

An Investigation of Link Quality Assessment for Mobile Multi-hop and Multi-rate Wireless Networks

Jinglong Zhou · Martin Jacobsson · Ertan Onur ·
Ignas Niemegeers

© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract Wireless ad hoc networks will be an important component in future communication systems. The performance of wireless ad hoc networks can be improved by link quality-aware applications. Wireless link quality is dynamic in nature, especially in mobile scenarios. Therefore, accurate and fast packet delivery ratio estimation is a prerequisite to good performance in mobile, multi-hop and multi-rate wireless ad hoc networks. In this paper, we propose a novel packet delivery ratio estimation method that improves the accuracy and responsiveness of the packet delivery ratio estimation. The proposed link quality estimation components are implemented in a IEEE 802.11b/g test-bed. The experiment results show that the accuracy of the packet delivery ratio estimation can improve up to 50% in mobile scenarios without introducing overhead. We also show the end-to-end performance impact of this improved estimation on route selection using different routing metrics and configurations. The measurement results show that our packet delivery ratio method leads to better route selection in the form of increased end-to-end throughput compared to traditional methods, which respond slowly to the link dynamics.

Keywords Link quality assessment and measurement ·
Mobile, multi-hop, multi-rate wireless networks · Link quality aware routing

J. Zhou (✉) · M. Jacobsson · E. Onur · I. Niemegeers
Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
e-mail: jinglong.zhou@ieee.org

M. Jacobsson
e-mail: m.jacobsson@ewi.tudelft.nl

E. Onur
e-mail: E.Onur@tudelft.nl

I. Niemegeers
e-mail: I.G.M.M.Niemegeers@tudelft.nl

1 Introduction

Due to its self-organizing capabilities, wireless mobile ad hoc networks (MANETs) offers a promising solution to wireless personal networking. However, the user demands for such systems are very high and expected to become even higher in the future. A lack of information about the quality of the wireless links affects the performance of MANETs, since higher layer application performance is very sensitive to link dynamics. Many higher layer functions, such as transmission rate adaptation, routing, QoS provisioning, gateway selection, video transmission optimization, require accurate link quality information. However, indoor wireless channels are dynamic in nature. Link quality may change dramatically in a short period due to small space, dynamic environments, and person movements. In mobile scenarios, the link dynamics get even more severe. Therefore, an accurate assessment of the link quality is required to increase the quality of communication.

Wireless link quality can be evaluated by packet delivery ratio. The problem of hello packet-based counting is that it reacts too slowly to link changes. If a link suddenly breaks or changes quality, maybe it take 4–5 s before the routing layer detects this and taking action. Hence, the packet delivery ratio measured by hello packets only is not accurate and fast enough for mobile scenarios. Furthermore, in IEEE 802.11 systems, the probe packets and data packets use broadcast and unicast methods to transmit the packets, respectively. The hello packets usually are broadcast using low data rates and they are small, while the data packets usually use higher data rates and have larger packet sizes [1]. Hello packets can be as small as 40 Bytes and use 2 Mbps, while data packets can be up to typically 1,500 Bytes and use 54 Mbps in 802.11g. Although the MAC layer provides a retransmission mechanism for lost frames, a lossy channel still decreases the throughput and increases the end-to-end delay for the data packet transmission. In most current systems, the packet delivery ratio of hello packets is used to predict the data packet delivery ratio despite the difference between these two types of packets and this results in inaccurate estimation of the link quality.

In this paper, we propose a novel link quality assessment method for mobile IEEE 802.11 multi-hop multi-rate networks. Several packet delivery ratio estimation method from different layers are combined to achieve a better estimation. Further, we tune SampleRate [2], which is a widely used rate adaptation mechanism, to adjust the transmission rate faster. Then different combinations of packet delivery ratio and the current transmission rate are used to improve the route selection and thereby the end-to-end performance.

We have two major contributions in this paper. The first is a new packet delivery ratio estimation method, which can accurately estimate the packet delivery ratio for all IEEE 802.11b/g transmission rates. Signal strength is introduced in the estimator, to the best of our knowledge, for the first time combined with hello packet delivery ratio and used in a linearization method. Significant performance enhancements are achieved compared to traditional hello packet probing. Second, we implemented the cross-layer architecture in a real test-bed, in which accurate packet delivery ratio estimation is provided to the routing layer. Using our test-bed, we demonstrate that our proposed packet delivery ratio estimation leads to significantly better route decisions and higher end-to-end throughput compared to traditional methods using the same amount of overhead in a mobile environment.

The paper is organized as follows. The related work is discussed in Sect. 2. We present our packet delivery ratio estimation mechanisms in Sect. 3 in detail. Section 4 introduces our test-bed and Sect. 5 presents the experimental results. The conclusion is in Sect. 6.

Table 1 Test-bed configuration parameters

	Values
<i>SampleRate</i>	
Smoothing factor	0.65
Information store time	2 s
Minimum rate switch time	1 s
<i>OLSRD</i>	
Topology control message interval	1 s
Topology control message valid interval	3 s
<i>Ppand</i>	
Hello packet interval	1 s
α for hello packet delivery ratio (R_H)	0.2
α for hello packet signal strength (S_H)	0.2
α for data packet delivery ratio (R_D)	0.5
Background traffic	10 UDP packets/s
Data packet size (MAC payload only)	1,500 Bytes

2 Related Work

2.1 Link Quality Estimation

Most ad hoc networks rely on hello packets. Usually, hello packets are small periodically broadcast packets. Neighboring nodes know the periodicity of those packets and can count the number of successfully received hello packets over a given time period. By dividing the number of received hello packets by the number of transmitted hello packets, a node can estimate the packet delivery ratio of a link.

Essentially the same, but a slight alternative is to use a EWMA filter [3,4] to calculate hello packet delivery ratio:

$$E_t = \alpha X_t + (1 - \alpha)E_{t-1}, \tag{1}$$

where $0 \leq \alpha \leq 1$, E_t is the estimated hello packet delivery ratio at time t , $X_t = 1$ represents a correctly received hello packet at time t , $X_t = 0$ means an incorrectly received probing packet at time t , and α is the smoothing factor. This method requires only one hello packet sent per time interval, thus it is fast and simple. The hello packet transmission interval is known by all nodes [3]. Therefore, the delivery ratio is measured at the receiver side. The α in Eq. 1 is to smooth the estimate. A larger α results in the estimation reacting faster to link dynamics. However, at the same time, the estimate becomes more jittery since it changes according to temporal and random events as well as using less historical values. The α value used in the experiments in this paper is deduced by our prior experiments and presented in Table 1. Since hello packets are used by a lot of previous work, we will call it the *traditional method* in the rest of this paper and we will use R_H to denote this estimate.

