QoS Preserving Topology Advertising Reduction for OLSR Routing Protocol for Mobile Ad Hoc Networks

Luminita Moraru, David Simplot-Ryl

To cite this version:

Luminita Moraru, David Simplot-Ryl. QoS Preserving Topology Advertising Reduction for OLSR Routing Protocol for Mobile Ad Hoc Networks. WONS 2006 : Third Annual Conference on Wireless On-demand Network Systems and Services, INRIA, INSA Lyon, IFIP, Alcatel, Jan 2006, Les Ménuires (France), pp.196-202. inria-00001028

HAL Id: inria-00001028

https://hal.inria.fr/inria-00001028

Submitted on 17 Jan 2006

QoS Preserving Topology Advertising Reduction for OLSR Routing Protocol for Mobile Ad Hoc Networks

Luminita Moraru and David Simplot-Ryl
IRCICA / LIFL, University of Lille 1, France
INRIA futurs, POPS research group
Email : {Luminita.Moraru, David.Simplot}@lifl.fr

Abstract— **Mobile ad hoc networks (MANET) are formed by mobile nodes with a limited communication range. Routing protocols use a best effort strategy to select the path between a source and a destination. Recently, mobile ad hoc networks are facing a new challenge, quality of service (QoS) routing. QoS is concerned with choosing paths that provide the required performances, specified mainly in terms of the bandwidth and the delay. In this paper we propose a QoS routing protocol. Each node forwards messages to their destination based on the information received during periodically broadcasts. It uses two different sets of neighbors: one to forward QoS compliant application messages and another to disseminate local information about the network. The former is built based on 2-hop information knowledge about the metric imposed by the QoS. The latter is selected in order to minimize the number of sent broadcasts. We provide simulation results to compare the performances with similar QoS protocols.**

I. INTRODUCTION

In the context of mobile ad hoc networks[1], new challenges are raised for routing protocols. Nodes are communicating through wireless links with limited range. Each message sent by a node will be received only by the nodes located in this communication range. Additionally, links between nodes are not stable due to the nodes mobility.

Routing protocols are finding paths between a source and a destination that do not communicate directly. They consider the number of hops as criterion for finding optimal routes between nodes. In the case of QoS routing [2], new constraints become prioritary (bandwidth, delay) and new metrics must be considered. When a packet coming from the application layer is routed to its destination, the links between nodes are relevant only if they are compliant with the QoS requirements. Many of the solutions that have been proposed to this problem are enhancements of existing routing protocols.

We consider the particular situation of proactive protocols, where each node stores routing tables with all known destinations in the network. Hosts are aware of network topology due to the routing related information, periodically propagated into the network. Each node sends periodically broadcasts about the links with its neighbors. Existing proactive protocols (e.g. OLSR [3]) minimize the number of broadcasts by selecting only a subset of neighbors, multipoint relays (MPR) [4], to relay messages containing routing related information. The MPR set of a node is computed between direct neighbors, by a greedy heuristic, to cover all neighbors at a distance of 2 hops. The same set of nodes is used for packets routing.

When guaranteed QoS is demanded, an option is to modify existing protocols to use only the links respecting QoS requirements. This will impose additional conditions to the neighbors subset selected as relays, thus the number of selected neighbors and the network traffic are increased.

This paper presents a method for QoS paths selection, based on network topology complexity reduction. Only the neighbors that are providing maximum bandwidth links are advertised. In our solution, we determine the 1-hop neighbors representing the best paths to the set of 2-hop neighbors, in terms of a specific metric. First we eliminate from redundant paths, the worst performance link. Since each node has complete knowledge only until the 2 hop distance neighbors, redundant paths are represented by nodes that are both 1-hop and 2-hop neighbors. Then, we are making the selection considering a specific QoS metric. By selecting only nodes providing optimal links, we are reducing the complexity of network topology, while preserving the connectivity of the network and the availability of paths. QoS enabled routing uses selected neighbors set when it forwards application messages. Therefore, the selection is flooded into the entire network. We use MPR sets to flood the selection of a node.

The paper is organized as follows: first a presentation of existing QoS protocols is made. Next section contains a description of OLSR protocol, for which we proposed an enhancement, followed by the description of the algorithm used for advertised set selection, for concave constraints (e.g. bandwidth) in section IV and for additive constraints (e.g delay) in section V. Experimental results are presented in section VI and conclusions in section VII.

II. PREVIOUS WORK

QoS routing protocols developed for mobile ad hoc networks [5] are extending classic, best effort routing algorithms for MANET.

