

Tools for performance assessment of OLSR protocol

Makoto Ikeda^a, Leonard Barolli^b, Giuseppe De Marco^c, Tao Yang^a, Arjan Durrresi^d and Fatos Xhafa^e

^a*Graduate School of Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan*
E-mails: {bd07001, bd07003}@bene.fit.ac.jp

^b*Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi, Higashi-Ku, Fukuoka 811-0295, Japan*
E-mail: barolli@fit.ac.jp

^c*Department of Engineering, Toyota Technological Institute, 2-12-1 Hisakata, Tenpaku-ku, Nagoya 468-8511, Japan*
E-mail: demarco@toyota-ti.ac.jp

^d*Department of Computer and Information Science, Indiana University Purdue University Indianapolis, 723 W. Michigan Street SL 280, Indianapolis, IN 46202, USA*
E-mail: durrresi@cs.iupui.edu

^e*Department of Languages and Informatics Systems, Technical University of Catalonia, C/Jordi Girona 1-3, 08034 Barcelona, Spain*
E-mail: fatos@lsi.upc.edu

Abstract. In this paper, we evaluate the performance of Optimized Link State Routing (OLSR) protocol by experimental and simulation results. The experiments are carried out by using our implemented testbed and the simulations by using *ns-2* simulator. We also designed and implemented a new interface for the ad-hoc network testbed in order to make more easier the experiments. The comparison between experimental and simulation results shows that for the same parameters set, in the simulation we did not notice any packet loss. On the other hand, in the experiments we experienced packet loss because of the environment effects and traffic interference.

Keywords: Ad-hoc network, Optimized Link State Routing protocol, traffic interference

1. Introduction

Ad-hoc networks are continuing to attract the attention for their potential use in several fields. Most of the work has been done in simulation, because, in general, 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 [2].

In this paper, we concentrate on the performance analysis of a small testbed of seven computers acting as nodes of a wireless ad-hoc network. We use Optimized Link State Routing (OLSR) which is a pro-active protocol, and it has been gaining great attention within the scientific community. Furthermore, the *olsrd* [3] software we have used in our experiments is the most updated software we have knowledge of.

In our previous work, we proved that while some of the OLSR's problems can be solved, for instance the routing loop, the self-interference problem still remains. 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 effect is not caused only by IEEE 802.11 MAC, but also by the routing protocol.

In this work, we deal with experimental and simulation evaluation of OLSR protocol. The simulation results showed that there is not any packet loss for our settings. However, in the experiments we experienced packet loss because of experimental environment and traffic interference.

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. In Section 4, we present the testbed and simulation system description, by describing the software tools we used to assess the performance of OLSR. In Section 5, we introduce the hidden node problem, and briefly explains the mechanisms which affect the throughput in ad-hoc networks. In Section 6, we discuss the experimental and simulation results. In Section 7, we present experimental and simulation evaluation. Finally, conclusions are given in Section 8.

2. Related works

In [4], the authors analyze the performance of an outdoor ad-hoc network, but their study is limited to reactive protocols such as AODV and DSR. The authors in [5] 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 [6].

The closest work to ours is that in [7]. However, that work was concerned with the analysis of TCP parameters in an indoor scenario, only. Moreover, it looks like that the authors did not care about the routing protocol. In [8], the disadvantage of using hysteresis-based selection of routes 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 with simulation results in an indoor scenario.

3. OLSR overview

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 are 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.

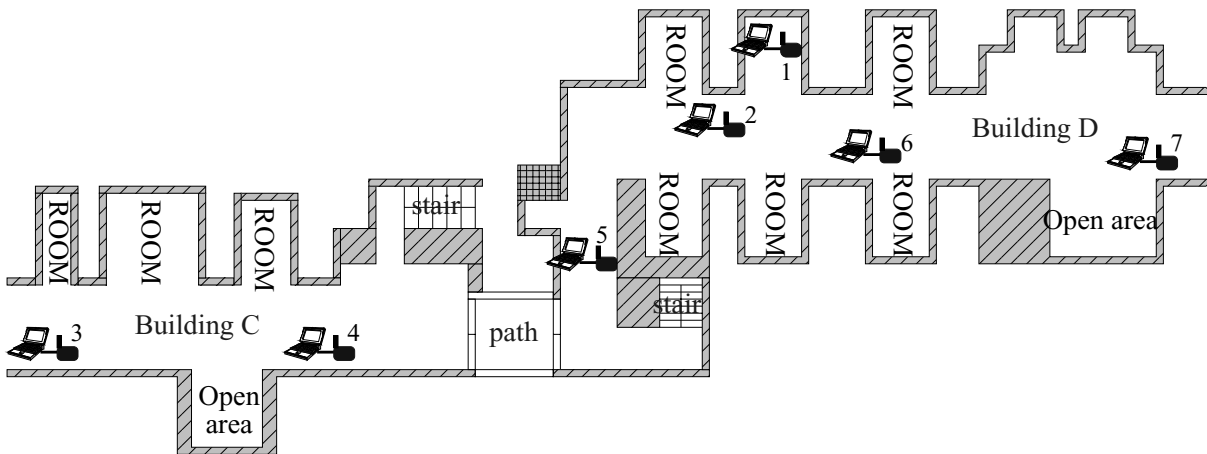


Fig. 1. Experimental model with 7 nodes.

