Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 31 (2008) 1400–1412
computer communications
www.elsevier.com/locate/comcom

Multicast overlay spanning trees in ad hoc networks: Capacity bounds, protocol design and performance evaluation ☆

Georgios Rodolakis [a,*], Anis Laouiti [b], Philippe Jacquet [a], Amina Meraihi Naimi [a]

[a] INRIA, BP 105 Rocquencourt, 78153 Le Chesnay Cedex, France
[b] GET/INT, 91000 Evry, France

Available online 26 January 2008

Abstract

We study the benefits of multicast routing in the performance of wireless ad hoc networks. In particular we show that if a node wishes to communicate with n distinct destinations, multicast can reduce the overall network load by a factor $O(\sqrt{n})$, when used instead of unicast. One of the implications of this scaling property consists in a significant increase of the total capacity of the network for data delivery. Hence, we show that the aggregate multicast capacity of wireless ad hoc networks is $O(\sqrt{n})$ larger than the unicast capacity, when the group size n is small compared to the total number of nodes in the network. We discuss how these information theoretic results can be taken into consideration in the operation of a multicast protocol for wireless mesh networks using Multicast Overlay Spanning Trees (MOST). We perform simulations of the MOST protocol to compare with the theoretical results, and we present a fully working implementation for real network environments.

Keywords: Multicast; Overlay; Capacity; Ad hoc; Wireless

1. Introduction

An ad hoc or mesh network can be defined as a set of nodes that communicate using the wireless medium, and does not require any pre-existent infrastructure. The main challenge is to offer connectivity between the participating nodes in a multi-hop manner by sharing the same wireless channel. Wireless mesh networks consist of mesh routers and mesh clients, where mesh routers have minimal mobility and form the core of the network. Mesh routers establish an ad hoc network and maintain the mesh connectivity in order to offer network access to mesh clients. Hence, mesh networks can be seen as a special case of ad hoc networks which deal with limited router mobility, while ad hoc networks cover more general mobility issues. Several protocols have been proposed to provide unicast communications in such environments.

Multicast offers an elegant way to establish group communication between users by using the concept of multicast groups, which are defined by their corresponding address. Interested *clients* can join and leave those groups in order to send and/or receive data from other group members. Moreover, the mechanisms which enable multicast communication ensure that an efficient strategy is used to deliver the data packets to all the members simultaneously. Therefore, multicast communication is adequate for a large class of applications, such as video-conferences, multi-player games, streaming applications *etc*. The previously described requirements make multicast routing an important and difficult challenge in the Internet, and even more so in ad hoc networks. In fact, mainly due to the dynamic nature of the routes, multicast protocols developed for wired networks cannot operate in the harsher wireless environment. This creates a need for protocols which are specially adapted to mesh networks. However, although the

☆ Some results of this work were presented in "Multicast Scaling Properties in Massively Dense Ad Hoc Networks", SANSO, Fukuoka, 2005, and "Multicast Overlay Spanning Tree for Ad Hoc Networks", WWIC, Coimbra, 2007.

* Corresponding author. Tel.: +33 1 39 63 58 56.
E-mail address: georges.rodolakis@inria.fr (G. Rodolakis).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.01.019

capacity of wireless networks has been a very active research area since the seminal paper of Gupta and Kumar [11], the specific impact of multicast routing has not attracted too much attention, with the exception of [25]. Our main contributions in this paper are as follows:

- we derive analytically multicast capacity bounds for wireless ad hoc networks;
- we propose a practical overlay spanning tree algorithm and a protocol solution (MOST) taking into account these information theoretic bounds;
- we verify the theoretical analysis with numerical simulations and we evaluate the performance of the protocol using the ns-2 simulator.

One of the advantages of multicast routing is that it reduces the total bandwidth required to communicate with all group destinations, since some links can be common to several destinations. In wired networks, the gain of multicast communication has been studied in [1,5,21], by estimating the ratio of the number of links in a multicast tree to n destinations over the average unicast hop distance between two random nodes. The resulting normalized multicast cost has been found experimentally to scale in $n^{0.8}$. The gain of multicast is reflected by how far the normalized multicast cost deviates from linear growth. Except from evaluating the protocol performance, such analytical cost estimates can also be useful for the efficient management and accounting of multicast services in the network [23]. However, the topology of mobile ad hoc networks is significantly different and one would expect a much different scaling law too. Indeed the average unicast hop distance in wired networks is usually of the order $\log N$, where N is the total number of nodes in the network, while in ad hoc networks the average distance grows proportionally to $\sqrt{N/\log N}$, since the optimal neighbor degree increases in $O(\log N)$ when the capacity increases [11].

In this paper, we establish performance bounds on the expected size of multicast trees as a function of the number of multicast destinations n, both via analytical methods and via simulation. In random mobile ad hoc networks, the gain of multicast communication compared to unicast is significantly larger than in wired networks. We show that a scaling law in $O(\sqrt{n})$ holds for the normalized multicast cost and, based on the analysis, we propose a protocol to be used in conjunction with the unicast routing protocol OLSR. We also show that the performance of the protocol is significantly better than MPR flooding, *i.e.*, the optimized broadcast mechanism which is already implemented in OLSR, for a vast scale of group sizes. These results can provide further motivation in supporting multicasting in mobile networks, besides the advantages of group-oriented communication. The implications of this scaling law consist in a significant increase of the total capacity of the network for data delivery, while the total amount of generated data will actually decrease (compared to the case where each node transmits data to one single destination), and both are proportional to $\sqrt{\frac{N}{\log N}}$.

The remainder of this paper is organized as follows. In Section 2 we present the network model and provide analytical results on the scaling law of the normalized multicast cost. The impact of multicasting in the capacity of the network is discussed and we present measurements on multicast scaling derived from simulations in generated graph models of wireless ad hoc networks. In Section 3, we introduce MOST, a new multicast protocol for ad hoc networks, which is based on the previous analysis. We evaluate the performance of the protocol through simulations, which we compare to the analytical results. We overview how the protocol was implemented for use in real network environments in Section 5. We conclude and present some interesting directions for further research in Section 6.

2. Asymptotic multicast properties in ad hoc networks

2.1. Multicast cost scaling law

In this section we will quantify the cost of multicast communication vs the average unicast cost. We assume that nodes have a complete knowledge of the network topology. In order to optimize the control traffic we will see in a further section how we can somewhat relax this hypothesis in the use of the OLSR link state routing protocol.

2.1.1. Model description

We assume that N nodes forming a massively dense ad hoc network are distributed according to a Poisson process[1] in an area of arbitrary size $\mathcal{A}$. In this case, *i.e.*, when N is large, routes can be considered as continuous lines between nodes, and the number of retransmissions needed for a packet to reach its destination is $\Theta(\frac{d}{r})$, where r is the typical radio range and d is the Euclidean distance from the source to the destination [14,3]. Hence, we can represent a massively dense ad hoc network with an Euclidean graph, in which the edge costs are proportional to hop distances between nodes. The result of Gupta and Kumar [11] states that the maximum bandwidth is attained when the radio range is $r = k\sqrt{\frac{\log N}{N}}$, where k is a constant which depends on signal propagation and medium access control. A source and a multicast group of size n are chosen uniformly at random among the N nodes. We assume here that $n \ll N$. As a result, the $n+1$ multicast nodes are distributed in the area according to a Poisson process of intensity $\frac{n+1}{\mathcal{A}}$.

