Realism in Design and Evaluation of Wireless Routing Protocols*

André Herms Georg Lukas
Svilen Ivanov
University of Magdeburg
Institute of Distributed Systems
{aherms, glukas, svilen}@ivs.cs.uni-magdeburg.de

Abstract: In this paper we consider the problem of reliable communication in wireless routing protocols. Many routing protocols for wireless networks use assumptions that are known to be invalid in this special kind of networks. These shortcomings are not detected in the protocol evaluation because common simulation tools are also based on these assumptions. In this paper we present a more realistic simulation model and a simple way of enabling existing protocols to handle real conditions of wireless propagation. A proof-of-concept implementation is presented and evaluated using simulation and measurements in a real and emulated network.

1 Introduction

In the development of wireless communication protocols simulation is the generally accepted and most widely used evaluation mechanism. Design and implementation decisions are based on results collected in simulation. When protocols are later tested in real setups, the observed behaviour significantly differs from the simulated one. Especially packet loss rates and topology stability are often much worse than observed in simulation.

Liu et. al. show in [LYN+04] that the delivery ratio of a simulated AODV network exceeds 80 % with the commonly used two-ray ground model, while the same scenario only yields around 42 % packet delivery in a real experiment. This is supported e.g. by [Bor05]. In [KNG+04] they give the reasons for this unforeseen behaviour: wrong assumptions on the properties of the wireless propagation and oversimplified simulation models. Still, when designing wireless routing protocols, most groups rely on default simulation settings without thoroughly considering their quality. Here, we follow a systematic approach to increase the quality of simulation by applying more realistic propagation models. Then we present a routing protocol, which was designed according to this understanding of the real wireless medium and verify the protocol on a real network.

In section 2 of this paper, we discuss the most common misconceptions used in simulation and during protocol development. Sections 3 and 4 suggest better models for wireless

*This work has been partially supported by the German Research Foundation (DFG), grant no. NE 837/3-2 and the German federal state of Saxony-Anhalt.

connectivity and determination of stable links. In section 5 we integrate these concepts into a MANET routing protocol and describe its implementation. The correctness of the link quality and symmetry algorithms is evaluated by comparing the simulation results with measurements in a real network in section 6. The paper is completed by an overview of related work in section 7 and a conclusion in section 8.

2 Common Pitfalls in Wireless Simulation Setups

The first generation wireless routing protocols like AODV and DSR are derived from protocols designed for wired networks, so they inherited their assumptions on the packet delivery, which are correct on the wired medium, but do not reflect wireless communication. The critical assumptions are:

Assumption 1 *Links are either perfectly usable or not all. If we can transmit a packet to a node once, it will be possible later, as long as the topology does not change.*

Assumption 2 *Links are bidirectional. If a station can receive packets from a neighbour node, it can be sure that the neighbour will receive its packets, too.*

Assumption 1 and 2 are e.g. used by AODV in for generating route replies ([PBRC03], section 6.6.1). DSR uses them for during the route discovery ([JMH04], section 3.1). It has been shown in several publications that these assumptions are unrealistic[KNE03, ZHKS04]. Nevertheless, many wireless protocols still rely on them, either explicitly or implicitly. The consequences will be explained in the following examples:

Neighbour Discovery Most protocols use HELLO messages for discovery of neighbouring stations. Every station periodically transmits broadcast packets, including at least its station identification number. Other stations in communication range receive them and can thus identify their direct neighbours. This information is used for calculating routes to distant nodes. However, the communication range in wireless networks has no strict boundary. At the border areas of the communication range the receive probability decreases. Thus, stations in this area will not receive *all* HELLO message, but only some of them.

An example is depicted in figure 1. The links between A and C and between C and B have a good quality, represented by their high packet delivery ratio of about 98 %, while the link between A and B has a weak connectivity of about 10 %. Such constellations are typical in wireless multi-hop networks.

