PARAMETER TUNING OF ROUTING PROTOCOLS TO IMPROVE THE PERFORMANCE OF MOBILE ROBOT TELEOPERATION VIA WIRELESS AD-HOC NETWORKS

Florian Zeiger, Nikolaus Kraemer and Klaus Schilling
University of Wuerzburg, Department of Robotics and Telematics, Am Hubland, Wuerzburg, Germany

Keywords: Networked robots, mobile robot teleoperation, mobile robot ad-hoc networks, wireless network, ad-hoc network, wireless teleoperation, wireless remote control.

Abstract: Currently, the use of wireless networks is very common in the field of networked robotics and can be considered as a key issue for capable multi robot systems with a high grade of mobility. Nevertheless, this mobility requests for special features of the communication infrastructure, which leads to the integration of mobile robots into wireless ad-hoc networks. Since the late nineties, more than 80 ad-hoc routing protocols were developed and nowadays some of them are implemented and ready to use in real world applications. A comparison of four ad-hoc routing protocols (AODV, DSR, OLSR, and BATMAN) showed some shortfalls of the default parameter settings not allowing a reliable teleoperation of mobile robots while using AODV, OLSR, or BATMAN. This work is focused on the parameter tuning of the routing protocols to use them in wireless ad-hoc networks of mobile robots. The time required for route reestablishing, as well as the packet loss during rerouting is investigated in hardware tests of a network with dynamic network topology consisting of mobile robots. It could be demonstrated, that an appropriate parameter setting of OLSR and AODV allow the teleoperation of mobile robots in outdoor environments via a wireless ad-hoc network.

1 INTRODUCTION

Currently, more and more research is done in the field of teleoperation of mobile robot teams via wireless networks. As now a larger number of mobile robots are developed which are capable to operate in impassable or hazardous environments with little or no communication infrastructure, the communication infrastructure is set up by the robots itself on demand. Within these wireless ad-hoc networks, different types of nodes might be present: human workers or rescue personnel equipped with modern communication devices, mobile robots, or even some stationary nodes. All of them are able to act as data source, data sink, and communication relay and must support a dynamic network topology. In the field of networked robotics, several approaches are using wireless ad-hoc networks in many different areas of robot teleoperation. Multi-robot exploration with robots using wireless networks (Rooker and Birk, 2007) or a mobile robot team connected via wireless network which performed localization and control tasks (Das et al., 2002). Also in the field of rescue robotics (Rooker and Birk, 2005), or for integrating UAVs into IP based ground networks (Zeiger et al., 2007), the use of wireless networks is quiet common nowadays.

An example for the network topology of these future scenarios is given in Figure 1. The network consists of several stationary nodes or ground stations and several mobile nodes which can be ground vehicles, aerial vehicles, or humans equipped with com-

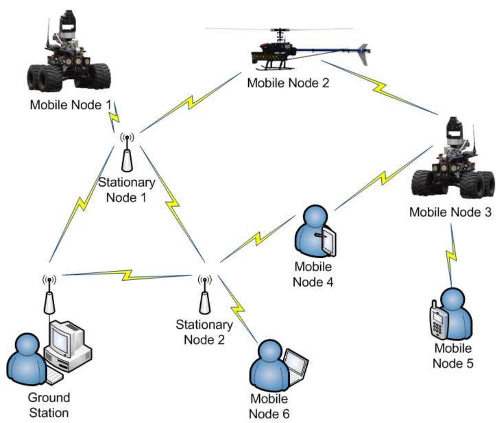

Figure 1: Future scenario of a heterogeneous network of mobile robots and human.