An optimal multicast tree is a Steiner tree, *i.e.*, a tree of minimal cost connecting all of the multicast nodes via an arbitrary subset of the remaining nodes that are not in the multicast group. Therefore, the problem of finding

[1] When N is large, the model is equivalent to a network consisting of N nodes distributed uniformly at random in an area $\mathcal{A}$.

the optimal tree is NP-complete, even in Euclidean graphs, although in this case there is a polynomial time approximation scheme [2]. Note that since we assumed here that $n \ll N$, the wireless multicast advantage only provides an asymptotically marginal improvement. In case the number of clients n becomes significant with respect to the total number of nodes N in the network, the wireless environment would permit to further reduce the total number of retransmissions in order to reach all group destinations. This is made possible by using a connected dominated set (as is the case with MPR flooding), or other algorithms specially adapted to the wireless environment (see [27] for optimized algorithms taking also into account energy efficiency). However, although some experimental results are provided in the simulations section, we do not consider this case analytically when $n \ll N$. In any case, the upper bounds we present remain valid.

Since the problem of finding the optimal tree is intractable, we will use an approximation. We consider the two more common cases of minimum spanning trees and shortest path trees.

2.1.2. Minimum spanning trees

First, we consider multicast trees corresponding to minimum spanning trees on the $n+1$ multicast nodes. In a minimum spanning tree branching is constrained only to multicast nodes, and the computation can be performed in polynomial time. On the other hand, in Steiner trees, branching can occur on any node (or any point in the plane in the Euclidean case). In metric graphs, the cost of a minimum spanning tree is within twice the cost of an optimal Steiner tree [26]. However, it can be shown that the Euclidean minimum spanning tree is not longer than $\frac{2}{\sqrt{3}}$ times the optimal Euclidean Steiner tree [8]. Hence, in the case of massively dense ad hoc networks, minimum spanning trees yield results which are very close to the optimal. In Fig. 1 we depict an example of a minimum spanning tree as well as a Steiner tree, in an Euclidean graph of four nodes. The red nodes in this example would be the clients that must be covered by the multicast tree. As we can see, the possibility of using the blue non-multicast node in the Steiner tree offers a length improvement of $\frac{\sqrt{3}}{2}$, which corresponds in this particular case to the worst case bound.

To proceed we will compute the path length (in meters) of a minimum spanning tree on $n+1$ points in an area $\mathcal{A}$. We denote this length $L(n+1)$.

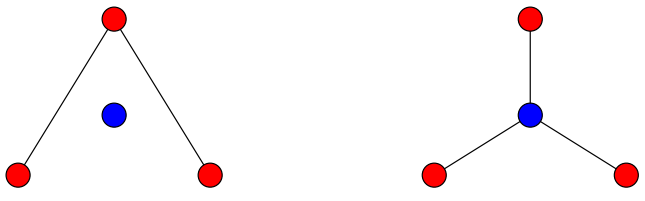


Fig. 1. Comparison of a Minimum Spanning Tree (MST) with an optimal Steiner tree in an Euclidean graph.

From the analysis in [4,24], in the two-dimensional case, it comes that an upper bound holding with high probability (approaching 1 as the number of nodes tends to infinity), for the path length (in meters) of a minimum spanning tree is

$$L(n+1) \leq \gamma n \sqrt{\frac{\mathcal{A}}{n+1}} \sim \gamma \sqrt{\mathcal{A} n}. \tag{1}$$

where γ is a constant that depends on the shape of the network domain. For a disk or a square we can set $\gamma = \frac{1}{\sqrt{2}}$.

We now define the normalized multicast cost $R(n)$ for a multicast group of size n as

$$R(n) = \frac{\text{multicast cost}}{\text{average unicast cost}},$$

where the costs are expressed in number of hops. In other words the multicast cost is the number of links in the multicast tree, and the average unicast cost is the average route length from a random source in the multicast group to a random destination in the multicast group.

We base our analysis on the observation that routes can be considered as continuous lines between nodes, and the number of hops needed for a packet to reach its destination is $\Theta\left(\frac{d}{r}\right)$, where r is the optimal radio range as stated by Gupta and Kumar.

The expected path length of the multicast tree in number of hops is $\Theta\left(\frac{L(n+1)}{r}\right)$, while the average unicast cost is $\Theta\left(\frac{L(2)}{r}\right)$. This implies that, for the normalized multicast cost, it holds

$$R(n) \simeq \frac{L(n+1)}{L(2)}. \tag{2}$$

Quantity $L(2)$ is highly dependent on the shape of the network domain, but is of order $\sqrt{\mathcal{A}}$ when the network domain shape stays within some reasonable model. In all rigor we have $L(2) = \beta\sqrt{\mathcal{A}}$. For the disk we have $\beta = \frac{128}{45\pi^{3/2}} \approx 0.51$, for the square it becomes $\beta = \frac{1}{15}(2+\sqrt{2}+5\log(1+\sqrt{2})) \approx 0.52$.

Combining (1) and (2), we get

$$R(n) \leq \frac{\gamma n}{\beta\sqrt{n+1}} = O(\sqrt{n}). \tag{3}$$

Hence, we obtain the multicast scaling law $R(n) = O(\sqrt{n})$. It comes that the gain of multicast over unicast, which is reflected by how far $R(n)$ deviates from linear growth, is also $O(\sqrt{n})$. This result is in contrast with similar comparisons in wired networks [1,5,21] where the gain of multicast communication is significantly smaller. In that case, the multicast cost scales, according to experimental studies, following a power law with exponent between .8 and .9.

More generally, we can show, using the same approach, that for a network spanning on a domain in dimension D

$$R(n) = O\left(n^{1-\frac{1}{D}}\right).$$

In [24] it is shown that the length of minimum spanning trees on points randomly placed in a hypercube is $O(n^{1-\frac{1}{D}})$, even when the point distribution is not uniform, with some mild constraints. This implies that the multicast scaling law still holds when the multicast nodes are not distributed uniformly among the nodes of the network.

2.1.3. Shortest path trees

A popular approach in building multicast trees in wired networks consists in pruning shortest path trees rooted at the source node. In this case, we cannot prove worst case bounds on the total cost of shortest paths trees, compared to the cost of optimal Steiner trees. However, in practice, shortest path trees achieve a satisfactory performance. Moreover, shortest path trees minimize the maximum path length from the source to any destination. In the currently considered model of mobile ad hoc networks, when $n \ll N$, a shortest path tree is equivalent to n unicasts, since the expected number of branching nodes is very small. Hence for a small number of destination nodes, the gain of multicast communication is negligible.

On the other hand, when $n \to N$, the total number of hops in the tree also tends to N, since we consider a tree spanning on almost all the nodes. The average unicast distance in hops is $O\left(\frac{1}{r}\right)$, where the radio range $r = \alpha\sqrt{\frac{\log N}{N}}$. Hence, the normalized multicast cost tends to

$$R(N) = O(Nr) = O\left(\sqrt{N \log N}\right).$$

This is the expected behavior for any method used to construct a tree spanning on all the nodes of an ad hoc network with unit cost links. Consequently, for large multicast group sizes we still expect to observe a multicast scaling law of the form $R(n) = O(n^{\frac{1}{2}+\varepsilon})$, for any $\varepsilon > 0$. In Section 2.3, we study the normalized multicast cost of shortest path trees experimentally.

2.2. Capacity of multicast communication

In this section we investigate the impact of the multicast cost scaling law in the capacity of the network, when nodes communicate with multicast. We are interested in the order of magnitude of the maximum attainable bandwidth. We show that similar bounds to the ones described in [25] can be obtained without the need of particularly complex additional routing mechanisms.