In OLSR, control packets are flooded within the network by promoting 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 2-hops neighbors. In our OLSR code, a simple RFC-compliant heuristic is used [9] 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. Recently OLSRd has been equipped with the Link Quality (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 Rate (ETX), which is defined as $ETX(i) = 1/(NI(i) \times LQI(i))$. Given a sampling window W , $NI(i)$ is the packet loss probability seen by a node on the i -th link during W . Similarly, $LQI(i)$ is the estimate of the packet loss 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 [10].

4. Testbed and simulation system description

4.1. Testbed description

Our testbed is composed by six laptops and one gateway machine as shown in Fig. 1. The operating system mounted on these machines is Linux with kernel 2.6, suitably modified in order to support the wireless cards. These IEEE 802.11 g cards are all from the same vendor, except for two nodes, the gateway (GW) and the first hop from the gateway (clio), which use a LynkSys card. However, the chipset is the same and is based on the Broadcom series. The driver can be downloaded from the Web sites in references [11,12]. This wireless USB card is equipped with an external antenna.

The GW machine serves as DNS system and Internet router for the nodes in the ad-hoc network. By this way, the network 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

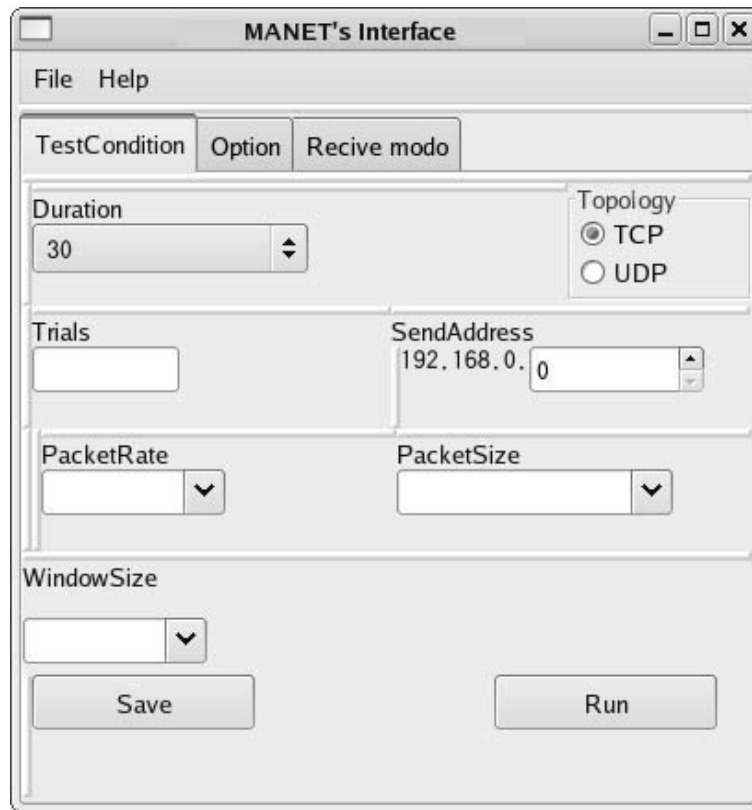


Fig. 2. GUI interface.

the `ssh` program, which is used to remotely start and control the measurement programs on the GW machine. The other traffic is a kind of interference which is typical in an academic environment.

We used the UDP protocol and studied the Mesh Topology (MT). In the MT scheme, the MAC filtering routines are not enabled. This means that every node can connect to any other node. We collected data for four metrics: the throughput, Round-Trip Time (RTT), jitter and packet loss. These data are collected by using D-ITG [13], which is an open-source traffic generator. D-ITG computes the packet loss as the number of lost packet divided by the effective number of sent packets.

Every experiment is repeated 50 times. In the experiments, we suppose only one single test-flow. The source is always located at the GW and the destination is changed along the experiments. By this way, we fairly focus the study on the performance of the routing protocol and not on other phenomena, such as congestion or throughput saturation.