Zeiger F., Kraemer N. and Schilling K. (2008).
PARAMETER TUNING OF ROUTING PROTOCOLS TO IMPROVE THE PERFORMANCE OF MOBILE ROBOT TELEOPERATION VIA WIRELESS AD-HOC NETWORKS.
In *Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics - RA*, pages 53-60
DOI: 10.5220/0001488800530060
Copyright © SciTePress

munication devices. All these nodes are connected by an ad-hoc wireless network which should guarantee a transparent any-to-any communication. Nevertheless, wireless communication always implies unpredictable communication delays, packet loss, or in worst case the loss of the link which makes the provision of the required quality a challenging task (Hsieh et al., 2006). To avoid the loss of communication, research focused on a dynamic setup of the required telecommunication infrastructure by placing relay nodes on demand (Nguyen et al., 2004)(Pezeshkian et al., 2007) or using mobile robots as relay nodes (Nguyen et al., 2003)(Pezeshkian et al., 2006). These approaches are using communication relays in wireless ad-hoc networks to setup communication networks with dynamic topologies. In these wireless networks no fixed infrastructure exists, and each mobile node not only works as host but also as router for data packets of other nodes. These dynamic topologies of wireless communication networks have advantages like providing direct and indirect any-to-any communication of each network node, redundant communication links in larger networks, no central administration, and a distribution of the traffic load in large networks. Of course, these advantages can only be used with rather complex and special routing protocols providing each node the necessary information about the network topology. The nodes itself are working as routers and must store the routing information of the complete network locally. In the field of wireless telecommunication, more than 80 adhoc routing protocols for wireless networks were developed (Johnson and Maltz, 1996)(Redi and Welsh, 1999)(Das et al., 2003)(Chakeres and Belding-Royer, 2004). Also some simulations for performance evaluations for larger scale telecommunication networks were done in the past (Broch et al., 1998)(Das et al., 2001)(Dyer and Boppana, 2001). (Johansson et al., 1999) compared several ad-hoc routing protocols in a simulation study and (Kiess and Mauve, 2007) gives a survey of currently existing real-world implementations of ad-hoc routing protocols, including some real-world scenario tests.

This work is based on the results of a former publication (Zeiger et al., 2008), which compares several ad-hoc routing protocols with respect to mobile robot teleoperation. The standard parameter settings of the routing protocols AODV, OLSR, DSR, and BATMAN were investigated. Unfortunately, only DSR showed to be an appropriate solution for mobile robot teleopreation and the performance other three routing protocols had to be improved by parameter tuning. This work is focused on the parameter tuning of OLSR, AODV, and BATMAN. The objective is an acceptable packet loss and time for rerouting in a highly dynamic network topology. Therefore, existing protocol implementations of AODV and DSR (http://core.it.uu.se/core/index.php) from Uppsala University and the University of Basel, OLSR (http://www.olsr.org) and BATMAN (https://www.open-mesh.net/batman) are used in real-world test scenarios where mobile robots are teleoperated in an outdoor environment.

The presented work is structured as follows. in Section 2, the used hardware and the real-world test scenarios are described. Section 3 gives a brief summary of a comparison of AODV, DRS, OLSR, and BATMAN with respect to mobile robot teleoperation with standard parameter settings. The results of this work – the parameter tuning of ad-hoc routing protocols to enable mobile robot teleoperation via wireless ad-hoc networks – are presented in Section 4. A conclusion is given in Section 5.

2 HARDWARE AND TEST SETUP

2.1 Hardware

For the performed tests, several mobile nodes were used. One node is a PC for the operator. Up to 4 MERLIN robots (standard version) were used as stationary communication relay nodes, and one Outdoor MERLIN was used (cf. Figure 2) (Eck et al., 2007).

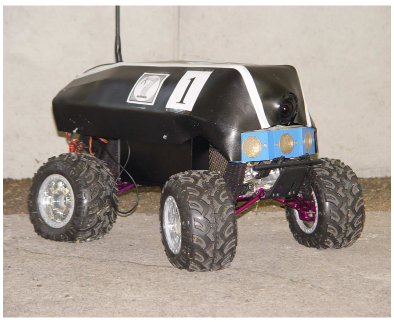

Figure 2: The Teleoperated OutdoorMERLIN Robot.

