

Energy Level Accuracy in Mobile Ad-Hoc Networks Using OLSR

Rana Alhalimi
School of Computer Science
Carleton University
1125 Colonel By Drive
Ottawa, Ont., K1S 5B6 CANADA
ralhali2@scs.carleton.ca

Thomas Kunz
Systems and Computer Engineering
Carleton University
1125 Colonel By Drive
Ottawa, Ont., K1S 5B6 CANADA
tkunz@sce.carleton.ca

ABSTRACT

To support energy-efficient routing, accurate state information about energy level should be available. But due to bandwidth constraints, communication costs, high loss rate and the dynamic topology of MANETs, collecting and maintaining up-to-date state information is a very complex task. In this work, we use Optimized Link State Routing (OLSR) as the underlying routing protocol. We report the quantification of state information accuracy under different traffic rates. We are focusing on energy level as QoS metric, which has been used for routing decisions in many energy-efficient routing protocol proposals. State information accuracy is defined as the average difference between perceived energy level (by the node making a routing decision) and its actual value. The results show that state information is inaccurate, especially under high traffic rates. Tuning the OLSR protocol parameters has no noticeable impact on inaccuracy levels. Based on our inaccuracy level analysis, we propose three additional techniques as an attempt to reduce inaccuracies. We compare the different techniques against each other and against the basic OLSR protocol. Two of our proposed techniques show significant improvements in inaccuracy levels. In particular, a technique we call smart prediction achieves highly accurate perceived energy levels under all traffic loads.

Categories and Subject Descriptors

C.2.2 [Network Protocols]: Routing protocols

General Terms

Algorithms, Measurement, Performance, Design, Experimentation.

Keywords

MANETs, state information accuracy, energy level, Quality of service (QoS).

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

WICON 2007, October 22-24, 2007, Austin, Texas, USA.
Copyright 2007 ACM 987-963-9799-04-2 \$5.00.

1. INTRODUCTION

Optimized Link State Routing (OLSR) is a routing protocol used for Mobile Ad-Hoc Networks (MANET) [3]. It is a best-effort proactive protocol. Proactive protocols are characterized by all nodes maintaining routes to all destinations at all times through the periodic exchange of protocol messages. This gives them the advantage of having pre-computed routes available when needed and to propagate topology changes in bulk updates to many nodes. OLSR performs hop-by-hop routing, where each node uses its most recent topology information for routing.

OLSR is highly focused on reducing the protocol overhead. As a result, information about QoS-related state is not propagated throughout the network. But with the rising popularity of multimedia applications and the potential commercial usage of MANETs, QoS support in ad-hoc networks has become a very critical issue and a range of QoS signaling and routing protocols have been proposed.

To support QoS routing, state information such as energy level, bandwidth or queue length should be available when making routing decisions at a node. But because the quality of wireless links changes quite frequently due to mobility and changes in surroundings, coupled with the limited wireless bandwidth, collecting and updating such knowledge is a nontrivial task.

Since OLSR works on periodic exchange of messages, QoS-related state information might not be up to date at any instance of time. Therefore, nodes might have inaccurate information about other nodes in the network, which might have a negative impact on the performance of the network. The motivation of this research is to quantify the accuracy of the QoS-related state information in ad-hoc networks under different conditions and if possible, devise techniques to reduce inaccuracies.

State information represents a QoS-related state. It could be a node attribute such as energy level and queue length, or a link attribute such as bandwidth, delay, or error rate. In this study, we are interested in node attributes, and specifically energy level. Due to energy constraints, a number of energy-aware routing protocols have been proposed [2][13][14]. Our work investigates the possibility of making OLSR an energy-aware routing protocol by evaluating how accurately individual nodes learn about the current energy levels for other nodes when piggybacking this information onto OLSR control messages.

State information accuracy specifies how accurate is the available QoS-related state information relative to their actual

values. Throughout this paper, we refer to the latter as the actual value and the former as the perceived value.

The main contributions of this work are:

- Quantification of the inaccuracy of energy level information in ad-hoc networks under different traffic loads, using OLSR as the underlying routing protocol.
- Studying the impact of tuning the OLSR protocol parameters on the inaccuracy level of the energy level metric.
- Suggesting and developing techniques to reduce inaccuracies.
- Evaluating the performance of the proposed techniques and comparing them to the basic OLSR protocol performance.

The paper is organized as follows: A review of the state of the art is presented in Section 2. Section 3 summarizes the core functionality of the Optimized Link State Routing Protocol (OLSR). Section 4 describes how we propagate QoS state information in OLSR and Section 5 shows, using simulation, the inaccuracy level we achieve with OLSR under different traffic loads. This section also analyzes the effect of varying various protocol parameters on the inaccuracy level and suggests ways to improve on the inaccuracy problem for energy level. Section 6 discusses three techniques for improving the overall energy inaccuracy level and evaluates their performance through simulations. The final section concludes our work and suggests possible future work.

2. RELATED WORK

OLSR is a well-known routing protocol for ad hoc networks. It has been broadly examined [4][11], implemented and deployed [9][15]. [11] provides performance measurements in a real test-bed and concludes that OLSR suffers from high variability of performance depending on how far apart are the nodes, and from unfairness depending on the topology and on the nature of the traffic. It suggests that QoS features could complement the performance of the OLSR protocol.

