R
A
P
P
O
R
T

D
E

R
E
C
H
E
R
C
H
E

L R I

OPTIMAL PATH SELECTION ANALYSIS IN AD HOC NETWORKS

BADIS H / AL AGHA K

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud – LRI

08/2004

Rapport de Recherche N° 1391

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 490
91405 ORSAY Cedex (France)

Optimal Path Selection Analysis in Ad Hoc Networks

Hakim Badis*† and Khaldoun Al Agha*†
*LRI Laboratory, University of Paris XI, 91405 Orsay, France
†INRIA Laboratory, Rocquencourt, 78153 Le Chesnay, France

Abstract

This paper aims to describe a novel QoS routing protocol for ad hoc networks based on the QOLSR protocol. The proposed QoS routing protocol is based on the standardized OLSR. The OLSR protocol is an optimization of the classical link state algorithm tailored to the requirements of a mobile wireless LAN networks. The standard OLSR introduces an interesting concept, the multipoint relays (MPRs), to mitigate message overhead during the flooding process. The heuristic for the selection of MPRs limits their overall number in the network and ensures minimum overhead. However, there is no guarantee that OLSR finds the optimal path in terms of QoS requirements. We propose two heuristics for the selection of MPRs based on QoS measurements that allow QOLSR to find optimal widest paths on the known partial network topology. We demonstrate that these paths are also optimal widest paths in the whole network topology. Afterwards, we introduce the QOLSR routing table calculation based on multiple metrics. The idea is to introduce more appropriate metrics like bandwidth and delay. The implications of routing metrics on path computation are examined. We first consider algorithms for the single-metric approach, and then we present two distributed algorithms for multiple metrics. The performance evaluation of our protocol is extensively investigated. Mathematical analysis and simulation results show that the QOLSR protocol yields better performance compared to the best-effort OLSR protocol using our proposed heuristics.

I. Introduction

A link state routing approach makes available detailed information about the connectivity and the topology found in the network. Moreover, it increases the chances that a node will be able to generate a route that meets a specified set of requirements constraints. OLSR protocol [1] is an optimization over the classical link state protocol, proposed for the Mobile Ad hoc NETworks (MANET) [2]. It performs hop-by-hop routing, i.e., each node uses its most recent information to route a packet. Therefore, each node selects a set of its neighbor nodes as MultiPoint Relays (MPRs) [3]. In the OLSR protocol, only nodes, selected as such MPRs, are responsible for forwarding control traffic, intended for diffusion into the entire network. MPRs provide an efficient mechanism for flooding control traffic by reducing the number of transmissions required. Nodes, selected as MPRs, also have a special responsibility when declaring link state information in the network. Nevertheless, no QoS information is taken into account.

In [4], [5], we have presented the QOLSR protocol, which is an enhancement of the OLSR routing protocol to support multiple-metric routing criteria. Such a work details how the metrics measurements and the routing table calculation are achieved by QOLSR protocol. They include quality information in *TC* messages about each link, between a node and its MPR selectors. The heuristic for the selection of MPRs limits its number in the network and ensures that the overhead is as low as possible. However, there is no guarantee that OLSR finds the optimal path in terms of QoS requirements. With this heuristic, the good quality links may be hidden to other nodes in the network. In this paper, we present two heuristics for the selection of MPRs based on QoS measurements. We show that the proposed heuristics find optimal widest paths (paths with maximum bandwidth) based on the QoS over the links. The idea is to introduce more appropriate metrics, such as bandwidth and delay. The QOLSR protocol calculates these metrics between each node and its neighbors having direct and symmetric link. These metrics information are stored in the neighbor table and used to calculate the MPRs. Afterwards, each MPR node generates a topology control message including its MPR selector' bandwidth and delay. Based on these information, the routing table is deduced to find the optimal routes in terms of bandwidth and delay. Our methods are backed with mathematical proofs and extensive simulations.

The remainder of this paper is organized as follows. Section II presents the QoS routing in ad hoc networks, describing some related work and the QOLSR protocol. The routing table calculation based on multiple metrics is presented in Section IV. Section V presents the disadvantages of the standard MPR selection heuristic and introduces the new heuristics for the selection of MPRs algorithms including bandwidth and delay information. Results from extensive simulations showing their performance is presented in Section VI. Finally, we conclude in Section VII.

II. QoS ROUTING IN AD HOC NETWORKS

A. Related work

The existing research on QoS routing for ad hoc networks can be divided into two categories: QoS route information and QoS route computation. QoS route information provides the QoS information over the paths using traditional best-effort routing algorithms. Such information helps the source node to fulfill the call admission task. QoS route computation calculates feasible routes based on various QoS requirements.

For the QoS route information, [6] proposes a bandwidth-constrained routing algorithm. Each node calculates the available bandwidth over the wireless links to the destination. Such bandwidth information is piggybacked in the Destination Sequence Distance Vector (DSDV) routing algorithm [7]. Thus, each node knows the bottleneck bandwidth over the paths calculated by DSDV to all known destinations.

In [8] a similar approach using DSDV is presented. Focusing on bandwidth control, bandwidth information is embedded in the nodes' routing tables and sent to the neighbors. Upon receiving a routing table from a neighbor, a node updates its own routing table and the path bandwidth information. With the bandwidth information, a node can decide whether or not it should accept a new connection request based on the bandwidth requirement of that connection.

In [9], the authors have developed an architecture that supports accurate permissible throughput explicit feedback to multimedia transports and call admission applications. The motivation is for the architecture to base a cost/benefit analysis of end-to-end measurements versus lower level explicit feedback in 802.11 access networks. They use the fragment acknowledgement and link-fail message used for unicast in DCF to produce link-by-link (source, destination) pair throughput measurement. Link layer queue utilization measurements are then combined to produce a permissible throughput measurement, in a suitable way for a wireless multi-hop network.

For the QoS route computation, [10] considers a number of issues in QoS routing. The authors first examine the basic problem of QoS routing, namely, finding a path that satisfies multiple constraints, and its implications on routing metric selection. They present a centralized algorithm that is suitable for source routing and two distributed algorithms that are suitable for hop-by-hop routing based bandwidth and delay constraints.

In [11], the author proposes a heuristic algorithm for the delay-cost constrained routing problem which is *NP-complete*. The main idea of this algorithm is to first reduce the NP-COMPLETE problem to a simpler one which can be solved in polynomial time, and then solve the new problem by either using an extended Dijkstra's algorithm or extended Bellman-Ford algorithm.

In [12], the authors propose a routing algorithm to find the shortest path between one source and one destination node while considering the criteria of multiple metric constraints. They have three goals to achieve, 1) Sorting the QoS metrics according to the requested service, 2) Finding the possible routes between the source and the destination and 3) Speed up the path determination by using sliding window and hierarchical clustering technique.

In [13] authors propose and analyze the performance of a distance-vector QoS routing algorithm that takes into account three metrics: propagation delay, available bandwidth and loss probability. Classes of service and metric combination are used to turn the algorithm scalable and as on a two-metric scheme.

[14] gives an algorithm to find a path that meets two requirements in polynomial time. The algorithm reduces the original problem to a simpler one by modifying the cost function, based on which the problem can be solved using an extended shortest path algorithm. The shortcoming of this method is that the algorithm has to use high granularity in approximating the metrics, so it can be very costly in both time and space, and it cannot guarantee that the simpler problem has a solution if the original one has a solution.

The algorithms in [15], [16] are based on calculating a simple metric from the multiple requirements. By doing so, we can use a simple shortest-path algorithm based on a single cost aggregated as a combination of weighted QoS parameters. The main drawback of this solution is that the result is quite sensitive to the selected aggregating weights.