In presence of traffic density of λ bits per time unit per square area unit, the typical radius of correct reception r decays in $O\left(\frac{1}{\sqrt{\lambda}}\right)$ [11,14]. If C is the capacity generated by each node, the density of traffic generated per square area unit is $\Theta(CN)$. The maximum bandwidth attainable for unicast traffic is $C = O\left(\frac{1}{\sqrt{N \log N}}\right)$.

We have shown that each multicast packet in a group of size $n \ll N$ will be retransmitted with high probability $\Theta(\sqrt{n}\frac{1}{r})$ times. This yields a traffic density (including retransmissions) $\lambda = \Theta\left(CN\sqrt{n}\frac{1}{r}\right)$. Therefore

$$r = O\left(\frac{1}{\sqrt{\lambda}}\right) = O\left(\sqrt{\frac{r}{CN\sqrt{n}}}\right)$$
$$\Rightarrow C = O\left(\frac{1}{rN\sqrt{n}}\right).$$

As a result, the maximum rate at which a node can transmit multicast data is $O\left(\frac{1}{\sqrt{nN \log N}}\right)$ and it is obtained for the minimum $r = O\left(\sqrt{\frac{\log N}{N}}\right)$.[2] In this case, the total rate at which data is received by the n destinations in the multicast group is $O\left(\sqrt{\frac{n}{N \log N}}\right)$.

When all nodes in the network communicate in unicast (each node with one single destination), the total capacity of the network increases with network size in $O\left(\sqrt{\frac{N}{\log N}}\right)$. However, when there are $O(N)$ nodes in the network acting as multicast sources in groups of size n (*e.g.* in teleconferences between n users), the total rate at which data is transmitted in the network is $O\left(\sqrt{\frac{N}{n \log N}}\right)$. Similarly, the total rate at which data is received is $O\left(\sqrt{\frac{nN}{\log N}}\right)$. Hence, compared to unicast traffic, multicast traffic results in an increase by a factor $O(\sqrt{n})$ of the capacity of the network (and per node) for receiving data, although the total distinct data transmitted will in fact decrease by the same factor.

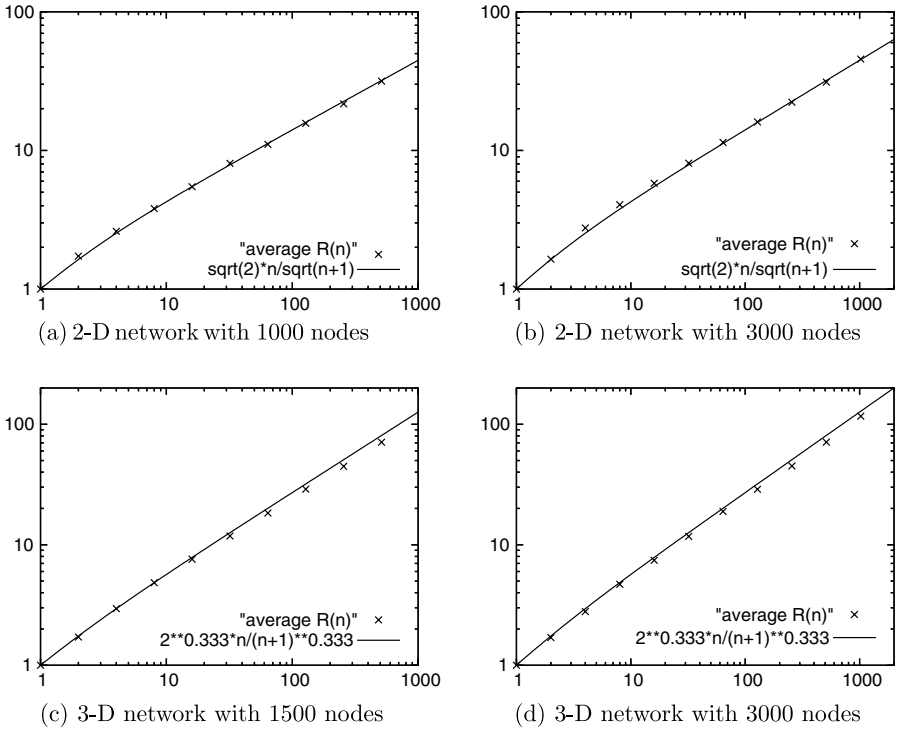
2.3. Numerical results

In this section, we present numerical simulations on the comparison of multicast performance with unicast and MPR flooding, respectively.

2.3.1. Comparison with unicast

In this section, we present simulations that verify the theoretical results on generated graph models of mobile ad hoc networks. We measure the normalized multicast cost $R(n)$ for various sources and multicast groups, and take the average for each group size n. The graphs are generated by placing nodes randomly in a square for 2D networks and in a cube for 3D networks, and then connecting the nodes which are in a distance smaller than the critical radius for connectivity r, such that the average number of neighbors for each node is $\log N$. Figs. 2 and 3 depict the theoretically predicted behavior of the multicast power law derived in Section 2.1. Since the analysis predicts a power law, the plots are presented in log log scale, where the power laws correspond to straight lines. The simulation results are compared to a

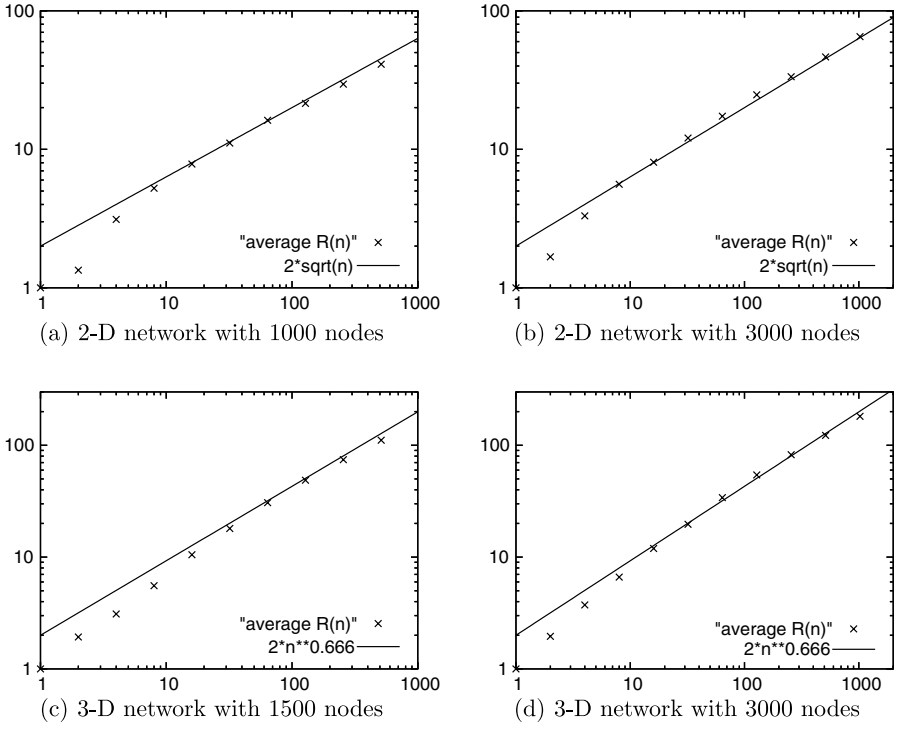
[2] The $\log N$ factors can be dropped if we assume optimally placed nodes, or if we relax the network connectivity requirement to the existence of a giant component.



(a) 2-D network with 1000 nodes (b) 2-D network with 3000 nodes

(c) 3-D network with 1500 nodes (d) 3-D network with 3000 nodes

Fig. 2. Multicast cost $R(n)$ vs multicast group size n.