[6] develops a QoS version of the OLSR protocol, based on link bandwidth as QoS metric. This QoS OLSR protocol attempts to find paths with maximum bottleneck bandwidth. In order to support QoS (provide optimal bandwidth path), changes in the link bandwidth must be propagated for the correct computation of the best bandwidth route. [6] evaluates the performance of this QoS OLSR model under different bandwidth change threshold values and compares it to the basic OLSR protocol performance. These threshold values define a tradeoff between the accuracy of link-bandwidth information and the additional overhead the routing protocol introduces. Three threshold values (20%, 40% and 80%) are used. The results show that amongst the proposed QoS OLSR algorithms, 20% QoS OLSR calculates the routes that are closest to the optimal routes compared to the 40% and 80% QoS OLSR. This is due to the fact that 20% QoS OLSR updates the bandwidth condition most frequently and consequently gets the most accurate bandwidth information. [6] demonstrates that OLSR has a potential for QoS routing. It also shows that the availability of more accurate state information throughout the network, via more frequent updates, improves the performance of QoS routing. However, their work does not investigate quantitatively the level of accuracy of the QoS metric (link bandwidth).

[4] investigates the impact of extending topology knowledge on the OLSR protocol performance. In an OLSR network, nodes have partial topology knowledge in which only a subset of the

links are known to the node to reduce the protocol overhead. Increasing the partial topology information provides a more robust and accurate topology view. It is achieved by increasing the number of links advertised and number of nodes advertising links. In OLSR this can be done by varying two protocol parameters (named MPR-coverage and TC-redundancy). In order to determine the effect of advertising redundant and more accurate topology information on the performance of the OLSR routing protocol, [4] studies the impact of increasing the MPR-coverage parameter. Their results show higher packet delivery rates under moderate node mobility when increasing the redundancy of topological information and retransmissions provided by a higher MPR-coverage.

The research in [16] expands the work done in [4] and entirely focuses on understanding the trade-offs of increasing accurate topology knowledge. It investigates the impact of tuning the MPR-coverage and TC-redundancy parameters on the OLSR performance. It shows that delivery rates are not affected by the overhead resulting from advertising redundant information. Both [4] and [16] focus on having more accurate information at the topology (network) level and how it affects the routing protocol performance. In other words, they study the effect of tuning the OLSR protocol parameters on accuracy in terms of network status (existing nodes and links) and not in terms of state information available at nodes and links.

The most relevant body of work to the problem of QoS routing in the presence of inaccurate information is a set of papers aimed at exploring state-aggregation issues and their impact on routing performance in large networks. They emphasize on developing good aggregation techniques that minimize inaccuracy in network state information, while allowing substantial reductions in the amount of state data. In particular the work in [8] addresses the information collection problem for QoS-based services in mobile environments. Specifically, it proposes a family of information collection policies that vary in the granularity at which system state information is represented and maintained. It evaluates the impact of information collection algorithms on the performance of QoS-based resource provisioning. The work in [8] proposes two approaches to collecting location information for mobile applications. Fine-grained approaches maintain current location of each individual mobile client, while coarse-grained collection captures information at an aggregate level of multiple clients. [8] concludes that coarse-grained mobility information is sufficient for effective resource provisioning, whereas fine-grained mobility information introduces a very high overhead. The work in [8] however does not evaluate the impact of fine-grained mobility information on resource provisioning and how it compares to coarse-grained mobility information. Since we are dealing with relatively smaller networks, our work investigates the impact of collecting fine-grained data on the accuracy of state information as an upper bound on achievable accuracy.

[1], [7] and [12] investigate the impact of inaccuracies, in the available network state and metric information, on the path selection process for flows which require QoS guarantees. In particular [12] evaluates the impact of inaccurate state information on the performance and overheads of QoS routing by evaluating periodic and triggered updates. They use connection blocking as their performance measure. Connection blocking defines the percentage of times a connection request from a source to a destination fails. They draw a distinction between routing failures and setup failures. Routing failures occurs when

the source cannot compute a feasible path for the new connection. In contrast, setup failures occur when the source selects a seemingly feasible path that ultimately cannot support the new connection. With a periodic update policy, larger periods substantially increase connection blocking, ultimately outweighing the benefits of QoS routing. The results show that a periodic update policy alone cannot meet the dual goals of low blocking probability and low overhead in realistic networks. In contrast, experiments with triggered updates show that coarse-grained triggers do not have a significant impact on the overall blocking probability, although larger triggers shift the type of blocking from routing failures to more expensive setup failures.

In summary, none of the prior works fully address the problem of quantifying the inaccuracy in state information. On the other hand, our work focuses on understanding the impact of how the QoS-related state is collected and how it is updated.

3. DESCRIPTION OF OLSR

The IETF Working Group introduced the Optimized Link State Routing Protocol (OLSR) for mobile ad hoc networks [3]. The protocol is an optimization of the pure link state algorithm. The key concept used in the protocol is that of multipoint relays (MPRs). Each node selects a set of its neighbor nodes as MPRs. Only nodes selected as 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 (re-)transmissions required.

Nodes selected as MPRs also have a special responsibility when declaring link state information in the network. Indeed, the only requirement for OLSR to provide shortest path routes to all destinations is that MPR nodes declare link-state information for their MPR selectors.

Nodes which have been selected as multipoint relays by some neighbor node(s) announce this information periodically in their control messages. Thereby a node announces to the network that it has reachability to the nodes which have selected it as an MPR. In route calculation, the MPRs are used to form the route from a given node to any destination in the network. Furthermore, the protocol uses the MPRs to facilitate efficient flooding of control messages in the network.

