Journal of Mobile Multimedia, Vol. 9, No. 1&2 (2013) 027–038
© Rinton Press

EXPERIMENTAL RESULTS OF A MANET TESTBED FOR DIFFERENT SETTINGS OF HELLO PACKETS OF OLSR PROTOCOL

MASAHIRO HIYAMA SHINJI SAKAMOTO ELIS KULLA
Graduate School of Engineering
Fukuoka Institute of Technology
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan
masahiro.hiyama@gmail.com, shinji.t.sakamoto@gmail.com, eliskulla@gmail.com

MAKOTO IKEDA LEONARD BAROLLI
Department of Information and Communication Engineering
Fukuoka Institute of Technology
3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan
makoto.ikd@gmail.com, barolli@fit.ac.jp

In Mobile Ad-hoc Networks (MANETs) the mobile terminals can be used in cooperation with each other, without having to depend on the network infrastructure. Recently, these terminals are low-cost, have high-performance and are mobile. Because the terminals are mobile, the routes change dynamically, so routing algorithms are important for operation of MANETs. In this paper, we investigate the behaviour of OLSR routing protocol for different values of `HELLO` sending interval and validity time. We conduct real experiments in a MANET testbed. We design and implement two experimental scenarios in our academic environment and investigate their performance behaviour for different number of hops.

Keywords: MANET; DITG; OLSR; HELLO; Send Interval; Validity Time; Testbed; Throughput; Indoor and Outdoor Environment

1 Introduction

A Mobile Ad hoc Network (MANET) is a collection of wireless mobile terminals that are able to dynamically form a temporary network without any aid from fixed infrastructure or centralized administration. In recent years, MANET are continuing to attract the attention for their potential use in several fields. Mobility and the absence of any fixed infrastructure make MANET very attractive for mobility and rescue operations and time-critical applications.

Most of the work for MANETs 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 results 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 packetloss. 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.

There is a lot of work done on routing protocols for MANET. In [3], 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) and Dynamic Source Routing (DSR). The authors of [4] perform 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 [5].

In [6], the authors present an experimental comparison of OLSR using the standard hysteresis routing metric and the Expected Transmission Count (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 [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. In our previous work [10, 11, 12, 13], we carried out many experiments with our MANET testbed. We proved that while some of the OLSR's problems can be solved, for instance the routing loop, this protocol still have the self-interference problem. 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.

We carried out the experiments with different routing protocols such as OLSR and BATMAN and found that throughput of TCP were improved by reducing Link Quality Window Size (LQWS), but there were packet loss because of experimental environment and traffic interference. For TCP data flow, we got better results when the LQWS value was 10. Moreover, we found that the node join and leave operations affect more the TCP throughput and RTT than UDP.

In this work, we compare the performance of OLSR in a MANET testbed in an indoor-outdoor environment, considering different values of `HELLO` packet send interval and validity time. We implement two MANET scenarios and evaluate the performance considering throughput.

The structure of the paper is as follows. In Section 2, we introduce OLSR routing protocol. In Section 3, we explain the implementation of our testbed. Experimental results are shown and discussed in Section 4. Finally, conclusions are given in Section 5.

2 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 proactive 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 bidirectional link.

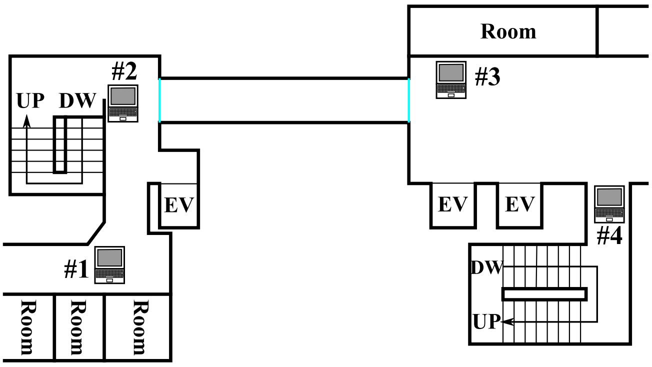
OLSR makes use of `HELLO` messages to find its one hop neighbors and its two hop neighbors through their responses. The sender can then select its Multi Point Relays (MPR) based on the one hop node which offer the best routes to the two hop nodes. By this way, the amount of control traffic can be reduced. Each node has also an MPR selector set which enumerates nodes that have selected it as an MPR node. OLSR uses TC messages along with MPR forwarding to disseminate neighbor information throughout the network. OLSR checks the symmetry of neighbor nodes by means of a 4-way handshake based on `HELLO` messages. This handshake is inherently used to compute the packetloss probability over a certain link. This can sound odd, because packetloss is generally computed at higher layer than routing one. However, an estimate of the packetloss is needed by OLSR in order to assign a weight or a state to every link. Host Network Address (HNA) messages are used by OLSR to disseminate network route advertisements in the same way that TC messages advertise host routes.