Previous work tried to estimate the packet delivery ratio with two methods: direct measurement or indirect estimation. Direct measurement means proactively sending hello packets to realize neighbor discovery and to measure the number of received hello packets in a certain period of time to estimate the packet delivery ratio. To overcome some of the problems with

hello packets, Kim et al. [5] proposed to use unicast instead of broadcast packets to probe the links and for the other nodes to overhear the pass-by traffic. When unicast packets are used, the data rate difference may be alleviated but the connectivity lessens. This is because the communication range for high data rates is shorter and the periodic probe packets are still too infrequent. Recent research [6] started to looking at short term link quality variance based on packet counting, still, the problem of different packet types exists.

Since direct measurements have the problem of slow reaction, some papers propose an indirect method instead. The most common approach is to use signal strength from the received packets as a link quality indicator. Chin et al. [7] carried out research that directly uses the signal strength to find the optimal gateway. In their design, signal strength from four different gateways was measured in a mobile node and the mobile node then chose the gateway with the best signal to be its only gateway. It is good to use signal strength as a key parameter for gateway selection since it is fast. However, the authors directly use the signal strength value to select the link without considering the data rate difference for a certain link. The 6-hop project [8] implemented a test-bed using signal strength. Based on received signal strength, they divided the link into good, medium, and bad with the costs 1, 3, and 5. Then they used these link metrics to compute the lowest cost routes. In their experiment results, they showed some system performance enhancement. The latest work we can find which uses signal strength is [9] and [10], however, the link quality information is not used in routing experiment.

2.2 Link Quality Based Routing

A direct application of wireless link quality is found in routing. Hop count based routing may produce routes with poor links, since using long links generally implies fewer hops but longer links also generally means poorer link qualities. Therefore, some prior proposals suggest minimizing the number of transmissions needed to reach the destination, such as the expected transmission count (ETX) proposed by De Couto et al. [3]. ETX is proposed in [3] as a routing metric that leads to a routing path requiring a minimum amount of transmissions, including retransmissions. Having the packet delivery ratio from both sides of the link, ETX becomes

$$ETX = \frac{1}{P_f \cdot P_r}, \quad (2)$$

where P_f and P_r are forward and reverse packet delivery ratios of the link as estimated by the packet delivery ratio estimation method, respectively. However, ETX needs the packet delivery ratio for each link and to acquire that, the authors used simple hello packets. Because the limitations of using hello packets, ETX will benefit from an accurate packet delivery estimation as well as rate adaptation. The protocol works best in small networks, such as clusters, where almost all nodes are within interference range. According to experiments conducted in the ns-2 simulator, ETX performs much better than shortest path routing in such networks.

In [11], the authors proposed to use what they call medium time metric (MTM) which uses the link data rate as metric. The authors claim that this can select more optimal throughput paths and avoid long unreliable links. Simulation results show that MTM outperforms the shortest path algorithm. To be able to consider both packet delivery ratio and data rate of a link in the route computation, In [4], the authors combined both ETX and MTM and defined the expected transmission time (ETT) metric. It uses both the data rate and the link quality in one metric tries to find the route with the minimum end-to-end medium occupying time. The ETT routing metric in [4] is computed as:

$$ETT = ETX \cdot \frac{L}{B}, \quad (3)$$

where L and B are the packet size and the current transmission rate, respectively. In our implementation, we use the packet delivery ratio information estimated at lower layers and feed it to the routing layer. Experiments were done with real wireless channels which demonstrated the suitability of this metric. Microsoft [12] further improved this routing metric for use in multi-channel environments and defined the Weighted Cumulative ETT (WCETT). It uses cumulative ETT as a basic metric but assigns different weights to tune channel diversity. They showed that this algorithm further outperforms ETX in a two channel environment. Further, interference aware route selection is proposed in [13].

For all these works, accurate packet delivery ratio estimation and smart data rate selection is a prerequisite. Moreover, most of the previous works consider only stationary networks. However, wireless ad hoc networks are mobile and many mobile scenarios are envisaged. Examples include people roaming in a building with all kinds of personal wireless devices and other fixed devices in the building. In such scenarios, a packet delivery ratio estimation method still must be fast enough and detect link changes and failures in the mobile environment in due time. Meanwhile, data rate selection, which is also based on packet delivery ratio estimation, should also respond fast to link dynamics. Previous works share one common, that is the link quality estimation used in the ETX or ETT formulas still uses the traditional hello packet-based method. If we use our method, which is more accurate, the route selection will be more accurate and faster, which can trade in higher throughput. The improvement of throughput in routing is investigated in Sect. 5.5.

3 Proposed Link Quality Assessment

Various information sources can be used to estimate link quality, such as counting received hello packets, observing the amount of MAC-level retransmissions, or using received packet signal strength [14].

Received signal strength is another important link quality estimation source. Previous work claims this source is inaccurate due to its fluctuation. However, it has the advantage that its value is not affected by different data rates and packet sizes. The signal strength can be obtained from any received packet. Usually the signal strength is read only from the hello packets [8], since the hello packets are transmitted at fixed intervals.

In this paper, we combine the hello packet delivery ratio, R_H and the received hello packet signal strength, S_H to estimate the packet delivery ratio, \hat{R} and compare it with the actual data packet delivery ratio, R_D . The measurements of R_H , S_H , and R_D are described in Sect. 4.2.2.

Each packet delivery estimation source has its own advantage. For example, R_H is simple and stable, while S_H does not need to inject new packets into the network, and hence, is costless. S_H is fast, does not depend on packet type and data rate and can be used with other methods to improve the accuracy and speed. R_D is the most accurate one. However, in real networks, there is not always data traffic on every potential link. Therefore, R_D is not always available.

To leverage the advantages of all measures, we propose the following method to estimate the packet delivery ratio based on different situations:

1. When there is enough data traffic on the link, use the feedback about the MAC-level retransmissions, R_D ,

2. When a node first joins the network, use S_H to estimate the packet delivery ratio, since it is instantaneous,
3. Otherwise, use both R_H and S_H to adapt faster to link dynamics and compensate the differences of transmission rates and packet sizes between hello and data packets.

For method 3, we propose a packet delivery ratio estimation as follows:

$$\widehat{R} = C \left(1 - \frac{S_H}{S_{min}} \right) R_H, \tag{4}$$

where C is a constant multiplier and S_H is normalized with the minimum signal strength level, $S_{min} = -95$ dBm. Two extra rules to this formula are: (1) If the resulting \widehat{R} is larger than 1, then the packet delivery ratio is estimated to be 1. (2) If the signal strength is larger than -50 dBm, the \widehat{R} is also estimated to be 1, no matter what the R_H value is. In this paper, this formula with the two extra rules is called the *proposed method*.