Due to its proactive nature, OLSR works with a periodic exchange of messages. The key messages are Hello and TC messages. Hello messages are periodically exchanged to inform nodes about their neighbors and their neighbors' neighbors and are 1-hop broadcast messages. The 2-hop neighborhood information is then used locally by each node to determine MPRs. In contrast, TC messages are flooded through the network to inform all nodes about the (partial) network topology. At a minimum, TC messages contain information about MPRs and their MPR selectors.

To control the protocol overhead, OLSR defines a few parameters. The Hello-interval parameter (default: 2 seconds) represents the frequency of generating a Hello message. Increasing the frequency of generating Hello messages leads to more frequent updates about the neighborhood and hence a more accurate view of the network.

The TC-interval parameter (default: 5 seconds) represents the frequency of generating a TC message. TC messages are one of the major sources of overhead in MANETS, as they are flooded throughout the network, but they facilitate the topology discovery process. Since nodes learn about the whole topology by

exchanging TC messages, the more frequently nodes generate TC messages, the more recent the knowledge nodes have about the topology.

The MPR-coverage parameter (default: 1) allows a node to select redundant MPRs. The criterion for selecting MPRs is that all 2-hop neighbors must be reachable through at least one MPR node. Nodes should select their MPR set to be as small as possible in order to reduce protocol overhead. Redundancy of the MPR set affects the overhead through affecting the amount of links being advertised, since a node will be selected by more neighbor nodes as an MPR, the amount of nodes advertising links, since more nodes will be selected as MPRs, and the efficiency of the MPR flooding mechanism. On the other hand, redundancy in the MPR set ensures that reachability for a node is advertised by more nodes.

The TC-redundancy parameter (default: 0) specifies, for the local node, the amount of information that may be included in the TC message. A TC-redundancy of 0 specifies that the advertised link set of the node is limited to links to its MPR selectors. A TC-redundancy of 1 specifies that the advertised link set of the node is the union of links to its MPR selectors and to other MPRs. A TC-redundancy of 2 specifies that the advertised link set of the node is the full neighbor link set. The TC-redundancy parameter affects the overhead through affecting the amount of links being advertised as well as the amount of nodes advertising links.

4. OLSR MODIFICATIONS

In order to quantify the accuracy of state information, the QoS-related state needs to be propagated throughout the network. There are two ways in which QoS-related state can be propagated throughout the network. Either we define a new message type to carry the QoS-related state information, or we include it in the OLSR protocol messages (Hello and TC messages) to be available to other nodes in the network. With the first approach, a new message type has to be defined and exchanged. This will incur a potentially large overhead in the network since more messages will be exchanged. In addition, these messages will include a lot of redundant information and the same gain can be obtained by including the QoS-related state information in the OLSR protocol messages. Therefore, the second approach was taken to propagate the QoS-related state information to be available to other nodes in the network.

Through the exchange of OLSR control messages, each node accumulates information about the network. This information is stored according to the OLSR specifications. However, to store the QoS-related state associated with a node, a new field was added to the neighborhood information base and to the topology information base maintained by the protocol. To populate these fields, the message format of Hello and TC messages was extended as well.

Extended Hello messages are broadcast to all one-hop neighbors. They contain not only a list of addresses of neighbors, but also the most recent QoS-related state associated with those neighbors from the sender node's perspective. In addition to that, the message also contains the QoS-related state of the sender node itself at the time the message is generated. The other fields are loaded according to the OLSR specifications.

Extended TC messages are broadcast and retransmitted by the MPRs in order to diffuse topology information into the entire network. TC messages contain not only a list of addresses of a node's MPR selectors, but also the QoS-related state associated

with those nodes from the originator node’s perspective. In addition to that, the message also contains the QoS-related state of the originator node at the time the message is generated.

Moreover, a timestamp of when the data was sent out (created) is collected along with the QoS-related state. Hello and TC messages were extended to carry timestamp information. Timestamps will be used to analyze delays and “knowledge age”. This implies that the network nodes are time-synchronized. Since there are many clock synchronization solutions available such as Network Time Protocol (NTP), we will assume that the network is synchronized. As our analysis deals with delays in the order of seconds, no particularly tight clock synchronization (i.e. at the level of micro seconds) is required.

5. SIMULATION RESULTS

To quantify the accuracy of state information in different conditions, we ran extensive simulations using the NS2 simulator with the OOLSR implementation of OLSR provided by the Hipercom project (NS2 version 2.27 with OOLSR version 0.99.15) [9].

For each sample point, 10 random network snapshots are generated. The simulation results presented are an average over these 10 scenarios. The same set of 10 scenarios are used for all simulations for a given sample point, hence the different parameters are evaluated under identical conditions.

Table 1: Simulation Parameters

Simulator Parameters	
Propagation model	TwoRayGround
Network Type	IEEE 802.11
Transmission Range	250 m
Mobility Model	Static Network
Scenario Parameters	
Topology area	1000*1000
Number of nodes	50
Simulation time (secs)	200 seconds
Energy Model Specifications	
Initial Energy (Joules)	1000
Transmission Power (Watt)	1.4
Receiving Power (Watt)	1.0
Idle Power (Watt)	0.83

In the following, we describe the simulation conditions that we generally used in our simulation experiments. Unless otherwise specified, while discussing a certain experiment, these are the conditions under which the experiment was conducted. In each run, we have 20 communication pairs. Each source sends 128 bytes CBR packets at different intervals. The simulations are done with 5 different intervals to study the effect of low, medium and high traffic rates on the QoS metric. The intervals are 0.2, 0.14, 0.09, 0.04 and 0.02 seconds – an interval of 0.2 means a node will send a packet every 0.2 of a second (i.e. 5 packets per second). At the lower intervals, significant number of packets (data and control packets) will be lost due to network congestion, testing the performance under stressful overload scenarios. These may not be representative of long-term sustained traffic, but often occur in MANETs due to the link bandwidth limitations and the

bursty nature of data traffic. The energy model values are based on the study conducted in [5].