B. QOLSR: QoS in OLSR protocol

OLSR [1] is a proactive routing protocol, which inherits the stability of a link state algorithm [17] and has the advantage of having the routes immediately available when needed due to its proactive nature. In a pure link state protocol, all the links with neighbor nodes are declared and are flooded in the whole network. The OLSR protocol is an optimization of the pure link state protocol for the mobile ad hoc networks. First, it reduces the size of the control packets: instead of all links, it declares only a subset of links with its neighbors that are its multipoint relay selectors [18]. Secondly, it minimizes the flooding of its control traffic by using only the selected nodes, called multipoint relays, to broadcast its messages. Therefore, only the multipoint relays of a node retransmit the packets. This technique significantly reduces the number of retransmissions in a flooding or broadcast procedure [3], [19]. OLSR protocol performs hop by hop routing, i.e., each node uses its most recent information to route a packet.

We have proposed the QOLSR protocol which is an extension introduced to the OLSR protocol. Additional fields about QoS conditions are added to *HELLO* and *TC* messages but no additional control messages are generated. As in the standard OLSR, link state information is generated only by nodes selected as MPRs. This information is then used for route calculation. QOLSR requires only partial link state to be flooded in order to provide optimal paths in terms of bandwidth and delay.

III. Metrics Measurements

The delay and bandwidth metrics are taking into account as QoS constraints for the proposed QOLSR protocol. Such metrics are included on each routing table entry corresponding to each destination.

A. Delay Metric

With the QOLSR protocol, a route is immediately available when needed satisfying the QoS requirements. So, before sending data traffic we must inform each node about the delay information between any node and its MPRs. The only possibility to calculate the delay is to use the control traffic messages. Each node in the ad hoc network periodically broadcasts locally its *HELLO* messages. These control messages are transmitted in broadcast mode without acknowledgements in response. Consequently, indeed to measure the round trip time rtt (time passed from sending the message in the sender to arrival the acknowledgment in the sender), each node calculates the one-way time (time between generation of the *HELLO* message in the source node and arrival in the neighbor node) from its neighbors. Now, there are two cases two consider: synchronized and asynchronized ad hoc networks.

In synchronized networks, measured delay's computing is very simple. Each node includes in the *HELLO* message, during the neighbor discovery performed by the QOLSR, the creation time of this message. When a neighbor node receives this message, it calculates the difference between such a time and the current time, which represents the measured delay. For instance, if we use the IEEE 802.11b as the medium access control, the measured delay represents the follow formula:

$$\text{measured delay} = t_q + (t_S + t_{CA} + t_{Overhead}) \times R + \sum_{r=1}^{R} B_T$$

Where: t_q = Mac queuing time; t_S = Transmission time of S bits; t_{CA} = Collision Avoidance phase time; $t_{Overhead}$ = Control Overhead time (e.g. RTS, CTS, etc); R = Number of necessary transmissions; B_T = Backoff time for r.

In asynchoronized networks, due to the characteristics of sparse ad hoc networks, classical clock synchronization algorithms are not applicable. Time synchronization in ad hoc networks is a wide subject of research, e.g., the work presented in [20]. Here, we propose a simple algorithm to estimate an average of measured delays and not to synchronize the local computer clocks of the devices. Figure 2 shows the *HELLO* messages exchanged between a node a and its neighbor b and how to measure the measured delay.

We suppose that the absolute time when the node a is generating its *HELLO* message is t_0. So, the local time in a is $t_a(t_0) = t_0 + \delta_a(t_0)$ and $t_b(t_0) = t_0 + \delta_b(t_0)$ in b where $\delta_a(t_0)$ and $\delta_b(t_0)$ are the differences between the local times in a and b respectively and the absolute time t_0. The node a

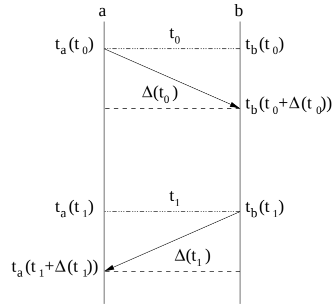

Fig. 1. HELLO message delay estimation.

includes $t_a(t_0)$ in its *HELLO* message. When the neighbor node b receives this message, it has a local time $t_b(t_0+\Delta(t_0)) = t_0+\delta_b(t_0+\Delta(t_0))+\Delta(t_0)$, it calculates the difference between its local time and the attached time $t_a(t_0)$, which represents the $\Delta_{\mathrm{pars(b,a)}} = \delta_b(t_0+\Delta(t_0))+\Delta(t_0)-\delta_a(t_0)$. We suppose that the absolute time when the node b is generating its *HELLO* message is t_1. At this time, the local times in a and b are respectively $t_a(t_1) = t_1 + \delta_a(t_1)$ and $t_b(t_1) = t_1 + \delta_b(t_1)$. The node b includes $t_b(t_1)$ and its $\Delta_{\mathrm{pars(b,a)}}$ in the *HELLO* message. When the neighbor node a receives this message, it has a local time $t_a(t_1+\Delta(t_1)) = t_1+\delta_a(t_1+\Delta(t_1))+\Delta(t_1)$, it calculates the difference between its local time and the attached time $t_b(t_1)$, which represents the $\Delta_{\mathrm{pars(a,b)}} = \delta_a(t_1+\Delta(t_1))+\Delta(t_1)-\delta_b(t_1)$. Then, the node a gets the average between its $\Delta_{\mathrm{pars(a,b)}}$ and the attached $\Delta_{\mathrm{pars(b,a)}}$ which represents the measured delay. By the assumption that $\delta_b(t_0+\Delta(t_0)) = \delta_b(t_1)$ and $\delta_a(t_0) = \delta_a(t_1+\Delta(t_1))$ (the hardware clocks in a and b are perfect, i.e., there is no clock drift in a and b) $\frac{\Delta(t_0)+\Delta(t_1)}{2}$ represents the measured delay. It is clear that this measured delay has no information about Δ_a and Δ_b individually.

After calculating the measured delay, each node must smooth the measured delay due to variations of delay in the ad hoc networks. Like the well-known Congestion Control of TCP [23], the average delay is calculated as follows:

$\mathrm{Average\ delay} = \alpha \times \mathrm{Average\ delay} + (1-\alpha) \times \mathrm{measured\ delay}$

where α is a constant filter gain. In [4], we havee shown that the best performance are achieved using $\alpha = 0.4$. the Average delay is updated every time a new measurement is made. Forty percent of each new estimate is from the previous estimate and 60% is from the new measurement to take with more importance the actual topology.

B. Bandwidth Metric

The bandwidth measurement in the on demand protocols is made when a new route is established between a source and a destination host. A remarkable work is presented in [9], considering the acknowledgement time from the data packets. We have started the bandwidth measurements based on their statements.

We consider in our proposal a link-state paradigm. For our approach, the bandwidth will be calculated between a node and its neighbors having direct and symmetric link. We consider for our analysis data packets and signaling traffic that also uses the available bandwidth and then must be taken into account. (e.g., *HELLO* messages and Traffic Control messages in the OLSR protocol).

Measuring the throughput over the link in multi-hop networks, as presented in [9], needs the analysis of three aspects: the unique contention for each source-destination link pair, the distributed knowledge of the contention and the rapid changes due to mobility.

For our purpose, as we have taken into account a link-state paradigm, afterward the considered parameters are in function of source node and its neighbor, such as a hop by hop model, differing to the final destination considered in on demand approach.

Let be a node i and j its neighbor, then we apply the follow statement:

$Bw_{(i,j)} = (1-u) \times Throughput_{(i,j)}$, where u is the link utilization.