In our previous OLSR code, a simple RFC-compliant heuristic was used to compute the MPR nodes [15]. 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 ETX, which is defined as $\mathrm{ETX}(i) = 1/(NI(i) \times LQI(i))$. Given a sampling window W, $\mathrm{NI}(i)$ is the packet arrival rate seen by a node on the i-th link during W. Similarly, $\mathrm{LQI}(i)$ 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 [16].

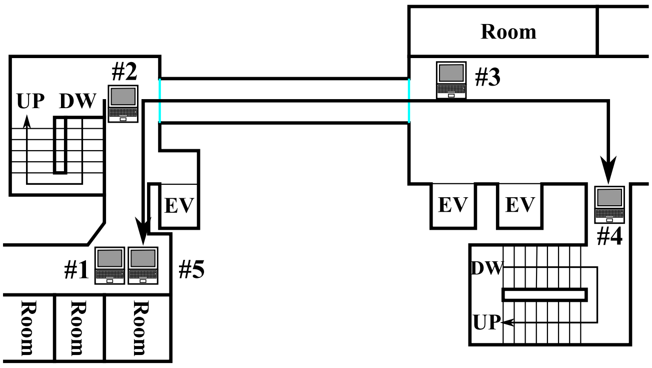
ETX_ff (ETX Funkfeuer/Freifunk) is the current default LQ algorithm for OLSRd. It uses the sequence number of the OLSR packets (which are link specific) to determine the current packet loss rate. ETX_ff includes a hysteresis mechanism to suppress small fluctuations of the LQ and NLQ values. If no packets are received from a certain neighbor at all, a timer begins to lower the calculated LQ value until the next packet is received or the link is dropped. ETX_ff uses only integer arithmetic, so it performs well on embedded hardware having no FPU.

An important parameter of OLSR protocol is the `HELLO` packet send interval. Each OLSR node multicasts `HELLO` packets at a certain interval (default for *olsrd* is 2 seconds), in order to keep the topology information updated in the network. If this parameter is set to lower values, the topology informtaion is more reliable, but in return there will be more overhead from this frequent packets. Therefore, a trade-off between the two values is required.

Another important parameter of OLSR protocol is the `HELLO` packet validity time. This parameter indicates for how long time after reception a node must consider the information contained in the message as valid, unless a more recent update to the information is received. Thus it is better that the value of this parameter is greater than the `HELLO` send packet interval.



(a) Static scenario (STA)



(b) Moving scenario (MOV)

Fig. 1. Experimental scenarios.

Table 1. Number of nodes for each experimental scenario.

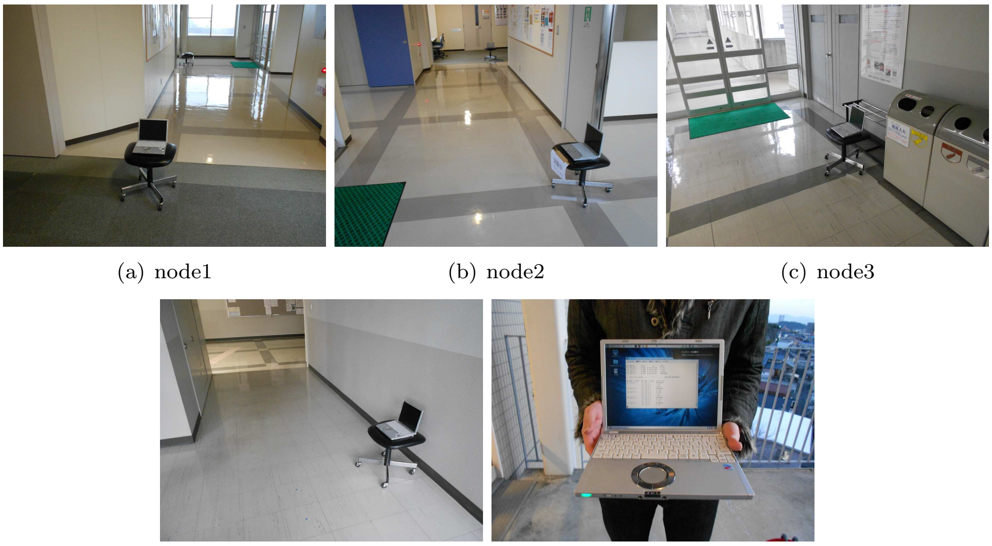
Scenario	Number of Nodes			
	Building D	Bridge	Building C	Moving
STA	2	0	2	0
MOV	2	0	2	1