The neighbour discovery of station A will detect B and C as adjacent and usable for routing, since even with a lower probability, HELLO packets of B are received. However, when this link is used for forwarding packets, most of them are lost due to the low delivery ratio. This problem is further amplified by routing protocols that use shortest path routing and thus choose relatively long (and thus lossy) links. In our example it would also be

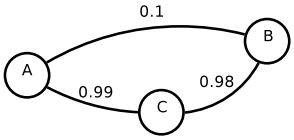

Figure 1: Example – weak link detection

possible to forward packets from A to B via $A \rightarrow C \rightarrow B$ with a better delivery ratio. However, the shortest path algorithms prefer the one-hop path.

Link Asymmetry The communication range of wireless stations differs, caused e.g. by obstacles like walls, by different transmit powers, or by different antennas. A critical case occurs when the endpoints of a wireless link have different transmission ranges. Expressed by packet delivery probabilities, this can result in the constellation shown in figure 2. Station A has a high probability of receiving packets from B, but B has a low probability of receiving packets from A. Obviously such links can not be used for bidirectional communication.

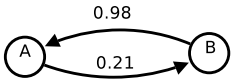

Figure 2: Example – asymmetric links

However, the normal beaconing mechanism is not able to detect asymmetric links and to exclude them from the routing topology. The problem is even worse for protocols using the count of *received* HELLO messages as a quality metric — thus, node A routes data to node B because it perceives the link as good. For unidirectional communication, here for packets from node B to node A, this problem also remains. Wireless devices following the IEEE 802.11 standard use MAC layer acknowledges for unicast packets. These acknowledge frames are lost on the way back, which results in a very high number of retransmissions and effectively limits the link quality in both directions. Similar problems arise in routing protocols using path reversal techniques, like AODV [Bor05] or DSR [ZHKS04].

The described problems are often not observed during the evaluation, which is normally done using simulation. As described by Kotz et. al. [KNE03], one major reason is the used propagation model. A statistic of the typically used models is presented, based on the MobiCom Proceedings of the years 1995 – 2003. It shows that with one or two exceptions all simulations use distance-based propagation models. The most popular member of this class is the two-ray ground reflection model that will serve as an example for discussing the consequences for evaluation of wireless protocols.

Figure 3 includes the receive probability of a wireless link with varied distance, using the two-ray ground reflection model. We can see two properties of the propagation model: First, the two-ray ground model has a receive probability of 1 in communication range and

a receive probability of 0 outside. This means: a node in communication range can receive *every* packet and a node outside will *never* receive one. This way the simulated packet delivery corresponds to the critical assumption 1.

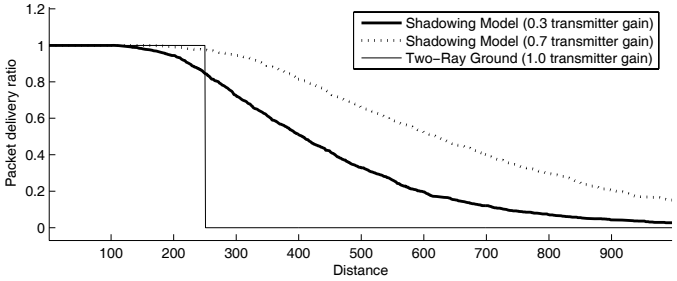

Figure 3: Receive probabilities of wireless propagation models

Furthermore, the receive probability is a function depending only on the distance between transmitter and receiver. Because the same distance is used for both directions of a link, the packet delivery ratio in both directions is the same, and thus the link is *always symmetric*. The consequence is that the simulation perfectly fulfils assumption 2.

Conclusion Many routing protocols for wireless networks are based on assumptions about the packet propagation that do not match real wireless networks. In most cases, only simulation is used to evaluate these protocols, where the utilised simulation models contain the same assumptions about the medium, thus creating the illusion that the protocols are adequate for real network use.