(a) 2-D network with 1000 nodes (b) 2-D network with 3000 nodes

(c) 3-D network with 1500 nodes (d) 3-D network with 3000 nodes

Fig. 3. Shortest path tree cost $R(n)$ vs multicast group size n.

continuous line which illustrates the predicted asymptotic growth.

In Fig. 2, we present results corresponding to minimum spanning trees. The algorithm used to construct the minimum spanning trees will be presented in the following section, when we describe a protocol which can achieve these performance estimates. The measured cost is compared to the function $\frac{n\sqrt{2}}{\sqrt{n+1}}$, obtained from (3) by setting $\gamma = \frac{1}{\sqrt{2}}$ and $\beta = 1/2$, which corresponds to the approximate case where we do not consider border effects. The sample points (1.357).

Fig. 3 depicts measurements of the normalized cost of shortest path trees. The multicast cost $R(n)$ is compared

to function $2\sqrt{n}$ (where the constant 2 was chosen empirically). Observe that the cost is always higher than in minimum spanning trees, although the plot grows linearly with a slope close to 0.5 for large n.

In the case of three-dimensional ad hoc networks, for both minimum spanning trees and shortest path trees, the normalized multicast cost scales in $O(n^{\frac{2}{3}})$.

We note that, in all cases, the figures show that the simulations fit very well with the analysis for all network sizes, and for a wide range of group sizes.

2.3.2. Comparison with MPR flooding

In this section we compare the multicast overlay trees with MPR flooding, *i.e.*, the optimized broadcast mechanism which is already implemented in OLSR [6] and takes advantage of multi-point relay nodes (MPR). MPR nodes are elected by their neighbors because they cover their two-hop neighborhood. That way broadcast traffic consumes less resources in order to be forwarded to all destinations. Detailed performance studies of MPR flooding can be found in [17,12]. This comparison is of interest because, although using overlay multicast trees achieves significant performance gains, this would happen at the cost of some extra protocol complexity, which could be avoided by using the existent MPR-flooding technique. Hence, it is important to identify in which situations such a compromise is justified.

We perform simulations in ad hoc networks generated in the same manner as in the previous section. We measure the total multicast cost, *i.e.*, the total number of forwarding retransmissions needed to reach all the group destinations, for various sources and multicast groups, and take the average for each group size n. These measurements are compared with the average number of retransmissions that are generated using MPR flooding, initiated from the same source nodes as before. In this case, the number of retransmissions is independent of multicast groups and their size. The algorithm used for the selection of MPR nodes is the greedy algorithm described in [17,12].

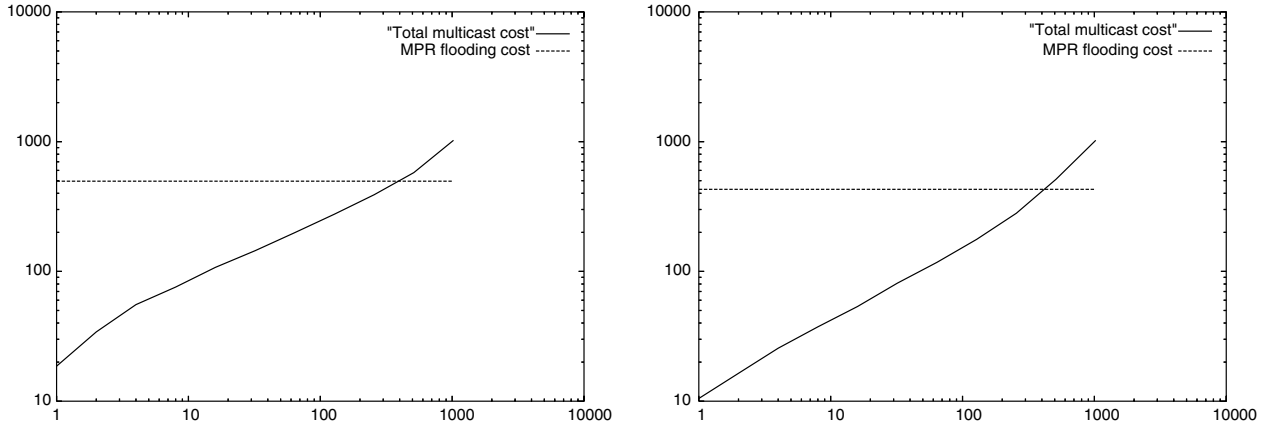
In Fig. 4(a), we present results obtained from simulations in a sparse network where the critical radius for connectivity r, is such that the average number of neighbors for each node is $\log N$. In this case, the number of retransmissions in MPR flooding corresponds to approximatively half the nodes in the network and is depicted by the straight lines in the graphs. However, multicasting performs better in almost the entire range of possible group sizes, except for multicast groups that constitute a large portion of the network. The same qualitative result is obtained if we repeat the simulation with a different network size.

In Fig. 4(b), we repeat the simulations in a denser graph where the average node degree is twice the number of neighbors corresponding to the critical connectivity limit. Suggestively, the average node degree observed in the simulation corresponding to Fig. 4(b) is approximatively 14, while in Fig. 4(a) it is approximatively 7. Note that this does not influence the comparative advantage of multicast over the same range of group sizes as in sparse graphs.

3. Specification and simulation of MOST protocol

In this section, we present the Multicast Overlay Spanning Tree (MOST) protocol, in which we take into consideration the previously derived results and simulations.

As discussed earlier, multicast protocols proposed for wired networks are not adapted to ad hoc networks, because of the frequent changes in tree structure due to the dynamic network topology, in addition to the group membership changes. Hence, multicast ad hoc routing is a challenging research domain, and many possible approaches have been proposed in the research literature [7]. Multicast ad hoc protocols can be classified according to the underlying routing structure to tree-based protocols and mesh-based protocols. The routing structure can be either group shared, or source dependent. Some tree-based protocols are MAODV [22] which is an extension to the



(a) Network with 1000 nodes (b) Denser Network with 1000 nodes

Fig. 4. Total number of multicast retransmissions vs group size n, compared with the number of retransmissions using MPR flooding (straight lines).

unicast routing protocol AODV [20] based on a group shared tree, MOLSR [16] which is an extension to OLSR unicast routing protocol based on a Dijkstra tree, and Adaptive Demand-driven Multicast Routing Protocol (ADMR) [15]. A protocol which is based on overlay trees is AMRoute [19]. As an example of mesh-based routing protocols we mention On-Demand Multicast Routing Protocol (ODMRP) [18] and Core-Assisted Mesh Protocol (CAMP) [10].

In contrast to these solutions, we propose a protocol aiming at achieving the theoretical capacity bounds derived in the previous section, being based on overlay group shared trees. In the next section, we describe the algorithms used by the protocol in order to maintain the multicast overlay spanning trees.

3.1. Overlay tree construction algorithms

As we saw previously, it is more efficient to consider minimum spanning trees. We discuss two algorithms for the overlay tree construction, which achieve optimal normalized multicast cost. The algorithms do not require any more information than what is provided by a link state unicast routing protocol, like OLSR.

In Algorithm 1 the construction of the minimum spanning tree (Step 3) can be implemented using Prim's algorithm. The resulting tree is an overlay multicast tree, since it consists only of multicast nodes and its links are in fact tunnels in the actual network. Multicasting is achieved when each node forwards multicast packets to its successors in the overlay tree. It must be noted that in a fully distributed protocol, each node must be able to compute the same minimum spanning tree independently from the others. Therefore, we impose an ordering to the multicast nodes based on their IP addresses when executing Prim's algorithm. The computed minimum spanning tree will be directed and rooted to the node with the smallest IP address. However, this fact has no practical importance in the protocol's operation, where the tree will be treated as a shared tree with no root. Step 1 corresponds to n Dijkstra algorithm iterations. Therefore, the total complexity is $O(n(M + N\log N))$, where n is the multicast group size, N and M are the number of nodes and edges in the network, respectively.