During the simulation, a snapshot for the whole topology is taken every second. It contains information such as:

- Which other nodes a node can hear and what it believes to be their energy levels. Node n_2 is said to be heard by node n_1 if there exists a route from n_1 to n_2 .
- If a node n_2 is heard by node n_1 , how many hops away is it from node n_1 .
- The actual energy level for each node.

Also during the simulation, we keep records of all Hello and TC messages sent and received. We are interested in determining the average overall inaccuracy level. We define this as the average difference between a node’s actual energy level and what other nodes believe its energy level is. More formally, overall inaccuracy level is calculated as: for each pair of nodes (n_1, n_2) in the network such that n_1 can hear n_2 , the sum of the absolute difference between the actual energy level of n_2 and what n_1 believes to be the energy level of n_2 , for all time points (every second of the simulation) divided by the total number of pairs. This is done for 10 scenarios and the overall inaccuracy level is the average over the 10 scenarios. We ignore the first 50 seconds of the simulation to have the network stabilized and to ignore transient startup conditions. We also are only considering pairs of nodes (n_1, n_2) such that n_1 can hear n_2 . During the simulation and due to message loss and delays, some nodes get temporarily disconnected from other nodes. Therefore they are not considered part of the network as they are not used for routing. For overall inaccuracy level calculation, only visible nodes are considered.

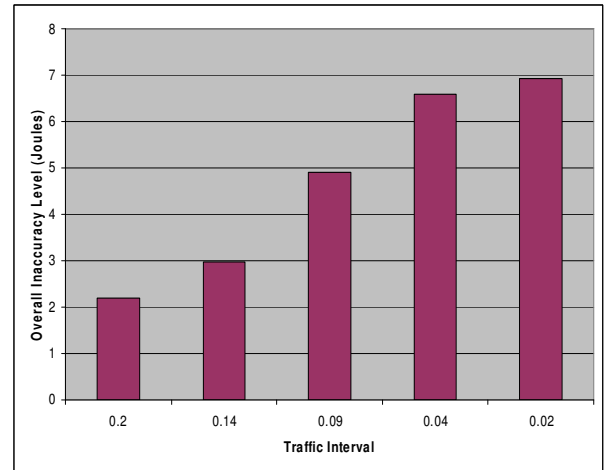


Figure 1: Energy Overall Inaccuracy Level under Different Traffic Rates with Default OLSR Parameters

Figure 1 shows the overall inaccuracy level under low, medium and high traffic rates using the default OLSR parameters (i.e. Hello-interval 2, TC-interval 5, MPR-coverage 1 and TC-redundancy 0). As expected, traffic does introduce a considerable level of inaccuracy to the network. And as traffic rate increases, the level of overall inaccuracy increases. As traffic rate increases, the network becomes more congested. Since network buffers have limited capacity in terms of storage and processing of arriving packets, this affects significantly the performance of the network, causing long delays and packet loss as packets will be waiting in the queues for processing or will be dropped due to overflow.

And as nodes send/receive traffic at a higher rate, they consume energy at a higher rate. Consequently, information available to the nodes in the network becomes outdated and no longer accurate.

These observations are confirmed by our experiments. Table 2 shows the average nodal energy consumption, Hello/TC message delay and Hello/TC message loss as well as average knowledge age under the different traffic rates. Hello/TC message

loss is calculated as the percentage of Hello/TC messages received compared to the number of received messages using traffic interval 0.2 as base case (assuming that no messages are lost in this case). Average knowledge age is calculated as: for all pairs of nodes (n1, n2) in the network, the average of how old is the information n1 has about n2.

Table 2: Hello/TC Message Delay and Loss, Average Knowledge Age and Nodal Energy Consumption

Traffic Interval	Nodal Energy Consumption (Joules/ Sec)	Hello Message Delay (secs)	TC Message Delay (secs)	Hello Message Loss (%)	TC Message Loss (%)	Average Knowledge Age (secs)
0.2	0.926	0.016	0.114	-	-	2.94
0.14	0.948	0.12	0.542	4%	3%	3.68
0.09	0.956	0.478	1.586	9%	17%	5.46
0.04	0.965	0.866	2.127	15%	36%	7.03
0.02	0.971	1.046	2.143	18%	45%	7.06

Table 3: Overall Inaccuracy Level (in Joules) under Different Traffic Rates with Different OLSR Parameters