3 Modelling of wireless connectivity

Because simulation with standard parameters is not able to optimally represent wireless communication, other models have to be introduced, which better map real connectivity to the simulator and have a minimal number of assumptions. This section shows how the simulation can be configured to reduce the gap between simulator and reality.

Mullen et al. [Mul04] discusses this in detail and suggests models with simple probabilistic components. The shadowing model is recommended, which is also available in the commonly used network simulators NS-2[1] and OPNET Modeller[2]. For comparison with the two-ray ground reflection model, the receive probabilities are also depicted in figure 3. It can be seen that this model avoids the strict receive range by modelling a smooth transition of the receive probabilities. Thus, simulations using this model do not reinforce assumption 1. The second assumption (symmetric receive probabilities) still remains unresolved in the simulation model. Even though the shadowing model adds randomness

[1] `http://www.isi.edu/nsnam/ns/`
[2] `http://www.opnet.com/products/modeler/home.html`

to the reception of packets, the mean value of the observed packet delivery rate is only dependent on the distance.

In order to avoid the symmetry of all links an additional randomness must be added to the model. There exist several possible solutions for this. The Radio Irregularity Model (RIM) in [ZHKS04] uses a non-isotropic antenna model that adds a direction-dependent random factor to the antenna gain and thus creates a direction-dependent variation in the receive probability. Another means to model heterogeneousness of the wireless hardware is the use of varying receive power thresholds or transmitter powers. In figure 3 the receive probability of the shadowing model is shown with varying transmit gains. It can be seen that this is one simple but effective way of generating different receive probabilities for the same distance. It enables the occurrence of asymmetric links, which is required for evaluating routing protocols.

Conclusion Using simulation for evaluation of wireless protocols is possible, but the following must be considered: A propagation model must with probabilistic transmission ranges should be used. Asymmetric links should be included in the simulation, e.g. by individually initialised antenna transmit gains.

4 Generating stable symmetric links

After eliminating the unrealistic assumptions from the simulation model, the question arises how a routing protocol can be designed that is able to cope with the more realistic representation of the wireless medium. One option would be the development of a new class of protocols that can handle these fuzzy links, but this would require new developments and modifications of existing protocols. Instead, we use another approach that fulfils the assumptions of the existing protocols, so that these do not have to consider the specific properties of the wireless networks. In this section we present a simple methods for classifying links based on their symmetry and reliability and only to use suitable links for communication.

Stable links are defined by a high delivery ratio. We use periodic broadcast messages for identifying stable links. If the routing already uses periodic HELLO messages, these can be utilised. Otherwise, additional periodic packets must be generated. Every station tracks its neighbouring stations based on these messages. Because single packets are not sufficient for distinction of weak and stable links, a long-term observation is required. Therefore, sequence numbers must be added to the broadcast packets to allow the detection of packet loss based on sequence number differences.

A k-bit integer variable H_n per neighbour is maintained, with the i-th bit indicating whether the i-th previous packet was received. A simple update rule can be applied, when a new packet with a sequence number s is received and the previous sequence number was $\hat{s}$:

$$H_n := 2^{\hat{s}-s} H_n + 2^{k-1} \tag{1}$$

or using C-like boolean bit operations: $H_n = (H_n >> (\hat{s} - s)) \,|\, (1 << (k-1))$. Links are classified by two states: STABLE or WEAK with initially every link being a WEAK one.

For becoming STABLE a link has to win a challenge: It must propagate a certain number T^+ of consecutive packets without any losses. This can be identified by constantly checking the last T^+ of its packet history H_n. The other direction – becoming WEAK – is triggered by a sequence of T^- consecutive packet losses, which can be detected by a timeout of T^- periods since the last received packet. Typical values that have proven to create stable networks are $T^+ = 12$ and $T^- = 3$.