Algorithm 1. Basic Minimum Spanning Tree Algorithm

Input: Network graph.
Output: Overlay tree.

(1) Find shortest paths between all pairs of multicast nodes.
(2) Build complete graph on multicast nodes with costs $c_{ij} = \{\text{length of shortest path between } i \text{ and } j\}$.
(3) Build minimum spanning tree on the complete graph, rooted at the source node.

The algorithm's expected complexity can be improved because it is not necessary in practice to compute all shortest paths from each node to all other nodes to build the minimum spanning tree. We propose Algorithm 2 as a faster alternative to compute minimum spanning overlay trees. The algorithm is essentially equivalent to Algorithm 1, but the shortest paths are calculated in conjunction with the minimum spanning tree. Hence, it is not necessary to compute shortest paths between all pairs of multicast nodes. In fact, according to tests in wireless network topologies, this algorithm has an average running time comparable to a Dijkstra algorithm, even when the number of clients increases.

We first present a high level description of Algorithm 2. The algorithm manages a min-priority queue, as in a Dijkstra algorithm, with the difference that the attribute is the distance of the nodes from the multicast tree (instead of the root). Algorithm 2 covers the network graph in the following general steps:

(1) initialize all distances to infinity and insert the nodes in a priority queue;
(2) set the root's distance to 0;
(3) while there are uncovered multicast nodes:
(4) extract from the queue the node v with the smallest distance;
(5) if v is a multicast client: update its distance to 0;
(6) update the distances and predecessors of v's neighbors;

We now give a detailed and complete description of Algorithm 2.

We denote $G(V, E, w)$ the network graph, where V is the node set, E is the edge set, and each edge e is associated with a cost $w(e)$. We also denote S the set of multicast nodes. The array d associates each node with a distance to the multicast overlay tree, *i.e.*, $d[v]$ corresponds to the minimum distance of node v to the multicast nodes that are already part of the tree. This distance is initialized to 0 for the root node and to ∞ for all other nodes. The array π associates each node with a predecessor multicast node. When this table has been computed, it contains the information needed to represent the overlay tree, since each multicast node will be associated with another multicast node (except from the root). The predecessors of the other nodes in the graph need only be maintained during the computations.

The algorithm manages a set F of multicast nodes that have not been covered yet by the tree, and a min-priority queue Q which includes all nodes, with the priority attribute being equal to their distance d. In each iteration the algorithm chooses a node with the smallest distance to the overlay tree (Step 6), and checks whether it is a multicast node (Step 7). In this case, the node's distance is updated to 0 (because the node is added to the overlay tree) and it is removed from the set F. Afterwards, for each chosen node, Steps 11 and 15 check its adjacent nodes on

whether their distance can be improved, and update the predecessors appropriately, similarly to Dijkstra's algorithm. However, in this case there are two important differences: the overlay predecessors can only be multicast nodes, hence Steps 14 and 15perform an additional check; moreover, previously extracted non-multicast nodes might be re-inserted in the priority queue in case their distance to the tree has improved due to the addition of new overlay nodes. The iteration ends when multicast nodes have been covered, hence the improvement in the average case complexity.

Algorithm 2. Efficient Minimum Spanning Tree Algorithm

Input: Weighted Graph $G(V, E, w)$, Multicast Node Set S, Root Node s.
Output: Predecessor Table π.

```
(1)  for all (v ∈ V) {d[v] ← ∞; π[v] ← NIL;}
(2)  d[s] ← 0;
(3)  Q ← V;
(4)  F ← S;
(5)  while (F ≠ ∅){
(6)      u ←EXTRACT-MIN(Q);
(7)      if (u ∈ S){
(8)          d[u] ← 0;
(9)          DEL(F, u);}
(10)     for each (v ∈ adj[u]){
(11)         if (d[v] > d[u] + w(u, v)){
(12)             d[v] ← d[u] + w(u, v);
(13)             if (v ∉ Q) {INSERT (Q, v);}
(14)             if (u ∈ S) π[v] ← u;
(15)             else π[v] ← π[u];}}}
```

3.2. Specification of MOST protocol

We now present the MOST multicast routing protocol which is based on overlay group shared trees. The protocol must be used in conjunction with a link state protocol, hence we choose to develop an extension to OLSR. One of the advantages of the overlay approach is that only the multicast nodes need to participate in the construction of the multicast tree, while the other nodes serve merely as relays and are not necessarily aware of the multicast communication. This fact facilitates the development of a peer to peer protocol which can be run only by the participating multicast nodes, hence it could be downloaded dynamically by a node whenever it decides to join a multicast communication.

The Optimized Link State Routing protocol [6] is a proactive table-driven routing protocol. An OLSR node uses Hello messages in order to detect neighbors and informs the entire network of its local topology by broadcasting TC messages. Broadcast traffic is relayed via Multi-Point Relay (MPR) nodes. MPR nodes are elected by their neighbors because they cover their two-hop neighborhood. That way broadcast traffic consumes less resources in order to be forwarded to all destinations. A node can either advertize the full neighbor link set (this mode is called full-OLSR), or the advertized link set can be limited to MPR links, *i.e.*, the neighbors that have elected this node as an MPR, while still guaranteing that the shortest paths can be computed.

In a fully distributed spanning tree design, the tree computation is performed independently by each group member. To proceed to the correct computation of the overlay tree, multicast nodes need to know the membership of their multicast group. Therefore, when a node wants to join or leave a group, it broadcasts a Join or Leave message to the entire network, via the optimized MPR-flooding mechanism used in OLSR. Join messages are sent periodically according to the *Join_Interval*, which is set by default to be equal to the TC message interval, *i.e.*, 5 s. The total protocol overhead is limited in these messages, independently of the number of groups and sources. In fact, MOST is well suited for managing numerous groups of small size, with arbitrary sources. Each group member must compute periodically the multicast tree to discover and maintain its overlay neighbors. In order to compute the multicast tree, the node needs information about network topology which is delivered by OLSR. The overlay neighbor set is a subset of clients sharing the same tree with that member, which are linked to it via unicast tunnels. The distance between overlay neighbors can be of one or several hops. Each client receives/retransmits multicast data from/to its overlay neighbors. The computed tree is a group shared tree, hence it must always be the same for all the clients. However, because of changes in the network topology and group membership, there is no guarantee that all clients hold the same tree. Consequently, to avoid loops there is a need to maintain a duplicate table in each client node. Moreover, some redundancy is introduced in data forwarding after tree updates to avoid packet losses. Notice that for any routing protocol to function properly, the rate of topology change must not be greater than the rate of state information propagation. This applies equally to unicast as well as multicast routing. Consequently, protocol parameters should be tuned accordingly. For example, the overlay computation period in MOST can be tuned according to the mobility, without inducing any additional control overhead.