4.2. Testbed interface

Until now, all the parameters settings were done by using command lines of Linux, which resulted in many misprints and the experiments were repeated many times. In order to make the experiments easier, we implemented a software interface. The interface is shown in Fig. 2. For the Graphical User Interface (GUI) we used `wxWidgets` tool and for GUI Builder the `wxGlade`. The definition for each button was done by `wxGlade` and each operation were implemented by Perl language.

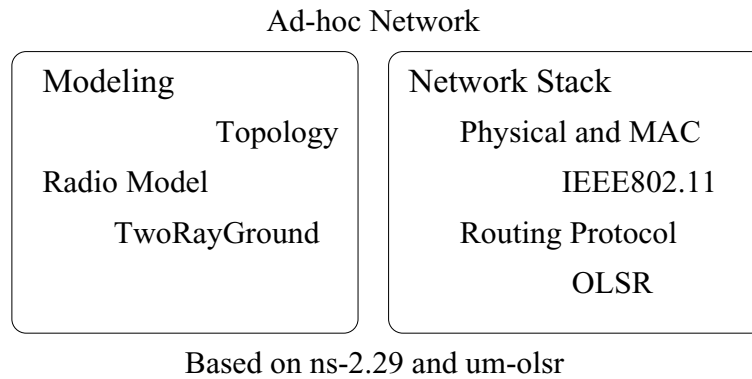
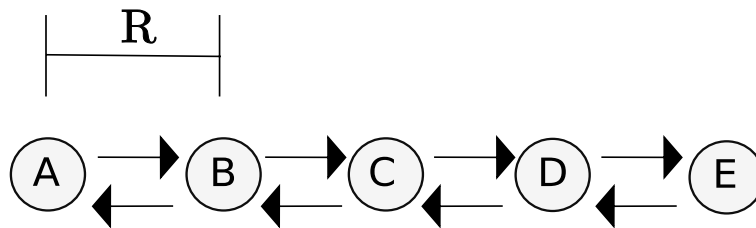


Fig. 3. Simulation model.

Fig. 4. Horizon effect of IEEE 802.11 when radio range is $2R$ m.

We defined 6 parameters in the interface: transmission duration, trial number, source address, packet rate, packet size and window size, W . We can save the data for these parameters in a text file and can manage in a better way the experimental conditions.

4.3. Simulations

The simulations have been performed in *ns-2*. The radio model used is the TwoRayGround propagation model, i.e. the radio channel is a deterministic model. For example, the power loss of the electromagnetic signal between two nodes is a deterministic function of their separation distance. It is clear that this model does not reflect the reality, and one of the aim of this work is to show to what extent this disparity impacts on the performance evaluation. The simulation models of other network protocols, e.g. MAC and link layer, are those implemented in the simulator and used as are.

5. Hidden node problem

The 802.11 MAC is a CSMA/CA access protocol based on carrier sensing with collision avoidance. The contention of the radio medium is controlled by means of a back-off timer which in turn depends on the value of a contention window. In multi-hop network, the DCF 802.11 is used. It is well known that the throughput per node scale as $O(\frac{1}{n})$. This can be easily explained by an example. Consider a linear network with step R , as shown in Fig. 4. We suppose that based on the transmission range every node can be listen up to a distance of $2R$.

Node A can hear Node B and node C. The node B can listen the nodes A, C and D. In general, every node can be interfered by up to 4 nodes. Therefore, the available throughput scales as $1/n$ of the total

