

## Performance Analysis of OLSR and BATMAN Protocols Considering Link Quality Parameter

Leonard Barolli<sup>†</sup>, Makoto Ikeda<sup>‡</sup>, Giuseppe De Marco<sup>††</sup>,  
Arjan Duresi<sup>†‡</sup>, Fatos Xhafa<sup>‡‡</sup>

<sup>†</sup>Department of Information and Communication Engineering,  
Fukuoka Institute of Technology (FIT)  
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan  
barolli@fit.ac.jp

<sup>‡</sup>Graduate School of Engineering, FIT  
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan  
bd07001@bene.fit.ac.jp

<sup>††</sup>Department of Engineering, Toyota Technological Institute  
2-12-1 Hisakata, Tenpaku-ku, Nagoya 468-8511, Japan  
demarco@toyota-ti.ac.jp

<sup>†‡</sup>Department of Computer and Information Science  
Indiana University Purdue University at Indianapolis (IUPUI)  
723 W. Michigan Street SL 280, Indianapolis, IN 46202, USA  
duresi@cs.iupui.edu

<sup>‡‡</sup>Department of Languages and Informatics Systems  
Polytechnic University of Catalonia  
Jordi Girona 1-3, 08034 Barcelona, Spain  
fatos@lsi.upc.edu

### Abstract

*In this paper, we present the implementation and analysis of our testbed considering the Link Quality Window Size (LQWS) parameter of Optimized Link State Routing (OLSR) and Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.) protocols. We investigate the effect of mobility in the throughput of a Mobile Ad-hoc Network (MANET). The mobile nodes move toward the destination at a regular speed. When the mobile nodes arrive at the corner, they stop for about three seconds. In our experiments, we consider two cases: only one node is moving (mobile node) and two nodes (intermediate nodes) are moving at the same time. We assess the performance of our testbed in terms of throughput, round trip time, jitter and packet loss. From our experiments, we found that throughput of TCP was improved by reducing LQWS.*

### 1. Introduction

In recent years, Mobile Ad-hoc Networks (MANET) are continuing to attract the attention for their potential use in several fields. Most of the work has been done in simulation, as in general, a simulator can give a quick and inexpensive understanding of protocols and algorithms. However, experimentation in the real world are very important to verify the simulation result and to revise the models implemented in the simulator. A typical example of this approach has revealed many aspects of IEEE 802.11, like the gray-zones effect [1], which usually are not taken into account in standard simulators, as the well-known *ns-2* simulator. So far we can count a lot of simulation results on the performance of MANET, e.g. in terms of end-to-end throughput, delay and packet loss. However, in order to assess the simulation results, real-world experiments are needed and a lot of testbeds have been built to date [2]. The baseline criteria usually used in real-world experiments is guaranteeing

the repeatability of tests, i.e. if the system does not change along the experiments. How to define a change in the system is not a trivial problem in MANET, especially if the nodes are mobile.

In this paper, we concentrate on the performance analysis of a small testbed of five computers acting as nodes of a MANET. We use Optimized Link State Routing (OLSR) and Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.), which is a pro-active routing protocol, and it has been gaining great attention within the scientific community. Furthermore, the *olsrd* [3] and *batmand* [4] software we have used in our experiments are the most updated software we have encountered.

In our previous work, we proved that while some of the OLSR's problem can be solved, for instance the routing loop, this protocol still have the self-interference problem. Moreover, there is an intricate inter-dependence between MAC layer and routing layer, which can lead the experimenter to misunderstand the results of the experiments. For example, the horizon is not caused only by IEEE 802.11 Distributed Coordination Function (DCF), but also by the routing protocol.

In this work, we deal with experimental evaluation of these protocols. The results showed that throughput of TCP were improved by reducing Link Quality Window Size (LQWS). Moreover, in the experiments we experienced packet loss because of experimental environment and traffic interference.

In our previous work, we carried out the experiments considering stationary nodes of ad-hoc network. In this work, we consider the node mobility and carry out experiments for OLSR and B.A.T.M.A.N. protocols.

The structure of the paper is as follows. In Section 2, we present the related work. In Section 3, we give a short description of OLSR and B.A.T.M.A.N.. In Section 4, we present the testbed system description and explanation of implemented testbed interface. In Section 5, we present experimental evaluation. Finally, conclusions are given in Section 6.

## 2. Related Work

In [5], the authors analyze the performance of an outdoor ad-hoc network, but their study is limited to reactive protocols such as Ad hoc On Demand Distance Vector (AODV), Dynamic Source Routing (DSR). The authors of [6], performs outdoor experiments of non standard pro-active protocols. Other ad-hoc experiments are limited to identify MAC problems, by providing insights on the one-hop MAC dynamics as shown in [7].

