



HAL
open science

Adaptive overlay for P2P membership management in MANET

Mohamed Karim Sbai, Emna Salhi, Chadi Barakat

► **To cite this version:**

Mohamed Karim Sbai, Emna Salhi, Chadi Barakat. Adaptive overlay for P2P membership management in MANET. [Research Report] 2008. inria-00342691v1

HAL Id: inria-00342691

<https://inria.hal.science/inria-00342691v1>

Submitted on 28 Nov 2008 (v1), last revised 20 Jan 2009 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive overlay for P2P membership management in MANET

Mohamed Karim Sbai
EPI Planete, INRIA, France
Email: mksbai@sophia.inria.fr

Emna Salhi
EPI Planete, INRIA, France
ENSI, Tunisia
Email: esalhi@sophia.inria.fr

Chadi Barakat
EPI Planete, INRIA, France
Email: 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 the best candidates to run over such networks. To profit from the service provided by a P2P overlay (e.g. file sharing), a node needs to be permanently informed about its members (e.g. other peers interested in the same file). This P2P membership management is a costly and difficult task in such dynamic and resource limited environment. We focus on this problem and we propose robust, network-friendly and decentralized algorithms to construct and maintain adaptive topology-aware overlays allowing peer discovery and update. Compared to classical flooding or client-server based approaches, our membership protocol achieves significantly lower network overhead and lower pollution of caches caused by peers who have left. Moreover, and as network split is very frequent in MANETs, our protocol is designed to be partition-aware. It mainly allows separate overlays providing the same service to efficiently merge together when communication opportunities occur. The efficiency of our solution is validated through extensive NS-2 simulations.

I. INTRODUCTION

A Mobile Ad-hoc NETWORK (MANET) is a wireless network that continually re-organizes itself as a function of its environment conditions and nodes' mobility without requiring any pre-existing infrastructure. It consists of a set of mobile devices that communicate, collaborate and interact in order to complete an assigned mission. The main challenge of a MANET is to provide wireless multi-hop end-to-end connectivity. Nodes have to communicate using bandwidth limited wireless links, with potential intermittent connectivity, as compared to stable wired links and infrastructures. Since the nodes are mobile, the network topology may change rapidly and unpredictably, and the connectivity among the terminals may vary with time. Due to this dynamic topology, the network can split into many disconnected partitions making some end-to-end communications impossible. Thanks to the recent technological developments, new wireless communication devices are able to offer limited dimensions, user friendly interfaces, long life batteries as well as computational power comparable to that of some desktop computers. This is encouraging the development of new sophisticated services for MANET such as content sharing or video streaming applications. These services should be completely decentralized as a MANET is comprised of nodes having symmetric roles without any dedicated ones. And

even if any node decides to offer a service, because of its mobility and network partitioning this centralized service will not scale and may even stop. Therefore, the designers of applications for such networks must propose architectures for their services that are scalable, efficient and that adapt to the mobility of nodes and the scarcity of resources. Another important design factor to be considered is the risk of network partitioning that might be caused by the high dynamicity of the network and the limited range of the wireless.

It is clear that the classical client-server architecture is not convenient for the MANET environment for the reasons aforementioned. The P2P architecture, largely deployed in the Internet, seems to be the best candidate for these networks. In fact, there are many similarities between the goals and the principles of P2P applications and MANETs since they are both based on the P2P paradigm. The first ones apply the paradigm at the service level and the second ones at the network level. A P2P overlay is namely a community of users (peers) that have agreed on a specific goal. For example, in BitTorrent [1], a well-known P2P file sharing protocol for the Internet, a torrent is a set of peers cooperating together for the download of the same file. The communication pattern among peers in an overlay (who connects to whom?) is generally specified by the P2P application mechanisms. For instance, in BitTorrent, peers periodically choose to exchange data with a subset of peers of the overlay based on the history of bitrates of exchanges. But before constructing the service overlay, a peer needs to know about the other members or peers without the need to rely on a client-server approach. First, a peer discovery phase is necessary to connect him to existing peers. In addition, as some peers can join or leave the overlay later, the other peers in the overlay should learn about these changes during the session lifetime. This urges for a membership management protocol to ensure the continuity and the efficiency of the service provided by the overlay. It consists in keeping peers informed about the members of the overlay, namely it announces the arrival of new peers and the departure of those that have left the overlay. More importantly, it should adapt to any partitioning of the overlay caused by network partitioning and be able to reunify the overlay when the partitions are again in contact with each other.