On demand routing protocols are using different communication models in order to satisfy the QoS requirements, e.g. TDMA (Time Division Multiple Access) or CDMA (Code Division Multiple Access) over TDMA. The issues raised are bandwidth or delay calculation and resource reservation during path discovery. An enhanced version of Ad-hoc On demand Distance Vector (AODV) protocol for QoS support [6] introduces a mechanism for resource reservation simultaneous with path discovery. An extension of Dynamic Source Routing (DSR) protocol is presented in [7]. It deals with common problems in TDMA environment for bandwidth reservation (e.g. race condition, parallel reservation problem). Temporally Ordered Routing Algorithm (TORA) extension [8] chooses from the available paths the shortest path compliant with the QoS requirements. The disadvantage is that they are operating not only into the network but also into Medium Access Control (MAC) layer.

From the reactive protocols category, an extension of OLSR for optimal routes in terms of QoS requirements was proposed in [9]. QOLSR proposes a heuristic for MPR selection and imposes several conditions for these nodes, in order to provide an optimal path, both in terms of hop distance and QoS metric. QOLSR has the disadvantage of increasing the number of MPR relays, thus the number of broadcasts in the network.

Another approach is core-extraction distributed ad hoc routing (CEDAR) protocol [10]. It determines a core dominating set. Only the nodes in this set are aware of core topology and of the metric of the neighbor links. This limits the number of broadcasts, compared with the control flooding of reactive protocols.

III. OLSR PROTOCOL ADAPTATION

Optimized Link State Routing (OLSR) protocol is a table driven protocol for MANET.

It maintains tables containing all the necessary data for finding a path to any other node in the network. In order to keep up to date routes, it regularly propagates routing information. It uses two types of messages: *HELLO messages* for neighborhood discovery and *topology control (TC) messages* for entire network topology discovery. *HELLO messages* are advertising the neighbors and MPR sets, while *TC messages* are disseminating network topology information necessary for building routing tables. MPR sets are enough to compute best routing path.

By using different sets of nodes for routing and topology advertising, new data structures are added to the *information base* of each node. Similarly to OLSR each node stores the 1 and 2-hop neighbors, MPR and MPR selector sets. Additionally each node will maintain the QoS Advertised Neighbor Set (QANS), which provides optimal connectivity based on the imposed metric and a list of QANS selectors: neighbors that selected it in their QANS set.

Topology information maintained at each node is retrieved from the TC messages and contains the list of all know destinations in the network together with the list of the last hop used to reach them. In OLSR this list contains the links of a node with its MPR selectors. In our case, these links are replaced in the TC messages by the QANS selectors set. Each node that receives a TC message will broadcast it only if it is in the MPR list of the last sender of the message.

IV. TOPOLOGY FILTERING FOR BANDWIDTH

A. Graph density reduction

Bandwidth constraint routing is based on finding routes in a network that maximize this criterion. A node has at most information regarding the presence of 1-hop and 2-hop neighbors and the metric of all 1-hop neighbors links. Based on link metric each node reduces the broadcasted information only to information needed to compute paths with the respect to constraints.

We consider the model of a network represented by a graph $G = (V, E)$, where V is the set of vertices in the graph, associated to the network nodes and E is the set of edges, representing links between nodes. Each communication link is characterized by a bandwidth value. Let B be the value of the maximum bandwidth link in the network. Then, we can define b, the bandwidth function that maps the set of edges E to the interval $]0, B]$. If the links are bidirectional, function b is considered to be symmetric (*i.e.* $b(u, v) = b(v, u)$). Bandwidth is a concave constraint, the bandwidth of a path p is defined by the minimum bandwidth link on that path. This means that for $p = \{a_0, a_1, \dots, a_n\}$, the bandwidth b_p of p is equal to:

$$b_p = \min_{0 \le i < n} \{b(a_i, a_{i+1})\}.$$

We will present below the method used for reducing the density of the graph. It is based on the situation where a node n_2 is a common neighbor for both a node u and another 1-hop neighbor of u, n_1. A triangle is generated in the graph. This is often the case of networks represented by a dense graph. Each node will maintain locally two paths to both neighbors (*e.g.* between n_1 and n_2 there are $p_1 = \{n_1, n_2\}$ and $p_2 = \{n_1, u, n_2\}$), characterized by the bandwidths: b_{p_1} and b_{p_2}. We can reduce the density of the graph by eliminating from the triangle formed by u, n_1 and n_2 the link with the minimum bandwidth.

Fig. 1 represents an example. In 1(a), $b_{p_1} = 3$ and $b_{p_2} = 4$. This makes p_2 the preferred option when maximum bandwidth routes are necessary. Both (n_1, n_2) and (n_2, n_3) have redundant paths with better metric value, as shown in 1(b) and they are eliminated.