3 Testbed Description

Our testbed is composed of five laptops machines. The operating system mounted on these machines is Fedora 14 Linux with kernel 2.6.35, suitably modified in order to support the wireless communications. In our testbed, we have two systematic background or interference traffic we could not eliminate: the control traffic and the other wireless Access Points (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 source node. The other traffic is a kind of interference, which is typical in an academic scenario.

3.1 Scenario Description

We constructed two experimental scenarios in our testbed. Node states for each scenario are shown in Table 1. In Fig. 1(a), all nodes are in a static state. Two nodes (node 1 and 2) are in the fifth floor of building D of our campus and two other nodes (node 3 and 4) are inside building C. We call this Static (STA) scenario. In Moving (MOV) scenario, node 5 moves



(a) node1 (b) node2 (c) node3

(d) node4 (e) node5

Fig. 2. Snapshot of nodes in the testbed.

Table 2. Experimental parameters.

Function	Value
Number of Nodes	4 or 5
Logical Link	mesh
MAC	IEEE 802.11b
Traffic Generator	D-ITG-2.8.0-rc1
Flow Type	CBR
Packet Rate	122 pps
Packet Size	512 bytes
Number of Trials	5
Duration	80 sec
Routing Protocol	OLSRd 0.6.5.4
LQWS of OLSR	10
HELLO Packet Interval	0.5, 1.0, 2.0,3.0 and 4.0 s
HELLO Validity Time	10 s, 20 s, 30 s

from position of node 1 to the position of node 4 and back, for 80 seconds, as shown in Fig. 1(b). In Fig. 2, is shown a snapshot of each node in the network.

3.2 Testbed Interface

In our previous work, all the parameters settings and editing were done 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 such as 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 make easier the experimenter's work.

4 Experimental Results

4.1 Experimental Settings

The experimental parameters are shown in Table 2. We study the impact of best-effort traffic for Mesh Topology (MT). In the MT scheme, the MAC filtering routines are not enabled. We collected data for throughput metric. These data are collected using the Distributed Internet Traffic Generator (D-ITG) [17], which is an open-source Internet traffic generator.

The transmission rate of the data flow is 122 pps = 499.712 Kbps, i.e. the packet size of the payload is 512 bytes. All experiments have been performed in the fifth floor of our department buildings. The experimental time for one experiment is about 80 seconds. We conducted our experiments for two different settings of OLSR protocol.

- Case 1: Different `HELLO` send interval values (0.5, 1.0, 2.0, 3.0, 4.0 seconds), while `HELLO` validity time is 20 s, and
- Case 2: Different `HELLO` validity time values (10, 20, 30 seconds), while `HELLO` send interval is 2.0 s

For OLSR, $T_{\text{HELLO}} < T_{\text{Exp}}$, where T_{Exp} is the total duration of the experiment, i.e., in our case, $T_{\text{Exp}} = 80$ seconds, and T_{HELLO} is the rate of the `HELLO` messages. However, the testbed was turned on even in the absence of measurement traffic. Therefore, the effective T_{Exp} was much greater.

As MAC protocol, we used IEEE 802.11b. The transmission power was set in order to guarantee a coverage radius big enough to cover all one-hop physical neighbors of each node 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 2.412 GHz (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 APs disseminated within the campus. In general, the interference from other APs is a non-controllable parameter.

4.2 Results Discussion

Here, we show the measured data by time-domain plots. In this way we can have a better observation of the oscillations occurring during transmission as well as the effects of mobility. Moreover, we show the average data for each experiment in Tables 3 and 4.