The multiplier C is a constant value that may depend on different conditions, such as transmission rate and packet size. Moreover, the C value can also be adjusted when there is huge amount of traffic over the link. A machine learning technique can be used to decide when the nodes need to adjust each C value towards the neighbors. To find the proper value for C for our experiment environment, we did a set of measurements in a real test-bed, which are presented in Sect. 5.4.

Some work [9] uses signal to noise ratio (SNR) to estimate the channel. If the SNR is used instead of signal strength, there will be a different C value.¹ In our experiment observations, the noise level is constant for most of the time, the maximum difference in an office environment is 1 or 2 dbm. Hence, using SNR or signal strength directly will achieve the same result. Therefore, we may just as well use the SNR value in the place of S_H in our method. We can directly change the formula as follows:

$$\widehat{R}_{SNR} = C \cdot SNR_{current} \cdot R_H, \tag{5}$$

$$\widehat{R} = \min(1, \widehat{R}_{SNR}); \tag{6}$$

The C value in Eqs. 4 and 5 is actually a linearization of the correlation between the S_H , R_H and R_D or between SNR_H , R_H and R_D . We show this approximation for SNR in Fig. 1 with some different values of C . We can see that a C between 0.06 and 0.07 best approximates the measurements.

To get this result, we carried out an experiment in which we collected 1,800 SNR and PDR values during a communication process. In the experiments, the mobile node moves with a constant speed on a certain track back and forth. Node marked with R1 is a stationary node. Different tracks for the mobile node are identified when different data rates are employed as shown in Fig. 2. The start position for the mobile node is marked as R2 in Fig. 2. At each interval of one second, they nodes send a hello packet to each other. The R_H , S_H and R_D are computed by both of the nodes and stored. While the mobile node moves in one direction, 1,800 PDR values are collected and categorized based on the SNR values. For a certain SNR value, the cumulated PDR values are averaged.

The R_H , S_H and R_D may change within a second in a mobile environment, if we set the interval to be smaller (less than 1 second), accuracy may be further improved. However, this will definitely introduce more overhead in the measuring. Meanwhile, if the PDR is computed frequently, the estimated \widehat{R} can hardly be used in higher layers, since the rapid link

¹ When a node has neighbors, it can gather the information of the S_H , R_H and R_D via the initial communication with its neighbors or by overhearing the communication of its neighbors.

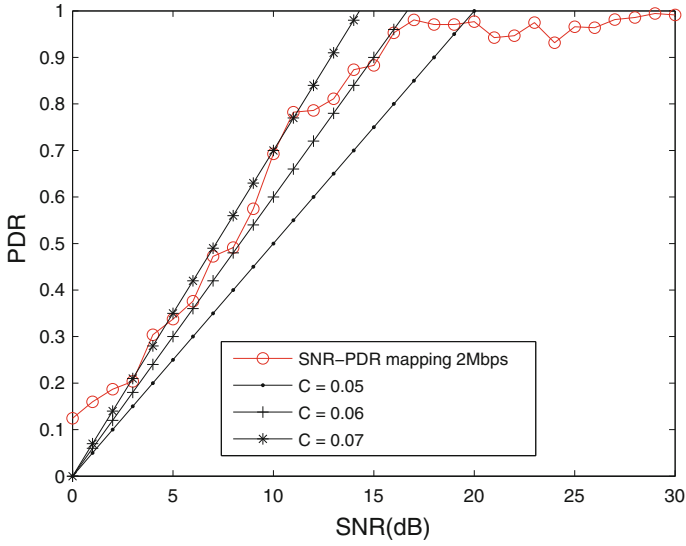


Fig. 1 The C value for SNR formula

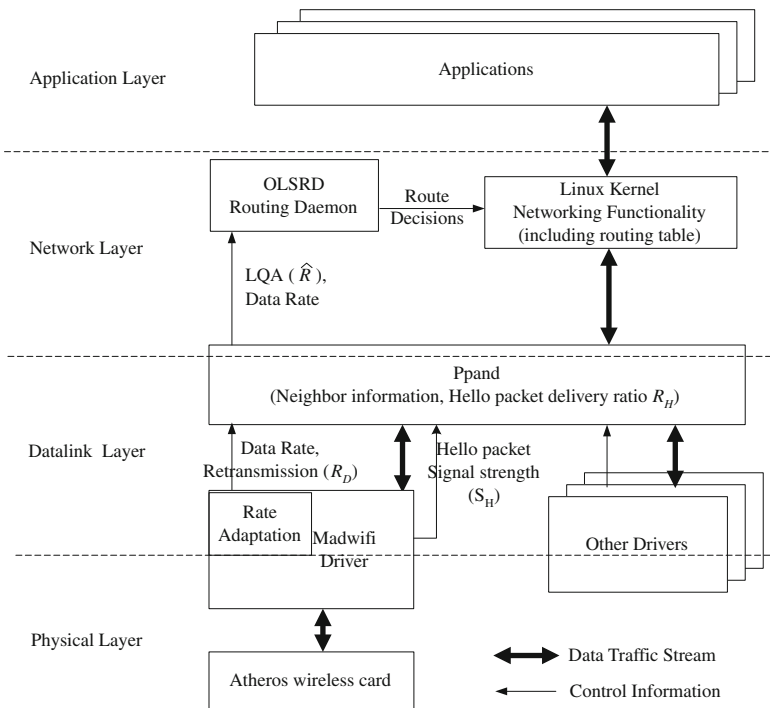
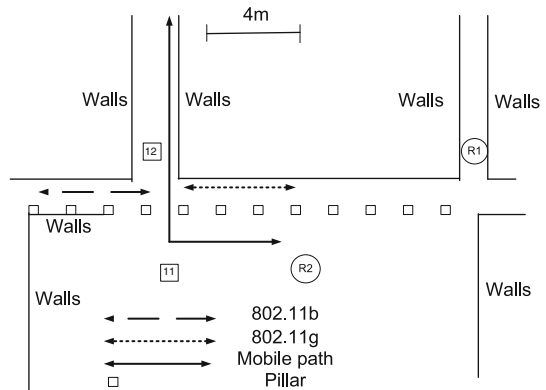


Fig. 2 Test-bed architecture

Fig. 3 Experiment topology

quality information updates will considerably increase the overhead of the routing protocols, both for information exchange and route discovery.