Let us define the graph $G' = (V', E')$ containing the remaining set of edges:

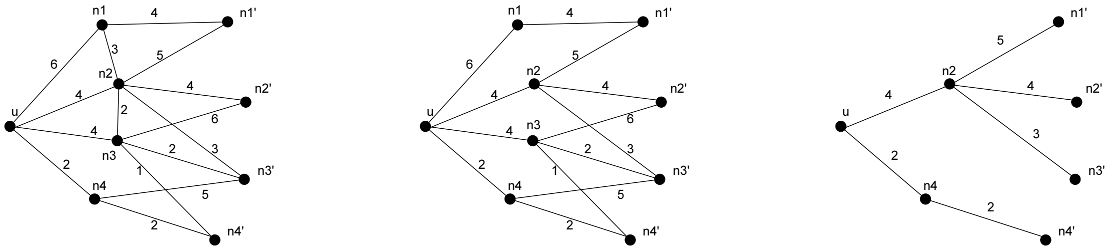
$$\begin{aligned} E' \quad = \quad & \{(u, v)\, in\, E \mid \nexists w\, such\, that\, (u, w), (v, w) \in E \\ & \wedge b(u, v) \le \min(b(u, w), b(v, w))\}. \end{aligned}$$

This graph reduction is a variation of Relative Neighborhood Graph (RNG) [11].

For a weight function f, the RNG graph,$G_{RNG} = (V, E_{RNG})$ of G, imposes the following condition, for an edge $(u, v) \in E$ between vertices u and v to exists:

$$\forall w \in V, w \neq u\, and\, v, f(u, v) \ge \max(f(u, w), f(v, w)).$$

Similarly, for the bandwidth metric, G' will represents the initial graph reduced to the RNG, which uses the bandwidth as weight function instead of distance.



(a) (b) (c)

Fig. 1. Example of bandwidth QANS selection for a node

In the case of two equal minimum links, another two criteria are evaluated in order to choose the link that will be eliminated. They are based on nodes IDs comparison, since each node is identified by an ID, unique in the network. First, the nodes with the minimum ID of each link are compared. The link with the smallest value for the minimum ID node of the link is eliminated. If the minimum is defined by a common node of the both links, the elimination is based on maximum ID node.

Let us consider $f(u,v) = (b(u,v), \min(id(u), id(v)), \max(id(u), id(v)))$, and the order relation $\leq$ defined on triples:

$$(x,y,z) \leq (x',y',z') \Leftrightarrow \quad x < x' \vee (x = x' and y < y') \vee (x = x' \wedge y = y' \wedge z < z'). \quad (1)$$

By applying all the three criteria, we are assured that all the triangles are eliminated, and none of the 1-hop neighbors is also in the 2-hop neighbors list.

Similar with the properties of a RNG graph, G' preserves the connectivity and the maximum bandwidth paths between any two vertices, while reducing the density of the graph.

The heuristic is presented in *Algorithm 1*.

Algorithm 1 Graph density reduction

```
Let $N(u) = \{n_1, n_2, \ldots, n_n\}$ be the list of 1-hop neighbors of the current node $u$.

function GET_BWRNG(u)
    N'(u)=N(u)
    for each $v$ in $N'$ do
        for each $w$ in $N(v) \cap N(u)$ do
            if $f(u,v) < f(u,w) \wedge f(u,v) < f(w,v)$ then
                remove $v$ from $N'(u)$
                break
            end if
        end for
    end for
return $N'(u)$
end function
```

B. Advertised neighbor set selection

From the reduced graph, we will select the neighbor set that preserve maximum bandwidth paths. It is computed by each node, base on 2-hop neighbors information.

The 1-hop neighbors are evaluated in the descendant order of the bandwidth of the link with the current node, u. A 1-hop neighbor of u, n_i is added to the set of advertised neighbors A only if it provides a maximal bandwidth path between the node u and at least one of its 2 hop neighbors. The evaluation stops when all the maximal bandwidth paths between the node u and the 2 hop neighbors are found.

Let n_j be the 1-hop neighbor that represents the path with maximum bandwidth between u and the 2 hop neighbor n'_i. It is equivalent with:

$$\min\{b(u,n_j), b(n_j,n'_i)\} \geq \min\{b(u,n_k), b(n_k,n'_i)\}, \\ \forall k = \overline{1:n} \wedge n_k \in N(u) \cap N(n'_i)$$

This relation is used to evaluate each 1-hop neighbor. Algorithm 2 returns the set of neighbors defining maximum bandwidth paths.