Traffic Interval		0.2	0.14	0.09	0.04	0.02
OLSR Parameters						
Hello-interval	2	2.6225 [2.37,2.87]	3.4287 [2.96,3.9]	5.1793 [4.8,5.56]	6.7602 [5.94,7.58]	6.839 [5.88,7.81]
	1	2.213 [1.99,2.43]	2.956 [2.47,3.44]	4.846 [4.36,5.33]	6.992 [6.16,7.82]	7.366 [6.37,8.36]
TC-interval	5	2.6225 [2.37,2.87]	3.4287 [2.96,3.9]	5.1793 [4.8,5.56]	6.7602 [5.94,7.58]	6.839 [5.88,7.81]
	4	2.196 [1.95,2.43]	2.978 [2.49,3.46]	5.014 [4.59,5.44]	6.774 [5.9,7.64]	6.896 [5.76,8.03]
	3	1.981 [1.76,2.2]	2.865 [2.38,3.35]	5.097 [4.66,5.53]	7.004 [6.2,7.8]	7.273 [6.15,8.39]
MPR-coverage	1	2.6225 [2.37,2.87]	3.4287 [2.96,3.9]	5.1793 [4.8,5.56]	6.7602 [5.94,7.58]	6.839 [5.88,7.81]
	2	2.241 [2.05,2.43]	2.888 [2.47,3.31]	4.718 [4.22,5.21]	6.925 [5.99,7.86]	6.927 [5.89,7.96]
TC-redundancy	0	2.6225 [2.37,2.87]	3.4287 [2.96,3.9]	5.1793 [4.8,5.56]	6.7602 [5.94,7.58]	6.839 [5.88,7.81]
	1	2.283 [2.06,2.5]	3.062 [2.6,3.52]	5.014 [4.51,5.52]	6.81 [5.98,7.64]	6.832 [5.9,7.77]
	2	2.143 [1.98,2.3]	3.005 [2.51,3.5]	4.89 [4.47,5.31]	6.79 [6.01,7.57]	7.257 [6.18,8.33]

OLSR is very focused on overhead reduction. Therefore OLSR default parameters are set to achieve an acceptable performance (without intensions for QoS support) while keeping the overhead as low as possible. We therefore investigated if it is possible to trade off cost (overhead) to gain better performance (more accurate state information). We analyzed the impact of sending more frequent Hello and TC messages (by reducing Hello and TC intervals) as well as more redundant topology information (by increasing TC-redundancy and MPR-coverage parameters).

As shown in Table 3, a 95% confidence interval is calculated for each parameter under the different traffic rates. Increasing the number of protocol messages, including both Hello and TC messages, or increasing the amount of information advertised

improves the overall inaccuracy level under low traffic rate. Under medium to high traffic rate, a 95% confidence interval test shows that the difference between the overall accuracy levels under the different OLSR parameters is not statistically significant. Under medium traffic rates (traffic intervals 0.14 and 0.09) we observe a trend towards better inaccuracy levels when varying the OLSR parameters. On the other hand, under high traffic rates (traffic interval of 0.04 and 0.02), the trend is towards less accurate energy levels when varying the OLSR parameters.

The results observed here are a direct consequence of the increased level of congestion in the network which results in high message loss and delay and hence less accurate state information.

Table 3 shows that among the different OLSR parameters, under low traffic rates, improvement in inaccuracy level is best achieved by a TC-interval of 3 with some improvement achieved by lowering the Hello-interval from 2 to 1. According to the OLSR specifications, the ratio between Hello-interval and TC-interval is 2 to 5. Therefore using a combination of TC-interval of 3 and Hello-interval of 1 seems appropriate. Figure 2 compares the overall inaccuracy level using the default parameters for OLSR, called default OLSR, versus using a combination of Hello-interval 1 and TC-interval 3, keeping the other parameters unchanged (MPR-coverage 1 and TC-redundancy 0), called Hello1TC3 OLSR.

Hello1TC3 OLSR improves the overall inaccuracy level under low traffic rates. As we move towards high traffic rates, Default OLSR starts to outperform Hello1TC3 OLSR due to the overhead added to the network (more frequent HELLO and TC messages).

The obvious approach of increasing the frequency of protocol messages or the amount of information advertised improves the energy overall inaccuracy level under low traffic rates only. Therefore, we further analyzed the results based on knowledge age to obtain better insights of what can be done to improve inaccuracies under higher traffic rates or even improve it further under low traffic rates.

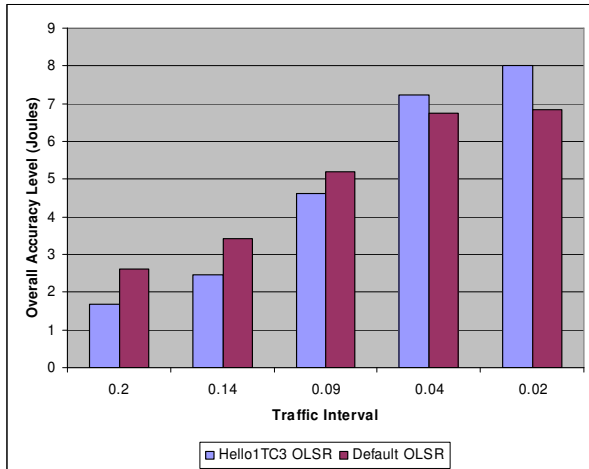


Figure 2: Overall Inaccuracy Level of default OLSR vs. Hello1TC3 OLSR

As a first step, we correlated knowledge and inaccuracy level. Knowledge age inaccuracy level represents the average difference between a node’s actual energy level and what other nodes believe its energy level is, categorized by how old the knowledge is. The knowledge between all pairs of connected nodes is categorized into 22 different groups based on how old the data is. The 1st group has all pairs with knowledge that is 0 (inclusive) to 1 second old (exclusive). And the 2nd group has all pairs with knowledge that is 1 (inclusive) to 2 seconds old (exclusive) and so on. All nodes that have data which is above 21 seconds old are put in the last group since we observed that it is very rare to have data that is older than 21 seconds. According to the OLSR implementation with default parameter values, it is typical to have data that is up to 6 seconds old which is the TC-interval default (every 5 seconds) plus 1 second to propagate the message. Each tuple in a node’s topology database will be expired after 15

seconds (or 3 times the TC-interval), so a reasonable upper bound on knowledge age under not heavily congested network conditions is therefore 21 seconds .