The closest work to ours is that in [8]. However, the authors did not care about the routing protocol. In [9], the disadvantage of using hysteresis routing metric is presented

through simulation and indoor measurements. Our experiments are concerned with the interaction of transport protocols and routing protocol, for instance OLSR. Furthermore, we compare the performance of the testbed for three scenarios: static nodes, one mobile node and two mobile nodes.

In [10], the authors presents an experimental comparison of OLSR using the standard hysteresis routing metric and the ETX metric in a 7 by 7 grid of closely spaced Wi-Fi nodes to obtain more realistic results. The throughput results are similar to our previous work and are effected by hop distance [11, 12].

## 3. Routing Protocols

### 3.1. OLSR Overview

The link state routing protocol that is most popular today in the open source world is OLSR from *olsr.org*. OLSR with Link Quality (LQ) extension and fisheye-algorithm works quite well. The OLSR protocol is a pro-active routing protocol, which builds up a route for data transmission by maintaining a routing table inside every node of the network. The routing table is computed upon the knowledge of topology information, which is exchanged by means of Topology Control (TC) packets. The TC packets in turn are built after every node has filled its neighbors list. This list contains the identity of neighbor nodes. A node is considered a neighbor if and only if it can be reached via a bi-directional link. OLSR checks the symmetry of neighbors by means of a 4-way handshake based on the so called HELLO messages. This handshake is inherently used to compute the packet loss probability over a certain link. This can sound odd, because packet loss is generally computed at higher layer than routing one. However, an estimate of the packet loss is needed by OLSR in order to assign a weight or a state to every link.

In OLSR, control packets are flooded within the network by electing special nodes, called Multi Point Relays (MPRs), to the role of forwarding nodes. By this way, the amount of control traffic can be reduced. These nodes are chosen in such a way that every node can reach its neighbors 2-hops far away. In our OLSR code, a simple RFC-compliant heuristic is used [13] to compute the MPR nodes. Every node computes the path towards a destination by means of a simple shortest-path algorithm, with hop-count as target metric. In this way, a shortest path can result to be also not good, from the point of view of the packet error rate. Accordingly, recently *olsrd* has been equipped with the LQ extension, which is a shortest-path algorithm with the average of the packet error rate as metric. This metric is commonly called as the Expected Transmission Count (ETX), which is defined as  $ETX(i) = 1/(NI(i) \times LQI(i))$ . Given a sampling window  $W$ ,  $NI(i)$  is the packet arrival rate

seen by a node on the  $i$ -th link during  $W$ . Similarly,  $LQI(i)$  is the estimation of the packet arrival rate seen by the neighbor node which uses the  $i$ -th link. When the link has a low packet error rate, the ETX metric is higher. The LQ extension greatly enhances the packet delivery ratio with respect to the hysteresis-based technique [14].

### 3.2. B.A.T.M.A.N. Overview

In OLSR, there is a serious synchronization problem between the topology messages and the routing information stored inside every node. In other words, a mismatch between what is currently stored in the routing tables and the actual topology of the network may arise. This is due to the propagation time of the topology messages. Routing loops are the main effect of such problem. To solve this problem, B.A.T.M.A.N. has been introduced. In B.A.T.M.A.N., there is no topology message dissemination. Every node executes the following operations:

1. Sending of periodic advertisement messages, called OriGinator Message (OGM). The size of these messages is just 52 bytes, containing: the IP address of the originator, the IP address of the forwarding node, a TTL value and an increasing Sequence Number (SQ).
2. Checking of the best one-hop neighbor for every (known) destination in the network by means of a ranking procedure.
3. Re-broadcasting of OGMs received via best one-hop neighbor.

The timer in B.A.T.M.A.N. is used for sending OGMs. The bi-directionality of links is checked using the SQ of OGM. If the SQ of an OGM received from a particular node falls within a certain range, the corresponding link is considered bi-directional. For example, suppose that in a time interval  $T$ , the node A sends  $Tr$  messages, where  $r$  is the rate of OGM messages. The neighbors of A will re-broadcast the OGMs of A and also other node's OGMs. When A receives some OGMs from a neighbor node, say B, it checks if last received OGM from B has a SQ less or equal to  $Tr$ . If it does, then B is considered bi-directional, otherwise it is considered unidirectional. Bi-directional links are used for the ranking procedure. The quantity  $Tr$  is called bi-directional sequence number range. The ranking procedure is the same as the link quality extension of OLSR. In few words, every node ranks its neighboring nodes by means of a simple counting of total received OGMs from them. The ranking procedure is performed on OriGinator (OG) basis, i.e. for every originator. Initially, for every OG, every node stores a variable called Neighbor Ranking Sequence Frame (NBRF), which is upper bounded by a particular value called ranking sequence number range. We

suppose that there is a rank table in every node which stores all the information contained in the OGMs. Whenever a new OGM is being received via a bi-directional link, the receiving node executes the following steps.