Algorithm 2 Select advertised neighbors set

```
Let $N(u) = \{n_1, n_2, \ldots, n_n\}$ be the list of 1 hop neighbors of u.

procedure GET_BW_QANS(u)
    Start with empty sets $A$ and $N'_j$.
    for each 2 hop neighbor $n'_i$ do
        determine $b_{max}(u, n'_i)$
    end for
    for each node $n_j \in N(u)$ do
        for each node $n'_i$ in $N(N(u)) \cap N(n_j)$ do
            if $b(u,n_j) \geq b_{max}(u,n'_i)$ then
                if $b(n_j,n'_i) \geq b_{max}(u,n'_i)$ then
                    add $n'_i$ to $N'_j$
                end if
            end if
        end for
        if $N_j$ not empty then
            add $n_j$ to A.
        end if
    end for
end procedure
```

There can be more than one maximum bandwidth path to a 2 hop neighbor in the selected set A. Each 1-hop neighbor n_i will define a maximum bandwidth path for a set N'_i of neighbors such that:

$$\bigcap_{i=1}^{n} N_i = N(N(u)).$$

In order to further optimize the dimension of QANS sets, we consider the following greedy method (implemented by algorithm 3), for removing nodes providing redundant paths. At the beginning both the set A'of neighbors and the set N' of 2-hop neighbors covered by the nodes in A' are empty. Each time the node from A that provides the greatest number of maximum paths to 2 hop neighbors not already in N' is added to A' and the covered neighbors in N'. The selection stops when all the 2 hop neighbors are covered. A' will represent the QANS set. An example of selection for the presented algorithm is shown in Fig. 1. After the evaluation of all links bandwidth of the graph in 1(b), only n_2 and n_4 are selected in 1(c).

Algorithm 3 Optimized advertised neighbors set

```
Start with empty sets A' and N'.
procedure REDUCE_BW_QANS(u)
    while $N' \neq N$ do
        Add to A' $n_j$ for which
            $N_j/N' = \max_{0 \leq i < n} N_i/N'$
        .
        Add elements from $N_j$ to $N'$.
    end while
end procedure
```

C. Proof of correctness

We have to prove that our algorithm 3 generates topology information which are sufficient to compute maximum bandwidth paths. We can notice that this statement is only needed for nodes which are not directly connected. In order to obtain this proof of correctness, we use three steps: (a) prove that the *graph density reduction* preserves maximum bandwidth (this property includes connectivity preservation), (b) prove that *advertised neighbor set selection* preserves maximum bandwidth between 2-hop neighbors, and (c) prove that 2-hop maximum bandwidth preservation is enough to guarantee maximum bandwidth preservation for any couple of nodes distant of at least two hops.

Concerning *graph density reduction*, we show that for all couple of nodes (u, v) and paths p between u and v in G, then there exist a path p' between u and v such that $b(p) \leq b(p')$. For a path $p = \{a_0, a_1, \ldots, a_k\}$ in G, we show how to build the path p'. Let us consider removed edges in ascendant order (according to the order defined in eq. 1). Each time that an edge (x, y) contained in p is removed, we apply the following operation. If (x, y) is deleted from the initial graph, it means that there exist two links (x, z) and (z, y) such that $f(x, y) < f(x, z)$ and $f(x, y) < f(z, y)$. By definition of the function f and of the order, it implies that $b(x, z) \geq b(x, y)$ and $b(z, y) \geq b(x, y)$. Moreover, these two links have not been removed yet and we can simply replace the sub-path $\{x, y\}$ by $\{x, z, y\}$. Since the number of edges is finite, when the process ends, we have a path with higher or equal bandwidth.

For the optimality of our *advertised neighbor set selection* algorithm for 2-hops neighbors in G', it suffices to observe that maximum bandwidth paths in G' between 2-hops neighbors cannot be longer than two hops. Let us consider a loop-free path $p = \{a_0, a_1, \ldots, a_k\}$ in G between $u = a_0$ and $v = a_k$, one of its 2-hops neighbors in G, such that $\forall 1 \leq i < k$ the intermediate node a_i in a 1-hop neighbor of u in G. We show that k is equal to two. Indeed, if k is greater than 2, it means that a_2 is a 1-hop neighbor of u. It implies that the edges (a_0, a_1), (a_1, a_2) and (a_0, a_2) exist in G. However, triangles cannot exist in G because at least one of the edges satisfies the condition to be removed compared to the two other ones. Because our algorithm preserves maximum bandwidth 2-hop paths, it is enough to guarantee bandwidth preservation between 2-hop neighbors.