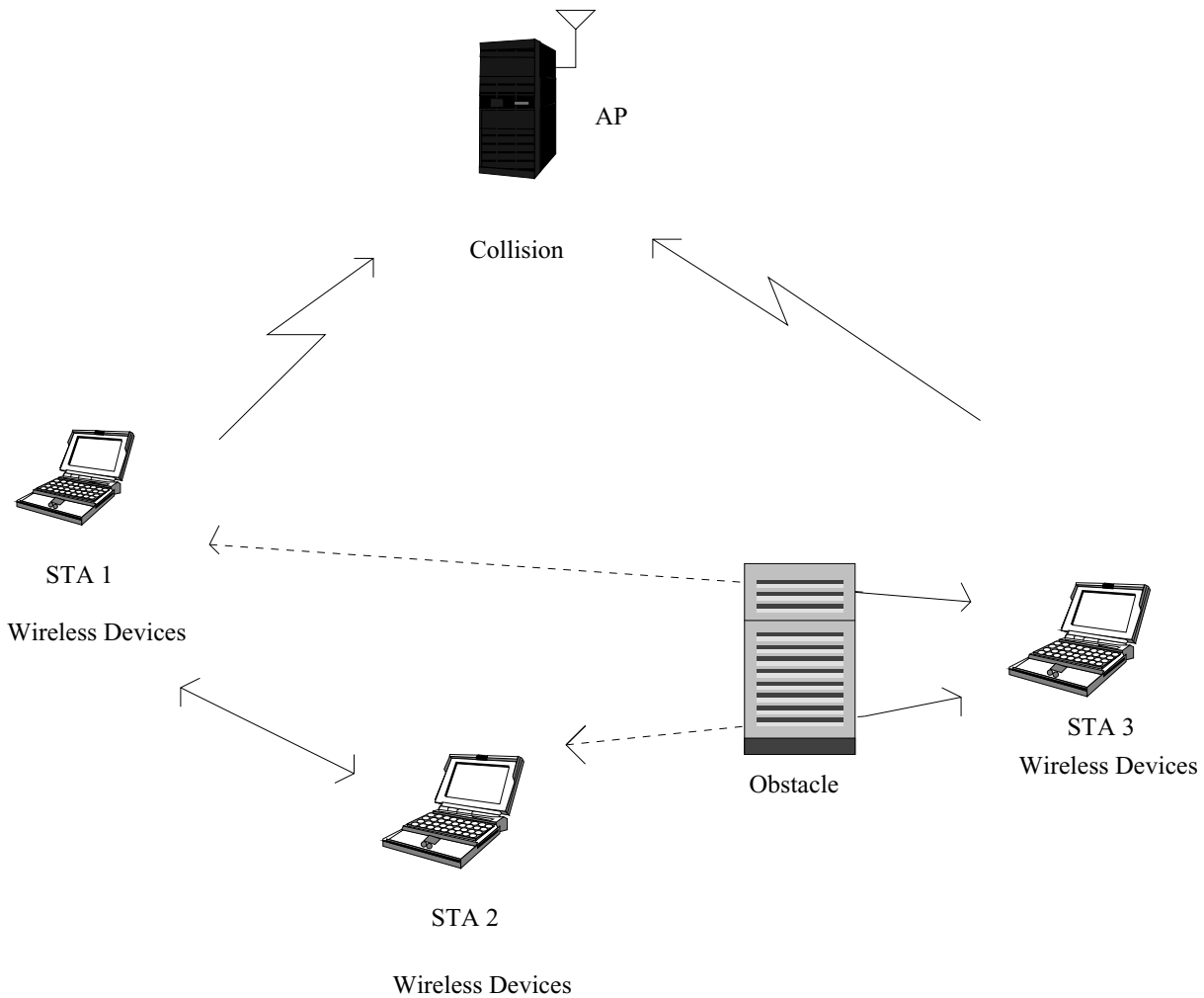


Fig. 5. Hidden node problem.

available capacity. For this configuration, the throughput at node i is $1/(i + 2)$, e.g. for node A it is $1/3$ and at node C it is $1/5$. The IEEE 802.11 has another well known problem which limits the sustainable throughput. It is caused by the hidden node problem which causes packet collisions. In general, this problem is more deleterious than the packet loss induced by carrier sense.

An example of the hidden node problem is showing in Fig. 5. As we can see, since between STA1 and STA3, and STA2 and STA3 there is an obstacle, STA1 and STA3 can't hear each other because of high attenuation, but they can both communicate with the same Access Point (AP). Because of this situation, STA2 may begin sending a frame without noticing that STA3 is currently transmitting (or vice versa). This will very likely cause a collision between STA2 and STA3 to occur at the AP. As a result, both STA2 and STA3 would need to retransmit their respective packets, which results in higher overhead and lower throughput.

In order to solve in some extent the hidden node problem we use the RTS (Request to Send)/CTS (Clear to Send) mechanism. By enabling RTS/CTS on a particular node, it will refrain from sending a data frame until the other nodes complete a RTS/CTS handshake, such as an AP. For example, when a

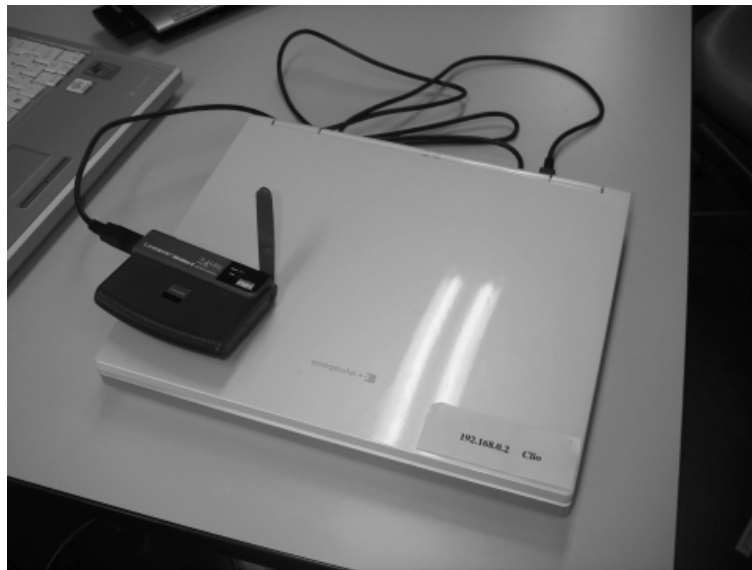


Fig. 6. Hardware settings.