The source-neighbor pair throughput is measured for a window of packets using existing traffic. Each successful packet transmission contributes its bits to the numerator of the throughput measurement and the time from when it was ready to transmission at the head of the link queues, to the acknowledgment receipt. Using the IEEE 802.11b as medium access control, this interval is packet size dependent as shown in the next equation:

$$Throughput_{packet} = \frac{S}{t_q + (t_S + t_{CA} + t_{Overhead}) \times R + \sum_{r=1}^{R} B_T}$$

where: S = Packet size. In order to filter another measurement issue, we use a packet window. As performed in [9] we also use a packet window to increase statistical robustness of the measurements. We have seen in such a work, that a packet window of 16 or 32 samples (packets) is adequate to produce fast enough, noise immune measurements. The idle time and window duration are calculated to produce the link utilization factor and the permissible throughput measurement as:

$$\frac{\text{idle\_time\_in\_window}}{\text{window\_duration}} \times \text{Throughput\_measured}$$

IV. Routing table calculation

Let $G = (V, E)$ be the network with $|V| = n$ nodes and $|E| = m$ arcs and met_{ij} a metric for link (i, j). The value of a metric over any directed path $p = (i, j, k, ..., q, r)$ can be one of the following compositions:

- ☞ *Additive metrics:* We say metric met is additive if $\text{met}(p) = \text{met}_{ij} + \text{met}_{jk} + ... + \text{met}_{qr}$. It is obvious that delay (del), delay jitter (dej), hop-count (hop) and cost (co) follow the additive composition rule.
- ☞ *Multiplicative metrics:* We say metric met is multiplicative if $\text{met}(p) = \text{met}_{ij} \times \text{met}_{jk} \times ... \times \text{met}_{qr}$. The probability of successful transmission (pst) follows the multiplicative composition rule. The composition rule for loss probability (Lp) is more complicated. It can be transformed to an equivalent metric pst, $\text{Lp}(p) = 1 - ((1 - \text{Lp}_{ij}) \times (1 - \text{LP}_{jk}) \times ... \times (1 - \text{Lp}_{qr}))$.
- ☞ *Concave metrics:* We say metric met is concave if $\text{met}(p) = \min\{\text{met}_{ij}, \text{met}_{jk}, ..., \text{met}_{qr}\}$. It is obvious that Bandwidth (Bw) follows the concave composition rule.

A. Single metric approach

In traditional data networks, routing protocols usually characterize the network with a single metric such as hop-count or delay, and use the shortest path algorithms for path computation. For delay metric, each arc (i, j) in the path p is assigned a real number del_{ij}. When the arc (i, j) is inexistent or j is not a MPR of i (Referring to the OLSR routing mechanism), then $\text{del}_{ij} = \infty$. Let $\text{del}(p) = \text{del}_{ij} + \text{del}_{jk} + ... + \text{del}_{qr}$. The routing problem is to find a path $p*$ between i and r so that $\text{del}(p*)$ is the minimum. In such a case, we use the well-known Dijkstra routing algorithm. For bandwidth metric, each arc (i, j) in the path is assigned a real number Bw_{ij}. When the arc (i, j) is inexistent or j is not a MPR of i, $\text{Bw}_{ij} = 0$. Let $\text{Bw}(p) = \min\{\text{Bw}_{ij}, \text{Bw}_{jk}, ..., \text{Bw}_{qr}\}$. The routing problem is to find a path $p*$ between i and r that maximizes $\text{Bw}(p*)$. In order to implement such a metric, we can use a variant-Dijkstra algorithm.

The worst-case complexity of Dijkstra's algorithm on networks with nonnegative arc length depends on the way of finding the labeled node with the smallest distance label. A naive implementation that examines all labeled nodes to find the minimum runs in $O(n^2)$ time [20]. The implementation using *k*-ary heaps [21] runs on $O(m\log n)$ time (for a constant k). The implementation using Fibonacci heaps [22] runs in $O(m + n\log n)$ time. The implementation using one-level R-hraps [21] runs in $O(m + (n\log C))$ time and the one using two-level R-heaps together with Fibonacci heaps, in $O(m + n\sqrt{\log C})$ time.

B. Multiple metrics approach

The second approach treats each metric individually. Such approach is not feasible due to the algorithm complexity. The problem of finding a path with n additive and m multiplicative metrics is *NP-complete* if $n + m \geq 2$ [23]. Including a single metric, the best path can be easily defined. Otherwise including multiple metrics, the best path with all parameters at their optimal values may not exist.

We consider bandwidth and delay routing problem. A path with both maximum bandwidth and minimum delay may not necessarily exist. Thus, we must decide the precedence among the metrics in order to define the best path. The delay has two basic components: queuing delay and propagation delay. The queuing delay is more dynamic and traffic-sensitive, thus bandwidth is often more critical for most multimedia applications. If there is no sufficient bandwidth, queuing delay and probably the loss rate will be very high. So, we define the precedence as bandwidth and then the propagation delay.

Our strategy is to find a path with maximum bandwidth (a widest path), and when there is more than one widest path, we choose the one with shortest delay. We refer to such a path as the shortest-widest path. The widest path problem is to find a path $p*$ between i and j that maximizes $\mathrm{Bw}(p*)$. For a given topology, there are usually many widest paths with equal width, and loops can be formed as a result. However, shortest-widest path is always free of loops. Intuitively, the delay metric eliminates the loops.

Theorem 1: Shortest-widest paths are loop-free in a distributed computation

Proof: By contradiction. Suppose that node A and node B are involved in a loop for destination C (Figure 2). Path p_1p_2 is the shortest-widest path from node A to node C and path $p_1^*p_2^*$ is the shortest-widest path from node B to node C.

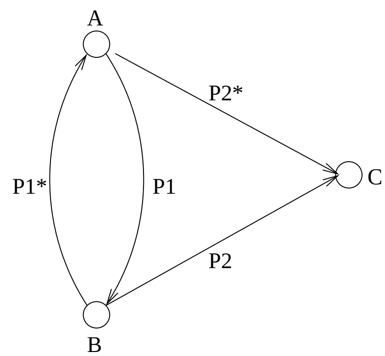

Fig. 2. A loop involving node A and node B.

By the definition of the shortest-widest paths, we have

$$\mathrm{width}(p_2^*) \leq \mathrm{width}(p_1p_2) \quad (1)$$
$$\mathrm{width}(p_2) \leq \mathrm{width}(p_1^*p_2^*) \quad (2)$$

Note that

$$\mathrm{width}(p_1^*p_2^*) = \min[\mathrm{width}(p_1^*), \mathrm{width}(p_2^*)] \leq \mathrm{width}(p_2^*) \quad (3)$$

Similarly,

$$\mathrm{width}(p_1p_2) = \min[\mathrm{width}(p_1), \mathrm{width}(p_2)] \leq \mathrm{width}(p_2) \quad (4)$$

From (1), (3) and (4), we have

$$\mathrm{width}(p_1^*p_2^*) \leq \mathrm{width}(p_2) \quad (5)$$

Comparing (2) with (5), we have

$$\mathrm{width}(p_1^*p_2^*) = \mathrm{width}(p_2) \quad (6)$$

Similarly, we have

$$\mathrm{width}(p_1p_2) = \mathrm{width}(p_2^*) \quad (7)$$

Equation (6) shows that path $p_1^*p_2^*$ and path p_2 are equal widest paths, since path $p_1^*p_2^*$ is the shortest-widest path, we have

$$\mathrm{length}(p_2) \geq \mathrm{length}(p_1^*p_2^*) > \mathrm{length}(p_2^*) \quad (8)$$

Similarly, equation (7) shows that path p_1p_2 and path p_2^* are equal widest paths, since path p_1p_2 is the shortest-widest path, we have

$$\mathrm{length}(p_2^*) \geq \mathrm{length}(p_1p_2) > \mathrm{length}(p_2) \quad (9)$$

Equation (8) and (9) contradict each other. This completes the proof. ■

Each arc (i, j) in the path is assigned the following values: del_{ij}, which is the propagation delay and Bw_{ij}, which is the available bandwidth. When the arc (i, j) is inexistent or j is not a MPR of i (due to the OLSR routing mechanism), $\mathrm{del}_{ij} = \infty$ and $\mathrm{Bw}_{ij} = 0$. Let $\mathrm{del}(p) = \mathrm{del}_{ij} + \mathrm{del}_{jk} + ... + \mathrm{del}_{qr}$ and $\mathrm{Bw}(p) = \min\{\mathrm{Bw}_{ij}, \mathrm{Bw}_{jk}, ..., \mathrm{Bw}_{qr}\}$. The shortest-widest path algorithm [4] based on Dijkstra algorithm finds at each iteration a node with maximum width from a source s, if there are more than one widest path found, it chooses the one with minimum length. The time complexity of the shortest-widest path algorithm is equal to that Dijkstra's shortest path algorithm.