Now, we show that the knowledge of maximum bandwidth path between 2-hop neighbors is enough to compute maximum bandwidth path between two arbitrary nodes distant of at least two hops. More precisely, for a loop-free path $p = \{a_0, a_1, \ldots, a_k\}$ in G with $k \geq 2$, we show by induction that that we can compute a path p based on 2-hop maximum bandwidth path such that $b(p) \leq b(p)$. If $k = 2$, the property simply holds because of previous statement. If $k > 2$, we know by induction that the subpath $p_1 = \{a_0, \ldots, a_{k-1}\}$ can be replaced by a subpath $p_1 = \{b_0, \ldots, b_l\}$ which use only knowledge of 2-hop maximum bandwidth path and such that $b(p_1) \leq b(p_1)$ (note that we have $a_0 = b_0$ and $b_l = a_{k-1}$). Because G does not contains triangles, the node b_{l-1} in p_1 is a 2-hop neighbor of a_k. From induction hypothesis, the subpath $\{b_{l-1}, b_l = a_{k-1}, a_k\}$ can be replaced by a 2-hop maximum bandwidth path $\{b_{l-1}, c, a_k\}$. In conclusion, we can compute a path $p = \{a_0 = b_0, b_1, \ldots, b_{l-1}, c, a_k\}$ with a higher of equal bandwidth.

These steps are enough to show that our algorithm guarantees bandwidth optimality for nodes distant of at least 2-hops (in G or G since G is a reduced graph of G). The proof of this optimality is simplified because of the use of G which does not contains triangles.

V. TOPOLOGY FILTERING FOR DELAY

A. Graph density reduction

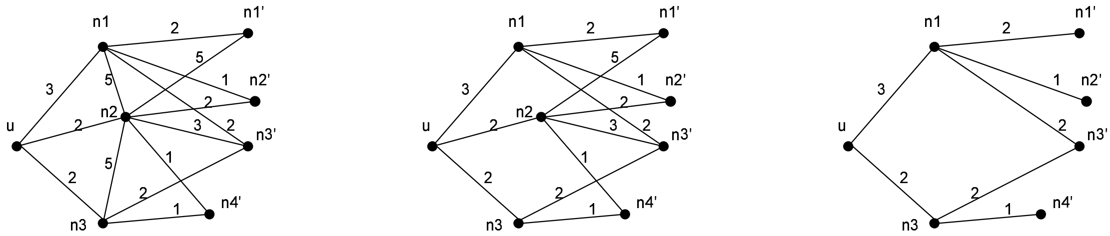
Delay is another demanding constraint for QoS routing, especially in the case of multimedia applications. The difference is that the delay of each link is added to the overall value.

For evaluating delay constrained routing we will use the same representation of a network by the graph $G = (V, E)$. If D is the value of the maximum delay link, then a link's delay value is defined by a function d defined on the set of edges E with values in the interval $[0, D]$. The delay is an additive metric. This means that for a path p between nodes u and v,

$$p = \{u, u_1, u_2, \ldots, v\},$$

the delay d_p is defined on $[0, D_p]$ and is

$$d_p = d(u, u_1) + d(u_1, u_2) + \ldots + d(u_n, v).$$



(a) (b) (c)

Fig. 2. Example of delay QANS selection for a node

For reducing the density of the graph we consider again the case of a triangle in the network, generated by u, a common neighbor of n_1 and of n_2, also neighbors.

Let $u, n_1 \, and \, n_2 \in V$ such that $(u, n_1), (n_1, n_2)$ and $(n_2, u) \in E$. Similar with the bandwidth we will reduce the density of the graph by removing the worst performance edge from the triangles generated by 1 hop neighbors. An edge is the worst performance edge if it has a delay greater or equal than a 2 hop path between the same nodes. An worst performance edge (u, n_1) is characterized by the property: $\exists n_2 \in V$ such that $d(u, n_1) \geq d(n_1, n_2) + d(u, n_2) \, if \, d(n_1, n_2) \neq 0 \, and \, d(u, n_2) \neq 0$.

Algorithm 4 Graph density reduction

```
Let $N(u) = [n_1, n_2, \ldots, n_n]$ be the list of 1 hop neighbors of $u$.
Let $N'_j$ be the set of 2 hop neighbors covered by $n_j$.

function GET_DELAYREDUCEDGRAPH(u)
    $N'(u) = N(u)$
    for each $v$ in $N'_u$ do
        for each $w \in N(v) \cap N(u)$ do
            if $f(u, v) \geq f(u, w) + f(w, v)$ then
                remove v of N'(u)
                break
            end if
        end for
    end for
return $N'(u)$
end function
```

By removing all the edges (u, n_1) with the property above from E, nor the connectivity neither the values of minimum delay paths are not affected.

Similar with the RNG, removing the greatest delay edge from a triangle does not influence the connectivity of the graph. If one of the edges has a delay equal with 0, then the other two links will be both removed. This situation is avoided by imposing the last condition.

In order to discuss the preservation of minimum delay paths value, we will consider a graph, G' obtained by removing all the edges in E with the property above. If the set of minimum delay paths is represented by P, then $\forall p \in P$, $\exists p'$ in P', the set of minimum delay paths in G' such that $d_p(p') = d_p(p)$. Indeed, if $d(n_i, n_{i+1}) \geq d(n_i, n'_i) + d(n'_i, n_{i+1})$, for each path $p = \{u, n_1, n_2, \ldots, n_i, ni + 1, \ldots, v\} \, in \, P$, there is a path $p' = \{u, n_1, n_2, \ldots, n_i, n'_i, n_{i+1}, \ldots, v\} \, in \, P$ with the property that $d_p \geq d_{p'}$.