4.2.1 Case 1: Different `HELLO` Send Interval Values

In Fig. 3, we show the throughput results for STA scenario. Source node is always node 1. In Fig. 3(a), we show the throughput of 1 → 2 flow. We can see that for every interval of `HELLO` packets, node 1 and 2 are near to each other, so the throughput is on maximum value. In 1 → 3 and 1 → 4 flows (see Fig. 3(b) and Fig. 3(c), respectively), for `HELLO` interval up

Table 3. Throughput results (kbps) for different `HELLO` packet send interval.

Scenario	Hello Interval	Source node→Destination node			
		1→2	1→3	1→4	1→5
STA	0.5 sec	499.71	189.78	200.23	-
STA	1.0 sec	499.71	266.48	208.20	-
STA	2.0 sec	488.09	272.71	285.72	-
STA	3.0 sec	499.71	285.05	350.66	-
STA	4.0 sec	499.71	184.50	243.05	-
MOV	0.5 sec	499.71	408.01	233.87	269.97
MOV	1.0 sec	499.71	393.62	287.39	239.96
MOV	2.0 sec	499.55	280.71	126.50	266.53
MOV	3.0 sec	499.55	220.38	129.93	177.58
MOV	4.0 sec	487.60	155.34	181.42	146.21

Table 4. Throughput results (kbps).

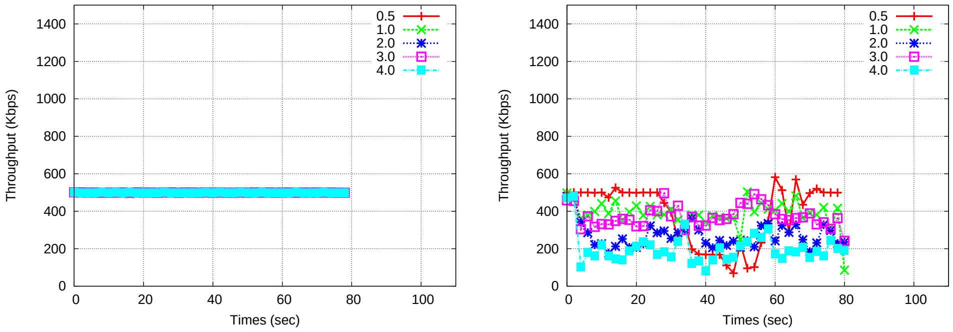
Scenario	Hello Validity Time (sec)	Source node→Destination node			
		1→2	1→3	1→4	1→5
STA	10	499.71	499.71	400.62	-
STA	20	499.71	499.71	397.84	-
STA	30	499.71	499.71	405.20	-
MOV	10	499.71	349.51	268.00	272.15
MOV	20	499.71	259.07	262.25	252.93
MOV	30	499.71	249.32	236.99	181.45

to 3.0 seconds, the throughput increases. When `HELLO` interval becomes 4.0, the throughput decreases noticeably, as we can also see in Table 3.

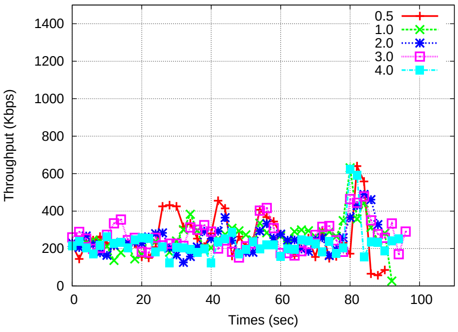
For MOV scenario, we show the throughput results in Fig. 4. For 1 → 2 flow (see Fig. 4(a)), the performance is maximum for all values of `HELLO` interval, similar to STA scenario. In MOV scenario, the topology is dynamic because of node movement. So it is important for nodes to have the information topology as soon as possible. For this reason, we notice in Figs. 4(b) and 4(c), that for low values of `HELLO` interval, the performance is better, compared to cases when `HELLO` interval is 3.0 or 4.0. However, becuase of topology changes, we still have oscillations in throughput.

We found some interesting results in the case when we sent data towards the moving node 5. Because node 5 is moving, from second 15 to second 25, the routes start to disconnect. In the period from 20 to 30 seconds, the graphs of `HELLO` interval 0.5 and 1.0 still have higher throughput, while the other graphs tend to go to zero faster. The low `HELLO` interval helps node 5 to still be connected with node 1 and it looses connectivity later, at about 30 seconds.