V. HEURISTICS FOR THE SELECTION OF MULTIPOINT RELAYS

Finding a MPR set with minimal size falls in the category of dominating set problem, which is known to be *NP-complete* [3]. The information needed to calculate the MPRs is the set of one-hop neighbors and two-hop neighbors. To select the MPRs for the node x, the following terminology is used in describing the heuristics:

- $\mathrm{MPR}(x)$: the multipoint relay set of node x which is running this algorithm;
- $\mathrm{N}(x)$: the one hop neighbor set of node x containing only symmetric neighbors;
- $\mathrm{N2}(x)$: the two hop neighbor set of node x containing only symmetric neighbors in $\mathrm{N}(x)$. The two hop neighbor set $\mathrm{N2}(x)$ of node x does not contain any one hop neighbor of node x;
- $\mathrm{D}(x, y)$: degree of one hop neighbor node y (where y is a member of $\mathrm{N}(x)$), is defined as the number of symmetric one hop neighbors of node y excluding the node x and all the symmetric one hop neighbors of node x, i.e., $\mathrm{D}(x, y)$ = number of elements of $\mathrm{N}(y) - x - \mathrm{N}(x)$;
- **Widest path**: is a path with maximum bandwidth, calculated by the source node with its known partial network topology. In the widest path, any intermediate node is MPR of its previous node;
- **Shortest-widest path**: is the widest path, and with shortest delay when there is more than one widest path;
- **Optimal widest path**: is the widest path between two nodes in the whole network topology. Any node in the network can be selected as an intermediate node in the optimal widest path;
- **Optimal shortest-widest path**: is the shortest-widest path between two nodes in the whole network topology. Any optimal shortest-widest path is an optimal widest path.

The heuristic used in the standard OLSR protocol computes a MPR set of cardinality at most $\log n$ times the optimal multipoint relay number, where n is the number of nodes in the network. The approximation factor of the upper bound can be given by $\log\Delta$ where Δ is the maximum number of two-hop nodes a one-hop node may cover.

The standard OLSR heuristic limits the number of MPRs in the network and ensures that the overhead is as low as possible. However, in QoS routing, by such a MPR selection mechanism, the good quality links may be hidden to other nodes in the network.

Theorem 2: There is no guarantee that OLSR finds the optimal shortest-widest or optimal widest path.

Proof: 1) By construction. The heuristic for the selection of multipoint relays in the standard OLSR does not take into account the bandwidth and delay information. It computes a multipoint relay set of minimal cardinality. So, the links with high bandwidth and low delay can be omitted. After, the path calculated between two nodes using the shortest-widest path algorithm has no guarantee that is the optimal widest path or shortest-widest path in the whole network. 2) By example, from Figure 3 and Table I:

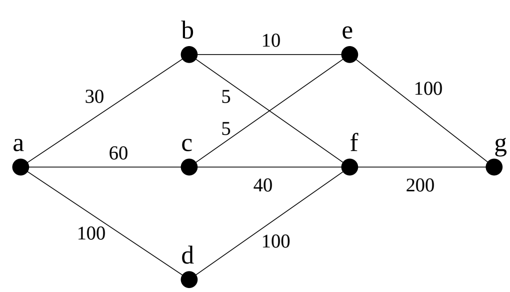

Fig. 3. Network example for MPR selection.

When g is building its routing table, for destination a, it will select the route (g, f, b, a) whose bandwidth is 5. The optimal widest path between g and a is (g, f, d, a). It has 100 as bandwidth. This completes the proof. ∎

The decision of how each node selects its MPRs is essential to determinate the optimal bandwidth and delay route in the network. In the MPR selection, the links with high bandwidth and low delay should not be omitted.

Node	1-hop neighbors	2-hop neighbors	MPRs
a	b, c, d	e, f	b
b	a, e, f	c, d, g	f
c	a, e, f	b, d, g	f
d	a, f	b, c, g	f
e	b, c, g	a, f	b
f	b, c, d, g	a, e	c
g	e, f	b, c, d	f

TABLE I

MPR SELECTED IN THE STANDARD OLSR.

A. QOLSR_MPR1

In this protocol, MPR selection is almost the same as that of the standard OLSR. However, when there is more than 1-hop neighbor covering the same number of uncovered 2-hop neighbors, the one with maximum bandwidth link (a widest link) to the current node is selected as MPR. If there is more than one widest link, we choose the one with the shortest delay. The heuristic used in QOLSR_MPR1 protocol is as follows:

Step 1: Start with an empty multipoint relay set $\mathrm{MPR}(x)$;

Step 2: Calculate $\mathrm{D}(x, y)$, $\forall$ nodes $y \in \mathrm{N}(x)$;

Step 3: First, select those one-hop neighbor nodes in $\mathrm{N}(x)$ as the multipoint relays which provide the only path to reach some nodes in $\mathrm{N2}(x)$, and add these one-hop neighbor nodes to the multipoint relay set $\mathrm{MPR}(x)$;

Step 4: While there still exist some nodes in $\mathrm{N2}(x)$ that are not covered by the multipoint relay set $\mathrm{MPR}(x)$:

Step 4.a: For each node in $\mathrm{N}(x)$ which is not in $\mathrm{MPR}(x)$, calculate the number of nodes that are reachable through it among the nodes in $\mathrm{N2}(x)$ and which are not yet covered by $\mathrm{MPR}(x)$;

Step 4.b: Select that node of $\mathrm{N}(x)$ as a MPR which reaches the maximum number of uncovered nodes in $\mathrm{N2}(x)$;

Step 4.c: In case of a tie in the above step, select that node with higher bandwidth as MPR;

Step 4.d: In case of a tie in the above step, select that node with minimum delay as MPR;

Step 5: To optimize, remove each node in $\mathrm{MPR}(x)$, one at a time, and check if $\mathrm{MPR}(x)$ still covers all nodes in $\mathrm{N2}(x)$.

The third step permits to select some one-hop neighbor nodes as MPRs which must be in the $\mathrm{MPR}(x)$ set, otherwise the $\mathrm{MPR}(x)$ will not cover all the two-hop neighbors. So these nodes will be selected as MPRs in the process, sooner or later. In step 5, an optimization is performed by reducing the number of MPRs, if possible.

This heuristic has the same time complexity of the standard OLSR heuristic. It computes a MPR set of cardinality at most $\log n$ times the optimal multipoint relay number where n is the number of nodes in the network.

Theorem 3: There is no guarantee that QOLSR_MPR1 finds the optimal shortest-widest or optimal widest path.

Proof: 1) By construction. The heuristic for the selection of multipoint relays in the QOLSR_MPR1 is almost the same as that of the standard OLSR. We use the bandwidth and delay information when there is more than one one-hop neighbor covering the same number of uncovered two-hop neighbors. So, the links with high bandwidth and low delay can be omitted. 2) By example, from Figure 3 and Table V-A, we have: Between b and c, c is selected as a's MPR because it has the larger bandwidth. When g is building its routing table, for destination a, it will select the route (g, f, c, a) whose bandwidth is 40. The optimal widest path between g and a is (g, f, d, a). It has 100 as bandwidth.