B. Advertised neighbor set selection

The next step is to select the subset QANS of nodes of G' that provides complete network connectivity through minimum delay links. Although the procedure above will not remove all the triangles from the network, it assures us that when they still exists, the minimum delay path is the direct one. Therefore, in order to find the QANS set, is necessary to remove from the list of 2-hop neighbors of u, those that are also 1-hop neighbors.

Similarly with the first algorithm, a 1-hop neighbor of u, n_i is added to the set A only if it provides a minimum delay path between the node and at least one of its 2 hop neighbors. The algorithm stops when all 1-hop neighbors are evaluated.

Algorithm 5 Select advertised neighbors set

```
Let $N(u) = [n_1, n_2, \ldots, n_n]$ be the list of 1 hop neighbors in G'.
Let $N'_j$ the set of 2 hop neighbors covered by $n_j$: $N'_j = N(N(u)) \cap N(n_j)$

procedure GET_DELAY_QANS
    start with empty sets QANS and $N'_j$.
    for each 2 hop neighbor $n'_i$ do
        determine $d_{min}(u, n'_i)$
    end for
    for each node $n_j \in N(u)$ do
        for each node $n'_i$ in $N(n_j)$ do
            if $d(n_j, n'_i) + d(u, n_j) = d_{min}(u, n'_i)$ then
                add $n_j$ to $N'_j$
            end if
        end for
        if $N_j$ not empty then
            add $n_j$ to QANS.
        end if
    end for
end procedure
```

The selected set will preserve the minimum delay paths. For each path p in the graph G, we can build a path p' in the graph G', with the length smaller or equal to the length of p and with the same delay.

Let $p = \{u, n_1, n_2, \ldots, n_{i-1}, n_i, n_{i+1}, \ldots, v\}$. Let us suppose that a node n_i it is not in QANS subset of n_{i-1}. Then it exists n'_i such that $n'_i \in$QANS and the delay $d_p((n_{i-1}, n'_i), (n'_i, n_{i+1})) \leq d_p = ((n_{i-1}, n_i), (n_i, n_{i+1}))$.

There can be more than one minimum delay path to a 2 hop neighbor in the selected set QANS. This means that the QANS set can be further minimized. We consider the same greedy method for selecting a smaller set. At each step the 1-hop neighbor that covers the maximum number of 2 hop neighbors not covered yet is selected. The selection stops when all the 2 hop neighbors are covered. The algorithm is identical with the bandwidth case.

Fig. 2 illustrates an example. The initial graph is represented in 2(a). In 2(b) the links with the worse performance metric are eliminated. In 2(c) is selected the minimum set of neighbors on best performance paths to the 2-hop neighbors set.

VI. SIMULATION

We implemented a simulator to evaluate the performances of the proposed algorithm. Tests were made with a static network of 200 nodes. Nodes are randomly distributed in order to obtain a given average number of neighbors. We compare our algorithm to QOLSR protocol.

Both QOLSR and OLSR-QANS are enhancements to OLSR protocol and aim at providing QoS routes. In a proactive protocol, each node declares the links with its neighbors, by sending broadcasts into the network. Network traffic is influenced by the size of packets and the number of broadcasts. The size of packets depends on the number of declared links. The number of broadcasts depends on the number of neighbors selected by a node to retransmit a message. We will compare the subset of neighbors selected for QoS routing and for network control messages retransmission. QoS performances are evaluated by the number of paths, that respect the QoS requirements, successfully found.

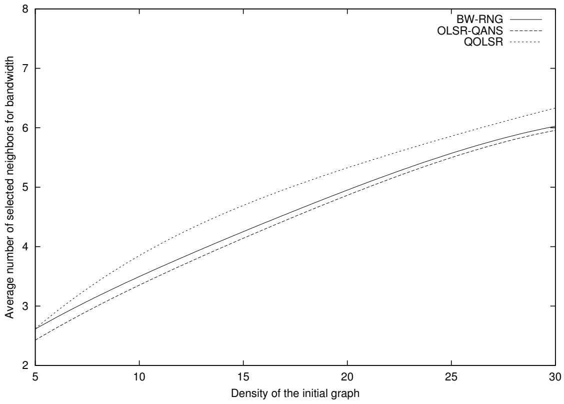


Fig. 3. Maximum bandwidth neighbors selection

We computed the number of neighbors selected to route messages. Fig. 3 compares the average number of 1-hop neighbors used for QoS path. The metric used is the bandwidth. The average size of 1-hop neighbors in the bandwidth RNG graph is smaller than the QOLSR selection. Accordingly, the 1-hop set selected by OLSR-QANS is smaller than QOLSR selection for bandwidth with 12%.