1. If the sequence number of the OGM is less than the corresponding NBRF, then drop the packet.
2. Otherwise, update the NBRF=SQ (OGM) in the rank table.
3. If SQ (OGM) is received for the first time, store OGM in a new row of the rank table.
4. Otherwise, increment by one the OGM count or make ranking for this OGM.

Finally, the ranking procedures select the best one-hop neighbor as that neighbor which has the highest rank in the ranking table. Let us note that the same OGM packet is used for: link sensing, neighbor discovery, bi-directional link validation and flooding mechanism. While this feature eliminates routing loops because no global topology information are flooded, the self-interference due to data traffic can cause oscillations in the throughput as we will see in our experiments. Other details on B.A.T.M.A.N. can be found in [15].

### 3.3. Re-broadcasting

Every node re-broadcasts received OGMs only once, and only those OGMs, which have been received via the best-ranked neighbor. This is a kind of selective flooding, which practically reduces the overhead of the flooding. Another analogy can be found in gossip protocols [15]. In gossip protocol, every node decides to re-broadcast received data with some probability,  $p$ . This is equivalent with eliminating some links in the network and then supposing that every node re-broadcast with probability 1. In gossip protocol there is a threshold for  $p$  and the density of nodes after which the success ratio<sup>1</sup> is almost surely 1. In B.A.T.M.A.N., the probability  $p$  is changed according to ranking procedure. It is the probability that an OGM is reached via the best rank neighbor. The expression of this probability is left for further analysis. Let us note that the selective flooding eliminates possible misbehavior of the ranking procedure. In fact, cumulative count of the OGM could be greater than the total number of OGM received via the current best neighbor.

<sup>1</sup>That is the probability that broadcast messages are received by all nodes in the network.

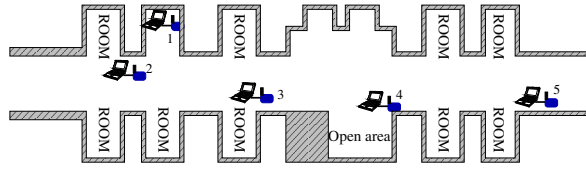


Figure 1. Static model.

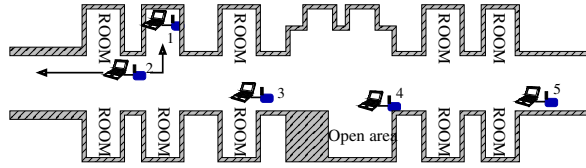


Figure 2. MV1 model.

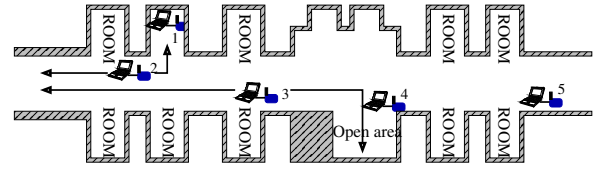


Figure 3. MV2 model.

## 4. Testbed Implementation

### 4.1. Testbed Description

Our testbed is composed of four laptops and one gateway (GW) machine as shown in Figs. 1, 2 and 3. In Fig. 1, all nodes are in a stationary state. We call this model Static model. On the other hand, in Fig. 2, one node is moving. We call this model MV1. In Fig. 3, two nodes are moving at the same time. We call this model MV2.

The operating system mounted on these machines is Fedora Core 4 Linux with kernel 2.6.x, suitably modified in order to support the wireless cards. The wireless network cards are from Linksys, and are usb-based cards with and external antenna of 2dBi gain, transmitted power of 16+/-1dBm and receive sensitivity of -80dBm as shown in Fig. 4. We verified that the external antenna improves the quality of the first hop link, which is the link connecting the ad-hoc network with the GW. The driver can be downloaded from the web site in reference [16].

The GW machine serves as HTTP, FTP, DNS and Internet router for the nodes in the MANET. This feature are provided by the `iptables` mechanism, readily available under Linux machines. By this way, the GW can be accessed ubiquitously from anywhere. Moreover, the GW hosts also all the routines used to coordinate the measurement campaign, as well as graphical tools to check network connectivity.

In our testbed, we have two systematic background or interference traffic we could not eliminate: the control traffic and the other wireless APs interspersed within the campus. The control traffic is due to the `ssh` program, which is used to remotely start and control the measurement software on the GW machine. The other traffic is a kind of interference, which is typical in an academic scenario.