node has data to send, it initiates the process by sending a RTS frame. The AP receives the RTS and responds with a CTS frame. The station must receive a CTS frame before sending the data frame. The CTS also contains a time value that alerts other nodes to hold off from accessing the medium while the node initiating the RTS transmits its data.

The RTS/CTS handshaking provides positive control over the use of the shared medium. The primary reason for implementing RTS/CTS is to minimize collisions among hidden nodes. This occurs when users and APs are spread out throughout the facility and a relatively high number of retransmissions occur on the wireless environment.

By activating RTS/CTS, the collision will not happen. Before transmitting, ST2 would send a RTS and receive a CTS from the AP. The timing value in the CTS (which ST3 also receives) will cause ST3 to hold off long enough for ST2 to transmit the frame. Thus, the use of RTS/CTS reduces collisions and increases the performance of the network if hidden nodes are present.

6. Experimental and simulation settings

Our testbed is composed of 6 laptops and one desktop machine acting as a GW. The GW provided us the possibility of control the testbed from anywhere within the campus. All machines run Linux Fedora with kernel 2.6. The wireless network cards are from Lynksys, and are usb-based cards with an external antenna of 2dbi gain as shown in Fig. 6.

In previous experiments [14], we realized that an external antenna improves radio signal reception. The transmission rate of the data flows is $122 \text{ Kpkt/s} = 499.712 \text{ Kbps}$, i.e. the packet size of the payload is 512 bytes. 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 s 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 [13]. We measured the throughput

Table 1
Experimental and simulation parameters

Number of Nodes:	7	MAC:	IEEE 802.11b
Packet Rate (Kpkt/sec):	122	Number of Trials:	50
Packet Size (Kbytes):	512	LQ Window Size:	100
Protocol:	OLSR	RTS Thresholds:	64, 128, 256, off

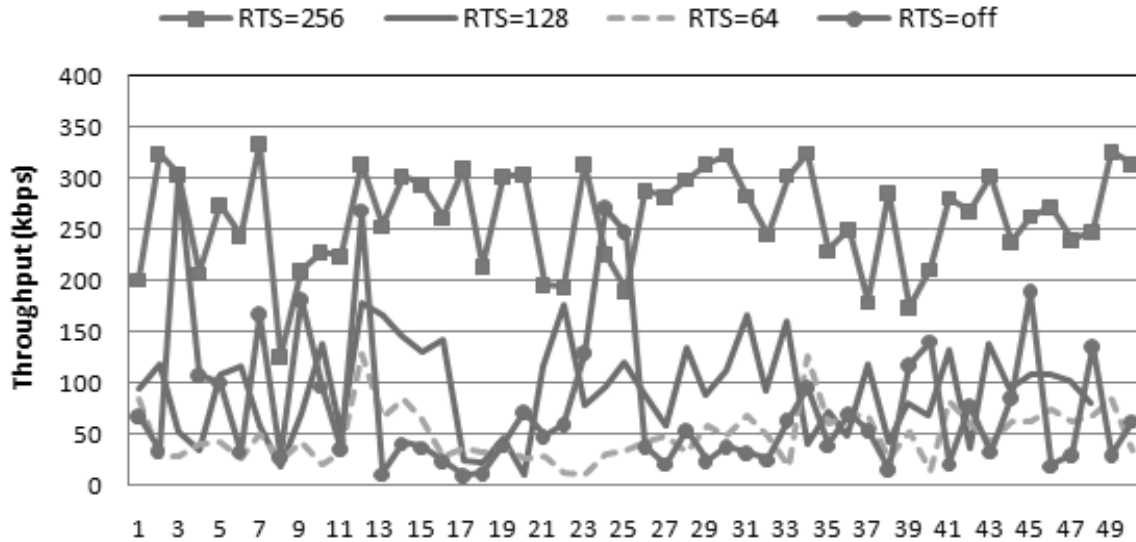


Fig. 7. Throughput.

for UDP, 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_{Exp} is the total duration of the experiment, i.e., in our case, $T_{\text{Exp}} = 500\text{s}$, 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.