Node	1-hop neighbors m	2-hop neighbors	MPRs
a	b, c, d	e, f	c

TABLE II

MPR SELECTED IN THE QOLSR_MPR1

This completes the proof. ■

B. QOLSR_MPR2

In this protocol, neighbors that guarantee maximum bandwidth and minimum delay among two-hop neighbors are selected as MPRs. The heuristic used in QOLSR_MPR2 protocol is as follows:

Step 1: Start with an empty multipoint relay set $\mathrm{MPR}(x)$;

Step 2: Calculate $\mathrm{D}(x, y)$, $\forall$ nodes $y \in \mathrm{N}(x)$;

Step 3: First, select those one-hop neighbor nodes in $\mathrm{N}(x)$ as the multipoint relays which provide the only path to reach some nodes in $\mathrm{N2}(x)$, and add these one-hop neighbor nodes to the multipoint relay set $\mathrm{MPR}(x)$;

Step 4: While there still exist some nodes in $\mathrm{N2}(x)$ that are not covered by the multipoint relay set $\mathrm{MPR}(x)$:

Step 4.a: For each node in $\mathrm{N}(x)$ which is not in $\mathrm{MPR}(x)$, For each node in $\mathrm{N}(x)$ which is not in $\mathrm{MPR}(x)$, calculate the number of nodes that are reachable through it among the nodes in $\mathrm{N2}(x)$ and which are not yet covered by $\mathrm{MPR}(x)$:

Step 4.b: Select that node of $\mathrm{N}(x)$ with the maximum bandwidth and minimum delay as a MPR;

Step 4.c: In case of a tie in the above step, select that node which reaches the maximum number of uncovered nodes in $\mathrm{N2}(x)$;

Claim 1: Let p =(a_1,...,a_{i-1}, a_i, a_{i+1},...,a_k) an optimal widest path, $k \geq 3$. For any intermediate node a_i ($i \neq 1$) in p that is not selected as MPR by its previous node a_{i-1}, we can find a node b_i selected as MPR by a_{i-1} such as the path (a_1,...,a_{i-1},b_i,a_{i+1}, ...,a_k) has the same bandwidth performance.

Proof: Let p =(a_1,...,a_{i-1}, a_i, a_{i+1},...,a_k), $k \geq 3$ an optimal widest path from a_i to a_k (Figure 4).

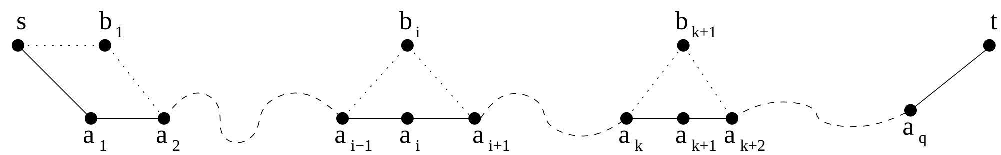

Fig. 4. Optimal widest path from s to t

Suppose that on the optimal widest path, the node a_i is not selected as MPR by its previous node a_{i-1}. We can assume that for each node on the path, its next node in the path is its 1-hop neighbor, and the node two hops away from it is its 2-hop neighbor. For example, a_i is a_{i-1}'s 1-hop neighbor, a_{i+1} is a_{i-1}'s 2-hop neighbor. Based on the basic idea of the MPR selection that all the 2-hop neighbors of a node should be covered by this node's MPR set. So, a_{i-1} must have another neighbor b_i, which is selected as its MPR, and is connected to a_{i+1}. Let p' =(a_1,...,a_{i-1}, b_i, a_{i+1},...,a_k), $k \geq 3$. According to the criteria of MPR selection specified on QOLSR_MPR2, a_{i-1} selects b_i instead of a_i as its MPR because:

$$\mathrm{Bw}_{a_{i-1}b_i a_{i+1}} > \mathrm{Bw}_{a_{i-1}a_i a_{i+1}} \tag{10}$$

Or

$$\begin{cases} \mathrm{Bw}_{a_{i-1}b_i a_{i+1}} = \mathrm{Bw}_{a_{i-1}a_i a_{i+1}} \\ \mathrm{del}_{a_{i-1}b_i a_{i+1}} < \mathrm{del}_{a_{i-1}a_i a_{i+1}} \end{cases} \tag{11}$$

From (10) we have $\mathrm{Bw}(p') \geq \mathrm{Bw}(p)$ and there is no guarantee about $\mathrm{del}(p') \geq \mathrm{del}(p)$.

From (11) we have

$$\begin{cases} \mathrm{Bw}(p') = \mathrm{Bw}(p) \\ \mathrm{del}(p') < \mathrm{del}(p) \end{cases} \tag{12}$$

In both cases, $\mathrm{Bw}(p') \geq \mathrm{Bw}(p)$. Based on our assumption, path p is an optimal widest path. So, path p' is also optimal widest. This completes the proof. ■

Claim 2: There is an optimal widest path in the whole network such that all the intermediate nodes are selected as MPR by their previous nodes.

Proof: By recurrence. Let p =(s,a_1,...,a_{i-1}, a_i, a_{i+1},...,a_k,...,a_q, t), $k < q$ an optimal widest path (Figure 4).

a) We demonstrate that the first intermediate node a_1 is selected as MPR by source s. By using the claim 1, we can find a node b_1 selected as MPR by s such as the path p' =(s,b_1,...,a_{i-1}, a_i, a_{i+1},...,a_k,...,a_q, t) has the same bandwidth performance of the optimal path (p' is also an optimal widest path). So, source's MPR are on the optimal widest path.

b) We assume that all the nodes $\{a_1,...,a_{i-1}, a_i, a_{i+1},...,a_k\}$ are selected as MPR by their previous node in the path p. We prove that the next hop node of a_k on p is a_k's MPR. Suppose that a_{k+1} is not an MPR of a_k. Same as above, by using the claim 1, we can find a node b_{k+1} selected as MPR by a_k such as the path p' =(s,a_1,...,a_{i-1}, a_i, a_{i+1},...,a_k,b_{k+1},...,a_q, t) has the same bandwidth performance of the optimal widest path (p' is also an optimal widest path). So, in an optimal widest route, the (k+1)th intermediate node is the MPR of the (k)th intermediate node.

Based on (a) and (b), all the intermediate nodes of an optimal widest path are the MPRs of the previous nodes. ■

By claim 2, there is an optimal widest path such that all the intermediate nodes are the MPR of the previous nodes on the same path. So the optimal widest path for the whole network topology is included in the partial topology the node knows. And by using the shortest-widest path algorithm, we can compute the optimal widest path in the partial network topology. We can conclude that the QOLSR_MPR2 finds the optimal widest path.

Theorem 4: QOLSR_MPR2 finds optimal widest paths using only the known partial network topology.

The heuristic used in the QOLSR_MPR2 finds exactly the optimal MPRs that guarantee maximum bandwidth and minimum delay. So, this heuristic is an algorithm. The upper bound of the time complexity of this algorithm is O(α) where α is the maximum number of two-hop nodes.

VI. Performance evaluation

In this section, we evaluate the performance of the QOLSR protocol and its extensions applying multiple metrics (bandwidth and delay) and the different MPRs Selection algorithms in different configurations and scenarios. We use the OPNET simulator for our evaluation.

A. Heuristic Evaluation in static networks

1) Average number of MPRs and MPR_2s per node: We assume that the ad hoc network topology is stable (a wireless network consisting of desktops, laptops and printers for home business may keep its original topology for a long time until someone moves one of the laptops to another room). We generate 100 random network snapshots for each scenario. Each node is placed in an area of $x_{\max}\mathrm{m} \times y_{\max}\mathrm{m}$ randomly selecting its x and y co-ordinates using uniform distribution. Each node is randomly assigned the idle time ratio ($idle\_time$), ranging from 0 to 1. All links have the same bandwidth, namely 2Mbps. The available bandwidth on a link between two nodes a and b amounts to $\min(idle\_time_a, idle\_time_b) \times bandwidth_{ab}$.

In the simulated scenarios, results are collected over 6 values of transmission range (50, 100, 150, 200, 250 and 300 meters).

Figures 5, 6, 7, 8, 9, 10 show the average number of 1-hop Neighbors, 2-hop Neighbors, MPRs and MPR_2s per node against the increasing number of nodes placed in an area of $1000\mathrm{m} \times 1000\mathrm{m}$ with different ranges of transmission. Obviously, the number of MPR_1s is the same as the number of MPRs, therefore results for the number of MPR_1s are omitted.