Figure 3 illustrates the results of knowledge age inaccuracy level under the five different traffic rates and using the default OLSR parameters. It can be clearly seen from this figure that the older the knowledge about other nodes is, the less accurate their knowledge about the node’s energy levels are. This trend is easily explained by the fact that the energy level is a monotonically decreasing metric. Based on these observations, we will explore ways to increase energy level accuracy.

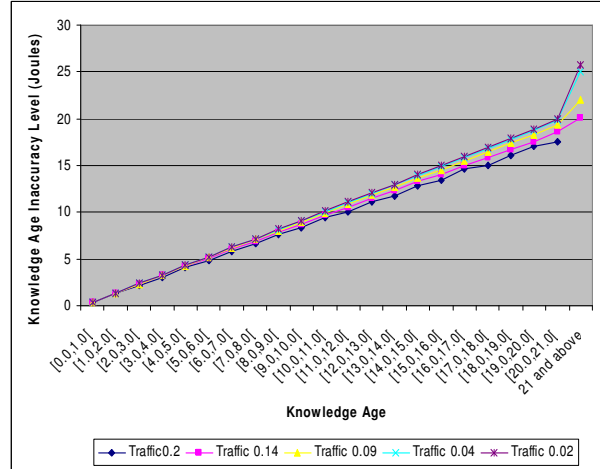


Figure 3: Knowledge Age Inaccuracy Level under Different Traffic Rates and Default OLSR Parameters

6. IMPROVING ACCURACY

Before exploring how to improve the accuracy levels, we first consider whether it is worthwhile propagating the energy level information at all.

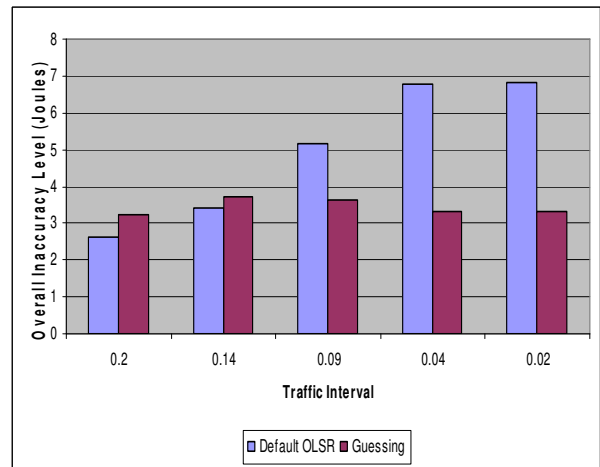


Figure 4: Guessing vs. Default OLSR

We “implemented” a *Guessing* mechanism in which, rather than propagating energy level information, each node will guess the energy level of the other nodes based on its own energy consumption. For this approach to work, we assume that the

networks under consideration are homogeneous: all nodes are configured with the same wireless card and therefore use the same energy model (same transmitting, receiving and idle power consumption) and they start with the same initial energy level. In this case every node trivially assumes that the energy level of the other nodes in the network is the same as its own energy level.

As shown in Figure 4, at first glance it seems that propagating the energy level information is better than Guessing under low traffic rate, whereas under higher traffic rates, Guessing outperforms the default OLSR. However, under the Guessing technique a node assumes that the energy level of all the other nodes is the same as its own energy level, leading to increased inaccuracies over time. A node in the centre of the network will frequently receive and relay many other packets, overestimating the energy consumption of nodes at the network edge and vice versa. Figures 5 and 6 plot the overall inaccuracy level as a function of time under traffic intervals of 0.2, and 0.02 (respectively). It is clear that the overall inaccuracy level increases linearly with time for Guessing. And as the traffic rate increases, the crossover point (at which point Guessing becomes less accurate than Default OLSR) will occur later. This means that the results in Figure 4 are somewhat oversimplified. If we ran the simulations a little longer under the higher traffic rates, Default OLSR will outperform Guessing and as a result the trend would be that Default OLSR outperforms Guessing under all traffic rates.

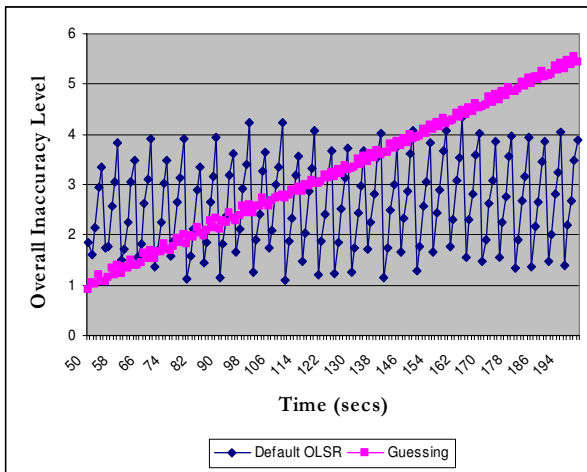


Figure 5: Overall Inaccuracy Level as a Function of Time under 0.2 Traffic Interval

Based on the energy level knowledge age inaccuracy level analysis in the previous section, the results show that the main source of inaccuracies is the existence of old data. A first approach to address this problem would be for a node to selective probe nodes for which it holds old information (above a certain threshold, say), hopefully obtaining more recent (and therefore more accurate) state information. In parallel work reported elsewhere [10], we experimented with such an approach and found that it, in general, is not capable of achieving significant reductions in inaccuracy.