Fig. 4 compares the number of nodes selected for broadcasting network information. Our protocol uses MPR sets for broadcasting, while QOLSR uses the same set of nodes as the one for QoS paths. MPR sets are smaller than QOLSR because they have only the constraint of 2-hop neighbors to cover. QOLSR selection has to fulfill additional requirements imposed by the QoS metric.

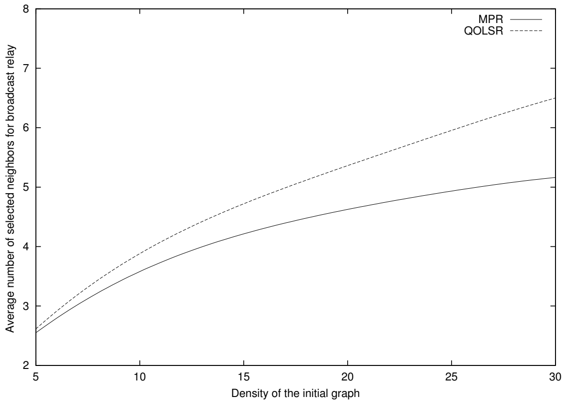


Fig. 4. Broadcast forwarding neighbors selection

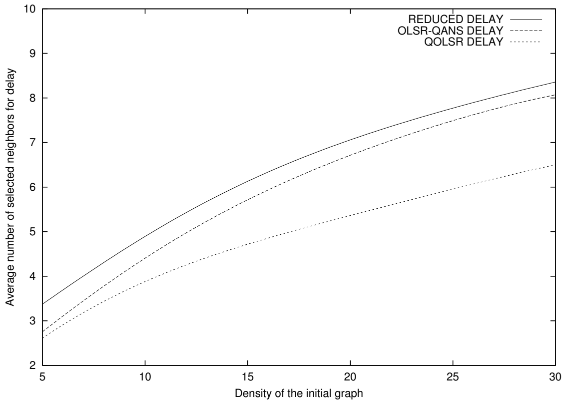


Fig. 5. Minimum delay neighbors selection

In Fig. 5 are presented the results of selection for delay. The selection of QOLSR is smaller with 18%. The size of 1-hop set in the reduced graph for delay is influenced by the conditions imposed to worse performance links, which are more restrictive than in the case of bandwidth.

In Fig. 6 we analyse the performances from the point of view of the bandwidth metric requirements. We present the dependence of path bandwidth on the average density. Paths are computed with a Dijkstra algorithm modified for concave constraints. The bandwidth gain obtained by using QoS protocols in OLSR-QANS compared with the bandwidth of the path in the QOLSR graph is relatively constant and has the average value of 8%. The bandwidth gain is obtained with a smaller set of 1-hop neighbors.

Similarly, fig. 7 shows the raport between the delay obtained for paths computed in the case of the two protocols. Paths are computed with Dijkstra algorithm, that considers the delay as the cost associated to links. The raport between the delays depends on the density of the network. For densities greater than 20, minimum delay of the paths in OLSR-QANS graph

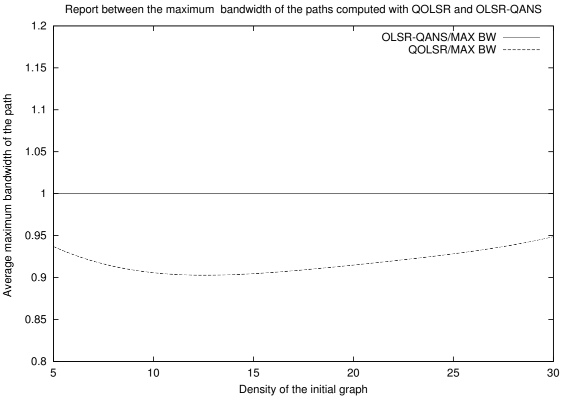


Fig. 6. Path average bandwidth comparison

is with 30% smaller than in QOLSR graph. This is obtained with the increase of 18% in the number of 1-hop neighbors used for QoS routing.