Figure 4. Hardware snapshot.

Table 1. Experimental parameters.

Function	Value
Number of Nodes	5
MAC	IEEE 802.11
Packet Rate	122 pps
Number of Trials	50
Packet Size	512 Kbytes
Duration	10000 ms
LQ Window Size	10 and 100
Protocol	OLSR and B.A.T.M.A.N.

### 4.2. Testbed Interface

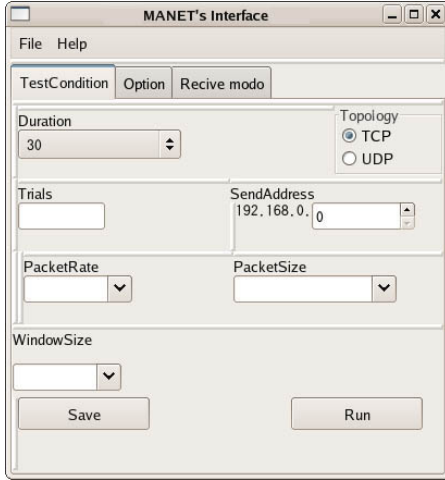
Until now, all the parameters settings and editing were done by using command lines of bash shell (terminal), which resulted in many misprints and the experiments were repeated many times. In order to make the experiments easier, we implemented a testbed interface. For the Graphical User Interface (GUI) we used wxWidgets tool and each operation is implemented by Perl language. wxWidgets is a cross-platform GUI and tools library for GTK, MS Windows and Mac OS.

We implemented many parameters in the interface: transmission duration, number of trials, source address, destination address, packet rate, packet size, LQWS, and topology setting function. We can save the data for these parameters in a text file and can manage in a better way the experimental conditions. Moreover we implemented collection function of experimental data in order to easy the experimenter's work. The testbed interface is shown in Fig. 5.

**Table 2. Each Treatments and groups of median results ( $\hat{T}$ ,  $\widehat{RTT}$ ,  $\hat{J}$ ,  $\widehat{P_L}$ )**

T	Factor			1 $\rightarrow$ 2				1 $\rightarrow$ 3			
	FLOW	MODEL	PROTOCOL	$\hat{T}$	$\widehat{RTT}$	$\hat{J}$	$\widehat{P_L}$	$\hat{T}$	$\widehat{RTT}$	$\hat{J}$	$\widehat{P_L}$
A	UDP	MV1	OLSR: LQWS10	498.572	0.021	0.000774	0.004	499.712	0.022	0.000762	0
B	UDP	MV1	OLSR: LQWS100	498.892	0.019	0.000617	0.002	499.712	0.022	0.000592	0
C	UDP	MV1	B.A.T.M.A.N.	497.664	0.0293	0.0010	0.1211	499.712	0.0207	0.0006	0
D	TCP	MV1	OLSR: LQWS10	498.217	0.0681	0.001	0	498.217	0.021	0.001	0
E	TCP	MV1	OLSR: LQWS100	498.217	0.115	0.001	0	497.721	0.398	0.002	0
F	TCP	MV1	B.A.T.M.A.N.	499.712	0.1154	0.0012	0	499.712	0.3940	0.0018	0
G	UDP	MV2	OLSR: LQWS10	497.708	0.023	0.001	0.005	477.422	0.026	0.001	0.055
H	UDP	MV2	OLSR: LQWS100	496.273	0.018	0.001	0.008	475.611	0.031	0.001	0.059
I	UDP	MV2	B.A.T.M.A.N.	496.640	0.0205	0.0007	0.008	476.397	0.0255	0.0008	0.0575
J	TCP	MV2	OLSR: LQWS10	498.217	0.089	0.001	0	498.217	0.087	0.001	0
K	TCP	MV2	OLSR: LQWS100	498.217	0.095	0.001	0	100.261	0.162	0.002	0
L	TCP	MV2	B.A.T.M.A.N.	150.907	0.0839	0.0011	0	497.721	0.1526	0.0013	0

T	1 $\rightarrow$ 4				1 $\rightarrow$ 5			
	$\hat{T}$	$\widehat{RTT}$	$\hat{J}$	$\widehat{P_L}$	$\hat{T}$	$\widehat{RTT}$	$\hat{J}$	$\widehat{P_L}$
A	499.712	0.031	0.000770	0	499.712	0.031	0.000708	0
B	496.478	0.029	0.000658	0.008	484.314	0.059	0.000985	0.037
C	499.712	0.0168	0.0006	0	499.712	0.0261	0.0005	0
D	497.721	0.210	0.002	0	497.721	0.087	0.001	0
E	227.579	0.188	0.003	0	102.770	0.179	0.003	0
F	499.712	0.0644	0.0006	0	499.712	0.0259	0.0006	0
G	452.663	0.101	0.001	0.109	444.777	0.061	0.001	0.122
H	410.711	0.489	0.003	0.118	451.014	0.117	0.001	0.096
I	473.425	0.0377	0.0011	0.0491	437.259	0.1390	0.0020	0.1016
J	495.266	0.178	0.002	0	399.355	0.163	0.003	0
K	198.765	0.146	0.002	0	77.912	0.181	0.003	0
L	385.432	0.1708	0.0027	0	445.100	0.1814	0.0021	0