By applying this technique nodes are able to detect and avoid weak links and are able to determine which neighbours can transmit packets to them with a high success rate. However, this does not guarantee that a link is usable for communication in both directions, as e.g required for IEEE 802.11 unicast frames. To determine whether the link has a good symmetric quality, additional communication with neighbouring stations is required. Thus, a list of all neighbours with the state STABLE is added to the broadcast packets. A station can identify that a stable, bidirectional link to a neighbour n exists when both the local state of neighbour n is STABLE and its own node number is included in the neighbours broadcast.

For use with an arbitrary protocol we suggest the use of an additional filtering layer between routing and MAC. It uses the described mechanism for detecting stable, symmetric links. In order to let the layers above only see these links, all higher level packets from unstable or unidirectional links are dropped. This solution is generic and can be applied to most protocols.

Next we analyse the effect of our protocol on control overhead, link setup time, and connectivity changes. The control overhead depends on two factors – the topology and the period of HELLO packets. The larger the set of neighbours of a station is, the larger the packet containing them becomes. A shorter period of the HELLO packets further increases the control overhead, but also influences the link setup time. A link setup requires at least T^+ times the HELLO period to establish a usable connection. Hence, after startup a station is logically disconnected and it takes at least the link setup time until communication with neighbours is possible. The same applies to connectivity changes, e.g. caused by station mobility or moving obstacles, which causes links to break and thus requires additional setup times. Because of this, connectivities that exist only for a short time are not able to establish logical links. On the one hand this can lead to a very low number of links in highly dynamic networks and even to parts of the network not being reachable. On the other hand it would not be advisable to use such links for routing because they often cause routing errors. There is a inherent trade off between control overhead and link setup time that depends on the HELLO period. A reasonable value must be chosen depending on the kind of network. For static networks a large period is advisable. We used 1 second in our static scenarios which causes a negligible overhead. In more dynamic and mobile networks a smaller value of about 100 ms is useful, but also involves more control overhead.

5 Implementation

To verify the described approach, we applied it to a simple link-state routing protocol. It can be summarised as follows: Every station announces its neighbours by flooding the information through the network. Based on this a station can build up a local view of the complete topology. A shortest path algorithm calculates the next hop for every station in the network. This is used for forwarding packets.

We implemented the protocol using an event based abstraction layer (GEA, [HM05]). It allows us to run the same implementation either in the NS-2 or on Linux-based devices. During the development the protocol was tested in the simulation. By using the shadowing model with enforced asymmetry in our extensive testing, we could ensure the protocol's correct handling of weak and asymmetric links.

Our software consists of multiple modules (see figure 4), which are dynamically composed using a plug-in mechanism of the abstraction layer. The raw-basic module is responsible for binding to the network interface. It allows to send and receive Ethernet Frames using the network driver of the operating system. The routing module uses the provided functions for communication. The advantage of this approach, in opposite e.g. the use of IP packets, is that we avoid the problem of assigning unique IP addresses to the stations. The AWDS module contains the link stability and symmetry component and the routing. For binding applications to the routing, a virtual network interface[3] is created by the Tap Iface module. The IP stack of the operating system generates Ethernet frames that are directed through the tap device to our AWDS component. Based on the MAC address the frames are routed to the corresponding station, after being encapsulated as payload of our routing protocol. On the destination node the packets are delivered to the IP stack, again. Broadcast packets like ARP request are flooded and delivered on all nodes. From the users point of view the routing behaves like a switched wired network. Thus, beside IP other transport protocols like IPv6, IPX, etc. work as well, and the automatic configuration of IP addresses via DHCP is possible. The last module to mention is the topology monitoring that allows to dump the link graph, which was used for the evaluation.