All MERLIN robots have a C167 micro controller for low-level operations and sensor data processing, as well as a PC-104 for more complex and computationally more intensive tasks. The PC-104 uses a Linux operating system and all nodes are equipped with 802.11b standard WLAN equipment (Atheros chip). More information on the MERLIN robots is given at www7.informatik.uni-wuerzburg.de. For steering the mobile robot, the operator's PC is running an application which generates command packets of a size between 6 and 22 bytes of payload. These packets are sent via UDP over the wireless network to the mobile robot. The onboard software of the mobile robot generates a UDP packet stream of packets with variable size containing the sensor data.

2.2 Test Setup

The scenario is set up in a way that the rerouting procedure will start with the mobile robot being at a certain location. Therefore, a large building is used as obstacle (cf. Figure 3). Relay nodes are placed at the corners of the building, such that they have always the neighbor nodes at the next and previous corner of the building within their communication range. As soon as the mobile robot is moved out of the line-of-sight of one node, the rerouting procedure is initialized. This scenario represents a worst case in terms of link redundancy, as only one route between operator PC and mobile robot is available. Relevant measurement categories are the packet loss and the duration of a communication drop-out during rerouting.

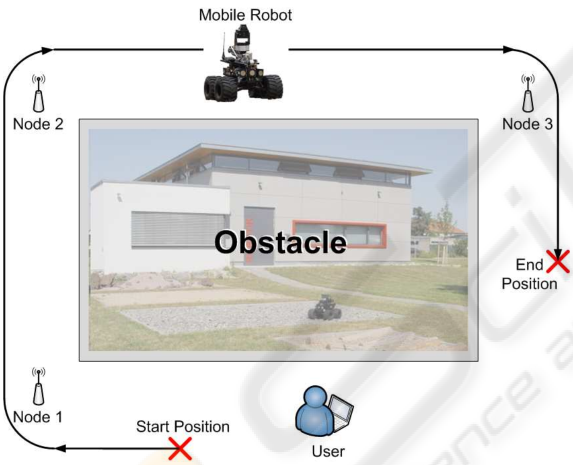

Figure 3: Test Setup.

3 AD-HOC ROUTING & TELEOPERATION

3.1 Investigated Protocols

This work investigates the parameter settings for different ad-hoc routing protocols: Ad-hoc On-demand Distance Vector (AODV), Dynamic Source Routing (DSR), and Optimized Link State routing (OLSR).

AODV (Das et al., 2003) (Chakeres and Belding-Royer, 2004) is a reactive routing protocol and determines required routes on-demand. To discover a route to an unknown destination, a Route Request (RREQ) message is broadcasted. Each intermediate node which is not the destination and without a route to the destination receiving a RREQ broadcasts it further. In case the RREQ is received more than once, only the first reception will result in a broadcast. To avoid uncontrolled dissemination of RREQs, each has a certain time to live (TTL) after which it is discarded. When the destination receives a RREQ message a Route Reply (RREP) message is generated and sent back to the source in unicast hop by hop fashion along the route which was determined by the RREQ message. After generating a RREP message, the RREQ message is discarded at this node. As the RREP propagates, each intermediate node creates a route to the destination. After the source receives the RREP, it records the route to the destination and begins sending data. In case the source receives multiple RREPs, the route with the shortest hop count is chosen. The status of each route is maintained in the local routing table and timers are used to determine link failures. In case a certain node is part of an active route, Hello messages are used to obtain the route status. These Hello messages are broadcasted periodically to all neighbors. If a neighbor does not send a Hello message within a specified time a link loss is detected and the node is deleted from the routing table. In addition, a Route Error message (RRER) is generated. More detailed information on AODV is given in (Das et al., 2003). In the test scenarios of this work, AODV-UU version 0.9.5 from Uppsala University (Sweden) is used (http://core.it.uu.se/core/index.php/AODV-UU).