In figure 5, the transmission range is 50 meters. The number of (1,2)-hop neighbors, MPRs and MPR_2s increases with the number of nodes, but there are less 1-hop neighbors than 2-hop neighbors per node (7 1-hop neighbors/node vs. 11 2-hop neighbors/node in a network of 1000 nodes). We note a maximal variation of 12.5% between the number of MPRs and MPR_2s.

We observe similar behavior with increasing number of nodes and transmission range in subsequent figures 6, 7, 8, 9, 10, but the maximal variation seems to remain stable (less than 40%). Since the number of MPRs converges to a constant value with the increase of the number of nodes, so do MPR_1s and MPR_2s.

Figure 11 summarizes all the preceding figures in three dimensions.

In figures 12, 13, 14, 15, with an area of $200\mathrm{m}\times 200\mathrm{m}$, we observe that the number of 2-hop neighbors decreases with the increasing transmission range, while the number of 1-hop neighbors increases. In

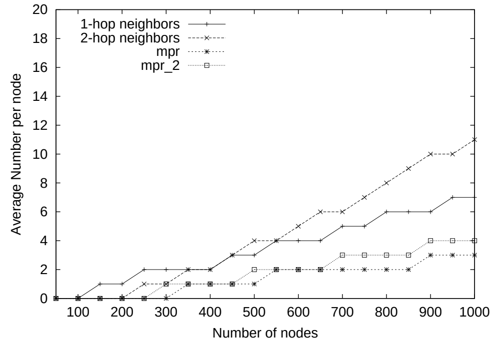

Fig. 5. Nodes in $1000\text{x}1000\text{m}^2$ with 50m as TR

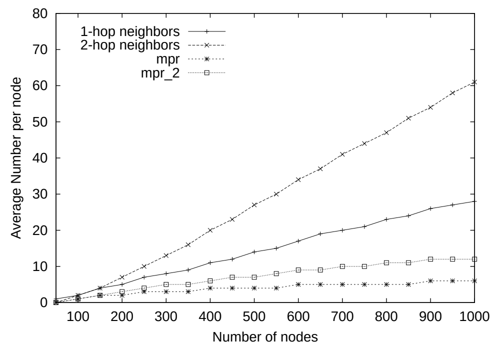

Fig. 6. Nodes in $1000\text{x}1000\text{m}^2$ with 100m as TR.

figure 13, the number of 2-hop neighbors has decreased down to the number of 1-hop neighbors and is decreases further down to 0 with increasing transmission range (see figures 14, 15). Consequently, the number of MPRs, MPR_1s and MPR_2s decreases along the number of 2-hop neighbors. The reason of the decrease of 2-hop neighbors is that the probability of a node to be a 1-hop neighbor of another node in a finite area tends to 1 as the transmission range tends to cover the whole area. It follows that multihop routing then becomes useless and hence the use of relays.

The average number of MPR_2s per node tends to be constant, amounting to less than 40% more than the number of MPRs. Hence, the overhead remains under acceptable limits as shown in the next section.

2) MPR selection performance: In our simulated scenarios, we collect results over three values of range transmission (100 meters, 200 meters, 300 meters) in area of $1000\text{m} \times 1000\text{m}$. Table III shows the average number of 1-hop neighbors and 2-hop neighbors. We can see that when the range transmission decreases, the number of (1, 2)-hop neighbors decreases. These values affect the MPR number in the network. By assuming high connectivity of the network: (1) the more 1-hop neighbors a node has, the less MPRs it may select, because with a high probability a small subset of its 1-hop neighbor can reach a high number of the 2-hop neighbors, (2) the more 2-hop neighbors a node has, the more MPRs may be needed to cover them all.

Transmission range	300 m	200 m	100 m
1-hop neighbors	21	10	2
2-hop neighbors	33	15	4

TABLE III
AVERAGE NUMBER OF (1,2)-HOP NEIGBORS

The next results show the performances of the routes found by the implemented algorithms (Standard OLSR, QOLSR_MPR1, QOLSR_MPR2, Pure link state algorithm: each node floods its link state information into the entire network). The results are given in two categories: performance and cost.

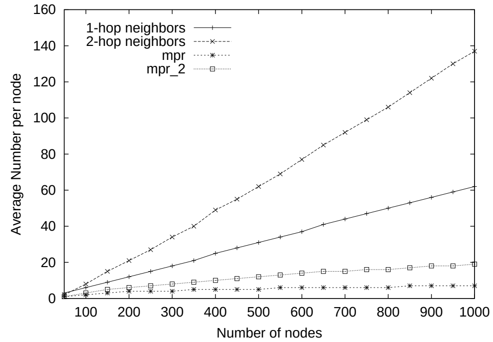

Fig. 7. Nodes in 1000x1000m^2 with 150 mas TR.

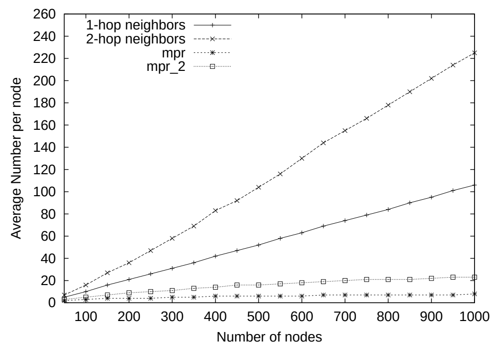

Fig. 8. Nodes in 1000x1000m^2 with 200m as TR.

Performance is characterized by: (a) Error rate: the percentage of the bad routes (bandwidth not optimal), (b) Average difference: the average of the difference between the optimal bandwidth and current bandwidth found in routing algorithms in percentage. The larger the value is, the worse the result. Cost is measured by: (a) Overhead: average number of the TC messages are transmitted in the network, (b) MPR number: average number of the MPRs in the network (**not per node**). Any MPR node is counted only once in the average number of MPRs.

Algorithm	Transmission Range	Performance		Cost	
		Error rate	Average Diff	Overhead	Nb MPR
Standard OLSR	300m	28%	46%	12	65
	200m	41%	51%	24	68
	100m	12%	45%	5	42
QOLSR_MPR1	300m	14%	22%	12	65
	200m	21%	26%	24	68
	100m	8%	44%	5	42
QOLSR_MPR2	300m	0%	0%	26	71
	200m	0%	0%	38	73
	100m	0%	0%	5.7	44
Pure link state	300m	0%	0%	1245	100
	200m	0%	0%	979	100
	100m	0%	0%	28	100

TABLE IV
PERFORMANCE AND COST

Table IV shows that for each transmission range the standard OLSR has the worst performance (it has the highest Error Rate and Average Difference). The bandwidth difference between the paths found by the standard OLSR and the optimal paths is large. QOLSR_MPR1 achieves a large improvement in performance than the standard OLSR. The explanation is that the shortest-widest path algorithm enhances the bandwidth of the found paths. However, QOLSR_MPR1 does not always find an optimal path, as its MPR selection heuristic may omit the optimal bandwidth link from the partial network topology the

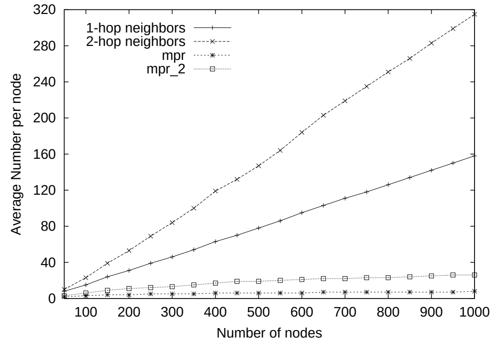

Fig. 9. Nodes in 1000x1000m^2 with 250m as TR.

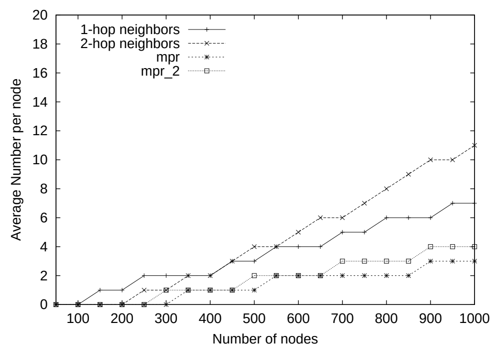

Fig. 10. Nodes in 1000x1000m^2 with 300m as TR.

node learned. QOLSR_MPR2 achieves the best performance at each time (it finds the optimal bandwidth route).