For the simulations, we used *ns-2* simulator with the same settings as the testbed. The experimental and simulation parameters are shown in Table 1.

7. Experimental and simulation results

We show the experimental results in Figs 7, 8 and 9. In Fig. 7, we show the experimental results for throughput. We use different RTS/CTS thresholds: 256, 128, and 64. The off status means that the RTS/CTS mechanism is not activated. The better throughput is for the threshold 256. When the RTS/CTS is not activated there are a lot of oscillations. It should be noted that an increase in performance using RTS/CTS is the net result of introducing overhead (i.e., RTS/CTS frames) and reducing overhead (i.e., fewer retransmissions). If we have small thresholds, then we have more overhead so the throughput is decreased.

In Fig. 8 are shown the experimental results for RTT. As can be seen from the figure, the RTT when the RTS/CTS mechanism is off is smaller than the other cases. However, as we show in the Fig. 8 the

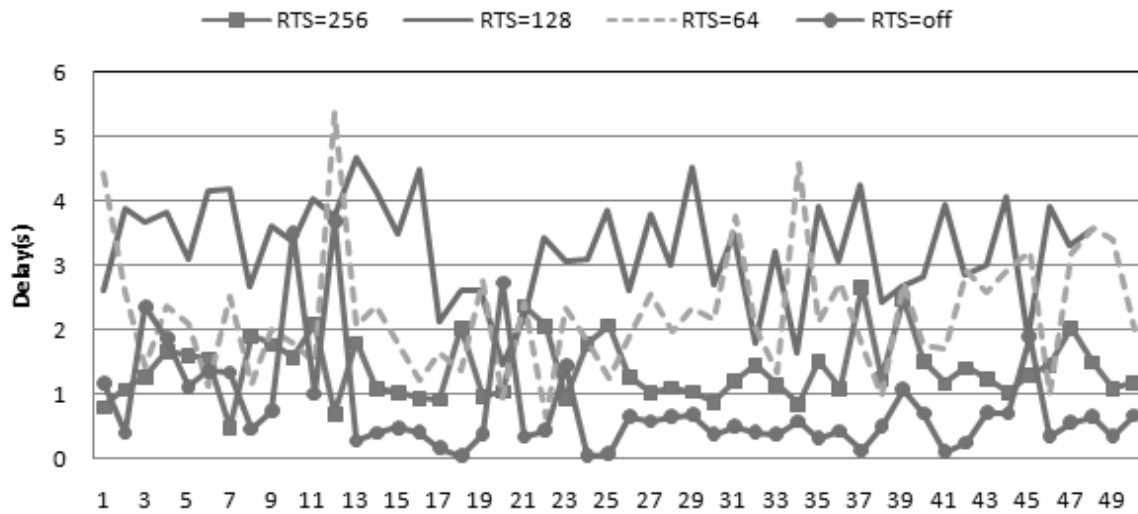


Fig. 8. RTT.

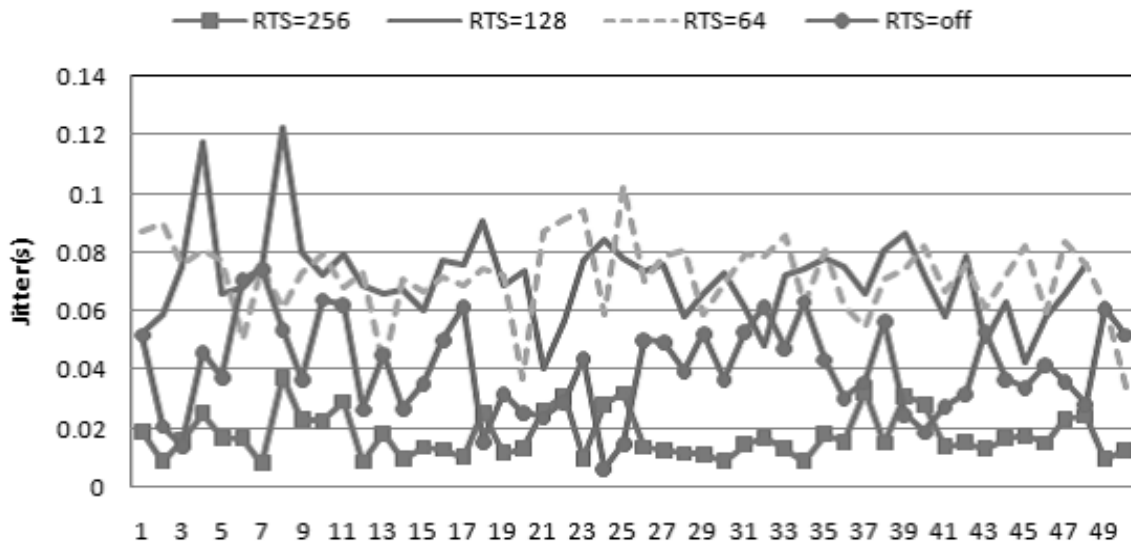


Fig. 9. Jitter.

packet loss is higher when the RTS/CTS is off. So we need to consider some trade-off relations when using the RTS/CTS mechanism.