DSR is also a reactive ad-hoc routing protocol which works similar to AODV but without using Hello messages for route maintenance. However, it uses source routing (DARPA Internet Program, 1981). DSR does not use any periodic routing advertisement, link status sensing, or neighbor detection packets, and does not rely on these functions from any underlying protocols in the network. DSR is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes in the ad-hoc network. In case source node (S) wants to send data to an unknown destination host (D), S initiates the route discovery mechanism. S broadcasts a route request message which identifies the source and destination of the route discovery to all neighbors. A route request also contains a record listing the address of each intermediate node which forwarded this particular copy of the route request. A node which receives this route request without being the destination looks up for a source route to the requested destination in its own route cache. Without any source route present in its route cache, the node appends its own address to the route record and broadcasts the route request message. In case this request message was received more than once, it is simply discarded. As soon as the route request message arrives at the desired destination D, a route reply message to S is created which contains an accumulated route record of the route request. After S receives this route reply, it caches the corresponding route in its route cache and S is ready to transmit data. Of course, there exist mechanisms to omit flooding of the network with route requests. A hop limit was introduced and every time a route request is forwarded, the hop limit is decremented by one. As soon as it reaches zero, the request is discarded. Also mechanisms for avoiding infinite recursion of route discoveries are implemented. A more detailed description of this protocol is given in (Johnson and Maltz, 1996) (Hu et al., 2004). The presented work uses DSR-UU version 0.2 from Uppsala University (Sweden) (http://core.it.uu.se/core/index.php/DSR-UU).

OLSR is a table-driven pro-active routing protocol for mobile ad-hoc networks. It uses hop-by-hop routing (each node uses its local information to route packets). OLSR minimizes the overhead from flooding of control traffic by using only selected nodes called Multipoint Relays (MPR) to retransmit control messages. Each node in the network selects a set of nodes in its neighborhood, which may retransmit its messages. This set of selected neighbor nodes is called the MPR set of that node. The neighbors of node N which are not in its MPR set, receive and process broadcast messages but will never retransmit broadcast messages received from node N. The MPR set is selected such, that every node in the 2-hop neighborhood of N has a link to the MPRs of N. OLSR continuously maintains routes to all destinations in the network by distributing link and neighborhood information (periodically exchange Hello messages). These messages are also used for link sensing and for checking the connectivity. More details on OLSR are given in (Clausen, 2003). The scenario tests in the present work are performed with OLSR version 0.5.3 (http://www.olsr.org/index.cgi?action=download).

BATMAN (Better approach to mobile ad-hoc networking) is a new approach to ad-hoc routing. Unlike other algorithms that exist right now, BATMAN does not calculate routes. It continuously detects and maintains the routes by receiving and broadcasting packets from other nodes. Instead of discovering the complete route to a destination node, BATMAN only identifies the best single-hop neighbor and sends a message to this neighbor. These messages contain the source address, a sequence number, and a time-to-live (TTL) value that is decremented by 1 every time before the packet is broadcasted. A message with a TTL value of zero is dropped. The sequence number of these messages is of particular importance for the BATMAN algorithm. As a source numbers its messages, each node knows whether a message is received the first time or repeatedly. More details on BATMAN are given in (B.A.T.M.A.N. (better approach to mobile ad-hoc networking), 2007). In the test scenarios of the presented work, BATMAN version 0.2 is used (https://www.open-mesh.net/batman).

3.2 Rerouting Time and Packet Loss with Standard Parameter Settings

In (Zeiger et al., 2008), four different ad-hoc routing protocols were investigated with respect to mobile robot tele-operation. A mobile robot was commanded in a test scenario which forced the routing protocols to increase the number of participating nodes in the communication link while the robot was moved around an obstacle. The four compared ad-hoc routing protocols were all used with the standard parameter settings and behaved quiet different. While it was not possible to accomplish the scenario with BATMAN, the other protocols at least allowed a teleoperation of the mobile robot – often with only very limited performance. AODV was originally designed for highly dynamic networks. Routes are established on demand. In some cases this rerouting took only a very short time (cf. Table 1), but sometimes, the communication drop-out duration was longer than 30 seconds. This is by far too long for the telecommand of a mobile robot. Compared to AODV, the minimum of the required re-routing time, OLSR is slower. Rarely, also communication drop outs were observed. With only half of the packet loss, OLSR showed a slightly better performance as AODV. Although OLSR worked more reliable than BATMAN or AODV, the observed minimum time for re-routing of 10.1 seconds is quiet high with respect to teleoperation and will not be appropriate for any kind of control via this network. DSR showed to be the most reliable and the fastest proto-

Table 1: Packet Loss & Times for Route Reestablishing from (Zeiger et al., 2008).