In the literature, many works have been conducted to implement P2P applications in MANETs e.g. [10] [11] [12]. We refer to the related work section for a short description of these implementations. The majority of them does not decouple the membership management from the construction of the service overlay itself but focuses more on the optimization of the global traffic generated by the service. The cost of the membership management part is often ignored compared to the overhead caused by the service traffic itself. The impact of network partitioning due to node's mobility has not been considered as well. This latter factor in particular is of major importance since it may split an overlay into multiple ones that exist in parallel and that do not know how to form again one single overlay. There is a strong need for a distributed and independent solution that allows peers to localize and update each other without being dependent on the application they run. Once this knowledge is built, peers can use it to agree on a way to communicate as a function of what they are willing to share. We dedicate this work to this problem of P2P membership management and we study it independently of the communication pattern used in the service overlay. Our contribution is a new standalone membership protocol convenient to MANETs as it is designed to be aware of the constraints and the features of such networks. By being standalone, our protocol can be used by different P2P applications as file sharing or video streaming. It provides an adaptive and efficient overlay for membership management to be used in peer discovery and update. This dedicated overlay only serves for exchanging information on peers (arrivals, departures, etc) and is not used for the content itself. We choose to structure this membership overlay in a topology-aware fashion that reduces the generated traffic and the overhead on the underlying network. This is done via the calculation and the maintenance of an optimal covering tree in terms of the number of hops connecting all peers of the P2P network. No central node is responsible of computing the optimal tree, the decision is made in a scalable and distributed way by all peers.

When a new peer joins, it applies a simple join procedure to the membership management overlay, gets information on other peers, and lets the tree restructure itself in a distributed manner to account for its arrival. And when the distance between peers change due to the mobility of nodes, the tree is also restructured in a distributed way to stay efficient in terms of total number of hops and load on the underlying network. This also applies when a peer leaves the overlay. The objective is always to keep the tree connected and having a minimal cost.

Another important factor that we consider in our protocol is the frequent network splits that occur in MANET. Our protocol is designed to be partition-aware. To this end, we add a mechanism to discover nodes coming from separate overlays when there is a new communication opportunity. This allows two or more separate overlays for membership

management belonging to the same service (e.g. same content in BitTorrent) to merge together efficiently forming a new covering one.

To validate the efficiency of our membership management protocol, we add a module to the NS-2 network simulator [2] and conduct extensive simulations. The performance of a membership management solution can be measured in terms of the volume of traffic it generates and the level of freshness of the knowledge about the membership it allows. We compare our protocol to classical approaches namely the client-server approach and the flooding approach. Our results prove that our adaptive overlay achieves significantly lower network overhead than these methods and ensures lower pollution of caches by peers who leave the network.

The remaining part of this paper is organized as follows. Section II overviews the related work of our paper. Section III gives a short description of the graph theory concepts on which we base our work. Section IV explains the design of our adaptive membership overlay and includes a detailed presentation of our algorithms. Section V presents the simulation results. Section VI summarizes the paper and gives some ideas on our future work.

II. 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.

A. 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 membership management and those coupling them together. One can mention 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. This solution is very efficient in the Internet since the graph of communications is totally meshed and the bandwidth is abundant so that contacting the tracker does not steal bandwidth from other peers. Other P2P protocols do not take into consideration any quality of service criteria when constructing their overlays and so they use the same structure to do both peer discovery and data transmission (e.g. file sharing). Content-based routing P2P networks [4] 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 overlay construction like in BitTorrent.

Some other works [13] address the scalability problem inherent to membership management by deploying gossiping techniques. The solution that they propose is to contact a random sub-set of the peers and then to exchange some references to other peers with them. This technique generates random graphs over which peers exchange their knowledge about the overlay. It is an acceptable option when the knowledge of a sub-set of peers provides good quality of service and when the network is completely meshed like in the Internet. Otherwise the overhead on the underlying network will be very important and the P2P application will suffer from bad performance.

B. P2P overlays in MANET

Both structured and non-structured overlays have been implemented in MANET. Since nodes are both end-users and routers, some cross-layer design approaches have been introduced. These approaches suppose that P2P applications operate both at the network layer and at the application layer. In a structured overlay, 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 topology. One can divide the design space of P2P overlays in MANET into four subspaces: Non-structured and layered design [10], non-structured and cross-layer design [11], structured and layered design [12] and structured and cross-layer design [12].