3.2.1. Managing join and leave messages

The Join message contains the multicast group(s) address(es) that the sender has joined. The message format is the OLSR message format [6]. The Join message contains a list of multicast group addresses. Each multicast node maintains a membership table with the members of all the groups it belongs to. Upon receipt of a new Join message, each concerned node adds the new client to its membership table. The entries in the membership table are also associated with an expiration time, which is determined by the *VTime* field in the Join message header. After this time

period the entries are removed, unless another Join message is received, in which case the expiration time is updated. When a node wants to leave a group, it broadcasts a Leave message to the entire network but keeps acting as a group member if it receives data for a predefined transition period. Beyond this period, received packets are not retransmitted. Leave messages follow the same format as Join messages, hence the message contains the group addresses that the originator wishes to leave. We note that in case there is a multicast node failure or disconnection from the network, this event will be accounted for in the OLSR topology table. Therefore, other multicast nodes will act accordingly, as if the problematic node had left all multicast groups, and the multicast communication will not be affected.

3.2.2. Tree computation and maintenance

For each group, each client computes periodically the corresponding overlay tree, according to its membership table. Therefore, the client needs to maintain a table with its overlay neighbors. The update period is set by default to the Hello interval, *i.e.*, 2 s, which has been found empirically to provide optimal performance. If a new node joins the group or a client leaves the group, the tree is updated immediately. Whenever the overlay neighbors change after an update, the client considers both new, as well as older overlay neighbors for a transition period of 1 s, that is half the update period. The transition period is introduced to make it possible for all the clients to take into account tree changes, and to improve the packet delivery ratio. After that period, only new neighbors are considered. Finally, in order to be able to perform the overlay tree construction according to the previously described algorithm, nodes switch to full-OLSR mode whenever they join a multicast group. In this mode, nodes advertize their complete neighbor set. The reason for this is that it must be possible to compute the distance between each pair of group members.

3.2.3. Transmission and forwarding of data packets

Unlike common multicast protocols where data packets are transmitted in a broadcast mode, in MOST data packets are encapsulated in unicast packets before being forwarded to the overlay tunnels. Unicast transmissions present important additional benefits when the subnetwork layer in use is IEEE 802.11 [28], as is the case in most actual wireless networks. Firstly, the packet delivery ratio is significantly improved, since packets are retransmitted in case of collisions, while this is not the case for broadcast (or multicast) packets. Secondly, in most 802.11 variants (including b and g), the broadcast frames are transmitted by default at a lower rate than unicast frames. For instance, when 802.11b operates at a unicast data rate of 11 Mbps, the default broadcast rate is 2 Mbps, and most wireless card drivers do not offer the possibility to change it. Therefore, unicast tunnels can actually increase the available bandwidth, since data is sent at a much higher rate. Under these MAC constraints, MOST outperforms protocols using broadcast frames.

When a client receives a multicast data packet,[3] it checks whether the packet has already been received. If this is the case, the duplicate packet is dropped. Otherwise, the client forwards the packet to each of its overlay neighbors, except the one from which the packet was received. Source nodes act also as multicast group members, hence they simply send their data in unicast to their overlay neighbors.

Table 1
Common simulation parameters

MAC protocol	IEEE 802.11b (11 Mb rate)
Propagation model	Two ray ground
Transmission range	250 m
Packet size	1200 bytes
Traffic type	CBR
Number of iterations	5

4. Simulation results

In this section we perform ns-2 simulations in various scenarios aiming to verify the theoretical minimum spanning tree analysis. We also present some protocol performance measures in a mobile ad hoc network environment. In Table 1, we summarize the parameters that are common for all our simulations. We note that the 95% confidence intervals for the measured packet delivery ratios presented in the following are $\pm 2\%$ or less.

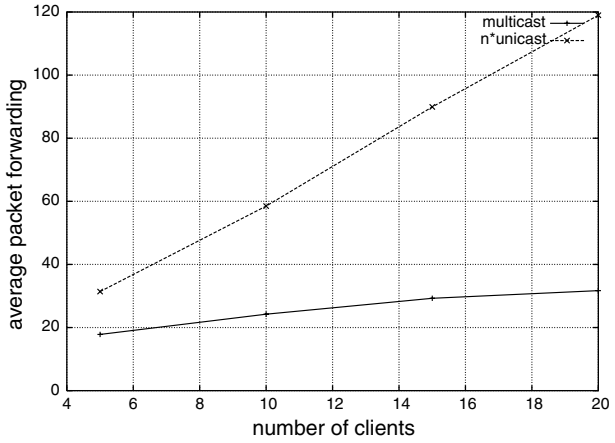
4.1. Comparison of multicast and unicast performance

We measure the average multicast cost for various group sizes, by counting the total number of times each packet is relayed to reach all destinations, using MOST. The unicast cost is determined in the same manner, by repeating the same simulations and considering OLSR unicast transmissions between each source–client pair.

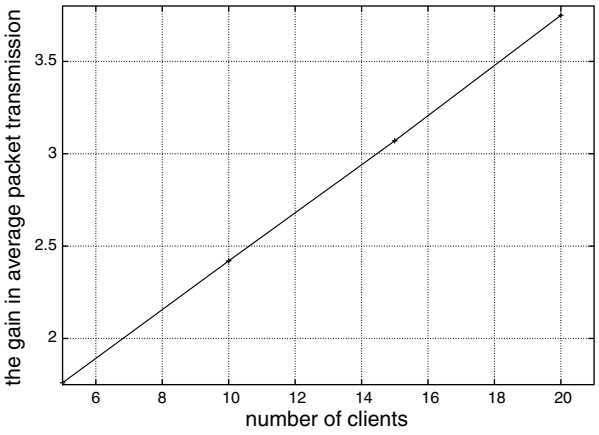
The simulation environment consists of a randomly generated topology of 100 wireless nodes forming an ad hoc network, in an area of 1500 m × 500 m. We consider group sizes ranging from 5 to 20 nodes (not including the source). Multicast groups and sources are chosen at random. The source node sends to the group CBR traffic of 64 kbps, with packets of 1200 bytes, for 150 s of simulated time. To obtain reliable results, simulations are conducted several times with five different seeds. The mean results are depicted in Fig. 5.

The analysis in [13] allows to compute an upper bound on the number of multicast packet retransmissions as a function of the number of group clients n. Namely, an upper bound for this parameter is: $\sqrt{2n} \times d_u$, where d_u is the average unicast distance between two nodes in hops. In Fig. 6, we compare the average number of retransmis-

[3] In fact all packets are received in unicast, so we refer here to the encapsulated multicast content.



(a) Average packet retransmissions.



(b) Gain of multicast routing.

Fig. 5. Comparison of multicast vs unicast to all destinations.

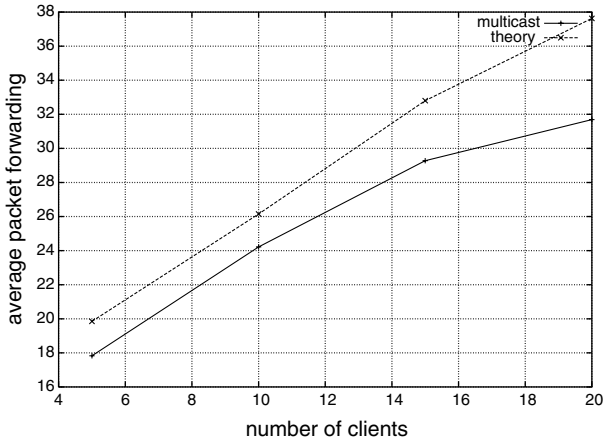


Fig. 6. Simulation results vs theoretical upper bound.

