficiency in MANET. We refer to the related work section for a detailed description of this protocol and of its membership management mechanism. In our simulations, we implement for PAST-DM the exchange of link state messages, the JOIN/LEAVE messages and the messages to discover peers in the near neighborhood. The period to exchange the link state tables is set to 30s in order to match the value of the pause time for nodes' mobility.
- Non-adaptive tree method:** This method is very similar to our membership management protocol. It implements the same algorithms but it does not adapt the constructed spanning tree to the topology and dynamicity of the network. In fact, a joining peer does not connect itself to the nearest peer but to the first responding peer and the constructed tree is not adapted to the topological changes of the underlying network. Hence, the constructed tree is a sub-optimal spanning tree. Our aim is to prove the need for topology awareness and the gain obtained from it.

We begin our comparison by analyzing the impact of the overlay density on the real cost of the membership management. The results for the four methods are presented in Figure 3. They show that the cost of the flooding-based method increases linearly with the number of interested nodes in the network and is quite high. This is due to the fact that information about arrivals and departures of peers is flooded in the entire network, creating a large number of redundant messages. In contrast, the cost of our protocol increases slowly with the overlay density while staying low. One can explain this behavior by the fact that the expanded-range technique used by our protocol for

discovering peers guarantees a low cost in dense overlays. Moreover, update messages circulate along shortest paths of the minimum spanning tree without generating redundant messages. Although PAST-DM implements a controlled flooding technique to overcome the limitations of classical flooding, periodic updates increase dramatically its membership management cost as the overlay density increases. Finally, unlike our protocol, the non-adaptive tree method does not scale when the P2P network grows because of the sub-optimality of the weights of the tree branches.

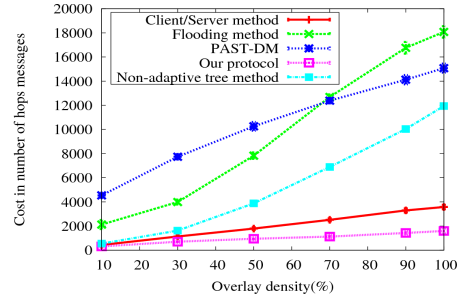


Figure 3. Real cost Vs. Overlay density

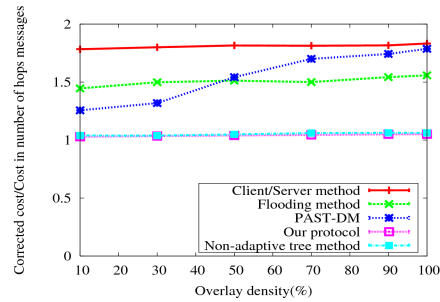


Figure 4. Ratio of corrected cost to cost Vs. Overlay density

As a second step, we examine the freshness of membership information of the different methods. We plot in Figure 4 the ratio of the corrected cost to the ordinary cost as a function of the overlay density. The higher this ratio is, the more out of date the peers are. A ratio equals to 1 means that peers have correct knowledge of the P2P service members over time. One can notice that our protocol and the non-adaptive tree method achieve a ratio value very close to 1. In fact, in these two methods, triggered updates and the tagging technique allow immediate information dissemination among peers. As expected, the client/server mechanism achieves a quite high ratio, even in sparse overlays. This confirms the idea that the client/server method does not scale in wireless environments. Concerning PAST-DM, update information is gradually propagated in the network through iterative exchanges between peers. Hence in dense overlays, where neighbors are physically close to each other, information needs several periods to reach all the peers. This explains the high cost ratio and the increasing trend seen in Figure 4. In Figure 5, we plot the cost corrected by freshness of the membership information as a function of the overlay density. By comparing the different methods in regard of this metric, one can decide which one is better than the others in terms of both network overhead and freshness of information. The figure shows that our protocol outperforms the other methods as it achieves the lowest network overhead while keeping a very high level of freshness of the membership information. Unlike our

protocol, the non-adaptive tree method has good freshness of information but pays a much higher cost for the overlay construction as we have seen in Figure 3 and Figure 5. The flooding-based method and PAST-DM achieve higher corrected costs as they have both higher network overhead and bad freshness of information.