C. P2P multicast overlays in MANET

The problem of constructing a P2P multicast overlay for MANET has some common challenges with the problem of constructing a P2P membership management overlay. Indeed, the management overlay can be seen as a tree to broadcast information on arrivals and departures of peers, which could be done using multicast. 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 need an overlay to disseminate membership information which is equivalent to only having the control packets of a P2P multicast overlay without the data. This is clearly suboptimal since in our case the optimization should target the control plane itself. Moreover some of the multicast protocols existing in the literature are centralized or require global knowledge which we want to avoid. There is also 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 networks. They represent random graphs linking peers of the network. This kind of overlays offers more connectivity and more robustness by maintaining redundant paths between peers. 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 network topology. Creating redundant paths between peers of the overlay is useless in our case since a link-failure (at the overlay level) means that the node is no longer interested in the overlay and then a departure procedure is executed to establish a new link. In case of multicast, this can mean a degradation of wireless link conditions that urges for another path. That is why we adopt for the membership management tree-based overlay. Our main contribution is a solution to construct an optimal spanning tree for membership management in a distributed and efficient way (all peers are responsible of the update of the tree and take decisions according to their local observations.). Moreover, our tree is resilient to departures of nodes and network splits. Here are some examples of overlay multicast protocols recently proposed for MANET.

- **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. The periodic exchange of link-state information is very costly and the computation method is not optimal since it must be done periodically by each node.
- **TrAM** [8] is a source-based multicast protocol. Each peer that wants to join the overlay must execute a parent discovery phase by flooding a QueryParent message. The peers that receive the QueryParent message answers by sending ParentAdvertise packets. After sometime, the new peer collects many answers from peers already in the overlay. It then chooses the nearest peer as its parent node. Periodically, peers flood QueryParent messages in order to discover the best possible parents and as a consequence the quality of the tree is ameliorated. When a peer wants to leave the overlay, it informs its current parents in the tree. These parents trigger parent discovery procedures in order to discover new parents. This multicast protocol results in important parent discovery traffic since it is done by all peers and in a periodic manner.
- **MOST** (Multicast Overlay Spanning Tree Protocol) [9] is an overlay multicast protocol based on the construction of a minimum spanning tree. This protocol requires impera-

tively 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 maintains a peer list 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 lists of peers of its multicast groups. Here also the cost of flooding JOIN messages periodically is very important and the solution is routing protocol dependent.

III. GRAPH THEORY CONCEPTS

In this section, we present some graph theory concepts that have been useful in our design of the membership management adaptive overlay. Let $G(V,E)$ be a graph. V and E are respectively the set of vertices and the set of edges. One calls:

- **Connected graph:** G is connected if there is a path connecting every pair of vertices.
- **Undirected graph:** G is an undirected graph if the vertices are connected by undirected edges.
- **Subgraph:** A subgraph of G is a graph whose vertices and edges are subsets of G .
- **Cut:** A cut is a partition of the vertices of a graph into two sets. More formally, a cut is a partition of the vertices V into two sets S and T . Any edge $e(u,v) \in E$ with $u \in S$ and $v \in T$ is said to be crossing the cut and is a cut edge.
- **Tree:** A tree is a graph in which any two vertices are connected by exactly one path. Alternatively, any connected graph with no cycles is a tree.
- **Spanning tree:** Given a connected, undirected graph, a spanning tree of that graph is a subgraph which is a tree and connects all the vertices together. A single graph can have many different spanning trees.
- **Minimum spanning tree:** One can assign a weight to each edge of a graph. The weight of a spanning tree can be then computed as the sum of the weights of the edges in that spanning tree. A minimum spanning tree or minimum weight spanning tree is then a spanning tree with weight less than or equal to 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 minimum spanning trees for its connected components.

The following proprieties of a minimum spanning tree have been profitable for the design of our approach:

- **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 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 belong to all minimum spanning trees of the graph.

IV. AN ADAPTIVE OVERLAY FOR P2P MEMBERSHIP MANAGEMENT IN MANET

In this section, we describe our solution for P2P membership management in MANET. Our solution is based on an adaptive and robust overlay that minimizes resource consumption and dissemination delay. This overlay is used for the exchange of membership information among peers in the P2P network. New information is generated after one of these events: A new peer wants to join the network or one of the peers already in the network wants to leave. An information is also generated when a set of peers disappear or reappear because of network partitioning caused by nodes' mobility. Generally, the exchanged information describes the identity of the joining and departing peer. This can range from a simple IP address to a set of data describing his profile and interests. The purpose is that all peers in the network maintain the same information about other peers and that this information be as fresh as possible.