**Figure 5. GUI interface.**

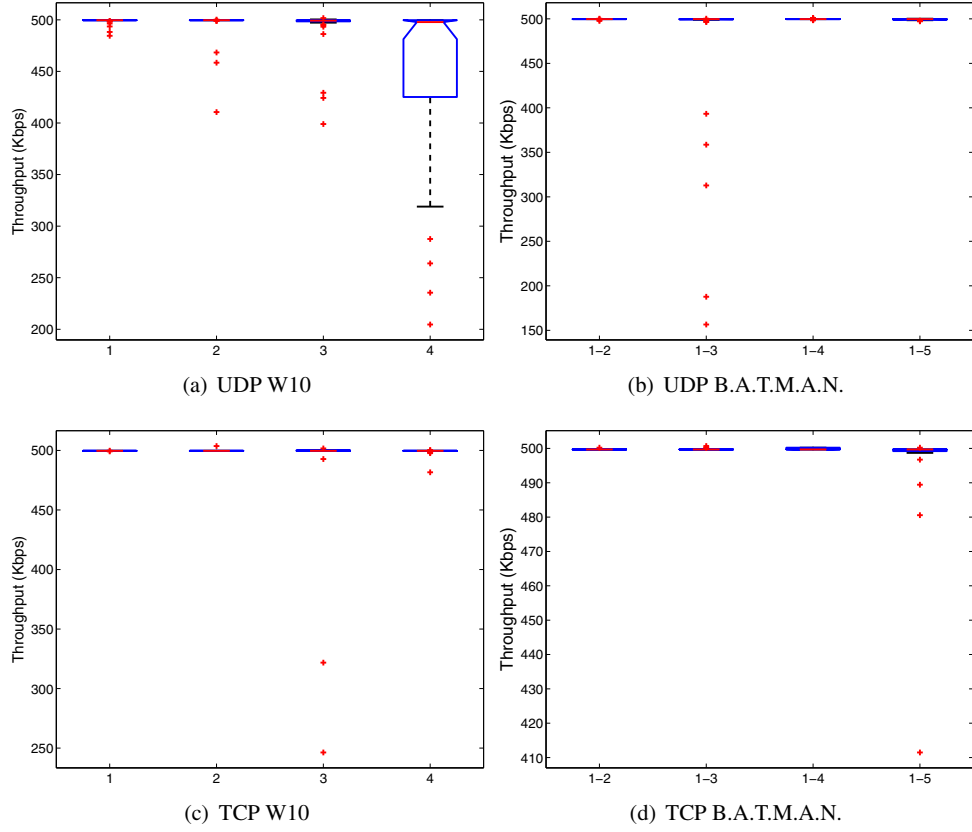
## 5. Experimental Results

### 5.1. Experimental Settings

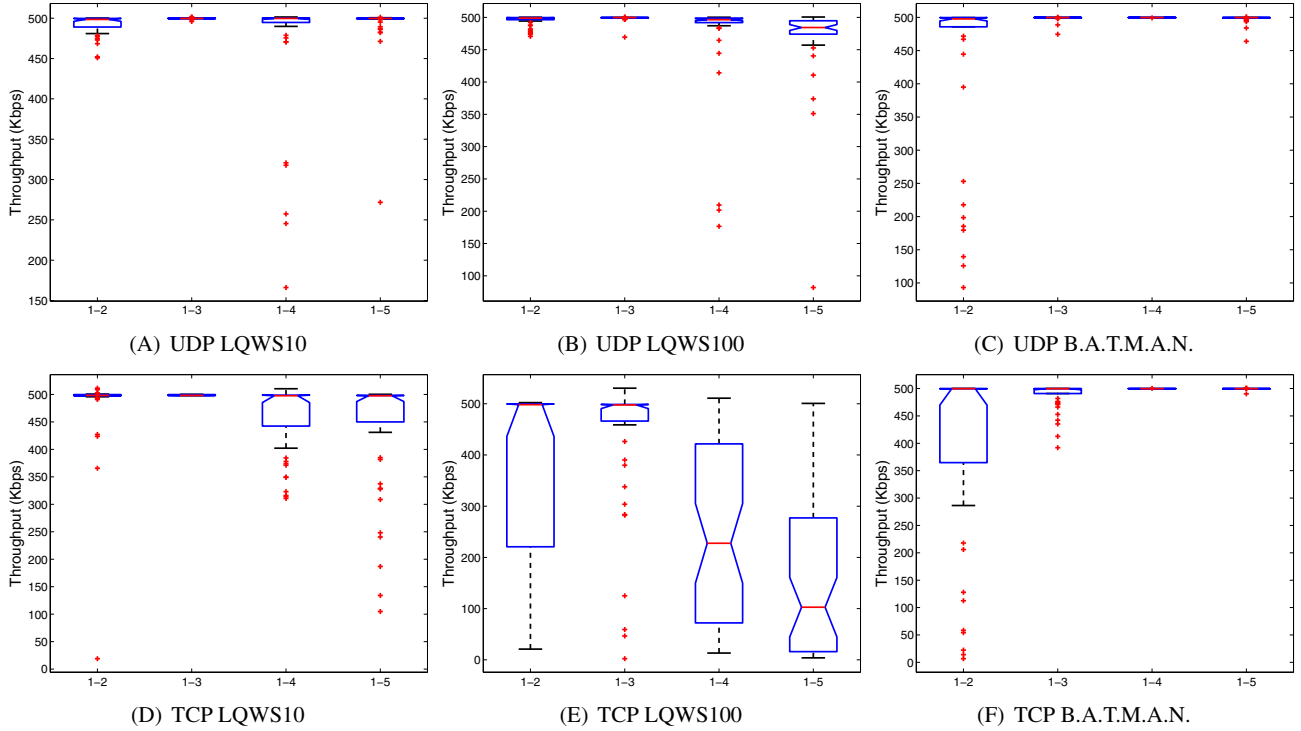
We consider multimedia traffic for experimental parameters as shown in Table 1. We study the impact of best-effort traffic on video streaming applications of the Mesh Topology (MT). In the MT scheme, the MAC filtering routines

are not enabled. We collected data for four metrics: the throughput, Round-Trip Time (RTT), jitter and packet loss. These data are collected by using the Distributed Internet Traffic Generator (D-ITG) [17], which is an open-source Internet traffic generator. D-ITG computes the packet loss as the number of lost packet divided by the effective number of sent packets.

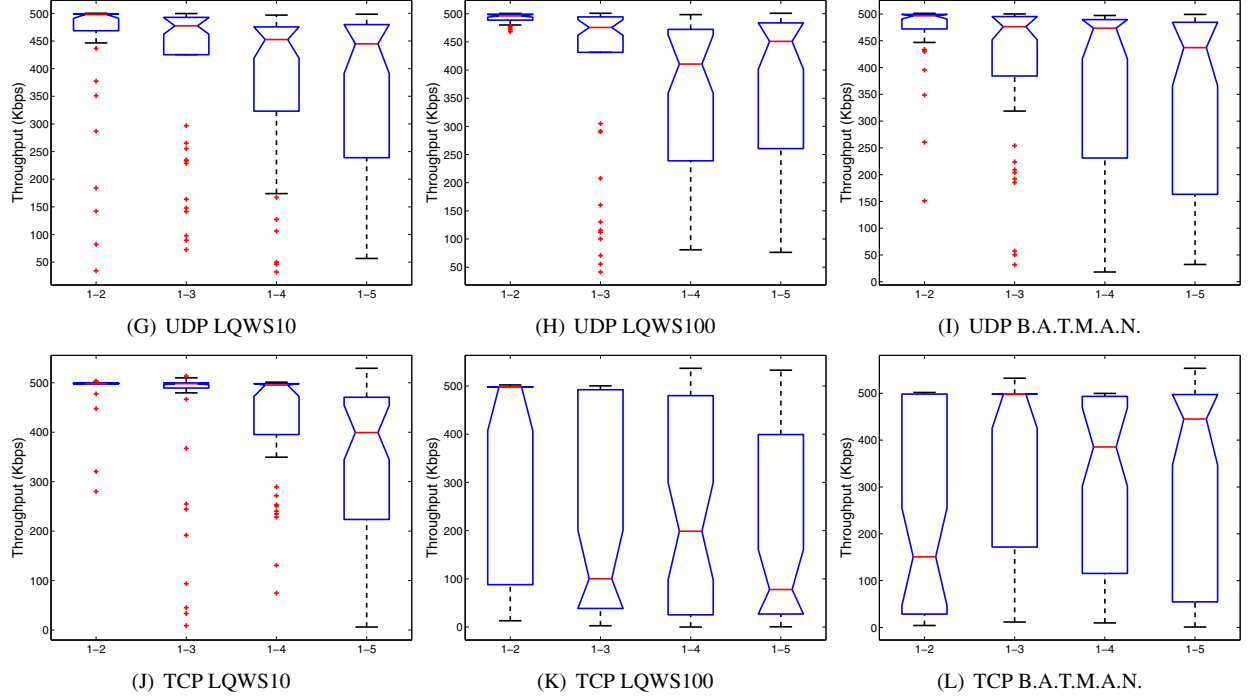
In previous experiments [11, 12, 18], we realized that an external antenna improves radio signal reception. The transmission rate of the data flows is 122 pkt/s = 499.712 Kbps, i.e. the packet size of the payload is 512 Kbytes. All experiments have been performed in indoor environment, within our departmental floor of size roughly 100 m. All laptops are in radio range of each other. We use the same method of analysis as our previous work. Every experiment lasted 10 second and it has been repeated 50 times. The injection of traffic has been carried out by means of D-ITG, which is a traffic generator for IP networks [17]. We measured the throughput for UDP and TCP (TCP NewReno), which is computed at the receiver. We estimate the packet loss to compute the link quality metrics, e.g. LQ. For OLSR,  $wT_{\text{HELLO}} < T_{\text{Exp}}$ , where  $T_{\text{Exp}}$  is the total duration of the experiment, i.e., in our case,  $T_{\text{Exp}} = 500\text{s}$ , and  $T_{\text{HELLO}}$  is the rate of the HELLO messages. However, the testbed was turned on even in the absence of measurement traffic.