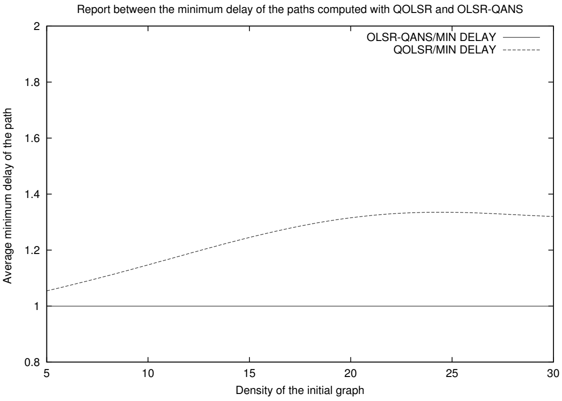


Fig. 7. Path average delay comparison

A concern in QoS routing is route computation. The length of the paths is influenced by the elimination of both links and nodes from the initial graph. We compared the distorsion of maximum bandwidth paths for the two protocols. For bandwidth the routes computed with QOLSR are smaller, as it can be seen in Fig. 8. For delay, the distorsion is influenced by the density of the graph, for higher densities, the distorsion of OLSR-QANS becomes smaller than QOLSR, as can be seen in Fig. 9 .

VII. Conclusions

In this paper we presented a QoS routing protocol. It is an extension of OLSR, a proactive routing protocol for MANET. We presented the modifications made to packets structure and the set of nodes selected for forwarding the messages. We explained the algorithm used to select the set of neighbors that respects the QoS requirements and we proved the correctness of the selection methods. Then we compared it with another extension of OLSR for QoS routing, QOLSR. The results shows that we obtained better performances in terms of QoS metric than QOLSR and a smaller number of broadcasts. Like all the other QoS protocols, our protocol has the drawback of routing QoS compliant packets on paths with a greater length that the best effort ones. Future works include the evaluation of the protocol when both bandwidth and delay are considered.

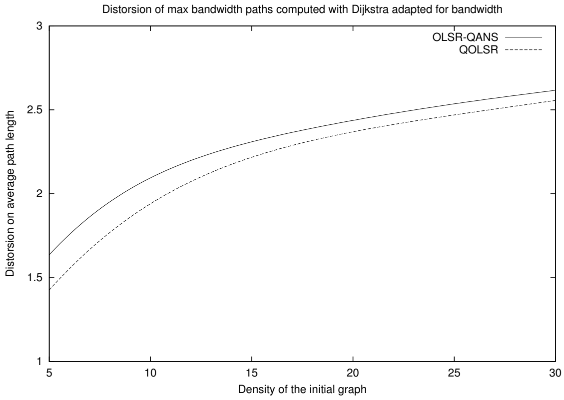


Fig. 8. Distorsion of the length of the maximum bandwidth paths

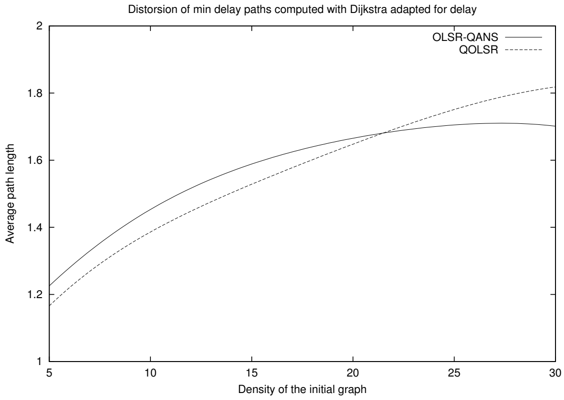


Fig. 9. Distorsion of the length of the minimum delay paths

References

[1] "Mobile ad-hoc network ietf working group." [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html

[2] B. R. E. Crawley, R. Nair and H. Sandick, "A Framework for QoS-based Routing in the Internet," RFC 2386, Aug. 1998. [Online]. Available: http://rfc.net/rfc2386.html

[3] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[4] L. V. Amir Qayyum and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobilewireless networks," in *Proceedings of the 35th Hawaii International Conference on System Sciences*, vol. 09.

[5] I. Jawhar and J. Wu, *Quality of Service Routing in Mobile Ad Hoc Networks*, 2004.

[6] I. Gerasimov and R. Simon, "A bandwidth-reservation mechanism for on-demand ad hoc path finding," in *Simulation Symposium, 2002. Proceedings. 35th Annual*, Apr. 2002, pp. 27 – 34.

[7] I. Jawhar and J. Wu, "Race-free resource allocation for qos support in wireless networks," pp. 179–206, 2005.

[8] I. Gerasimov and R. Simon, "Performance analysis for ad hoc qos routing protocols," in *mobiwac*, 2002, p. 87.

[9] H. Badis and K. A. Agha, "Optimal path selection analysis in ad hoc networks," LRI: Laboratoire de Recherche en Informatique, Tech. Rep., Aug. 2004.

[10] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm," in *INFOCOM (1)*, 1999, pp. 202–209. [Online]. Available: citeseer.ist.psu.edu/article/sinha99cedar.html

[11] G. Toussaint, *The relative neighborhood graph of a finite planar set*, 1980, pp. 261–268.