We want the membership overlay 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 structure this overlay as a minimum spanning tree in terms of number of hops, covering all peers of the P2P network (some MANET nodes might not be P2P members). This guarantees a minimal cost of the membership information dissemination in terms of number of messages. A message sent over a path is count as many times as the number of hops over the path. But since the members of a P2P network are continuously changing in time, the minimum spanning tree needs to adapt to the arrivals and departures of new peers. We introduce distributed and efficient mechanisms to take decisions on how to restructure the overlay when adding peers or when deleting them. Moreover, the nodes of a MANET are continuously moving which affects directly the weights of the spanning tree edges, and so our protocol needs to restructure the tree when needed in order to maintain the optimality property. The classical centralized Kruskal algorithm is not good in our case since it requires global knowledge about the logical-link states of all peers and the computation of the tree must be done by every node. Since the minimum spanning tree is not unique, every node will have a different spanning tree which disconnects the overlay. Our method guarantees the unicity of the overlay by making all decisions locally. Another important problem that we consider in the construction of our adaptive overlay is the fact that MANET is frequently subject to network partitioning. We add to our overlay a specific technique to merge separate overlays belonging to the same P2P network together when communication opportunities occur. The following paragraphs describe the detailed algorithms used to construct and update our adaptive overlay.

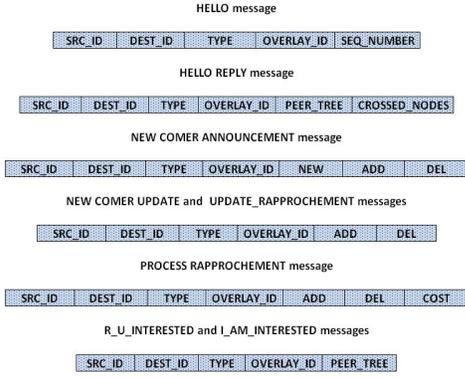


Fig. 1. Packet format

TABLE I
PACKET FIELDS DESCRIPTION

Field name	Field description
SRC_ID	The identifier of the peer sending the message
DEST_ID	The identifier of the peer that will handle the message
TYPE	The type of the message (eg. HELLO)
OVERLAY_ID	The identifier of the service overlay
SEQ_NUMBER	This field indicates the sequence number of the message. It is important in optimizing the flooding process. A node does not forward a message that it has already received.
PEER_TREE	A string representing the tree of the node sending the message.
CROSSED_NODES	A set of nodes tagged for the network partitioning awareness.
NEW	The identifier of the node joining the membership overlay.
ADD	A list of logical links to be added to the current tree.
DEL	A list of logical links to be removed from the current tree.
COST	The weight of the most costly logical link in a cycle.

A. Joining the membership overlay

We suppose in our work that each P2P network has a unique identifier (For example the file ID of a torrent in BitTorrent). Knowing this identifier, a node that becomes a peer interested in the service provided by the network triggers a join procedure. This procedure can be divided into two phases: discovering the nearest peer already in the network followed by the new arrival information dissemination to other peers and the overlay restructuring. By the end of this phase all peers are aware of this new arrival quickly and at low cost, and the overlay takes its new optimal shape.

1) *Discovering the nearest peer:* In order to discover a first attachment point to the membership overlay, 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 HELLO REPLY messages from any member located at one hop. HELLO REPLY messages are sent in unicast. In case there is no answer, it increments exponentially the value of the TTL of the HELLO message and waits again for at least one HELLO REPLY from one of the peers located at this new scope. If there are no answers, the TTL is incremented again and so on. If the maximum TTL is reached and no answer has been received, the node considers that the service is not provided in the network and that it is up to it to construct a new membership overlay. It automatically becomes the only node in the spanning tree. Now if an answer is received, the new member gets in the HELLO REPLY message a copy of the

actual tree. The tree is compressed in a string value obtained by an in breadth crossing of the tree where different levels of the tree are separated by a specific character (for instance a star). 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 the HELLO and HELLO REPLY messages is depicted in Table I and in Figure 1.

Using its current routing table, the new arriving peer compares the costs in number of hops to other peers belonging to the same tree without carrying out any measurement. This comparison allows it to identify the closest peer to it. The new peer should change its connection in the spanning tree if needed to attach to this closest peer as required by the cut property described in Section III. The connection to add is the lowest cost in the cut formed on one side by the old tree and on the other side by the new arriving peer. In practice and 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 an adaptation of the tree to be described next in order to conserve its optimality.