**Figure 6. Results for throughput (Static).**



**Figure 7. Results for throughput (MV1).**



**Figure 8. Results for throughput (MV2).**

Therefore, the effective  $T_{Exp}$  was much greater. The experimental parameters are shown in Table 1.

As MAC protocol, we used IEEE 802.11b. The transmission power was set in order to guarantee a coverage radius equal to the maximum allowed geographical distance in the network. Since we were interested mainly in the performance of the routing protocol, we kept unchanged all MAC parameters, such as the carrier sense, the retransmission counter, the contention window and the RTS/CTS threshold. Moreover, the channel central frequency was set to 2.412GHz (channel 1). In regard to the interference, it is worth noting that, during our tests, almost all the IEEE 802.11 spectrum had been used by other access points disseminated within the campus. In general, the interference from other access points is a non-controllable parameter.

## 5.2. Experimental Measurements

Results of our measurements are shown in Table 2. The observed outcome of a treatment is the tetralogy of the metrics medians,  $(\hat{T}, RTT, \hat{J}, \hat{P}_L)$ , except the case of the second group, i.e. with the TCP, where obviously the packet loss is always 0. From Table 2, by looking at “Factor” item, we can see that there is a significant difference in the treatments (T) of the first group (from A to F) where we use MV1 model. For instance, in treatments D, we have three factors: TCP data flow, MV1 model and LQWS10. All results showed the median values. In Table 2,  $1 \rightarrow 2$  means *source node id*  $\rightarrow$  *destination node id*.

In order to show the range of variability of the data, we also report the box plot of the metrics according to the treatment in action, as shown in Figs. 6, 7 and 8. The box plot shows the medians, the lower and the upper quartile (the 25th and 75th percentile, respectively) and the outliers. The ends of the whiskers can represent the lowest datum which is still within 1.5 inter-quartile range of the lower quartile, and the highest datum which is still within 1.5 inter-quartile range of the upper quartile.

In Fig. 6, we show the experimental results for throughput of Static model (stationary state model). The horizontal axis shows the source node id and destination node id. The vertical axis shows the throughput (Kbps). In this case, we did not find big influence, because in this experiment, we did not consider node mobility, but in Fig. 6(a) there are some oscillations, because of the effect of the environment.

In Fig. 7, we show the experimental results for throughput of MV1 (one mobile node: treatments A to F). As shown in Fig. 7, for groups A, B, D and E, if we use TCP data flow, we have more overhead so the throughput is decreased compared with UDP data flow. Moreover, the throughput drop about 50% after the third hop. Let us note that this happens also for group E, which is the MT case. It seems that the topology could not exploit direct links, e.g. from host 1 to 5. In this case, the hop-count can change because of impairments of the radio links and/or MAC problems, such as the gray-zones problem. The high variability for  $1 \rightarrow 5$  connection in group F confirms this fact. Due to the fixed sampling window of the link quality sensing mechanism, nodes

use routes with low quality. Consequently, a dynamic adaptation of the neighbors sensing messages rate could ameliorate the situation. In Fig. 6(a), a direct symmetric link exists between nodes 1 and 5, and OLSR chooses correctly the 2-hops route, i.e. 1-2-5. In this case, the hop-count threshold is 3, and, again, we find high variability for the connection 1-4, which incurs in the longest route, e.g. 1-2-3-4. This behavior is confirmed also by measurements of  $\hat{P}$  and  $\widehat{RTT}$ . Groups C show a packet loss is about 10%.