In Fig. 9 we show the experimental results for the jitter parameter. The case when the threshold is 256 has the smaller jitter than other thresholds and when RTS/CTS mechanism is off. This shows that by a good selection of the threshold value we can achieve a better performance of RTS/CTS mechanism.

In Fig. 10 we show the experimental and simulation results for the same settings. As can be seen from

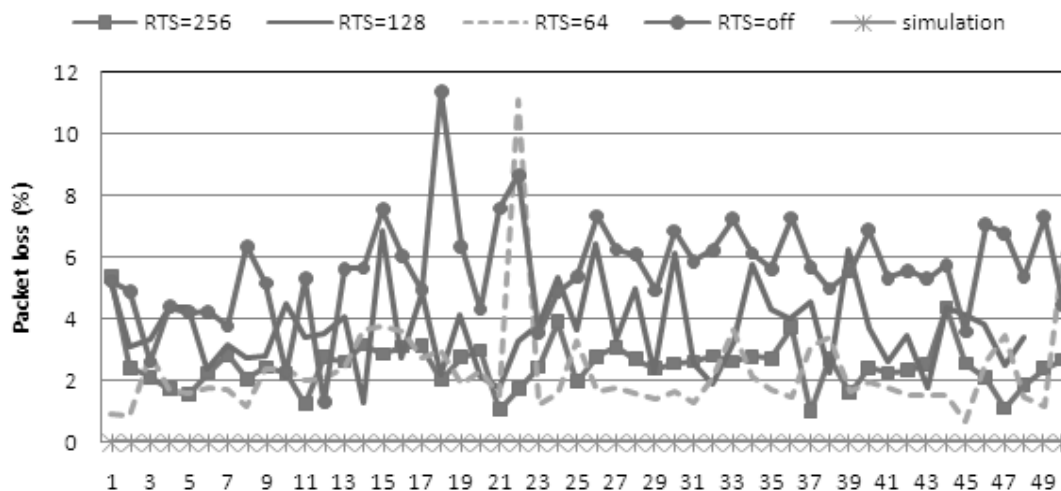


Fig. 10. Packet loss.

the figure, there are not packet losses during simulations although we experienced packet loss during experiments because of the effects of the environment and the traffic interference. This demonstrate that experiments with ad-hoc networks are a must for getting good evaluation data before implementing them in real environments.

8. Conclusions

In this paper, we carried out experimental and simulation results for a small wireless ad-hoc network with 7 nodes. We used OLSR protocol for experimental and simulation evaluation. We considered four parameters for performance evaluation: throughput, RTT, jitter and packet loss.

In order to deal with hidden node problem, we implemented RTS/CTS mechanism. The experimental results shows that we should be careful when implementing RTS/CTS. We got better results when the threshold value was 256.

For the same settings, we did not get any packet loss by simulations, but we experienced packet loss during the experiments because of the effects of the environment and the traffic interference.

In order to make easy settings of the testbed we implemented a new interface. In the future work, we would like to improve the interface and increase the number of nodes.

Acknowledgment

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

References

- [1] H. Lundgren, E. Nordstro and C. Tschudin, Coping with Communication Gray Zones in IEEE 802.11b Based Ad Hoc Networks, *Proc of the 5th ACM International Workshop on Wireless Mobile Multimedia* (2002), 49–55.