2) *New arrival information dissemination and overlay adaptation:* When a member of the overlay receives a CONNECT ME message from a new joining peer, it adds this peer to the current tree as a child node. This modification of the tree is then diffused to the other peers which can trigger other modifications to keep the tree structure optimal. The new parent sends to all its neighbors in the tree (its parents and its children) except the new arriving node a NEW COMER ANNOUNCEMENT message containing the identifier of the new node and the modifications it has made on the tree (It sends the added and removed logical links). We refer you to Figure 1 and Table I for details on this message. Every peer of the membership overlay receiving the NEW COMER ANNOUNCEMENT message updates its tree and verifies whether it can modify some of its logical links in order to optimize the current tree. This modification is described next. For now and after making these local decisions, it informs its neighbors in the tree except the peer which has sent the NEW COMER ANNOUNCEMENT message. It sends a new version of the NEWER COMER ANNOUNCEMENT message adding its own changes. The other peers of the tree (that do not receive the new version of the NEW COMER ANNOUNCEMENT message) are informed through a NEW COMER UPDATE message that the peer sends to the peer that has sent the original NEW COMER ANNOUNCEMENT message to it. This message contains only the logical links that the peer has added or removed from the tree. A 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 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 decisions that a peer must make when it receives a NEW COMER ANNOUNCEMENT message are a simple verification of the cycle property described in Section III. The cycle to consider is the cycle formed by the logical links on the path of the current tree starting from the intermediate peer taking the decisions 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 received the NEW COMER ANNOUNCEMENT message and adding the logical link to the newly joining peer. One can notice that all the decisions are made locally and in a distributed manner without compromising global optimality.

B. Adapting the membership overlay to mobility of nodes

Due to the mobility of the nodes of a MANET, the distances between the peers of the membership overlay 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. In this case, the weight of the spanning tree becomes smaller but it remains a minimum spanning tree. This movement has no impact on the optimality and structure of the membership overlay.
- Two peers that are not neighbors in the spanning tree get farther from each other. In this case, the weight 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 weight 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 of 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. So, this movement of peers needs an adaptation of the spanning tree seeking for the existence of the optimal one.

In the following paragraphs, we describe how we adapt our membership overlay in response to the two latter movements impacting the optimality of the tree:

1) *Two overlay neighbors get farther:* If two peers that are neighbors in the spanning tree get farther 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 rapprochement adaptation procedure which we will describe in the paragraph IV-B2. The second situation is that no one of these two peers gets nearer to other peers of the tree, 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 rapprochement problem and solved by the solution we come up for the latter one.

2) *Two peers that are not overlay neighbors get closer:*

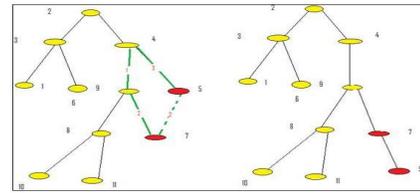


Fig. 2. Adapting the overlay after peers 5 and 7 get closer due to mobility

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 the nodes. Take the cycle formed by the actual logical path (in the tree) from P1 to P2, and by adding the logical link that connects directly P1 to P2. If there is a logical link L in this cycle such that $\text{cost}(L) > \text{cost}(P1,P2)$, the cycle property described in Section III 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 remove the logical link L and to add at its place the logical link (P1,P2). We illustrate this in Figure 2, where peers 5 and 7 represent respectively peers P1 and P2. The figure plots the minimum spanning tree before and after the adaptation. This adaptation procedure must be executed in a distributed and triggered manner. Each peer in the tree keeps tracking other peers and decides if another peer, which is not its neighbor in the current tree, becomes closer to it than normal. 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 with the lowest identifier executes the procedure. This is done by transiting a PROCESS RAPPROCHEMENT 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 the original sender of the message which then can decide which link to be removed and which link to be added. This modification is then disseminated to all peers of the tree. The peer sends an UPDATE RAPPROCHEMENT message to its neighbors in the tree which forward it to their neighbors and so on. The messages are described in Figure 1 and Table I.