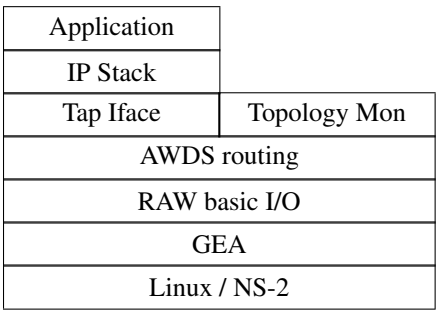

Figure 4: Composition of routing modules

[3] We use the universal tun/tap interface, available at `http://vtun.sf.net/tun/`

6 Evaluation

The objective of the experiments presented in this section is to test the proposed method for design of wireless ad-hoc routing protocols and the method for realistic evaluation of these protocols using network simulation. To do so, we evaluate the developed routing protocol described in section 5 in a real network and in a simulation scenario and show the differences between the results. In opposite to other works [Bor05, KUF+04] we use a complex topology that contains multiple paths to a destination and thereby enables the selection of wrong paths. First we describes the experimental setup and then show evaluations of typical network characteristics: end-to-end packet delivery ratio, network topologies, and end-to-end latencies.

6.1 Experimental Setup

We used the following setup: Sixteen wireless stations running our software were deployed in our office environment. Most of them were modified Netgear WGT 634U access points running Linux[4]. As communication endpoints served two DELL Latitude D410 laptops with 3Com PCMCIA wireless Cards using the madwifi-ng driver. All wireless interfaces were configured to use the IEEE 802.11g ad hoc mode in the same network cell. Our routing protocol was installed on all devices and bound to the wireless network interface.

Besides the wireless network the communication endpoints (D410 laptops) were connected to our intranet Gigabit Ethernet. The placement of all stations was recorded and reproduced in a simulation scenario. It was used to setup an emulated copy of the real scenario using the NS-2 emulation [MI04] with our software running on the emulated stations. The emulation host was also connected to the intranet.

For traffic generation a live stream from a MPEG4 web camera[5] was chosen and the video streaming software VLC[6] was used to transform the HTTP data stream into a MPEG4/TS RTP stream. The RTP stream was multiplied using the servers built-in duplication feature and directed to three destination IP addresses – the address of the receiver in the intranet, in the wireless network, and its representation in the emulation. This way we directed the same data stream though the real and emulated scenario and had a low latency data stream as reference for the evaluation. The traffic on client and server was recorded using tcpdump for later examination and reproduction.

A third version of the scenario was created based on pure simulation with the description taken from the emulation (table 1 shows the simulation parameters). The protocol stack was directly bound to the simulated station using the features of the abstraction layer GEA. A traffic generator on the simulated server was configured to load the traffic pattern from a tcpdump trace of the MPEG4 data stream. This allowed us to evaluate the protocol in

[4]The embedded Linux distribution OpenWrt (`http://www.openwrt.org`) is used, SVN revision 3936 from Jun 9th, 2006.

[5]AllNet ALL 2210

[6]VLC 0.8.5, `http://www.videolan.org`

simulation while avoiding effects generated by the emulation system, like latencies caused by the runtime operation system.

Table 1: Simulation parameters

Parameter	Ns-2 Setting (OTcl)	Value
802.11 MAC layer parameters	Mac/802_11	
Data Rate (unicast data frames)	dataRate_	11 MBit/s (54 MBit/s at one hop)
Basic rate (broadcast and management frames)	basicRate_	2 MBit/s
PLCP Data rate	PLCPDataRate_	1 MBit/s
RTS/CTS threshold	RTSThreshold_	250 bytes
Retry limit for management frames	ShortRetryLimit_	7
Retry limit for data frames	LongRetryLimit_	4
802.11 Physical layer parameters	Phy/WirelessPhy	
Transmission power	Pt_	25 dBm
Radio frequency	freq_	2.472 GHz
Carrier sense threshold	CSThresh_	-104 dBm
Receive threshold	RXThresh_	-94 dBm
Lambda	L_	1.0
Propagation model parameters	Propagation/Shadowing	
Path loss exponent (β)	pathlossExp_	5.7
Received power deviation (X_{dB})	std_db_	4
Reference distance	dist0_	1 meter
Seed for shadowing RNG	seed_	0

6.2 Experimental Results

For the protocol evaluation we measured two typical network properties that allow comparison with other works – latency and packet loss – depending on the number of hops. Another point of interest was the stability of the topology that is mainly influenced by our component for stable symmetric links.