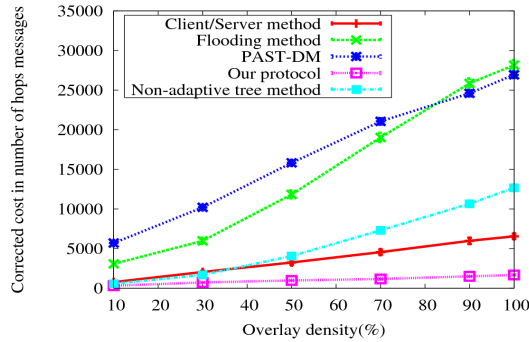


Figure 5. Corrected cost Vs. Overlay density

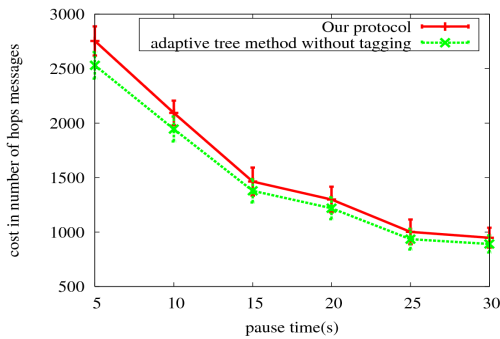


Figure 6. Split awareness: Cost Vs Pause time

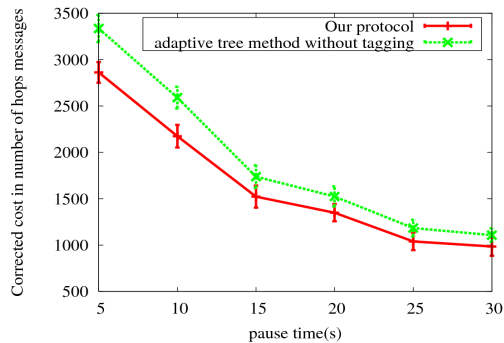


Figure 7. Split awareness: Corrected cost Vs Pause time

The last set of simulations aims to study the efficiency of our solution for overlay splits versus the frequency of topology changes by varying the pause time of nodes from 5 to 30s. A low value of pause time means frequent topology changes and more probable network splits. We evaluate the capacity of our protocol in handling network splits by simulating it in two modes: a mode that enables the splits' awareness mechanism and a second mode that disables it. shows that the extra cost for handling network splits is relatively small even for low values of the pause time. However, Figure 7 shows that the corrected cost of the split-unaware variant of the protocol becomes higher than

the corrected cost of the split-aware variant thanks to a better freshness of information. Hence, we conclude that our protocol provides an efficient and low-cost solution for MANET partitioning problem.

6. Conclusions

P2P membership management is a hard and costly task in MANET. In this work, we propose a scalable, robust and network friendly protocol to construct an adaptive topology-aware tree allowing peers to discover each other and to keep themselves informed about the arrivals and departures of other peers. The proposed protocol is a standalone service that can be used by any application requiring the sharing of up-to-date information among a group of users. Moreover, our protocol minimizes the number of exchanged messages and copes with node mobility and network partitioning, which makes it very useful for applications to know where peers are located in the network and how far they are from each other. The simulations show that our protocol outperforms classical solutions in terms of network load and freshness of information. The future work will be on the integration of this protocol within P2P applications as, for example, the trackerless BitTorrent. Our aim is to study the gain in performance that one can obtain by decoupling the construction of the membership management from the data plane.

7. REFERENCES

- [1] BitTorrent, <http://wiki.theory.org/BitTorrentSpecification>.
- [2] NS: The Network Simulator, <http://www.isi.edu/nsnam/ns/>.
- [3] Kruskal, http://en.wikipedia.org/wiki/Kruskal's_algorithm.
- [4] P.Jacquet, P. Mühlethaler, T Clausen, A. Laouiti, A. Qayyum and L. Viennot, Optimized Link State Protocol for Ad Hoc Networks., IEEE INMIC, 2001.
- [5] Gnutella, <http://dss.clip2.com/GnutellaProtocol04.pdf>
- [6] S.Ratnasamy, P.Francis, M.Handley, R. Karp and S.shenker, A scalable content-addressable networks, SIGCOMM, 2001.
- [7] I.Stoica, R.Morris, D.Karger, M.F.Kaashoek and H.Balakrishnan, Chord: a scalable P2P lookup service for internet applications, SIGCOMM, 2001.
- [8] C.Gui,P.Mohapatra,Efficient Overlay Multicast, WCNC, 2003.
- [9] G.Rodolakis, A.Naimi and A.Laouiti, Multicast Overlay Spanning Tree Protocol for Ad Hoc Networks, WWIC, 2007
- [10] L.B.Oliviera, I.G.Siqueira and A.A.Loureiro, Evaluation of ad hoc routing protocols under a P2P application, WCNC, 2003.
- [11] A.Klemm, C.Lindermann, O.Waldhorst, A special-purpose P2P file sharing system for MANETs, VTC, 2003.
- [12] S.M.Das, H.Pucha and Y.C.Hu, Ekta: an efficient P2P substrate for distributed applications in MANETs, 2004.
- [13] P2PSIP, <http://www.p2psip.org/>.
- [14] S.Boyd, A.Ghosh, B.Prabhakar, D.Shah, Gossip Algorithms: Design, Analysis, and Applications, INFOCOM, 2005