C. Leaving the membership overlay

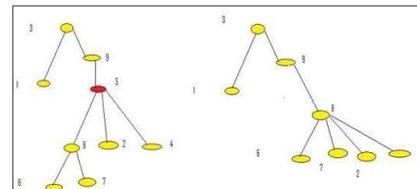


Fig. 3. The membership overlay before and after peer 5 leaves

Adapting the membership overlay after the departure of peers is very important for the efficiency and the unicity of the tree and for the freshness of the membership information. We ask every peer that decides to leave the application to

inform its logical neighbors in the spanning tree by sending 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 the optimal. The optimal is reached by having the peers apply the normal rapprochement adaptation procedure described earlier in paragraph IV-B2. 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 after the departure, are disseminated to all peers of the membership overlay together with the identifier of the departing peer. Figure 3 gives an example of adjustment of the spanning tree after the departure of peer 5. Sometimes a peer can leave the overlay improperly; in this case, it is the duty of the first neighbor detecting this phenomenon to trigger the tree adaption procedure described above.

D. 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 network split. Another scenario is that of two or more membership overlays constructed separately in different clusters but belonging to the same P2P network. At the first merging opportunity, it is very important to connect together the different partitions of a membership overlay in one large and efficient overlay. That is why; we add to our membership protocol a simple mechanism allowing peers to discover other peers coming from other overlays but belonging to the same application 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 overlay. Then, and using its routing table, it keeps tracking 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 overlay is a good candidate to ask for whether it belongs to the same application and comes from another cluster. The peer sends to this newly detected node a message R U INTERESTED in order to ask it whether it is interested in the application. 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 application, 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 overlay join procedure in order to connect to the biggest tree.

E. Global knowledge is not required

Until this point, we have considered that each peer must have a global knowledge about the other members in the same overlay and about how they are connected without the need to know the cost of the links among them. This is a strong assumption since the MANET can be of very large size. If it is the case, one can easily relax this assumption without impacting the optimality of the overlay neither the performance of the overlying application itself. In a large MANET it is almost of no importance if a peer knows about other peers located far away since the data exchange rate would drop to zero. Thus and to reduce the load, each peer can restrict its knowledge about the overlay to some scope. Here we explain that when applying the mechanisms described above while considering limited-scope knowledge, one can still construct global minimum spanning tree and the overlay partitioning problem is still correctly handled.

First, when a new peer wants to join the overlay, it is informed by the first peer it meets about its local knowledge of the membership overlay. With this information it can start constructing its own local knowledge of the overlay then completes this local knowledge by getting information from other peers. After that, it connects to the nearest peer among those in its local knowledge. This should be equivalent to connecting to the nearest peer in the global overlay. In an optimal overlay there is almost no chance that a peer connects to a distant peer beyond some limited scope. Now, in the case of two peers belonging to the global overlay that get close to each other, these two peers are generally in the local knowledge of each other otherwise this rapprochement is not important to be considered. But even if they are not, they will end up by being so in a progressive way after several applications of the rapprochement procedure with intermediate peers. Finally nodes move at a limited speed and a peer will not suddenly appear in the local knowledge of some other peer without being noticed. The case of separate overlays does not change since the tagging-querying mechanism we deploy to merge the same overlays together will still work in the case of local knowledge. We believe limiting the knowledge about the overlay is important in case of very large MANETs and P2P networks to reduce the load on the number of exchanged messages. The information on arrivals and departures will only propagate in the local scope interested by these changes. However the question of how much to limit the scope is still an important issue to balance accuracy and load. Further research is required in this direction.

V. PERFORMANCE EVALUATION

In this section, we study the performances of our method in comparison 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 [2].

A. Performance metrics

Let P_t be the set of peers interested in the application at an instant t and let p_t be the cardinal of this set of peers. When a peer is running the membership management protocol, it maintains at an instant t a set of peers N_t , consisting its view of the members of the P2P application. Let n_t be the cardinal of this set. Among these n_t peers, there are t_t peers belonging to P_t and f_t not belonging to it (e.g. due to peers which have left). 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 traversed by the exchanged messages during a fixed period of time. This cost varies as a function of the membership management method. This cost does not take into consideration the freshness of information maintained by the peers and then 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 metric named cost corrected by freshness of information which we note C_f . 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 on peers interested in the P2P application. At the end of the measurement time, one computes \tilde{p} , \tilde{n} , \tilde{t} and \tilde{f} the average values of p_t , \hat{n}_t , \hat{t}_t and \hat{f}_t on the samples.