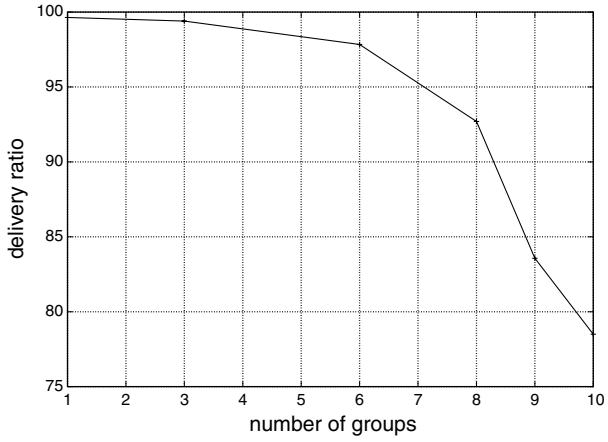


Fig. 7. Delivery ratio (%) vs number of groups in the network.

sions measured through simulations to the theoretical bound (where border effects are ignored). Although the analysis is performed in an asymptotic setting, we notice that the upper bound is also valid in these simulations. We note that the upper bound in Fig. 6 corresponds to the square root growth of the normalized multicast cost in Fig. 2.

4.2. Multicast performance vs throughput and group size

Simulations are conducted to determine the maximum source rate that maintains an acceptable delivery ratio in a multicast group. By acceptable we mean that its value is higher than 95%. We consider static topologies again since the goal is to find the saturation point of the network. We consider a 200 wireless nodes network in a 1800 m $\times$ 1800 m area, with one multicast group. We vary the number of clients as well as the source bit rate and we measure the packet delivery ratio, as shown in Fig. 7. We notice that the source node can transmit with a rate of up to 200 kbps with a delivery ratio higher than 99%. From a 250 kbps rate, the performance remains good for small groups but it decreases for large group sizes.

We also run other simulations by fixing the number of clients to 10 and varying the number of active multicast groups in a 300 nodes network. In each group a source is

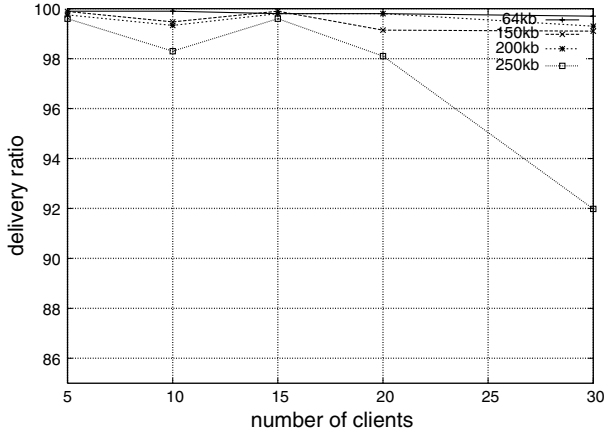


Fig. 8. Delivery ratio (%) vs group size for different source rates.

transmitting CBR traffic with a rate of 64 kbps. As shown in Fig. 8 we notice that the delivery ratio is very high until network saturation, with 8 groups of 10 clients.

4.3. Multicast performance with mobility

To evaluate the protocol performance with mobility, we consider a scenario consisting of a randomly generated topology of 200 wireless nodes in an area of 1850 m × 1850 m, and one multicast group in which an arbitrary source node sends a CBR traffic of 64 kbps for 300 s. We run simulations in which we vary the group members, the number of clients (from 5 to 20 nodes, not including the source) and the maximum mobility speed (from 1 m/s to 10 m/s). The mobility model is the Random Way-point model with a pause time of 10 s: nodes choose a random point in the network area and move to it with a constant speed chosen at random between 1 m/s and the maximum defined value; after they reach their destination, they remain idle for a period equal to the pause interval and then the same procedure is repeated. Moreover, we consider the following OLSR parameters: a Hello interval of 1 s and TC interval of 5 s; we note that the performance of MOST with mobility can be improved by using smaller intervals, at the cost of higher OLSR control message overhead. The simulations are again repeated several times with five different seeds. We measure the multicast packet delivery ratio and the traffic load caused by duplicate packets, and we depict the obtained results in Figs. 9 and 10.

As we can see, by varying the speed from 1 m/s, 5 m/s and up to 10 m/s, the delivery ratio remains acceptable, *i.e.*, higher than 95% for all groups of up to 20 clients. On the other hand, traffic load due to duplicate packets is higher when the mobility speed or the group size increase. This can be explained by the fact that any change in the topology due to mobility can affect the shared tree. In fact, each client is aware of these changes

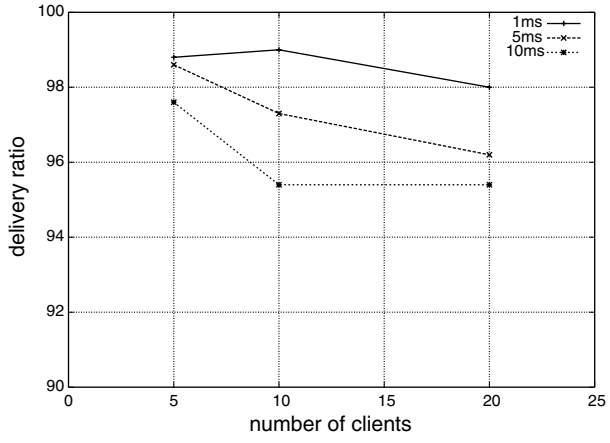


Fig. 9. Delivery ratio (%) vs group size with different mobility speeds.

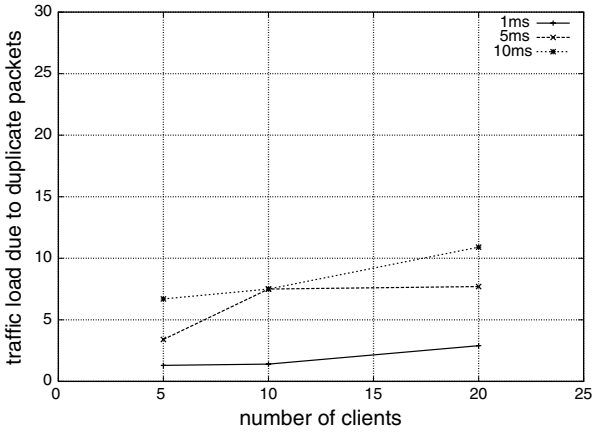


Fig. 10. Duplicate traffic load (%) vs group size with different speeds.

since it uses the OLSR protocol, and when it recalculates the overlay tree it enters a transition period, during which old and new overlay tree neighbors are maintained. A compromise can be found between the overhead due to duplicate packets and packet delivery ratio by setting a suitable transition period length. We notice that the duplicate traffic load in the network remains small compared to the total traffic load (10% in the worst case), hence the important advantage of improving the packet delivery ratio comes only with a modest performance cost.

5. Implementation overview

In this section we outline our complete implementation of the MOST protocol for Linux. An overview of the architecture is depicted in Fig. 11. The implementation consists of two modules: MDFP and OOLSR.

MDFP (Multicast Data Forwarding Protocol) is a forwarding protocol that enables point to multi-point data transfer. Multicast packets are captured and encapsulated in order to be forwarded inside a multicast tree. This module was developed for use with the MOLSR multicast protocol [29], and we adapted it to also support MOST. OOLSR

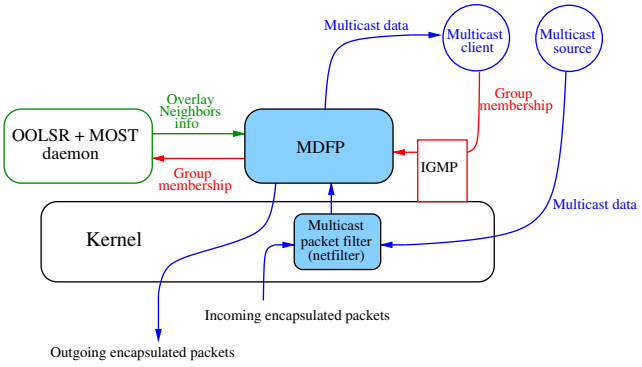


Fig. 11. Overview of multicast implementation.