The cost is directly related to the number of the re-transmitting nodes. If the number of the re-transmitting nodes increases, the cost increases. Pure Link State algorithm has the highest overhead, because each node re-transmits the messages it receives. As the MPR selection heuristic in the standard OLSR and QOLSR_MPR1 emphasizes on reducing the number of MPRs in the network, the standard OLSR and QOLSR_MPR1 have the same and the lowest MPR number, and so the lowest overhead compared with QOLSR_MPR2 and Pure Link sate algorithm. QOLSR_MPR2 selects more MPRs, so more overhead than the standard OLSR and QOLSR_MPR1.

We can see that the standard OLSR and QOLSR_MPR1 and QOLSR_MPR2 have more MPRs and so more overhead with transmission range of $200m$. In a higher density network (such as for a node transmission range of $300m$), node connectivity is also high (see Table III), so a node may need fewer MPRs to cover its 2-hop neighbors. In lower density network (such as for a node transmission range of $100m$), because the lower connectivity, a node may have fewer 2-hop neighbors; therefore, it also needs fewer MPRs. However, the transmission range of $200m$ falls within these two extremes, so it may well result in the largest number of MPRs to produce the highest overhead. This situation is not found in the pure link state algorithm, where a node's entire neighbor set is its MPR set.

3) Performances in varying load conditions: In this simulation, we study the behavior if QOLSR, QOLSR_MPR1 and QOLSR_MPR2 protocols and the maximum utilization of the bandwidth by varying the load in the network. We have taken a static network composed of 50 nodes without uni-directional links. This network operates in a stable state when each node has complete and correct information about the network. All nodes are packet generating source. We have taken the mean packet size as 1K bytes, 200 packet for each transmit buffer and the same size as receive buffer. We increase the data packet arrival rate from 100 packets per second (which represents approximately 100 k bytes per second) up to 1400 per second. With the arrival rate equal to 100 packets per second, each node generates 2 packets per second in the average. The terminology we use to describe the transmission of the packets is as follows: 1) sent: means the first time transmission of a packet when it is sent by the source, 2) re-transmission: means the forwarding of a data packet by an intermediate node to route the packet towards its destinations; or the re-transmission of a control packet when it is flooded in the network.

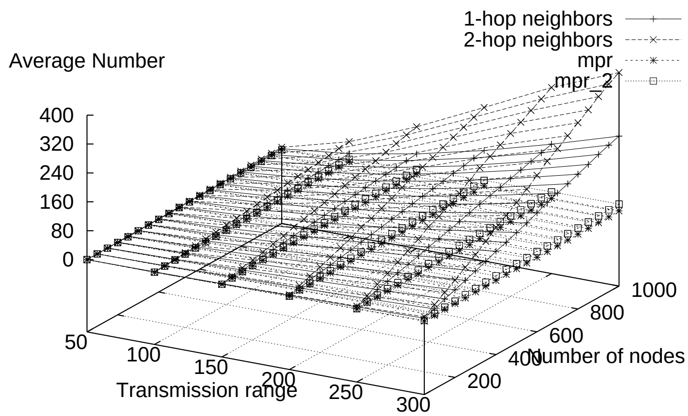

Fig. 11. Nodes in 1000x1000m^2 with 300 m as TR.

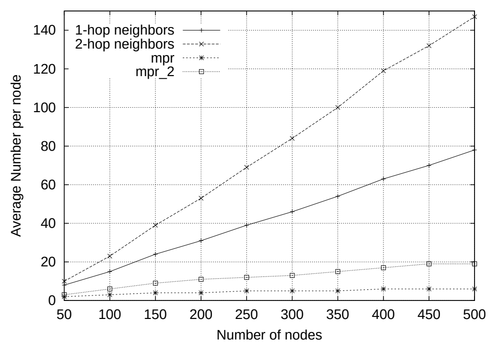

Fig. 12. Nodes in 200x200m^2 with 50m as TR.

The re-transmissions do not include the first transmission, 3) delivered: means a data packet delivered to its final destination.

In Figure 16, the data load delivered to the destination is plotted. As average route length is 3 hops, therefore the maximum throughput of data that we can obtain is:

$$\begin{aligned} \text{max throughput} &= \frac{\text{channel capacity}}{\text{average route length}} \\ &= \frac{(11000\text{kbps}/8)}{3} = 458.3 \text{ KBps} \end{aligned}$$

We can see the network attains almost the maximum throughput before saturation for both protocols. The drop in throughput after the saturation point using QOLSR and QOLSR_MPR1 is due to the absence of any congestion control mechanism as each node continues to generate data packet at a high rate. However, the throughput after saturation using the QOLSR_MPR2 remains stable in the maximum utilization bandwidth because all the paths founded are optimal widest paths and are between nodes that have enough space in their buffers.

B. Heuristic evaluation in mobile networks

The simulation model introduced in [24] is very close to a real Ad-Hoc network operations. Each node is represented by a subqueue and placed in the region by randomly selecting its x and y co-ordinates. The number of nodes can reach 100000 nodes. With this method, the simulation model is very optimized that enables to reduce the CPU time and consequently to increase the time of simulation.

The random mobility model proposed is a continuous-time stochastic process. Each node's movement consists of a sequence of random length intervals, during which a node moves in a constant direction at a constant speed. To calculate the co-ordinate of node n at t during an interval i of duration T_n^i, angle

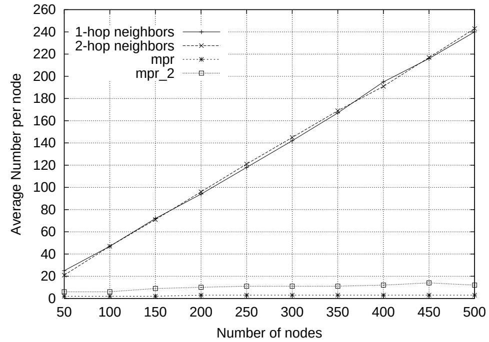

Fig. 13. Nodes in $200\text{x}200\text{m}^2$ with 100 m as TR.

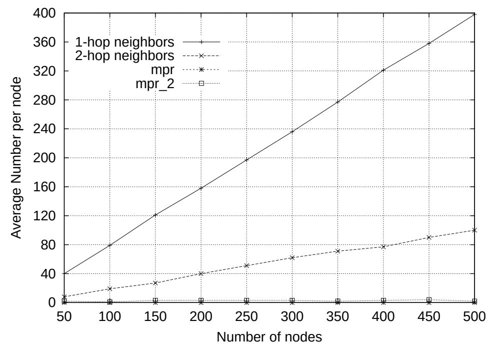

Fig. 14. Nodes in $200\text{x}200\text{m}^2$ with 150m as TR.

θ_n^i and speed V_n^i, we calculate at the first time the distance D covered by n, $D = V_n^i T_n^i$. Then, we calculate the x, y local co-ordinates, $x = D\sin(\theta_n^i)$, $y = D\cos(\theta_n^i)$. At the end, we calculate the global co-ordinates by changing scale.