The cost corrected by freshness of information C_f metric 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 $\tilde{p} - \tilde{t}$ remaining members. This term can be easily calculated considering that the members of the P2P application have paid $\frac{C}{n}$ to discover a peer. Hence, the lack

of information cost term will be equal to $\frac{C(\tilde{p}-\tilde{t})}{\tilde{n}}$. The second term to be added to the ordinary cost is a $\frac{C_f}{\tilde{n}}$ 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 $\frac{C_f}{\tilde{n}}$. The following equation gives the formula to compute the cost corrected by the freshness of information metric:

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

B. Scenario description

To conduct our simulations, we consider a MANET composed of 50 nodes moving inside a bounded area of width 100 m and height 500 m following the Random Waypoint mobility model. The speed and the pause time of each node are taken equal to 2 m/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 50 m and the data rate is set to 1 Mb/s. For ad hoc routing, we use the OLSR [3] protocol. 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. A node is then following an ON/OFF process until the end of the simulation. The durations of the OFF and the ON states follow, respectively, an exponential distribution of parameter μ and an exponential distribution of parameter λ . The density of the overlay is the ratio of interested nodes to the total number of nodes. One can easily show that this density equals $\frac{\lambda}{\lambda + \mu}$. When not varied, the density is taken equal to 50% taking both $\frac{\lambda}{\lambda + \mu}$ and $\frac{\mu}{\lambda + \mu}$ equal to 500 s. The simulation duration is set to 3600 s. The sampling period T_s used to compute the cost corrected by freshness of information is chosen as 10 s. The values of the parameters described in this paragraph can be varied in some simulations.

C. Simulation results

The classical client/server approach supposes that each peer contacts periodically the server to update its knowledge about the members of the P2P application. In our simulations, a random node plays the role of the server. Figure 4 plots the cost in number of hops messages as a function of the period of contacting the server. It shows that this cost is proportional to the inverse of the period. The same figure plots the cost corrected by freshness metric as a function of the period of contacting the server. One can easily notice that this corrected cost is higher than the ordinary cost. It does not continue diminishing when the period of contacting the server increases. In fact, the freshness of information decreases with the increase in the contacting period. In the following simulations, when client/server approach is used, the contacting period is set to 400 s.

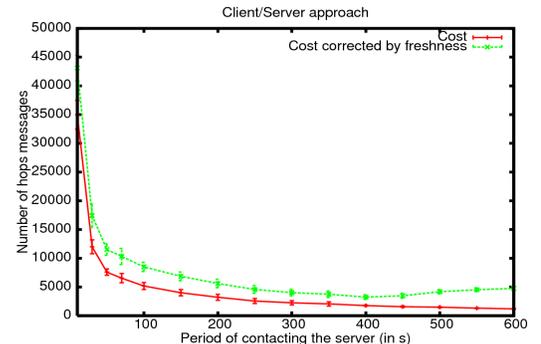


Fig. 4. Cost and corrected cost for the client/server approach as a function of the period of contacting the server

The following simulations compare our approach to three classical approaches of membership management: The client/server method, the flooding method and the non-adaptive overlay method. When the flooding method is used, every peer floods the network to announce its arrival or its departure.

Unlike our approach, the non-adaptive overlay method constructs a sub-optimal tree that does not adapt to the underlying topology. Figure 5 plots the cost corrected by freshness as a function of the density of the overlay for the four approaches. Figure 6 shows the ratio of the corrected cost to the ordinary cost as a function of the density of the overlay for the same four methods. These two figures show that the flooding method is the worst method in terms of the corrected cost. Unlike our adaptive overlay method, the non-adaptive overlay mechanism does not scale when the density of the overlay increases and as a result, generates a corrected cost higher than the client/server method. In terms of freshness of information, the two overlay methods have ratio of costs very close to 1 where as the client/server method is the worst one. Hence, the figures show then that our adaptive overlay method outperforms the other methods in terms of cost and freshness of information.