After a 40-second movement towards node 4, node 5 starts to go back to the position of node 1. In Fig. 4(d), around 50-55 seconds the communication starts to reconnect because node 5 is coming closer to source node 1. In this case, we notcie that for `HELLO` interval 4.0, the throughput will go back up slower than other values, because the routes are calculated according to `HELLO` packets, which need 4.0 seconds to start spreading around the network.



(a) 1 → 2 (b) 1 → 3



(c) 1 → 4

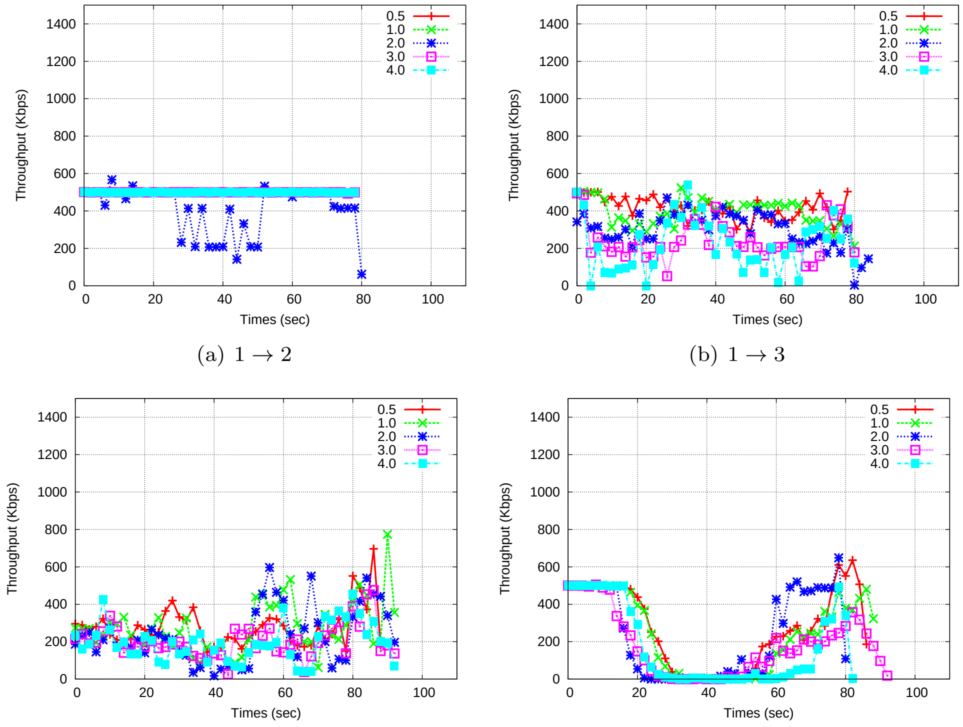
Fig. 3. Throughput results for Static scenario.

4.2.2 Case 2: Different `HELLO` *Validity Time Values*

In Fig. 5, we show the throughput results of STA scenario for three different values of validity time. We can see that the performance for 1 → 2 and 1 → 3 flows is very good, while for 1 → 4 flow the throughput has some oscillations and its average value decreases about 20% for all values of `HELLO` validity time. The average results are shown in Table 4. In the STA scenario, the topology is stable and the `HELLO` packets carry almost the same information during experimental time. Thus, the `HELLO` validity time does not effect the performance, while the hop distance has a strong effect in performance decrease.

The results for MOV scenario are shown in Fig. 6. When node 5 is moving the topology becomes more dynamic and there are more route changes. The performance of the network is decreased compared to the STA scenario. The oscillations and the decrease of throughput values start since the 1 → 3 flow. Obviously the throughput decreases more in the case of 1 → 4 flow (see Table 4).

The effect of `HELLO` validity time is visible because of topology changes. When the `HELLO` is low, the neighbors information carried by `HELLO` packets is updated more frequently, because the old information is discarded every 10 seconds. While, when `HELLO` validity time is 30 seconds, old information affects the calculation of routes causing not correct information of routes and lowest performance regarding throughput. This effect is more visible for 1 → 5 flow, where destination node 5 is moving and the routes are more dynamic. For `HELLO` validity time



(a) 1 → 2

(b) 1 → 3

(c) 1 → 4

(d) 1 → 5

Fig. 4. Throughput results for Moving scenario.