Network Topology The network topology was determined using the topology monitoring module of the protocol stack on the client station. Figure 5 and figure 6 show the corresponding graph of the three networks. As the topology influences the number of hops between two stations, which affects the end-to-end packet loss and delay, we first look at the similarity between simulated and real topology.

Our metric is the number of wrong links, which means the number of links that can be found in one topology, but not in the other one. In our scenario 23 of 240 possible links were wrong, expressed by an error of 9.6 %. Considering that our simulation cannot model obstacles like walls that have the major influence in indoor environments, the result is surprising good.

Network Latency The packet latency was extracted from the tcpdump traces by using the RTP stream on the Gigabit network as reference. Packets on the real and emulated wireless network were matched over the RTP sequence number with the Gigabit stream. Assuming a minimal delay of the reference network the difference represents the latency

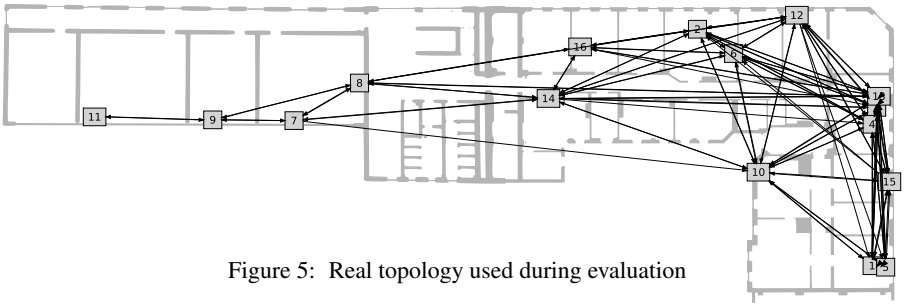

Figure 5: Real topology used during evaluation

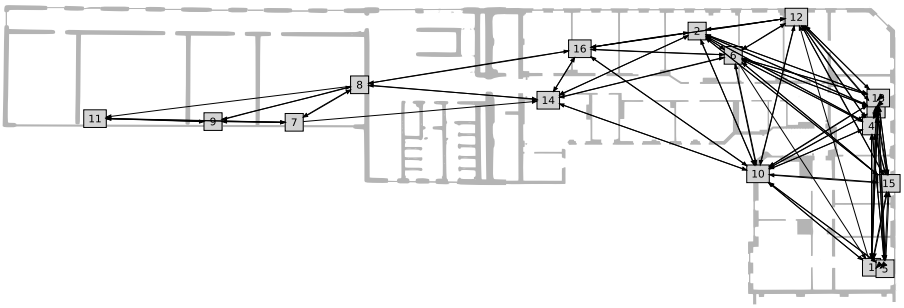

Figure 6: Emulated/simulated topology used during evaluation

of individual packets. For different locations and corresponding varying hop counts the delays are depicted in figure 7.

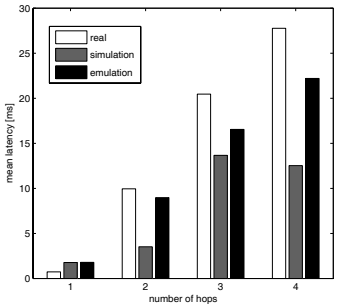

Figure 7: Mean packet latencies over different number of hops

The results show the expected linear increase of the latency with increasing number of hops. The only outlier is the simulated 4 hop scenario. In this case the number of hops between client and server differed because a 3 hop route was created instead of a 4 hop one. Is can also be seen that the simulation always underestimates the latency. We identified the following possible reasons:

1. The simulation model does not represent delays that occur in real operating systems

like process scheduling, interrupts, bus transfers and others.