Protocol	Packet loss during test run	Time for re-routing	
		min.	max.
AODV	29.2%	$2.1s$	$> 30s$
OLSR	14.2%	$10.1s$	$> 30s$
DSR	11.2%	$2.4s$	$2.7s$
BATMAN	conn. lost	–	–

col which was tested. A packet loss of about 11% and a re-routing time between 2.4 and 2.7 seconds make this protocol suitable for reliable telecommand of a mobile robot. With respect to the test scenario, it was expected that DSR performs best, as only one node (the robot) is mobile and all other nodes are stationary. Here, DSR discovers the topology quiet fast and only the changes due to the robots movement result in routing messages. The used test scenario also represents a worst case in the means of route redundancy due to the availability of only one possible route between controller and mobile robot. This could be the reason for the relatively poor performance of AODV and OLSR. Originally, these protocols were developed to handle much larger networks with higher node mobility and a higher grade of meshing as in the current test scenario. Nevertheless, the presented test scenario is quiet typical with respect to teleoperation. In Table 1, also the average packet loss during route reestablishing is given. Again DSR showed the best performance (11.2% packet loss) compared to OLSR (14.2% packet loss) and AODV (29.2% packet loss). BATMAN was not able to establish a new route via additional relay nodes.

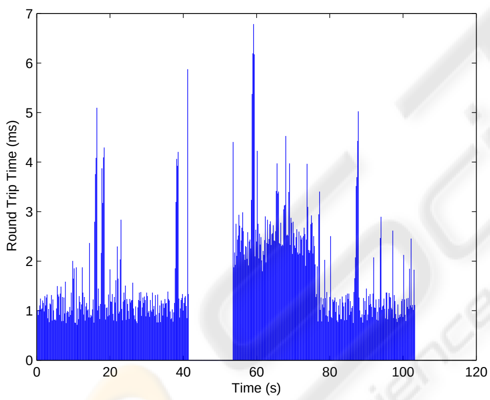

Figure 4: Example for OLSR round trip times for scenario 1 with default parameter setting.

Figure 4 displays an example of the round trip times for test scenario 1 and shows a clear communication drop out between 40 and 50 seconds test time. In Figure 5 the behavior of DSR in the same scenario is showed. Here, the communication drop-outs were significantly shorter. The compared ad-hoc routing protocols followed different principles (e.g. proactive vs. reactive) and were designed by different inspirations and for different application areas. Thus, they have also different parameters with different default settings, too. Nevertheless, this investigation showed that a better performance is required for mobile robot teleoperation. Based on these experiments, protocol

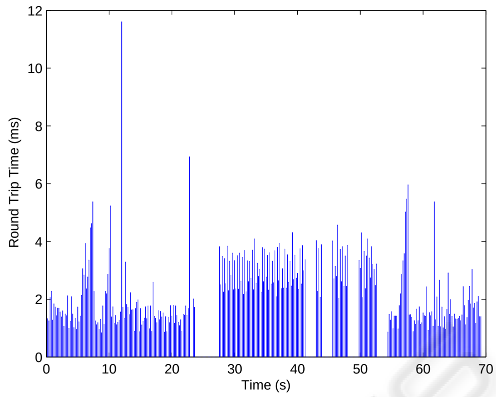

Figure 5: Example for DSR round trip times for scenario 1 with default parameter setting.

parameters are identified and tuned in order to increase the quality of the communication. The results are presented in the following sections.

4 PARAMETER TUNING

4.1 Variable Protocol Parameters

This section summarizes the default parameter settings of the investigated ad-hoc routing protocols OLSR (cf. Table 2), AODV, and BATMAN. In (Zeiger et al., 2008), these settings were chosen as a trade-off between generated routing overhead and fast topology discovery. Unfortunately, these parameter settings for OLSR, AODV, and BATMAN showed to be not suitable to use these ad-hoc routing protocols for mobile robot tele-operation.