We plotted each SNR value and its average PDR value in Fig. 1. During the communication process, the R_H does not change much, it is mostly the SNR_H or S_H that actually determine the estimation. The C value is actually the slope of the approximation lines, which we used to describe the SNR_H or S_H 's correlation with R_D . Different input values, such as SNR_H or S_H , or different data rates, environments and other impacting factors will impact this C value. Hence, our aim is to find a C value that brings the estimation line the closest to the actual correlation.

4 Test-Bed Implementation

Previous work demonstrates that current simulation models can not accurately reflect the channel conditions [15]. Therefore, we built a test-bed to evaluate our proposed packet delivery ratio estimation method and the route selection comparison. The test-bed architecture is shown in Fig. 3.

4.1 Hardware

The test-bed hardware consisted of four laptops equipped with an IEEE 802.11b/g card (3Com's OfficeConnect 108Mb 11g PC Card), which has a wireless chip from Atheros supported by the open source *Madwifi* driver [16]. The laptops ran the Linux 2.6.20 kernel and *Madwifi* driver version 0.9.2.1. Packet signal strength values were obtained from the driver by Linux's IWSPY interface. To obtain information related to the retransmissions of data packets and the data rate used for each neighbor, we adapted the driver. For rate adaptation, we used the *SampleRate* rate adaptation algorithm [2]. The default power level (15 dBm) was used. All our experiment nodes have the same capabilities.

4.2 Software

4.2.1 Rate Adaptation Mechanism

SampleRate is proposed in [2]. It stores the packet delivery ratio information for all IEEE 802.11b/g data rates. Based on history and current packet delivery ratio information,

it calculates the packet delivery ratio for all data rates to estimate the link quality. A smoothing technique is used to decide how much current information is taken into the packet delivery ratio calculation. Then, it selects the data rate that will occupy minimum medium time including the expect number of retransmissions, for a successful packet transmission. A certain amount of packets are sent not on the current data rate to probe the channel's performance on other possible data rates for better rate selection. SampleRate is implemented in a widely known IEEE 802.11 open source driver, namely Madwifi [16], which is used in our test-bed.

The problem of using SampleRate in our scenario is that some links do not carry any data traffic. Due to this, the rate selection is updated very slowly. Therefore, in some scenarios, we introduced background traffic between each node to allow the rate adaptation mechanism to react to the link changes faster.

4.2.2 Ppand

Ppand is the software designed to maintain the neighbor list and combine all the cross-layer information. As can be seen in Fig. 3, it sits in between the network interfaces and the rest of the networking stack as a layer 2.5 implementation. Its purpose is to discover neighbors and monitor the packet delivery ratio of the links to those neighbors using the packet estimation methods. Ppand generates and processes hello messages. To make sure that data traffic does not delay hello packets, we used a priority queue on each of the interfaces that gives hello packets the highest priority. Since hello packets are small (40 Bytes) and transmitted only once per second, the impact on the data packets is negligible. It stores and processes all cross-layer data, then forwards the information to the relevant layers. The Madwifi driver plays the role of physical and link layer, so the basic physical channel information such as signal strength and the link layer information including the available link types, number of transmitted and retransmitted packets with each neighbor is forwarded to Ppand. The packet delivery ratio estimation method estimates the link quality and Ppand forwards the packet delivery ratio as well as current data rate information to the routing layer. If necessary, Ppand also forwards the information to other layers for other cross-layer optimization. Ppand communicate with physical and network layer and do not have interaction with transport and application layer in this paper.

Every received hello packet causes Ppand to read the latest received signal strength from the driver. A lost hello packet is assigned the minimum receivable signal strength of -95 dBm since this is also an indication of low packet delivery ratio. The EWMA formula was used to smooth the signal strength, the smoothing factor for signal strength is also presented in Table 1. To measure R_D for each link in both directions, all nodes produced connectionless data traffic among neighbors referred to as background traffic. We sent 50 UDP packets per second and each second we calculated the R_D based on the number of transmission attempts and acknowledgements from the Madwifi driver. We also used the EWMA formula to filter per second R_D to generate a continuous R_D . In some of our experiments, data traffic is specially generated to measure the R_D as benchmark.

4.2.3 Routing Protocol: OLSR

Optimized Link State Routing Protocol (OLSR) [17] is a proactive ad hoc routing protocol. We used the open source routing daemon implementation of Tonnesen et al. [18] (version 4.10.0). The protocol is an optimization of the classical link state algorithm. The key concept used in the protocol is that of multi-point relays (MPRs), which are selected nodes that

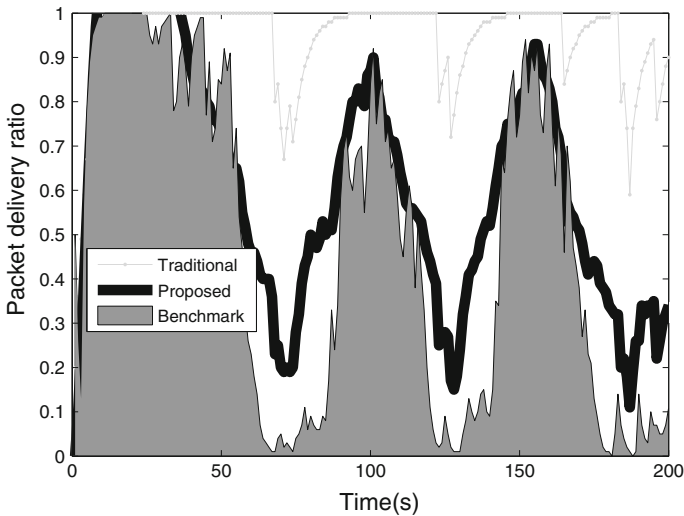


Fig. 4 The effectiveness of the estimation. (Rate is 36 Mbps, $C^* = 2.3$)

forward broadcast messages during the flooding process. OLSR uses Topology Control (TC) messages to propagate the link state information for non one-hop neighbors to realize optimal route calculation. A packet delivery ratio aware extension is made in the routing daemon that makes the route decisions based on link quality instead of number of hops.

We adapted the routing daemon and parameters (see Table 1) to get the link quality information from Ppand instead of using the daemon's own packet delivery ratio estimation method. OLSRD then uses its topology control messages to share the link quality information with the rest of the network. The topology control message interval is modified to 1 s instead of 5 s in our test-bed, this is due to the daemon is designed for stationary scenario in which link quality may not change much in this interval. However, in our mobile scenario, link quality is quite dynamic (see Fig. 4), so the default values reduce the improvement introduced by accurate link quality information. A smaller interval will cause very little overhead, but trade off for faster route selection decisions.

The proposed link quality assessment method estimates the link quality for any given link in a local fashion within the neighborhood of a node. Once the quality of a link is estimated locally, this information may be utilized by the routing protocols. Propagation of the link quality along with the topology information towards a large number of nodes is the concern of the employed routing protocol which is not in the scope of this paper.