Figure 17 shows the results of our simulation in which the data packets sent and successfully delivered are plotted against the increasing speed. The speed is increased from $50 meters/minute$ ($3Km/hr$) up to $500 meters/minute$ ($30Km/hr$). In this simulation, 50 nodes constitute the network in a region of $1000^2 m^2$, and all the 50 nodes are packet-generating sources. We also keep the movement probability as 0.3, i.e., only 20% of nodes are mobile and the rest are stationary. Each mobile node selects its speed and direction which remains valid for next 60 seconds. We can see that when the mobility (or speed) increases, the number of packets delivered to the destinations decreases. This can be explained by the fact that when a node moves, it goes out of the neighborhood of a node which may be sending it the data packets. There are about 99.92% of packets delivered for QOLSR at a mobility of 2 meters/minute (99.01% for QOLSR_MPR1 and 99.99% for QOLSR_MPR2). At a mobility of 500 meters/minute, 88% of packets delivered for QOLSR (90.9 % for QOLSR_MPR1 and about 97% for OLSR_MPR2). QOLSR_MPR2 has the highest packets delivered because the routes are optimal and chosen with minimal interferences. The data packets are lost because the next-hop node is unreachable. QOLSR with the classic MPR selection algorithm and QOLSR_MPR1 have the same performances in term of lost packets. A node keeps an entry about its neighbor in its neighbor table for about 6 seconds. If a neighbor moves which is the next-hop node in a route, the node continues to forward it the data packets considering it as a neighbor. Also, the next-hop is unreachable if there are interferences. Few of packets are also lost because of unavailability of route and it is the same for OLSR with or without QoS. This happens when a node movement causes the node to be disconnected from the network temporarily, until it re-joins the network again.

VII. Conclusions

This paper presented a link state QoS routing protocol for ad hoc networks. In order to improve quality requirements in routing information, delay and bandwidth measurements are applied. The implications of routing metrics on path computation are examined and the rationales behind the selection of bandwidth and delay metrics are discussed. The heuristic used in the standard OLSR finds a MPR set with minimal

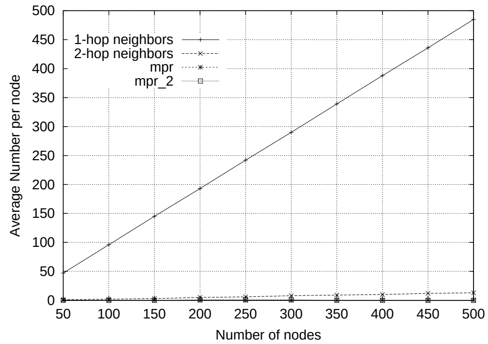

Fig. 15. Nodes in 200x200m^2 with 200m as TR.

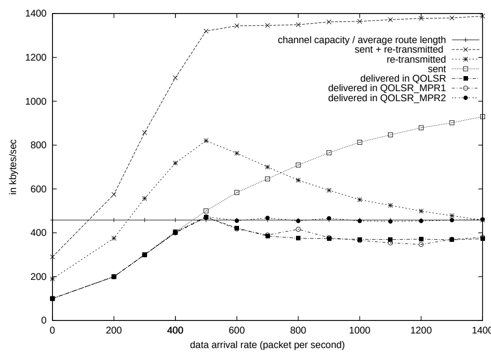

Fig. 16. Data load transmitted in varying load conditions.

size. There is no guarantee that OLSR finds the optimal widest path. We have proposed two heuristics that allow OLSR to find the maximum bandwidth path. In order to improve quality requirements in the MPRs selection and also in routing information, delay and bandwidth measurements are applied. Delay and bandwidth are calculated between each node and its neighbors having direct and symmetric link. We have demonstrated and also by simulations that QOLSR_MPR2 finds optimal widest paths using only the known partial network topology. From the analysis of the static and mobile networks simulation, QOLSR and QOLSR_MPR1 present the same performances.

REFERENCES

[1] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol," *RFC 3626*, October 2003.

[2] "http://www.ietf.org/html.charters/manet-charter.html."

[3] A. Qayyum, L. Viennot and A. Laouiti, "Multipoint relaying technique for flooding broadcast message in mobile wireless networks," *HICSS: Hawaii International Conference on System Sciences. Hawaii, USA*, January 2002.

[4] A. Munaretto, H. Badis, K. Al Agha and G. Pujolle, "QoS for Ad hoc Networking Based on Multiple Metrics: Bandwidth and Delay," *In the proceedings of IEEE MWCN2003, Singapore*, October 2003.

[5] H. Badis, A. Munaretto, K. Al Agha and G. Pujolle, "Optimal Path Selection in a Link State QoS Routing Protocol," *In the proceedings of IEEE VTC2004 spring, Italy*, May 2004.

[6] J. T. T. W. Chen and M. Gerla, " QoS Routing Performance in Multihop, Multimedia, Wireless Networks," *IEEE International Conference on Universal Personal Communications' 97*, pp. 451–557, October 1997.

[7] C. Perkins and P. Bhagwat, " Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Association for Computing Machinery's Special Interest Group on Data Communication'94*, pp. 234–244, 1994.

[8] C. R. Lin and J. S. Liu, "QoS Routing in Ad-Hoc Wireless Networks, Vol.17, No.8," *IEEE Journal On Selected Areas In Communications*, pp. 1426–1438, August 1999.

[9] Kazantzidis and M. Gerla, "End-to-end versus Explicit Feedback Measurement in 802.11 Networks," *In Seventh IEEE Symposium on Computers and Communications*, 2002.

[10] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. Vol.14, no.7, pp. 1228–1234, September 1996.

[11] S. Chen, "Routing Support for Providing Guaranteed End-to-End Quality of Service," *PhD, in Engineering Collage. Urbana*, 1999.

[12] M. Al-Fawaz and M. E. Woodward, "Fast Quality of Service Routing Algorithms with Multiple Constraints," *8th IFIP Workshop on ATM&IP, Ilkely, UK*, 2000.

[13] L. H. Costa, S. Fdida and M. B. Duarte, "Distance-vector QoS-based Routing with Three Metrics," *NETWORKING*, pp. 847–858, 2000.

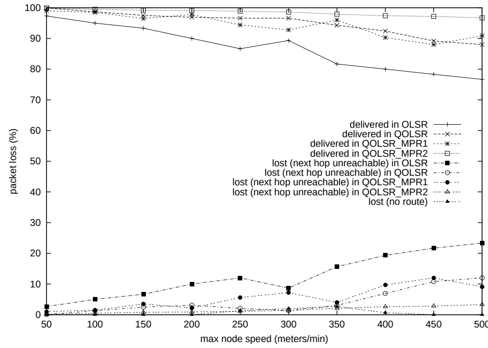

Fig. 17. Data load transmitted in varying node speeds.

[14] S. Cheng and K. Nahrstedt, “On finding multi-constrained paths,” *ICC’98, Atlanta, Georgia*, 1998.

[15] H. NEVE, P. Mieghem , “A multiple quality of service routing algorithm for PNNI ,” *IEEE ATM’98 Workshop, Fairfax, Virginia*, pp. 306–314, 1998.

[16] L. Guo and I. Matta, “Search Space Reduction in QoS Routing,” *Technical Report NU-CCS-98-09*, October 1998.

[17] E. Streenstrup, “Routing in Communication Networks,” *Prentice Hall International, Inc., Englewood Cliffs*, 1995.

[18] A. Qayyum, L. Viennot and A. Laouiti, “An efficient technique for flooding in mobile wireless networks,” *Research Report RR-3898, INRIA*, February 2000.

[19] P. Jacquet, P. Muhletaler, A. Qayyum, A. Laouiti, T. Clausen and L. Viennot, “Optimization Link State Routing Protocol,” *IEEE INMIC, Pakistan*, December 2001.

[20] E. W. Dijkstra, “A Note on Two Problems in Connection with Graphs,” *Numer. Math., 1:269-271*, 1959.

[21] T. H. Cormen, C. E. Leiserson and R. L. Rivest, “Introduction to algorithms,” *MIT Press, Cambridge, MA*, 1990.

[22] M. L. Fredman and R. E. Tarjan, “Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms,” *J. Assoc. Comput, 34:596-615*, Mach 1987.

[23] F. Kuipers, P. Van Mieghem, T. Korkmaz and M. Krunz, “An Overview of Constraint-Based Path Selection Algorithms for QoS Routing,” *IEEE Communications Magazine*, vol. Vol.40, No.12, December 2002.

[24] H. Badis and K. Al Agha, “ Scalable model for the simulation of OLSR and Fast-OLSR protocols ,” *Med-hoc-Net’03*, June 2003.