2. The ns-2.29 wireless model uses a fixed data rate at the MAC layer whereas the data rate in modern wireless networks is dynamically adapted to the quality of the medium.

The first point becomes evident from the emulation latencies. Here, a runtime environment is added to the simulation and thus corresponding effects occur. The emulation latencies are closer than the simulated one to the real latencies. The second point is a feature lack of the simulation. Even though some attempts exist[7], currently no transmit rate adaption is available for our setup. So we have chosen the same solution as other researchers – setting a fixed rate corresponding to the monitored[8] one of the real scenario.

These results show that simulation and emulation estimate the network latency relatively course-grained. One can use the simulation result to determine the latencies in the real network with an error in the range of 10 % to 125 %.

Packet Delivery Ratios The packet delivery ratios were determined again by observing the tcpdump streams on the both end-sites of the communication. Table 2 shows the packet delivery ratios of real, simulated and emulated network over different number of hops.

Table 2: Packet delivery ratios

Hop count	Real	Simulation	Emulation
1	100 %	99.70 %	100 %
2	99.86 %	99.74 %	100 %
3	99.86 %	99.78 %	99.80 %
4	98.73 %	99.58 %	99.73 %

First to be noticed is the very high packet delivery ratio (almost 100%) in all scenarios. This fact shows that the used method for generating stable and symmetric links described in section 4 results in high quality multi-hop connections through the wireless medium.

Furthermore, the simulated, emulated, and real delivery ratios are very similar. Compared to the results presented in [LYN+04], with e.g. a delivery ratio of about 40 % for the real and 85 % for the simulated scenario (using AODV and two-ray ground based simulation), this is a significant improvement. The reason is that due to our stability mechanism all used links have a high delivery ratio, which also applies to the end-to-end ratio.

[7] The current development status for the NS-2 can be found at `http://yans.inria.fr/ns-2-80211/`.

[8] The real rate was determined using a network sniffer (Ethereal) and a wireless card in monitoring mode.

7 Related Work

The problem of asymmetry and weak links is ignored in many protocols. Nevertheless, many protocols exist that solve or try to solve this problems in some way. We cannot present all of them, so only some should be discussed that are representatives for a certain solution approach.

AODV is a typical example of a protocols that assumes stable symmetric links. However, its specification contains a solution for detecting *unidirectional links* (see [PBRC03] section 6.8): If a route response packet is not acknowledged, the whole links a added to a blacklist. The underlying assumption is that the acknowledge is lost if and only if the link is unidirectional. As discussed, this is property is not met in real wireless networks. A single lost packet can lead to the blacklisting of a otherwise very good link, while bad links still can be used for communication because the route discovery packets were not lost.

The Optimized Link-State Routing (OLSR, [CeA$^+$03]) contains a components for weak links called hysteresis and for detection of asymmetric links, which are similar to our approach. In opposite to our approach, unusable links are suppressed during link announcement, instead of suppressing packet reception. OLSR is known to create stable and reliable connections, which is caused by the mentioned mechanism.

Instead of omitting weak links, some protocols use link quality metrics for preferring good links in the route selection. The Expected Transmit Count (ETX, [CABM05]) metric calculates the expected number of retransmissions for a link $a - b$ from the receive probabilities as $\frac{1}{P(a \to b)P(b \to a)}$. The result is used in a weighted shortest path algorithm for selecting optimal paths, regarding the expected number of required packets. This approach is limited to protocols that can include link weights into path selection.

Instead of excluding unidirectional links from routing, approaches exist that use them for routing. The SRL layer [RCM02] creates virtual bidirectional links from unidirectional paths. This helps to communicate with stations that are not reachable with only bidirectional links. The approach seems usable for very sparse networks, but also adds a high communication overhead.