5 Experimental Evaluation

To show the effectiveness of the proposed estimation method, We carried out real measurements. The experiment scenario is shown in Fig. 3 and is an indoor environment. Depending on the maximum range that can be achieved by a specific data rate, the experiments are grouped into two scenarios based on modulation differences. One of the nodes was stationary (R1 in Fig. 3) and the other was mobile (the mobile node's moving range is marked as arrow and other lines are the walls in the figure). The path is designed so that the mobile node moves from the point where it starts to drop packets until reaching the point where it can not

receive any data packets from the stationary node. That is, the sender tries to experience the packet delivery ratio variance from 0 to 1 and back again. The speed of the mobile node was one meter per second. Both nodes constantly sent hello and data packets at the same time. Samples were collected for a period of 300 s.

For performance comparison, we defined the average absolute percentage deviation D as:

$$D^{\text{method}} = \frac{100}{M} \sum_{i=1}^M |r_{d_i} - r_i^{\text{method}}|, \tag{7}$$

where M is the number of samples, \hat{r}_{d_i} is the sampled data packet delivery ratio, r_i^{method} is the estimated sample delivery ratio, where `method` is either the traditional hello packet estimation, R_H or, our proposed \hat{R} .

5.1 Link Quality Estimation Comparison

In this experiment, we used a fixed transmission rate of 36 Mbps. Based on the measurements, we could find the C constant that minimizes the error according to Eq. 7 as $C^* = 2.3$. As can be seen in Fig. 4, our proposed \hat{R} with $C^* = 2.3$ estimates R_D better than the traditional hello packet-based method R_H . R_H hardly predicts the link quality variance in this scenario. The errors, according to Eq. 7, are $D^{\hat{R}} = 18\%$ and $D^{R_H} = 50\%$ respectively. Hence, our method estimates the link quality much better than the traditional R_H .

The main reason that R_D is more accurate is that there is some correlation between the signal strength and R_D . When the channel changes very fast, signal strength adapts quickly. However, signal strength tends to fluctuate more. Therefore R_H is multiplied in Eq. 4 to generate more stable predictions.

5.2 Different Data Rates

One question that remains is how to determine whether C^* may change depending on the situation. The most obvious candidate is different transmissions rates. In this section, we analyze how different transmissions rates affect C^* .

To analyze this, we carried out the same experiment, but with different transmission rates. The optimal multiplier value C^* for all twelve IEEE 802.11 transmission rates are shown in Fig. 5. We can see that when higher data rates are used, a smaller C^* produces more accurate estimates. The data rate does not affect the delivery ratio of hello packets because the hello packets use the lowest data rate and a fixed small packet size. For the same experimental conditions, the data packet delivery ratio decreases when the data rate is increased and C^* decreases. Notice that C^* converges to a value around 2.0, as the rate increases.

In Fig. 6, we plot the error made by the proposed method and traditional method. The improvement is very obvious for the high data rate (36, 48, 54 Mbps). This improvement decreases with the data rate, which is due to the smaller difference between data packets and hello packets. An interesting effect is that for the IEEE 802.11b data rates, the traditional method causes a much smaller error while our proposed method still outperform it.

5.3 Outdoor Experiment

To find out what the performance of our LQA mechanism is in a large-scale outdoor environment, we carried out experiment in such an outdoor test-bed [19]. The test-bed network was located in Napa County, California, on a southern peninsula of Lake Berryessa and consisted

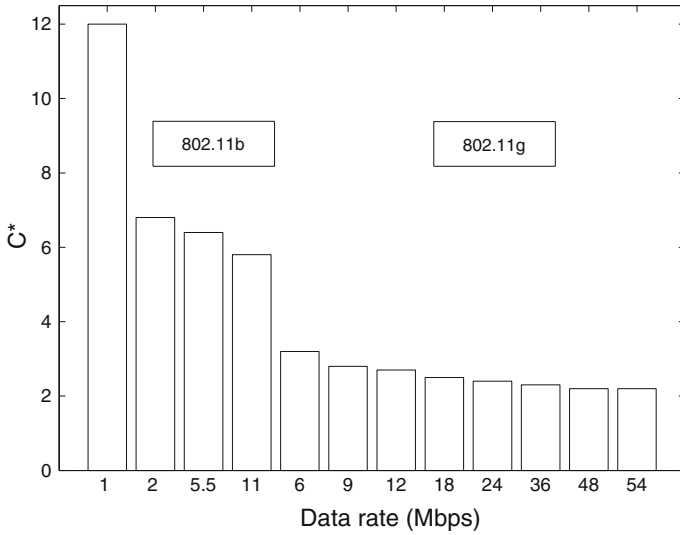


Fig. 5 The best constant value for 12 data rates

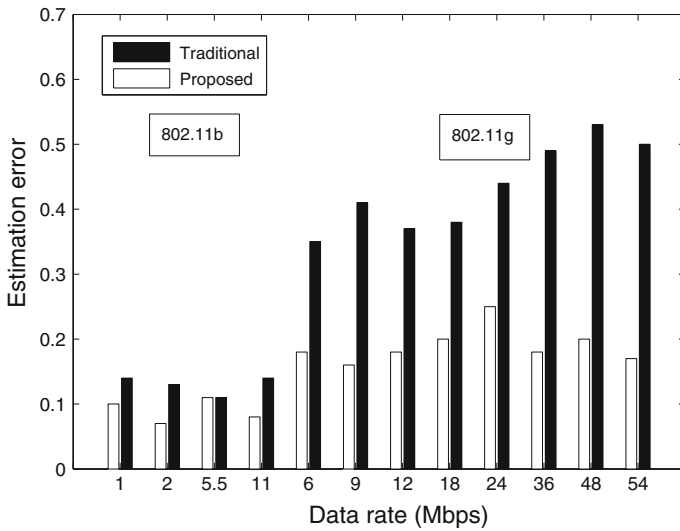


Fig. 6 The estimation error of the two methods

of hilly and densely forested terrain. There were 34 nodes at the time of testing, all placed at varying elevations due to the terrain spanning 2,000 acres. Distances between nodes ranged from a few hundred meters and up to a mile. Directional antennas were used for point-to-point links on top of the hills, while omni-directional antennas were used for lower elevations.

Due to that the distance was much larger than in our indoor environments, lower data rates had to be used. Therefore, we only used the four 802.11b data rates. We randomly selected several links to perform the experiment and the result is averaged into one error rate per data rate. The result is shown in Table 2, which shows that our proposed LQA method also is an improvement in outdoor environments and in large scale networks.