Table 2: Variable parameters for OLSR.

Name:	Default Value
Willingness	dyn. calc.
LinkQualityLevel	2
LinkQualityWinSize	10
Pollrate	0.05 sec
TcRedundancy	0
HelloInterval	2 sec
HelloValidity	6.0 sec
TCInterval	5 sec
TCValidity	15.0 sec

For AODV, relevant parameters are ”force gratuitous”, ”local repair”, and ”no wait on reboot ” which are disabled by default. The BATMAN protocol offers less possibilities for parameter changes. In the following sections, ”originator interval” will be used for optimization. By default, ”originator interval” is set to 1000 milliseconds.

4.2 OLSR

To increase the performance of OLRS in order to use it for mobile robot teleoperation, the four parameters were changed as shown in Table 3. As the parameter settings are interdependent, some simple rules must be followed as not all combinations of values are useful. The hello- and tc-intevals have to be smaller than half the corresponding validity times. Of course, the traffic for routing will be increased due to more hello and tc-messages but will not cause decrease the throughput significantly.

Table 3: Tuned parameters for OLSR.

Name:	Default Value:	New Value:
HelloInterval	2 sec	0.5 sec
HelloValidity	6 sec	1.5 sec
TcInterval	5 sec	2.5 sec
TcValidityTime	15 sec.	5.0 sec

The rerouting time was reduced from more than 10.1 to an average value of 5.96 seconds (with a minimum of 5.2 and a maximum of 7.4) for including the first relay node into the communication link. The packet loss during rerouting was reduced from 14.2% to 4%. Figure 6 shows an example of the behavoir of the round trip time during a rerouting process with OLSR. As the rerouting is initiated at 32.4 seconds, a short communication drop out occurred until the link is reestablished at 37.2 seconds simulation time. Thus, the OLSR performance could be increased but still, the observed performance of DSR during the comparison in (Zeiger et al., 2008) could not be reached.

4.3 BATMAN

As already mentioned, the number of variable parameters for BATMAN is very small. A suitable parameter to be tuned is the ”originator interval”, which is the time to wait sending one message and before the batman daemon sends the next message (default value is 1000 milliseconds). In (Zeiger et al., 2008), all BATMAN test runs ended up in a communication loss. Now, the value for ”originator interval” is decreased which should result in a faster response on topology changes by the routing protocol. Figure 7 shows an Example of a test run with ”originator interval” set to 125 ms. Of course, this setting increases the routing

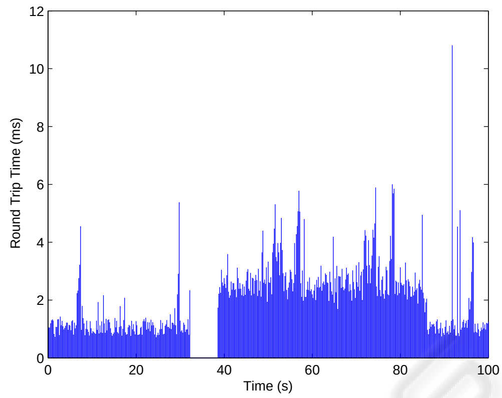

Figure 6: Example for OLSR round trip times for scenario 1 with HelloInterval=0.5, HelloValidity=1.5, TCInterval=2.5 and TCValidity=5.0.

overhead but as a result, BATMAN now can handle the rerouting without losing the communication. During the tests, the average rerouting duration is 7.78 seconds with a packet loss of 8%. This parameter setting significantly improved the performance of BATMAN but still, the duration of the rerouting procedure takes too long to be used for mobile robot teleoperation.