Alternatively, as energy level is a monotonically decreasing metric, we can adjust “old” information and predict the current value. Our idea is therefore to have every node locally adjusts nodes’ old energy levels based on their “behavior pattern”. We

propose a *Prediction* mechanism in which each node locally extrapolates an expected energy level based on old (reported) energy levels and energy consumption rate pattern for all other nodes. Every second of the simulation, instead of having every node report its perceived knowledge for every other node in the network as is, a node’s perceived value is first adjusted based on its past behavior (consumption rate) and then the adjusted value is reported. For example, at second 51 of the simulation, node 0 has an energy level of 958.581 associated with node 1 and this knowledge is timestamped 47.5916. At time 52, the perceived energy level for node 1 (from node 0 perspective) is 954.998, timestamped 49.5884. Node 0 computes the consumption rate of node 1 as: $(958.581 - 954.998) / (49.5884 - 47.5916) = 1.7943$. Then it adjusts the estimated energy level of node 1 to be: $954.998 - (1.7943 * (52 - 49.5884)) = 950.6707$ (at time 52). If no prediction is possible, as no consumption rate is known yet, the last reported energy level will be used without adjustment.

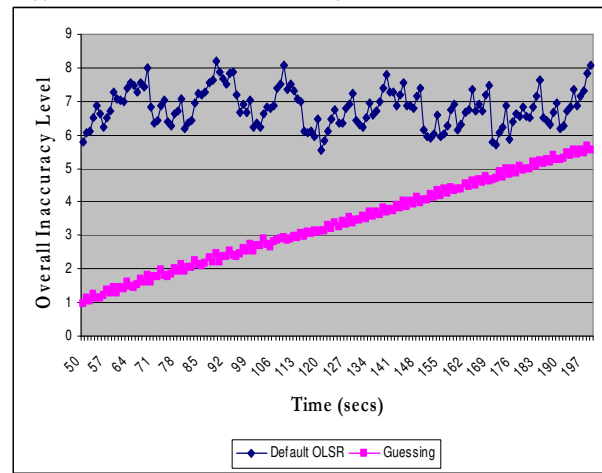


Figure 6: Overall Inaccuracy Level as a Function of Time under 0.02 Traffic Interval

The only drawback of the Prediction algorithm is the need to wait for two different perceived value readings, so a consumption rate can be calculated and used to adjust the perceived values. Table 4 shows the percentage of number of times an adjustment takes place under different traffic rates. Under high traffic loads, adjustments happen less rarely. Protocol control messages are lost/delayed, and as a result nodes will not “hear” other nodes. After a node is deemed unreachable, we go through the startup phase again, where we need at least two successive reports to be able to calculate a consumption rate. We therefore propose the *Smart Prediction* algorithm which is an enhanced version of the prediction algorithm so that adjustments take place almost all the time.

Table 4: Percentage of Times Adjustments Take Place under Different Traffic Rates

Traffic Interval	Number of Adjustments (in %)
0.2	97%
0.14	95%
0.09	91%
0.04	82%
0.02	78%

In the Smart Prediction algorithm, for every pair of nodes (n_1 , n_2), if n_2 's consumption rate is not yet known, n_1 adjusts the perceived value of n_2 based on the average of all known consumption rates for other nodes. If n_1 knows not a single consumption rate for other nodes, it adjusts n_2 's perceived energy level based on its (n_1 's) consumption rate.

Using all known nodes' consumption rates eliminates the domination of outliers and ensures closeness to the actual consumption rate, assuming again that nodes are somewhat homogeneous in their wireless cards. Unlike the Guessing approach, we do not assume or require that all nodes have the same initial energy level.

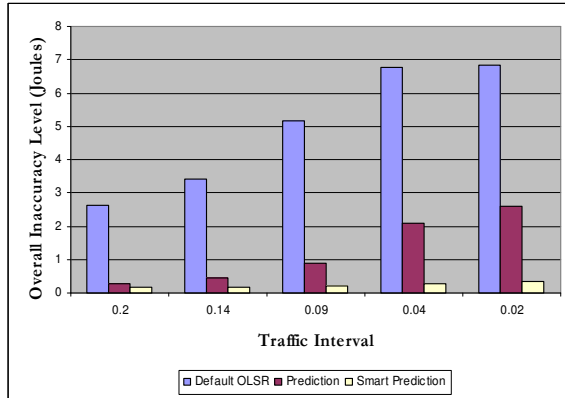


Figure 7: Overall Inaccuracy Level using Default OLSR vs. Prediction vs. Smart Prediction

According to Figure 7, our Prediction algorithms improve the overall inaccuracy level under different traffic rates. The improvement under higher traffic rates is not as high as it is under lower traffic rates. For an adjustment to take place, a node must wait for two different perceived values. But under high traffic rates, due to message loss and delays, the percentage of times adjustments take place is much lower, as shown in Table 4. Since the Smart Prediction algorithm takes care of the problem of not being able to adjust the perceived energy level value all the time, it achieves much better performance in terms of overall inaccuracy level, especially under higher traffic rates. Both the Prediction and the Smart Prediction algorithms outperform the Default OLSR protocol. At the same time, the Smart Prediction algorithm outperforms the Prediction algorithm in improving the overall inaccuracy level.

7. CONCLUSIONS AND FUTURE WORK

In this work, we used the Optimized Link State Routing (OLSR) protocol as the underlying wireless network routing protocol. We explored the quantification of state information accuracy (energy level) under different traffic rates. We showed that as traffic rate increases, the overall inaccuracy level increases. Tuning the OLSR protocol parameters did not have a noticeable impact on overall inaccuracy level.