- [2] NS-2: Network Simulator (Ver. 2), LBL, <http://www.isi.edu/nsnam/ns/>.
- [3] A. Tønnesen, OLSRd: Implementation Code of the OLSR, <http://www.olsr.org>, 2006.
- [4] D.A. Maltz, J. Broch and D.J. Johnson, Lessons from a Full-Scale Multihop Wireless Ad Hoc Network Testbed, *IEEE Personal Communications* **8**(1) (2001), 8–15.
- [5] 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 of the 7-th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM-2004)* (2004), 220–229.
- [6] G. Anastasi, E. Borgia, M. Conti and E. Gregori, IEEE 802.11b Ad Hoc Networks: Performance Measurements, *Cluster Computing* **8**(2–3) (2005), 135–145.
- [7] V. Kawadia and P.R. Kumar, Experimental Investigation into TCP Performance over Wireless Multihop Networks, *Proc of the 2005 ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND-2005)* (2005), 29–34.
- [8] T.H. 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* (2001), 6.
- [9] T. Clausen and P. Jacquet, Optimized Link State Routing Protocol, *IETF RFC3626*, 2003.
- [10] D. De Couto, D. Aguayo, J. Bicket and R. Morris, A High-throughput Path Metric for Multihop Wireless Routing, *Proc of ACM MobiCom-2003* (2003), 134–146.
- [11] Ndiswrapper: <http://ndiswrapper.sourceforge.net>.
- [12] The rt2x00 Project: <http://rt2x00.serialmonkey.com>.
- [13] D-ITG: Distributed Internet Traffic Generator, <http://www.grid.unina.it/software/ITG>.
- [14] 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, *Proc of IEEE AINA-2007* (2007), 7–14.

Makoto Ikeda received BE and MS from Fukuoka Institute of Technology (FIT), Japan in March 2005 and 2007, respectively. Presently, he is a PhD Student at Graduate School of Engineering, FIT. His research interests include high-speed networks, routing protocols, genetic algorithms, sensor and ad-hoc networks. He is a Student Member of IPSJ.

Leonard Barolli received BE and PhD degrees from Tirana University and Yamagata University in 1989 and 1997, respectively. Dr. Barolli is a Full Professor, at Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT). Dr. Barolli has published more than 250 papers in referred Journals and International Conference proceedings. He has served as a Guest Editor for many International Journals. Dr. Barolli has been a PC Member of many International Conferences and was the PC Chair of IEEE AINA-2004 and IEEE ICPADS-2005. He was General Co-Chair of IEEE AINA-2006 and AINA-2008, Workshops Chair of iiWAS-2006/MoMM-2006 and iiWAS-2007 /MoMM-2007, Workshop Co-Chair of ARES-2007, ARES-2008 and IEEE AINA-2007. Presently, he is General Co-Chair of CISIS-2009, and Workshop Co-Chair of ARES-2009. Dr. Barolli is the Steering Committee Chair of CISIS International Conference and is serving as Steering Committee Member in many International Conferences. He is organizers of many International Workshops. His research interests include network traffic control, fuzzy control, genetic algorithms, agent-based systems, ad-hoc networks and sensor networks. He is a member of SOFT, IPSJ, IEEE CS and IEEE.

Giuseppe De Marco received the laura degree (5 years) in Electrical Engineering from University of Bologna, Italy, in 1999, and the PhD degree in 2004 from University of Salerno, Italy. From 2001 to 2003 he has been involved in the research activities of CoRiTel consortium, partially funded by Ericsson Lab Italy. From 2004 to 2006, he held an appointment at the Dipartimento di Ingegneria dell'Informazione e Ingegneria Elettrica (DIIE) at University of Salerno, as Research Fellow. Since 2004, he spent two years as JSPS Post Doctor Fellow Researcher at the Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT), Japan. Currently, he is with Toyota Technological Institute, Japan, where he is involved in security analysis of wireless networks and VANET. His main scientific interests are protocol design, performance analysis and wireless networks. He is a member of IEEE.

Tao Yang received BE from Hunan University, China in 2001 and MS from Fukuoka Institute of Technology (FIT), Japan in March 2007. Presently, he is a PhD Student at Graduate School of Engineering, FIT, Japan. His research interests include ad-hoc networks and sensor networks.

Arjan Duresi is an Associate Professor at Indiana University Purdue University at Indianapolis (IUPUI). Previously, he was an Assistant Professor of Computer Science at Louisiana State University. He received BE, MS and PhD from Polytechnic University of Tirana, Albania and a Superior Specialization Diploma in telecommunication from La Sapienza University in Rome, Italy and Italian Telecommunication Institute. He has 20 years of experience in industry and academic research, in the

areas of network architectures, heterogeneous wireless networks, security, Grid computing, etc. He has published more than 40 journal and 70 conference papers. He also has over thirty contributions to standardization organizations such as IETF, ATM Forum, ITU, ANSI and TIA. He is on the editorial boards of Ad Hoc Networks Journal (Elsevier) and a Senior Member of the IEEE.

Fatos Xhafa received his Ph.D. in Computer Science from the Technical University of Catalonia (Barcelona, Spain) in 1998. He joined the Department of Languages and Informatics Systems of the Polytechnic University of Catalonia as an Assistant Professor in 1996 and is currently Associate Professor and member of the ALBCOM Research Group of this department. His current research interest include parallel algorithms, approximation and meta-heuristics, distributed programming, Grid and P2P computing.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