(Object oriented OLSR) is INRIA's implementation of the OLSR protocol in $C++$ [30]. The core of MOST was implemented as an extension inside this module.

The OOLSR module with MOST extension is in charge of sending and processing Join and Leave messages, as well as computing and maintaining the overlay multicast tree, based on the network topology. The MDFP module is in charge of the actual forwarding of multicast data packets in the overlay tree. For this purpose, it performs encapsulation and decapsulation of data packets, and maintains a table in order to detect duplicate packet receptions. As shown in Fig. 11, the two modules constantly exchange information. The OOLSR daemon provides MDFP with up to date overlay neighbors information, which is all that is needed to perform the transmission and forwarding of multicast data. Conversely, MDFP communicates to the OOLSR daemon the group membership information concerning the node's OLSR interfaces. In fact, multicast client applications update the interfaces' IGMP information (*cf.* Internet Group Management Protocol [9]), and this information is interpreted by MDFP. Incoming multicast data packets are captured by the *netfilter* module in the kernel, following predetermined rules (such as a predetermined UDP port number). MDFP decapsulates the packets and passes them to the client applications, while it re-encapsulates them in order to forward them to the overlay neighbors. Similarly, data transmitted by a local multicast source is also captured by netfilter and processed by MDFP.

Finally, we note that the OOLSR module (including the MOST extension) can be loaded as a plugin in ns-2, hence the simulator shares the same source code as the real implementation. The simulations presented in the previous section were performed using this plugin.

6. Conclusion

In this paper, we established performance estimates for multicast routing vs unicast in massively dense ad hoc networks. We showed that multicasting can reduce the overall network load by a factor $O(\sqrt{n})$, for n multicast group members. Consequently, the total capacity of the network for data delivery is significantly increased. Although we used geometric arguments to justify this behavior analytically, we also proposed a protocol which uses only the information provided by a link state routing protocol. The Multicast Overlay Spanning Tree routing protocol was implemented as an extension to OLSR. The protocol is fully distributed, in the sense that each group member computes and maintains the shared multicast tree independently. Being based on overlay trees, it guarantees robustness, while it achieves good performance since OLSR provides topology information allowing to construct an optimal multicast overlay tree. Finally, MOST was tested via ns-2 simulations, and it was shown to achieve the theoretical performance estimates. MOST has also been tested in real network environments, hence we intend to provide measurement studies of the protocol performance in future work. Another interesting direction for future work consists in enhancing MOST with quality of service mechanisms.

References

[1] C. Adjih, L. Georgiadis, P. Jacquet, W. Szpankowski, Is the internet fractal: the multicast power law revisited, in: SODA, 2002

[2] S. Arora, Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems, J. ACM 45 (5) (1998) 753–782.

[3] F. Baccelli, K. Tchoumatchenko, S. Zuyev, Markov paths on the Poisson–Delaunay graph with applications to routing in mobile networks, Adv. Appl. Probab. 32 (1) (2000) 1–18.

[4] D. Bertsimas, G. Van Ryzin, An asymptotic determination of the minimum spanning tree and minimum matching constants in geometrical probability, Oper. Res. Lett. 9 (1990) 223–231.

[5] J.C.-I. Chuang, M.A. Sirbu, Pricing multicast communication: a cost-based approach, Telecommun. Syst. 17 (3) (2001) 281–297.

[6] T. Clausen, P. Jacquet (Eds.), Optimized link state routing protocol (OLSR), RFC 3626, October 2003, Network Working Group.

[7] C. Cordeiro, H. Gossain, D. Agrawal, Multicast over wireless mobile ad hoc networks: present and future directions, in: IEEE Network, vol. 17, no. 1, 2003.

[8] D.Z. Du, F.K. Hwang, A proof of Gilbert–Pollak's conjecture on the steiner ratio, Algorithmica 45 (1992) 121–135.

[9] W. Fenner, Internet Group Management Protocol, Version 2, RFC 2236, 1997.

[10] J. Garcia-Luna-Aceves, E.L. Madruga, The core-assisted mesh protocol, J. Select. Areas Commun. 17 (8) (1999) 1294–1380.

[11] P. Gupta, P.R. Kumar, Capacity of wireless networks, IEEE Trans. Inform. Theory IT-46 (2) (2000) 388–404.

[12] P. Jacquet, A. Laouiti, P. Minet, L. Viennot, Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models, RSRCP, Special issue on Mobility and Internet, December 2001.

[13] P. Jacquet, G. Rodolakis, Multicast scaling properties in massively dense ad hoc networks, in: SaNSO, Fukuoka, Japan, 2005.

[14] P. Jacquet, Geometry of information propagation in massively dense ad hoc networks, in: MobiHoc, 2004.

[15] J. Jetcheva, D.B. Johnson, Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks, in: Proceedings of the Second Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001), 2001.

[16] A. Laouiti, P. Jacquet, P. Minet, L. Viennot, T. Clausen, C. Adjih, Multicast Optimized Link State Routing, INRIA research report RR-4721, 2003.

[17] A. Laouiti, A. Qayyum, L. Viennot, Multipoint relaying: an efficient technique for flooding in mobile wireless networks, in: 35th Annual Hawaii International Conference on System Sciences (HICSS'2001), IEEE Computer Society, 2001.

[18] S. Lee, W. Su, M. Gerla, On demand multicast routing protocol in multihop wireless mobile networks, ACM/Baltzer Mobile Networks and Applications, special issue on Multipoint Communication in Wireless Mobile Networks, 2000.

[19] M. Liu, R.R. Talpade, A. McAuley, E. Bommaiah, AMRoute: Ad-hoc Multicast Routing Protocol. UMD TechReport 99-8.

[20] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (AODV) routing, RFC 3561, 2003.

[21] G. Phillips, S. Shenker, H. Tangmunarunkit, Scaling of multicast trees: comments on the Chuang–Sirbu scaling law, in *SIGCOMM*, pp. 41–51, 1999.

[22] E. Royer, C. Perkins, Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing, IETF, Internet Draft: draft- ietf-manet-maodv-00.txt, 2000.

[23] H. Sallay, O. Festor, A highly distributed dynamic IP multicast accounting and management framework, in: IFIP/IEEE Eighth International Symposium on Integrated Network Management, 2003.

[24] M. Steele, Growth rates of Euclidean minimal spanning trees with power weighted edges, Ann. Prob. 16 (1988) 1767–1787.
[25] S. Toumpis, A.J. Goldsmith, Performance bounds for large wireless networks with mobile nodes and multicast traffic, in: Interational Workshop on Wireless Ad Hoc Networks, Oulu, Finland, 2004.
[26] V. Vazirani, Approximation Algorithms, Springer-Verlag, 2001.
[27] J.E. Wieselthier, G.D. Nguyen, A. Ephremides, On the construction of energy-efficient broadcast and multicast trees in wireless networks, in: INFOCOM, 2000, pp. 585–594.
[28] IEEE 802.11 Standard, Wireless LAN Medium Access Control and Physical layer Specifications, 1997.
[29] MDFP. Available from: <http://hipercom.inria.fr/smolsr-molsr/>.
[30] OOLSR. Available from: <http://hipercom.inria.fr/oolsr/>.