Table 2 Estimation error in an outdoor scenario

Data rate (Mbps)	1	2	5.5	11
Traditional	0.15	0.18	0.19	0.25
Proposed	0.08	0.10	0.11	0.20

5.4 Other Impacting Factors

In our experiments, we found that the transmission rate can impact the optimal C^* value. To analyze if other factors impact of the estimation method, we carried out experiments with different configurations, such as, packet sizes, traffic loads, speeds, and mobile patterns.

First, we conducted an experiment with four different packet sizes using the same data rate, 54 Mbps. Figure 7a shows that as the packet size increases, the optimal C^* value decreases. Larger packet sizes lead to larger packet loss ratios. For all packet sizes, the input parameters S_H and R_H are almost the same, since they are only affected by the channel conditions. Therefore, C^* is generally smaller for larger packet sizes. When the data packet size becomes smaller, the difference between hello and data packets become less and the estimate becomes less accurate.

Second, the data traffic between two testing nodes used in previous experiments has a rate of 50 UDP packets per second. Based on Fig. 7b, we can see that more traffic load leads to a larger C^* , because of larger collision probability between the data and hello packets. Meanwhile, the data packet delivery ratio is not affected when the traffic load increases. The right side of Eq. 4 becomes smaller and hence a larger C^* is required.

Third, another potential impacting factor is the speed of the nodes. We assume a constant speed of one meter per second for the moving node. In Fig. 7c, we show the effect of double node speed on the estimation. There is almost no impact of the speed on the optimal C value. The small difference in pedestrian speed will not lead to a variance in any of the component in Eq. 4.

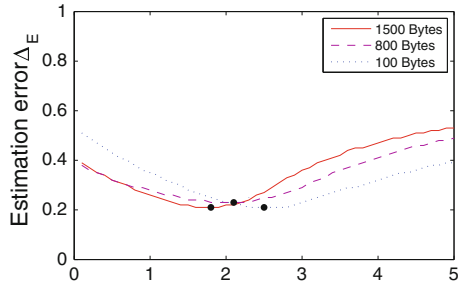
Fourth, in all previous experiments, we assumed a node moving with a constant speed and trajectory. In these experiments, we used the stop and wait scheme. That is, the node stops at each 4 m for a period of 5 s, than walks for another 5 s. Based on the results in Fig. 7d, we can see that mobility pattern does not have a large effect on Eq. 4 due to the same reason as we observed for speed.

These experiments show that the data rate, packet size, and traffic load have a large impact on C^* . Thus, for an optimal implementation of our packet delivery ratio estimation method in a real protocol stack, the effects of the data rate, mobility, and data traffic intensity should also be considered to achieve a better estimation. We can draw the conclusion that, these factors do have impact, however, the performance degradation caused by a sub-optimal C^* is not much. To obtain the optimal C^* for all scenarios, a machine-learning mechanism can be used to adapt all impacting factors and select the optimal C^* for a certain impacting factors combination.

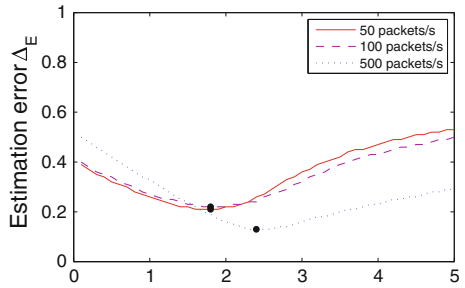
5.5 Routing Improvement

One real benefit of a more accurate LQA is the improved route selection. To compare the impact of different LQA methods as well as the rate adaptation response to the link dynamics, on the routing layer, we carried out experiments with the scenario shown in Fig. 2. To maintain a stable link, three stationary laptops were placed on tables one meter above the

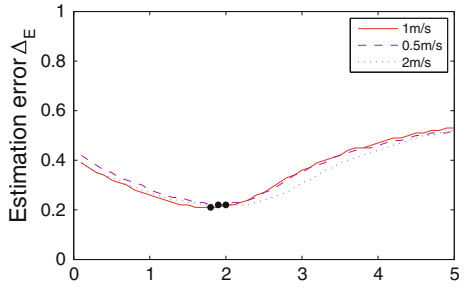
Fig. 7 Different impacting factors for the estimation method.
a Packet size. **b** Traffic load.
c Speed. **d** Mobility pattern



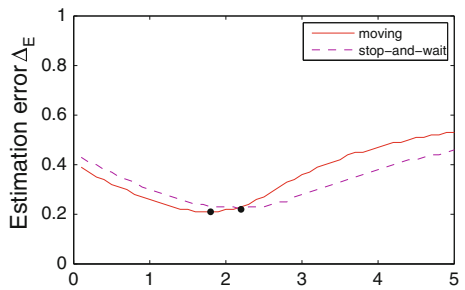
(a)



(b)



(c)



(d)

Table 3 Routing throughput performance comparison

Experiment	1	2	3
Routing metric	ETX	ETX	ETT
Rate (Mbps)	36	adaptive	adaptive
Rate adaptation	–	SampleRate	SampleRate
Proposed (Mbps)	4.87	4.40	1.23
Traditional (Mbps)	2.20	3.99	0.89
Improvement	121%	10%	38%

floor at positions R2, I1, I2. Two of them (I1, I2) were the intermediate nodes, while R2 was the receiver node. The mobile node is referred to as the sender. The sender followed the trajectory shown as an arrow in Fig. 2. It waited 30 s (warmup period to stabilize the SampleRate mechanism, then make a movement of 45 s following the trajectory. This process was repeated ten times with almost exact time control. We can assume each walk is independent. The sender kept sending UDP packets as fast as it could via the routing table provided by OLSRD. Experiments with three different configurations were carried out using the scenario described above. The three experiments used ETX (with one-rate), ETX (multi-rate), and ETT (multi-rate).

In the first experiment, all nodes used a fixed data rate of 36 Mbps. Two link quality metrics were used separately, our proposed \hat{R} and the traditional R_H . We plot the average throughput obtained in ten runs with a 95% confidence interval in Fig. 8. To analyze the route choice influenced by the two mechanisms, we also plot the number of hops taken to reach the receiver. At the beginning of the experiments, the two mechanisms made the same choice: the one-hop route. Thus, in the first 15 s, the performance is almost the same, so we do not plot this period and the performance comparison is only based on the period from 15 to 45 s for all experiments. From 25 to 35 s, the throughput achieved by our proposed \hat{R} is higher than traditional R_H , because our proposed packet delivery ratio estimation influenced the routing layer to use the interim nodes (multi-hop paths) while the traditional packet delivery ratio estimation continued to suggest the routing layer to use the low throughput one-hop route. Even though the receiver and sender did not have a line of sight and the link quality had almost dropped to zero, hello packets still had good delivery ratio and caused a slow reaction to the link dynamics. After the 35th second, the proposed packet delivery ratio estimation chose the three-hop route while the traditional packet delivery ratio estimation chose the two-hop route. Therefore, the throughput of the traditional packet delivery ratio estimation is still poor. We calculate the average throughput for these two methods in Table 3.