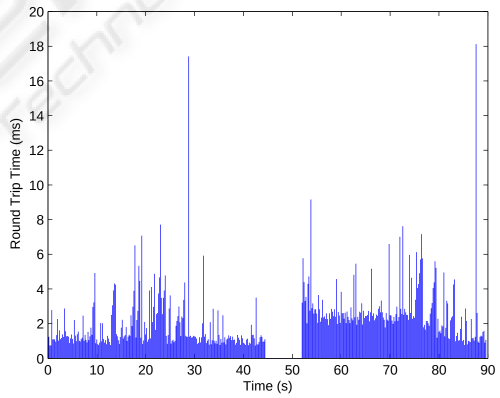

Figure 7: Example for BATMAN with ”originator interval” set to 125 ms.

4.4 AODV

For AODV, it is possible to tune several parameters. The performed tests showed, that only an combined change of several parameters might have the opportunity to show a positive effect with respect to teleoperation. Unfortunately, it was not possible to find a suitable parameter setting – all combinations made the protocol behaving very unstable. For example, enabling ”force gratuitous”, ”local repair”, ”no wait on reboot”, and setting ”treat node as neighbor” to 2 (cf.

Figure 8) lead to many communication losses.

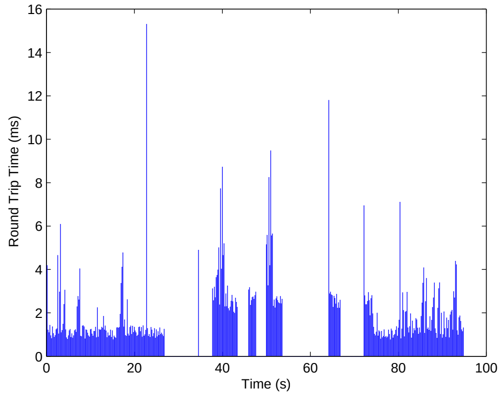

Figure 8: Example for AODV with "force gratuitous", "local repair", and "no wait on reboot" enabled and "treat node as neighbor" = 2.

5 CONCLUSIONS

This work uses results of a comparison of four ad-hoc routing protocols as a basis where the default parameter setting showed to be not appropriate for mobile robot teleoperation. Here, results of a study for parameter tuning of real implementations of the ad-hoc routing protocols OLSR, AODV, and BATMAN are presented. Real hardware tests of a mobile robots teleoperation scenarios were performed and the behavior of the communication link was analyzed with respect to mobile robot teleoperation. It is shown, that a tuning of a combination of relevant parameters for OLSR can improve its performance. Also the reduction of originator interval" of BATMAN results in a more reliable network than experienced with the default parameter settings. For AODV, sometimes a better reaction in terms of required time for rerouting could be observed. The evaluated protocol implementations are suitable for some teleoperation approaches – e.g. systems with local autonomy functions to ensure a defined behavior of the mobile robot during the rerouting times. Nevertheless, the behavior of all tested ad-hoc routing implementations was less than expected. The previously mentioned simulation results showed much shorter rerouting times (in a magnitude of some milliseconds). These short rerouting times were never observed in the presented real hardware tests. Future work will now be focused on a detailed analysis of the differences between the simulation studies and the effects described in this work to further improvement the usability of AODV, OLSR, DSR, and BATMAN for mobile robot teleopertation or remote control approaches via wireless multi-hop networks. In addition, the required interaction between applications (e.g. teleoperation interfaces or control algorithms) and the network status via the lower protocol layers must be analyzed.

REFERENCES

B.A.T.M.A.N. (better approach to mobile ad-hoc networking) (2007). https://www.open-mesh.net/batman, visited 2007/09/07.

Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., and Jetcheva, J. (1998). A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Mobile Computing and Networking*, pages 85–97.

Chakeres, I. D. and Belding-Royer, E. M. (2004). AODV Routing Protocol Implementation Design. In *Proceedings of the International Workshop on Wireless Ad hoc Networking (WWAN), Tokyo, Japan*, pages pp. 698–703.

Clausen, P. J. T. (2003). RFC 3626, Optimized link state routing protocol (OLSR). IETF, Network Working Group.

DARPA Internet Program (1981). RFC 791, Internet Protocol Specification.

Das, A., Spletzer, J., Kumar, V., and Taylor, C. (2002). Ad Hoc Networks for Localization and Control. In *Proceedings of the 41st IEEE Conference on Decision and Control, (CDC 2002)*, volume 3, pages 2978–2983.