8 Conclusion and Outlook

In this paper we consider the problem of differences between simulation-based evaluation of wireless routing protocols and their experienced performance in real networks. Kotz et. al. identify the reason – unrealistic assumptions during the protocol design. The design problems are often not detected in the simulation-based evaluation, because the used simulation models support the assumptions by over-simplified propagation models. We discuss how this situation can be overcome by applying more realistic simulation that enables the occurrence of weak and asymmetric links. Furthermore, a simple algorithm is presented that allows to filter those problematic links. This extension can be transparently embedded between the MAC layer and existing routing protocols, thus improving their

connectivity.

A proof-of-concept implementation was done and the evaluation showed a good performance in real networks. Especially the packet delivery ratios are at a high level that allows the use of sensitive protocols like TCP. Our results showed that using this methods one can design communication protocols closer to the reality and still be able to accurately evaluate them using simulation.

In this paper we showed that the design and evaluation of communication protocols for wireless ad hoc networks can be done closer to reality than it is currently done in the research community. With this work we appeal the researchers in this area to increase the efforts for realistic design and evaluation of communication techniques.

Based on the methods presented in this paper we plan to extend the presented routing protocol to support bandwidth reservation QoS in ad hoc networks with adaptive data rate on the MAC layer. We plan to design and evaluate our methods having the real networks in mind and we believe we are on the right way.

References

[Bor05] Eleonora Borgia. Experimental Evaluation of Ad Hoc Routing Protocols. In *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 232–236, Washington, DC, USA, 2005. IEEE Computer Society.

[CABM05] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. a high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11:419–434, July 2005.

[CeA+03] T. Clausen, P. Jacquet (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhletha ler, A. Qayyum, and L.Viennot. Optimized Link State Routing Protocol (OLSR). RFC 3626, OCT 2003. Network Working Group.

[HM05] André Herms and Daniel Mahrenholz. Unified Development and Deployment of Network Protocols. In *Meshnets '05*, Budapest, Hungary, July 2005.

[JMH04] David B. Johnson, David A. Maltz, and Yih-Chun Hu. *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*. IETF MANET Working Group, July 2004. `http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt`.

[KNE03] David Kotz, Calvin Newport, and Chip Elliott. The mistaken axioms of wireless network research. Technical Report TR2003-467, Dartmouth College, July 2003.

[KNG+04] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 78–82, New York, NY, USA, 2004. ACM Press.

[KUF+04] K. Kuladinithi, A. Udugama, N.A. Fikouras, A. Timm-Giel, and C. Görg. Experimental Evaluation of AODV Implementations. In *Proceedings of CEWIT 2004*, New york, USA, Oct. 2004.

[LYN+04] Jason Liu, Yougu Yuan, David M. Nicol, Robert S. Gray, Calvin C. Newport, David Kotz, and Luiz Felipe Perrone. Simulation validation using direct execution of wireless Ad-Hoc routing protocols. In *PADS '04: Proceedings of the eighteenth workshop on Parallel and distributed simulation*, pages 7–16, New York, NY, USA, 2004. ACM Press.

[MI04] Daniel Mahrenholz and Svilen Ivanov. Real-Time Network Emulation with ns-2. In *8th IEEE International Symposium on Distributed Simulation and Real-Time Applications*, 8, pages 29–36. IEEE Computer Society, October 2004.

[Mul04] John P. Mullen. Efficient Models of Fine-Grain Variations in Signal Strength. In *OPNETWORK 04*, Washington DC, USA, September 2004.

[PBRC03] Charles E. Perkins, Elizabeth M. Belding-Royer, and Ian Chakeres. *RFC3561 - Ad hoc On-Demand Distance Vector (AODV) Routing*. Request for Comment, July 2003. `http://www.ietf.org/rfc/rfc3561.txt`.

[RCM02] V. Ramasubramanian, V. Chandra, and D. Mosse. Providing a bidirectional abstraction for unidirectional ad hoc networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1258– 1267, 2002.

[ZHKS04] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of Radio Irregularity on Wireless Sensor Networks. In *Proceedings of the 2004 International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 125–138, Boston, MA, June 2004.