In the second experiment, we turned on the rate adaptation mechanism. Hence, the data rate between each link was determined by the SampleRate algorithm. We still used the ETX routing metric to decide the route. Since the procedure is the same as the first one, we do not plot the throughput with time for this scenario, but only provide the average throughput for different packet delivery ratio estimation methods in Table 3. As can be seen in Table 3, our proposed packet delivery ratio estimation still has better performance than the traditional packet delivery ratio estimation due to the selection of the better path. The three hop selection in later part of experiments can not have very high throughput since all the links at that time select the low data rate and performs much worse than the fixed 36 Mbps. It is also shown in previous work that rate adaptation does not outperform the fixed data rate in some scenarios [20]. This explains why the second experiment for the proposed method performs slightly

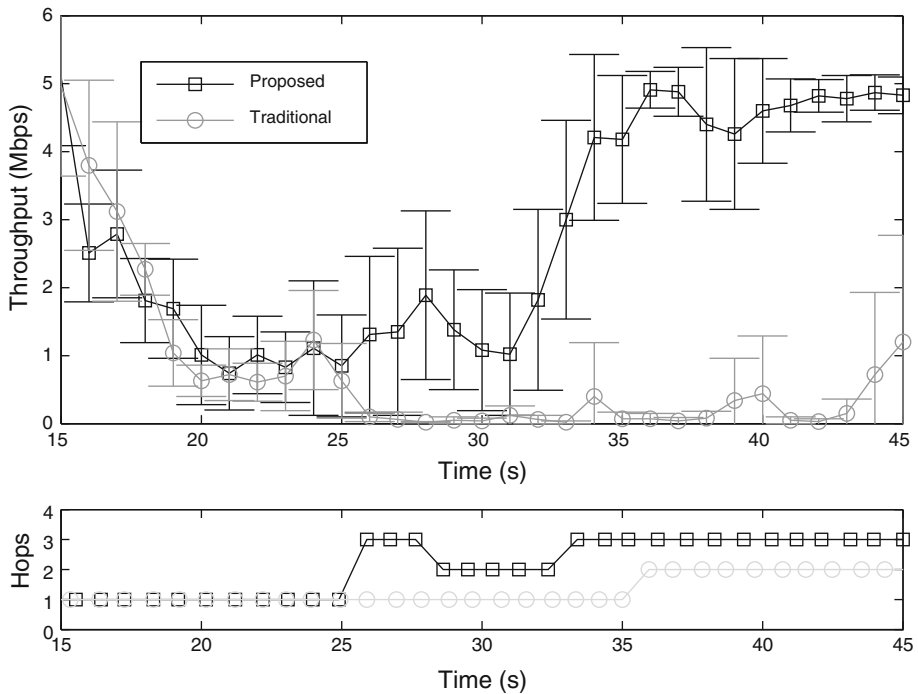


Fig. 8 The throughput with time (ETX, constant 36Mbps)

worse than the first experiment, which used a fixed data rate of 36Mbps. However, for the traditional method, one-hop was selected for a long time. Rate adaptation can select some lower data rate that is more suitable for long distance and non line of sight transmission, which results in higher end-to-end throughput. The result from experiment 1 and 2 indicate that, for the ETX metric, if the route selection is fast and proper, rate adaptation may degrade the end-to-end performance due to improper rate selection. On the contrary, if the route selection is not so suitable, rate adaptation can improve erroneous links by using the lower data rate in bad links, which increases the end-to-end throughput.

In the third experiment, the rate adaptation mechanism was also turned on, and both packet delivery ratio estimation and data rate were used in the route selection by using ETT. Due to SampleRate's problems in our scenario as described in Sect. 4.2.1, we sent background traffic between each nodes in this experiment to let the SampleRate update faster.² The result which is shown in Table 3 is similar to the second experiment and our proposed packet delivery ratio estimation still outperforms the traditional packet delivery ratio estimation. When using ETT, the slow reaction of rate adaptation also affects the route decision and that is why the throughput is lower than the other two experiments. Further, the background traffic also causes the multi-hop throughput (2 and 3 hop) to be lower.

Based on these experiments, it can be concluded that our proposed packet delivery ratio estimation achieved better performance than traditional packet delivery ratio estimation with a faster response to link dynamics. The rate adaptation that we used is not so good for mobile scenarios. In fact it does not always improve the end-to-end throughput compared to a fixed

² Based on trial experiment, without background traffic, the rate selection is very slow which causes the experiment results to be quite random.

data rate. Moreover, when using the selected rate in the ETT route selection metric, performance degraded even further. However, instead of looking for a better rate adaptation scheme, we propose to also use the LQA method for rate adaptation.

6 Conclusion

In this paper, we proposed a linearization based link quality estimation method, tuned a widely used rate adaptation mechanism, and used this for routing. We combined the available packet-delivery-ratio estimation sources to produce a new packet-delivery-ratio estimation method. The experiments in our test-bed showed the significance of an accurate packet delivery ratio estimation method in a mobile multi-rate IEEE 802.11 network and better performance in route selection. The proposed estimation method significantly improves the throughput performance without any additional overhead in the network.