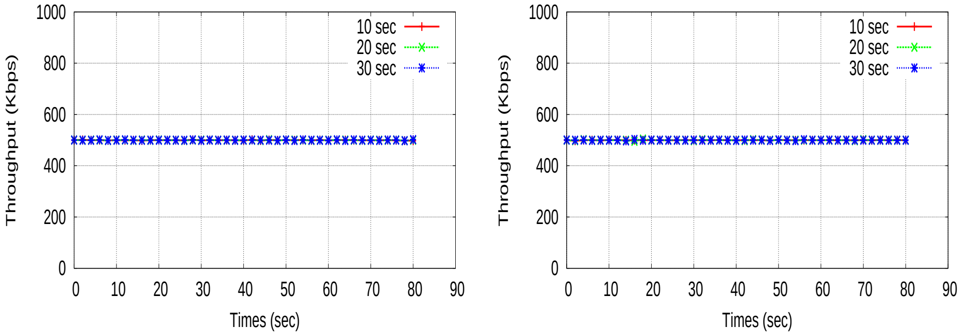
30 seconds, the throughput decreases to 181.45 kbps, which is 60% lower than the theoretical throughput.

5 Conclusions

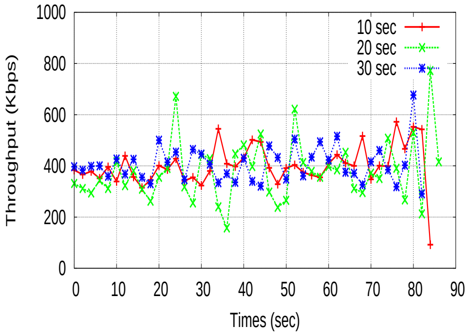
In this paper, we conducted experiments with our MANET testbed for two scenarios. We used OLSR protocol for experimental evaluation. We changed the `HELLO` send interval and validity time from the default values and compared the effect of different `HELLO` packet settings and mobility. We assessed the performance of our testbed in terms of throughput and from our experiments, we found the following results.

5.1 Case 1

- For near-zone communication, the throughput was not affected by HELLO interval in both scenarios.
- For STA scenario, the high value HELLO interval cases have better performance, regarding throughput and the overhead is low.
- For MOV scenario, the topology is dynamic, so for more frequent HELLO packets the performance was better.
- During route recovery, the cases with lower HELLO interval showed a better perfor-



(a) Flow 1 → 2 (b) Flow 1 → 3



(c) Flow 1 → 4

Fig. 5. Throughput results for STA scenario.

mance because nodes keep the communications for a longer time and return faster in touch with the lost communication.

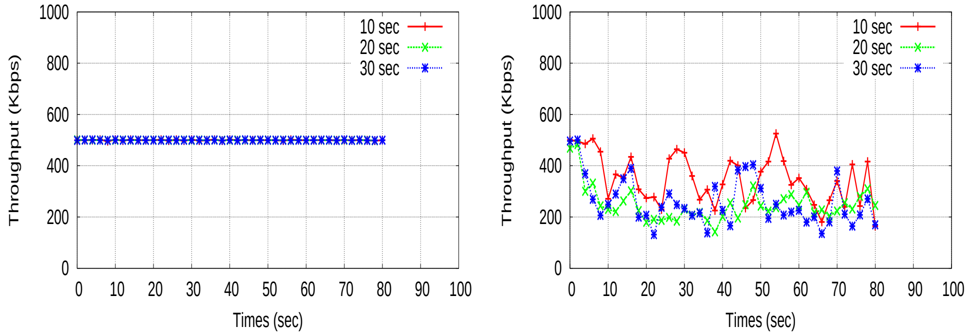
5.2 *Case 2*

- The `HELLO` validity time does not effect the performance for STA scenario.
- The hop distance has a strong effect in performance decrease, in STA scenario.
- For MOV scenario, the throughput decreases for 1→3 flow.
- In MOV scenario, the `HELLO` validity time affects the throughput performance. Thus, the throughput is decreased 60% than theoretical throughput.