Das, S., Perkins, C. E., and Belding-Royer, E. M. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. IETF RFC 3561.

Das, S. R., Perkins, C. E., Royer, E. M., and Marina, M. K. (2001). Performance Comparison of Two On-demand Routing Protocols for Ad hoc Networks. *IEEE Personal Communications Magazine special issue on Ad hoc Networking*, February 2001:16–28.

Dyer, T. D. and Boppana, R. V. (2001). A comparison of TCP performance over three routing protocols for mobile ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Long Beach, CA, USA*, pages 56–66.

Eck, D., Stahl, M., and Schilling, K. (2007). The Small Outdoor Rover MERLIN and its Assistance System for Tele-Operations. In *Proceedings of International Conference on Field and Service Robotics (FSR), Chamonix (France)*.

Hsieh, M. A., Cowley, A., Kumar, V., and Taylor., C. (2006). Towards the deployment of a mobile robot network with end-to-end performance guarantees. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'06), Orlando, FL (USA)*, pages 2085–2090.

Hu, Y.-C., Johnson, D. B., and Maltz., D. A. (2004). The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. IETF MANETWorking Group, IETF RFC 4728.

Johansson, P., Larsson, T., Hedman, N., Mielczarek, B., and Degermark, M. (1999). Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In *Proceedings of Mobicom'99, Seattle, Washington*.

Johnson, D. B. and Maltz, D. A. (1996). Dynamic Source Routing in Ad-Hoc Wireless Networks. *Mobile Computing*, 353.

Kiess, W. and Mauve, M. (2007). A Survey on Real-World Implementations of Mobile Ad-Hoc Networks. *Elsevier's Ad Hoc Networks*, 5(3):324–339.

Nguyen, H. G., Pezeshkian, N., Gupta, A., and Farrington, N. (2004). Maintaining Communication Link for a Robot Operating in a Hazardous Environment. In *Proceedings of ANS 10th International Conference on Robotics and Remote Systems for Hazardous Environments, Gainesville, Florida, March 28-31*.

Nguyen, H. G., Pezeshkian, N., Raymond, M., Gupta, A., and Spector, J. M. (2003). Autonomous Communication Relays for Tactical Robots. In *The 11th International Conference on Advanced Robotics, Proceedings of ICAR 2003, Coimbra, Portugal, June 30 - July 3*.

Pezeshkian, N., Nguyen, H. G., and Burmeister, A. (2006). Unamnned Ground Vehicle Non-Line-of-Sight Operations Using Relaying Radios. In *Proceedings of the 12th IASTED International Conference on Robotics and Applications, Honolulu, Hawaii (USA)*.

Pezeshkian, N., Nguyen, H. G., and Burmeister, A. (2007). Unmanned Ground Vehicle Radio Relay Deployment System for Non-Line-Of-Sight Operations. In *Proceedings of the 13th IASTED International Conference on Robotics and Applications, Wuerzburg (Germany)*.

Redi, J. and Welsh, R. (1999). Energy-Conservation for Tactical Mobile Robots. In *Proceedings of Military Communications Conference, MILCOM*.

Rooker, M. N. and Birk, A. (2005). Combining Exploration and Ad-Hoc Networking in RoboCup Rescue. In Nardi, D., Riedmiller, M., and Sammut, C., editors, *RoboCup 2004: Robot Soccer World Cup VIII*, volume 3276 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 236–246. Springer.

Rooker, M. N. and Birk, A. (2007). Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445.

Zeiger, F., Kraemer, N., and Schilling, K. (2008). Commanding Mobile Robots via Wireless Ad-Hoc Networks - A Comparison of Four Ad-Hoc Routing Protocol Implementations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'08)*.

Zeiger, F., Selbach, C., Ruderisch, B., and Schilling, K. (2007). An Application Protocol to Integrate a Small Size Helicopter into an IP based Ad-Hoc Network. In *Proceedings of ROBOCOMM 2007, Athens (Greece)*.