Acknowledgments This work is supported by Freeband PNP2008, funded by the Dutch Ministry of Economic Affairs. Parts of this work are presented in the 6th IEEE Consumer Communications and Networking Conference, 2009, which was given the best student paper award. We also would like to acknowledge Daniel Wu and Professor Prasant Mohapatra at UC Davis for their precious help in carrying out the outdoor experiment and their input during the testing period.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Lundgren, H., Nordstrom, E., & Tschudin, C. (2002). The gray zone problem in IEEE 802.11b based ad hoc networks. In *ACM SIGMOBILE mobile computing and communications review, Vol. 6* (pp. 104–105). New York, NY, USA: ACM Press.
2. Bicket, J. (2005). *Bit-rate selection in wireless networks*. Master's thesis, Massachusetts Institute of Technology.
3. Couto, D., Aguayo, D., Bicket, J., & Morris, R. (2005). A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4), 419–434.
4. Bicket, J., Aguayo, D., Biswas, S., & Morris, R. (2005). Architecture and evaluation of an unplanned 802.11b mesh network. In *Proceedings of ACM Mobicom'05* (pp. 31–42). Cologne, Germany.
5. Kim, K. S. K. (2006). On accurate measurement of link quality in multi-hop wireless mesh networks. In *Proceedings of ACM Mobicom'06* (pp. 38–49). Los Angeles, CA, USA.
6. Alizai, M., Landsiedel, O., Wehrle, K., & Becher, A. (2008). Challenges in short-term wireless link quality estimation. In *Proceedings of the 7th GI/ITG Fachgespräch Wireless Sensor Networks (FGSN'08)*. Berlin, Germany.
7. Chin, K., Judge, J., Williams, A., & Kermode, R. (2002). Implementation experience with MANET routing protocols. *ACM SIGCOMM Computer Communications Review*, 32, 49–60.
8. Colin, S. et al., Implementation of enhanced WAF versions, Deliverable d3.1, IST 6-hop Project.
9. Verma, L., Kim, S., Choi, S., & Lee, S. (2008). Reliable, low overhead link quality estimation for 802.11 wireless mesh networks. In *Proceedings of 5th IEEE annual communications society conference on sensor, mesh and ad hoc communications and networks workshops* (pp. 1–6). SECON Workshops' 08.
10. Lei, J., Greenstein, L., & Liu, H. (2009). Mapping link SNRs of real-world wireless networks onto an indoor testbed. *IEEE Transactions on Wireless Communications*, 8(1), 157–165.
11. Awerbuch, B., Holmer, D., & Rubens, H. (2006). The medium time metric: High throughput route selection in multi-rate ad hoc wireless networks. *Mobile Networks and Applications*, 11(2), 253–266.
12. Draves, R., Padhye, J., & Zill, B. (2004). Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of ACM Mobicom'04* (pp. 114–128). Philadelphia, PA, USA.

13. Subramanian, A., Buddhikot, M., & Miller, S. (2006). Interference aware routing in multi-radio wireless mesh networks. In *Proceedings of IEEE Workshop on Wireless Mesh Networks (WiMesh)*. Reston, Virginia.
14. Zhou, J., Jacobsson, M., & Niemegeers, I. (2007). Cross layer design for enhanced quality routing in personal wireless networking. In *Proceedings of IEEE MobiQuitous'07*. Philadelphia, PA.
15. Newport, C., Kotz, D., Yuan, Y., Gray, R. S., Liu, J., & Elliott, C. (2007). Experimental evaluation of wireless simulation assumptions. *Simulation*, 83(9), 643–661.
16. <http://www.madwifi.org>, Madwifi driver.
17. RFC 3626 on Optimized Link State Routing Protocol (OLSR).
18. Tonnesen, A., Lopatic, T., Gredler, H., Petrovitsch, B., Kaplan, A., & Turke, S. -O. (2008). *Olsrd: An adhoc wireless mesh routing daemon*. <http://www.olsr.org>
19. Wu, D., Gupta, D., & Mohapatra, P. (2007). Quail ridge reserve wireless mesh network: Experiences, challenges and findings. In *Proceedings of IEEE/Create-Net TridentCom*. Orlando, FL, USA.
20. Wong, S., Yang, H., Lu, S., & Bharghavan, V. (2006). Robust rate adaptation for 802.11 wireless networks. In *Proceedings of ACM Mobicom'06* (pp. 146–157). Los Angeles, CA, USA.

Author Biographies



Jinglong Zhou received his B.S. degrees in Electrical Information Engineering department from HuaZhong University of Science and Technology, Wuhan, China, in 2004. He received his M.S. degree from Delft University of Technology, Delft, The Netherlands, in 2006, graduated with honors. In October 2010, he received his doctor degree from Delft University of Technology, Delft, The Netherlands. His research interests include Wireless mesh networks, Wireless sensor networks, cellular networks and Personal Networks. He has been evolved in the Dutch Telecommunication project, PNP2008 project and he won the best student paper award in the IEEE CCNC 2009 conference. He is a member of IEEE and he is currently working at Vodafone Netherlands.



Martin Jacobsson graduated in Computer Science from University of Linköping, Sweden in 2002. In June 2008, he received his doctor degree from Delft University of Technology, Delft, The Netherlands. During his doctoral studies and afterwards, he participated in several Dutch and European research projects, including IST MAGNET and Freeband PNP2008 on Personal Networks. I have, as the main author, written a book about the results that was published by Wiley & Sons in 2010. His research includes ad hoc and self-organizing wireless networking techniques, such as link quality assessment for unicast routing and network-wide broadcasting, but also combinations between ad hoc networks and infrastructure-based networks for personal networks.



Ertan Onur received the B.Sc. degree in computer engineering from Ege University, Izmir, Turkey in 1997, and the M.Sc. and Ph.D. degrees in computer engineering from Bogazici University, Istanbul, Turkey in 2001 and 2007, respectively. After the B.Sc. degree, he worked for LMS Durability Technologies GmbH, Kaiserslautern, Germany. During the M.Sc. and Ph.D. degrees, he worked as a project leader at Global Bilgi, Istanbul and as an R&D project manager at Argela Technologies, Istanbul. He developed and managed many commercial telecommunications applications, has a patent and published more than forty papers. Presently, he is an assistant professor at EEMCS, WMC, Delft University of Technology, the Netherlands. He is the editor/convenor of the Personal Networks Group of Ecma International Standardization Body. Dr. Onur's research interests are in the area of cooperative networks, personal networks, wireless and sensor networks. He is a member of IEEE.



Ignas Niemegeers got a degree in Electrical Engineering from the University of Gent, Belgium, in 1970. In 1972 he received a M.Sc., B.E. degree in Computer Engineering and in 1978 a Ph.D. degree from Purdue University in West Lafayette, Indiana, USA. From 1978 to 1981 he was a designer of packet switching networks at Bell Telephone Mfg. Co, Antwerp, Belgium. From 1981 to 2002 he was a professor at the Computer Science and the Electrical Engineering Faculties of the University of Twente, Enschede, The Netherlands. From 1995 to 2001 he was Scientific Director of the Centre for Telematics and Information Technology (CTIT) of the University of Twente, a multi-disciplinary research institute on ICT and applications. Since May 2002 he holds the chair Wireless and Mobile Communications at Delft University of Technology, where he is heading the Centre for Wireless and Personal Communication (CWPC) and the Telecommunications Department. He was involved in many European research projects, e.g., the EU projects MAGNET and MAGNET Beyond on personal networks, EUROPCOM

on UWB emergency networks and, eSENSE and CRUISE on sensor networks. He is a member of the Expert group of the European technology platform eMobility and IFIP TC-6 on Networking. His present research interests are 4G wireless infrastructures, future home networks, ad-hoc networks, personal networks, cognitive networks.