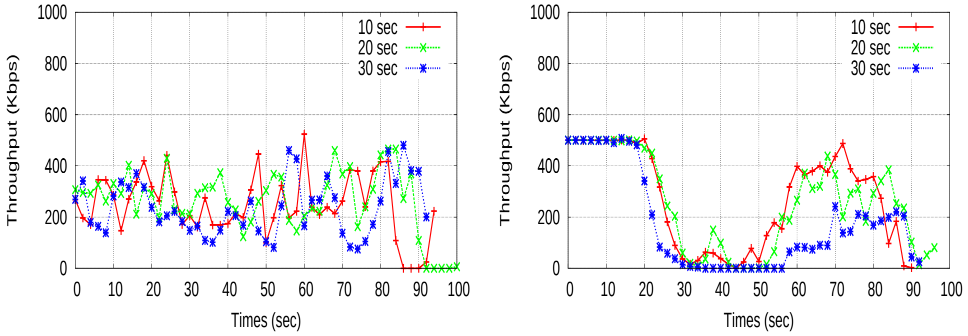
In our future work, we would like to compare the results with other proactive routing protocols and compare experimental and simulation results. We believe that MAC layer has an effect on the performance, so we would like to use IEEE 802.11g/n for future evaluations.

Acknowledgements

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.



(a) Flow 1 → 2 (b) Flow 1 → 3



(c) Flow 1 → 4 (d) Flow 1 → 5

Fig. 6. Throughput results for MOV scenario.

References

1. H. Lundgren, E. Nordstrom, and C. Tschudin, “Coping with Communication Gray Zones in IEEE 802.11b based Ad Hoc Networks”, Proc. of the 5-th ACM International Workshop on Wireless Mobile Multimedia (WOWMOM-2002), pp. 49-55, 2002.
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. 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.
4. 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”, Proc. MSWiM-2004, pp. 220-229, 2004.
5. 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.
6. D. Johnson and G. Hancke, “Comparison of Two Routing Metrics in OLSR on a Grid based Mesh Network”, Ad Hoc Networks, Vol. 7, No. 2, pp. 374-387, March 2009.
7. G. De Marco, M. Ikeda, T. Yang, and L. Barolli, “Experimental Performance Evaluation of a Pro-active Ad-hoc Routing Protocol in Outdoor and Indoor Scenarios”, Proc. of IEEE AINA-2007, pp. 7-14, May 2007.

8. V. Kawadia and P. R. Kumar, "Experimental Investigations into TCP Performance over Wireless Multihop Networks", Proc. of E-WIND-2005, pp. 29-34, 2005.
9. T. Clausen, G. Hansen, L. Christensen, and G. Behrmann, "The Optimized Link State Routing Protocol, Evaluation through Experiments and Simulation", Proc. of IEEE Symposium on Wireless Personal Mobile Communications, Available on line at `http://hipercom.inria.fr/olsr/wpmc01.ps`, September 2001.
10. L. Barolli, M. Ikeda, G. De Marco, A. Durresi, and F. Xhafa, "Performance Analysis of OLSR and BATMAN Protocols Considering Link Quality Parameter", Proc. of IEEE AINA-2009, pp. 307-314, May 2009.
11. E. Kulla, M. Ikeda, L. Barolli and R. Miho, "Impact of Source and Destination Movement on MANET Performance Considering BATMAN and AODV Protocols", Proc. of BWCCA-2010, pp.94-101, 2010.
12. M. Ikeda, L. Barolli, G. De Marco, T. Yang, and A. Durresi, "Experimental and Simulation Evaluation of OLSR Protocol for Mobile Ad-hoc Networks", Proc. of NBiS-2008, pp. 111-121, September 2008.
13. M. Hiyama, M. Ikeda, L. Barolli, G. De Marco, F. Xhafa, and A. Durresi, "Mobility Effects in Mobile Ad Hoc Networks", Proc. of International Workshop on Network Traffic Control, Analysis and Applications (NTCAA-2009), Vol. 2, pp. 679-684, December 2009.
14. A. Tonnesen, "OLSRd: Implementation Code of the OLSR", Available on line at http://www.olsr.org/.
15. T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC 3626 (Experimental), 2003.
16. D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High Throughput Path Metric for Multi-hop Wireless Routing", Proc. of MobiCom-2003, pp. 134-146, 2003.
17. A. Botta, A. Dainotti, and A. Pescape, "Multi-protocol and Multiplatform Traffic Generation and Measurement", Proc. of INFOCOM-2007, Demo Session, May 2007.