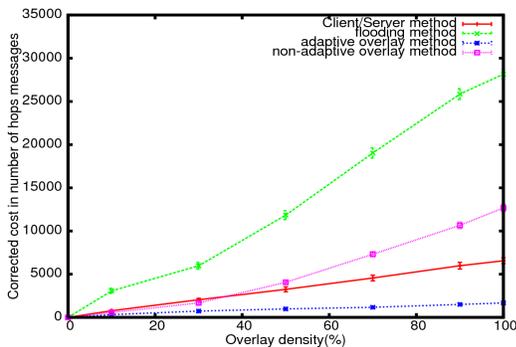


Fig. 5. Corrected cost as a function of the density of the overlay

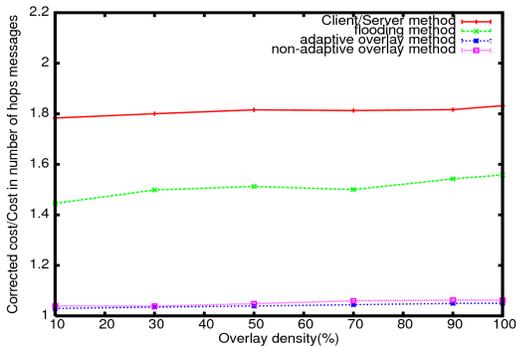


Fig. 6. Ratio of corrected cost to ordinary cost

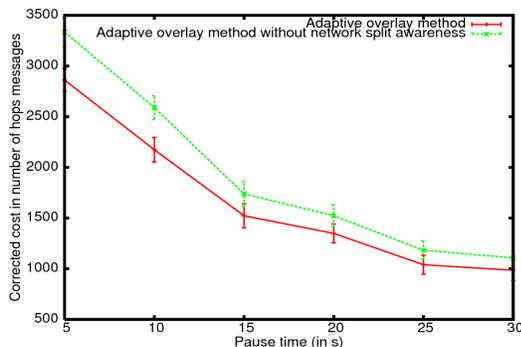


Fig. 7. Overlay split awareness

The last set of simulations aims to study the impact of network split on the membership management approaches. We validate our adaptive overlay approach by comparing

it to a similar adaptive approach but without the overlay split awareness mechanism. This means that for the second approach the requesting-tagging mechanism is not activated. Figure 7 compares the corrected cost of the two methods as a function of the pause time of the mobility model. It shows that paying extra cost for having overlay partitioning awareness; does not impact the corrected cost of our method. Indeed, it ameliorates the freshness of information and then it has a lower cost corrected by freshness.

VI. 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 overlay allowing peers to discover each other and keep themselves informed about the arrivals and departures of other peers. The overlay we propose to connect the peers is a standalone service that can be used by any application requiring the sharing of information among a group of users. Moreover, our overlay minimizes the number of exchanged messages and reflects faithfully the underlying topology, 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 approach outperforms classical solutions in terms of network load and freshness of information. The future work will be on the integration of this membership overlay within P2P applications as, for example, the trackerless BitTorrent file sharing application. Our aim is to study the gain in performance that one would obtain by decoupling the construction of the membership overlay from the data plane itself.

REFERENCES

- [1] BitTorrent protocol. <http://wiki.theory.org/BitTorrentSpecification>.
- [2] NS: The Network Simulator, <http://www.isi.edu/nsnam/ns/>
- [3] P.Jacquet, P. Mhlethaler, T Clausen, A. Laouiti, A. Qayyum and L. Viennot Optimized Link State Protocol for Ad Hoc Networks. IEEE INMIC Pakistan 2001.
- [4] The Gnutella specification, 2000, <http://dss.clip2.com/GnutellaProtocol04.pdf>
- [5] S.Ratnasamy, P.Francis, M.Handley, R. Karp and S.shenker. A scalable content-addressable networks, ACM SIGCOMM, 2001.
- [6] I.Stoica, R.Morris, D.Karger, M.F.Kaashoek and H.Balakrishnan. Chord: a scalable peer-to-peer lookup service for internet applications, ACM SIGCOMM, 2001
- [7] Chao Gui, and Prasant Mohapatra, Efficient Overlay Multicast for Mobile Ad Hoc Networks, In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), March, 2003.
- [8] Peter Baumung, TrAM: Cross-Layer Efficient Application-Layer Multicast in Mobile Ad-hoc Networks, Proceedings of IEEE Wireless Communications and Networking Conference, Hong Kong, China, Mar 2007.
- [9] G. Rodolakis, A. Meraihi Naimi, A. Laouiti. Multicast Overlay Spanning Tree Protocol for Ad Hoc Networks. In WWIC, Coimbra, Portugal, 2007
- [10] L.B. Oliveira, I.G.Siqueira, A.A.Loureiro, Evaluation of ad hoc routing protocols under a peer-to-peer application, WCNC, 2003.
- [11] A.Klemm, C.Lindermann, O.Waldhorst. A special-purpose peer-to-peer file sharing system for mobile ad hoc networks, VTC, 2003.
- [12] S.M. Das, H.Pucha, Y.C. Hu. Ekta: an efficient peer-to-peer substrate for distributed applications in mobile ad hoc networks. TR-ECE-04-04, Purdue University, 2004.
- [13] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, Gossip Algorithms: Design, Analysis, and Applications, Proc. INFOCOM 2005, March 2005.