In Fig. 8, we show the experimental results for throughput of MV2 (two mobile nodes: treatments G to L). In treatments G, H and I, the results of UDP are almost the same. In other words, when we use UDP data flow with different protocols or different LQWS, the throughput results are not affected. Also, the experimental results for models MV1 and MV2 have the same tendency. On the other hand, when we use TCP data flow, the experimental results are affected by LQWS. If we use small LQWS, then we have high throughput, so the  $\widehat{RTT}$  is decreased compared with the case when LQWS is 100. From experimental results, we found that for OLSR if we use TCP data flow, we got better results when the LQWS value was 10.

## 6. Conclusions

In this paper, we carried out experimental results for a small multihop wireless mobile ad-hoc networks with five nodes. We used OLSR and B.A.T.M.A.N. protocols for real experimental evaluation. We considered four parameters for performance evaluation: throughput, RTT, jitter and packet loss. From experimental results, we found that for OLSR if we use TCP data flow, we got better results when the LQWS value was 10.

These experiments were performed using a single flow through the network. In the future, we would like to consider the effect of multiple flows on the routing and increase the number of nodes. Moreover, in order to make easy settings of the testbed we will improve the setup interface.

## Acknowledgment

This work is supported by a Grant-in-Aid for scientific research of Japan Society for the Promotion of Science (JSPS). The authors would like to thank JSPS for the financial support.

## References

[1] H. Lundgren, E. Nordström, and C. Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *WOWMOM '02: Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*, 2002, pp. 49–55.

[2] W. Kiess and M. Mauve, "A survey on real-world implementations of mobile ad-hoc networks," *Ad Hoc Networks*, vol. 5, no. 3, pp. 324–339, 2007.

[3] A. Tønnesen, "OLSRd: Implementation code of the OLSR," <http://www.olsr.org/>.

[4] A. Neumann, C. Elektra Aichele, and M. Lindner, "BATMANd: Implementation code of the B.A.T.M.A.N.," <https://www.open-mesh.net/batman/>.

[5] D. A. Maltz, J. Broch, and D. B. Johnson, "Lessons from a full-scale multihop wireless ad hoc network testbed," *IEEE Personal Communications*, vol. 8, no. 1, pp. 8–15, February 2001.

[6] R. S. Gray, D. Kotz, C. Newport, N. Dubrovsky, A. Fiske, J. Liu, C. Masone, S. McGrath, and Y. Yuan, "Outdoor experimental comparison of four ad hoc routing algorithms," in *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2004, pp. 220–229.

[7] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "IEEE 802.11b ad hoc networks: Performance measurements," *Cluster Computing*, vol. 8, no. 2-3, pp. 135–145, 2005.

[8] V. Kawadia and P. R. Kumar, "Experimental investigations into tcp performance over wireless multihop networks," in *Proc. of E-WIND-2005*, 2005, pp. 29–34.

[9] T. Clausen, G. Hansen, L. Christensen, and G. Behrmann, "The optimized link state routing protocol, evaluation through experiments and simulation," in *Proc. of IEEE Symposium on Wireless Personal Mobile Communications*, September 2001.

[10] D. Johnson and G. Hancke, "Comparison of two routing metrics in OLSR on a grid based mesh network," *Ad Hoc Networks*, 2008, available online 4 May 2008.

[11] G. De Marco, M. Ikeda, T. Yang, and L. Barolli, "Experimental performance evaluation of a pro-active ad-hoc routing protocol in out- and indoor scenarios," in *Proc. AINA-2007*, Canada, May 2007, pp. 7–14.

[12] M. Ikeda, G. De Marco, T. Yang, and L. Barolli, "A BAT in the lab: Experimental results of new link state routing protocol," in *Proc. AINA-2008*, Okinawa, May 2008, pp. 295–302.

[13] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," RFC 3626 (Experimental), 2003.

[14] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. MobiCom-2003*, 2003, pp. 134–146.

[15] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, 2006.

[16] "Rt2x00 project," <http://rt2x00.serialmonkey.com/>.

[17] "D-ITG: Distributed Internet Traffic Generator," <http://www.grid.unina.it/software/ITG/>.

[18] M. Ikeda, L. Barolli, G. De Marco, T. Yang, and A. Durresi, "Experimental and simulation evaluation of OLSR protocol for mobile ad-hoc networks," in *Proc. NBS-2008*, Italy, September 2008, pp. 111–121.