A Guessing technique, in which a node simply guesses the energy level of other nodes based on its own energy level, shows that propagating the energy level information is indeed beneficial. Using Default OLSR, this does provide the nodes in the network with more accurate energy level information. Overall inaccuracy level using Guessing tends to drift linearly with time, increasing the gap between the guessed and actual energy level values.

We proposed two other techniques to reduce energy level inaccuracies, Prediction and Smart Prediction. Under the Prediction technique, a node's energy level is adjusted based on its past behavior (its own consumption rate). Smart Prediction is a modified version of the Prediction technique such that, if no behavioral pattern was known for a node, its energy level is adjusted based on the average of all known consumption rates for other nodes. The results show that both Prediction and Smart Prediction outperform the Default OLSR (OLSR with QoS-related state propagated, and using default parameters). Moreover, Smart Prediction outperforms Prediction since energy level adjustments take place all the time. In addition, the overheads associated with Prediction and Smart Prediction are exactly the same as the Default OLSR since no extra messages or fields are required.

The only restriction under the Smart Prediction technique is that nodes are assumed to be homogeneous in their wireless cards. As part of our future work, Smart Prediction could be evaluated when nodes have heterogeneous wireless interfaces. In other words, nodes are not required to have similar transmitting, receiving or idle power.

As the first step towards supporting QoS routing, our work reports the quantification of state information accuracy and proposes new techniques to reduce inaccuracies. But how effective is providing this more accurate information on the routing decision? Will providing more accurate energy level information help making better routing decisions compared to the Default OLSR solution? For example, will Prediction and Smart Prediction calculate the routes that are closest to the optimal routes, the routes with the highest remaining energy, compared to the Default OLSR? While the work in [6] showed that more accurate QoS state information leads to more accurate routing decisions, the authors only focused on available bandwidth as routing metric. Applying our techniques to energy-efficient variants of OLSR to quantify protocol performance impacts is therefore the main next step in our work.

8. ACKNOWLEDGMENTS

This work was supported by the National Sciences and Engineering Research Council of Canada.

9. REFERENCES

- [1] G. Apostolopoulos, R. Guerin, S. Kamat and S. Tripathi, Quality of Service Based Routing: A Performance Perspective, Proceedings of ACM SIGCOMM, Vancouver, Canada, Volume 28, No. 4, Pages 17-28, Oct. 1998.
- [2] J. H Chang and L. Tassiulas, Energy Conserving Routing in Wireless Ad-hoc Networks, Proceedings of IEEE INFOCOM, Tel Aviv, Israel, Volume 1, Pages 22-31, March 2000.
- [3] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol*, IETF RFC 3626, Oct. 2003.
- [4] T. Clausen, P. Jacquet and L. Viennot, *Investigating the Impact of Partial Topology in Proactive MANET Routing Protocols*, Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications, Volume 3, Pages 1374-1378, October 2002.

- [5] L. M. Freeneey and M. Nilsson, Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment, Proceedings of IEEE INFOCOM, Anchorage AK, Volume 3, Pages 1548-1557, April 2001.
- [6] Y. Ge, T. Kunz and L. Lamont, *Proactive QoS Routing in Ad-Hoc Networks*, Proceedings of the 2nd International Conference on Ad-Hoc Networks and Wireless, Montreal, Canada, October 2003.
- [7] R.A. Guerin and A. Orda, *QoS routing in networks with inaccurate information: theory and algorithms*, IEEE/ACM Transactions on Networking, Volume 7, Issue 3, Pages 350-364, June 1999.
- [8] Q. Han and N. Venkatasubramanian, *Information collection services for QoS-Aware mobile applications*, IEEE Transactions on Mobile Computing., Volume 5, Issue 5, Pages 518-535, May 2006.
- [9] Hipercom Project, OOLSR Implementation of the OLSR Optimized Link State Routing Protocol, <http://hipercom.inria.fr/oolsr/>, November 2004.
- [10] T. Kunz and R. Alhalimi, *Load-balanced routing in wireless networks: State information accuracy using OLSR*, to appear in Proceedings of the 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007), New York, USA, October 2007.
- [11] T. Plesse, et al., *OLSR Performance Measurement in a Military Mobile Ad-hoc Network*, Proceedings of the 24th International Conference on Distributed Computing Systems Workshops, Pages 704-709, October 2004.
- [12] A. Shaikh,, J. Rexford and K. G. Shin, *Evaluating the impact of stale link state on quality-of-service routing*, IEEE/ACM Transactions on Networking., Volume 9, No. 2, Pages 162-176, April 2001.
- [13] S. Singh, M. Woo and S. Raghavendra, *Power-Aware Routing in Mobile Ad Hoc Networks*, Proceedings of the ACM Mobile Computing and Networking Conference, Dallas, Texas, Pages 181-190, October 1998.
- [14] C.-K. Toh, *Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks*, IEEE Communications Magazine, Pages 2-11, June 2001.
- [15] A. Tonnesen, *Implementing and Extending the Optimized Link State Routing Protocol*, Master Thesis, Department of Informatics, University of Oslo, August 2004.
- [16] P.E. Villanueva-Pena, T. Kunz and P. Dhakal, *Extending network knowledge: Making OLSR a Quality of Service conductive protocol*, Proceedings of the International Conference on Communications and Mobile Computing, Vancouver, Canada, Pages 103-108, July 2006.