Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Semester Thesis

March 2007 – June 2007

Experimentation and evaluation of routing metrics for multi-hop ad-hoc wireless networks on an indoor wireless testbed

Author: Georgios Sotiropoulos

Professor: Bernhard Plattner

Advisor: Georgios Parissidis

List of figures

List of tables

Abstract

The main goal of this semester thesis is to evaluate the performance of routing metrics on an indoor wireless test-bed. At the first part of the thesis we built a stationary wireless test-bed (TikNet) of 20 commodity PCs distributed on the G-floor of the ETZ building at the ETH. The Tik-Net test-bed is a platform for wireless experimentation with real-world hardware that improves the understanding of wireless networks while circumventing conventional assumptions and possible misconceptions made by simulation tools with actual real-world recorded behavior.

Wireless testbed conclusions about higher layer mechanisms (like routing or transport mechanisms) imply also a concrete understanding of the wireless behavior. In this context we tried to identify and quantify through link level measurements the intensity of wireless link level phenomena, like the link asymmetry and the validity of “neighbor” abstraction. Our extended study shows that link asymmetries exist at the TikNet test-bed, whereas the “neighbor” abstraction could be roughly a valid assumption.

The performance evaluation of routing metrics on the test-bed complements the thesis. For the routing metric evaluation, we used the minimum hop-count metric and the ETX routing metric proposed by MIT. It turns out that ETX provides better connectivity but is only slightly better than minimum hop-count in terms of mean throughput for the specific experimental scenarios. We also proceeded to evaluate ETX in terms of mobility and prediction accuracy. This evaluation along with the wireless characterization of the testbed could provide a basis for understanding how routing metrics affect the performance of the routing protocols in practice and thus could lead to the design of new routing metrics that would make the mesh networking paradigm a viable realizable solution.

1. Introduction

Over the last decade wireless LAN technology such as 802.11 has been deployed at an explosive rate. Apart from the infrastructure mode networks that can now be found everywhere - from university campuses to airports, cafes and private homes - wireless ad-hoc networks have also drawn a lot of attention. There is a massive amount of activity in the research community to improve the performance of the wireless ad-hoc networks and develop new applications based on this technology, such as mesh networks. In particular, static multi-hop wireless ad-hoc networks, where users cooperate by forwarding each other's packets, have the big advantage of easy deployment and apart from providing connectivity between users they can also act as a gateway to the internet. The most representative example of this kind of networks is the Roofnet network deployed by MIT which provides connectivity to the area of Cambridge [1].

The majority of the wireless research over the past few years has been conducted with simulations. Although simulations have the obvious advantage of less effort when designing an experiment, control and repeatability, they are often not representative of the real world behavior, thus producing many times imprecise results [2]. This is mainly due to the simplifications made by simulation tools about the distributed and very complex nature of the wireless signal propagation, and the transmission and reception of the wireless signals. For example, initial work [3] has shown that the most common simulator (ns-2) produces results that vary significantly from real-world experimentation results. My motivation of building up a wireless testbed and using hardware-based experimentation lies exactly in the direction of trying to achieve the most physical layer realism and understand in depth the various phenomena in wireless networks. For example, my measurements along with already mentioned experiments of the research community [4, 5] show that wireless networks exhibit a variety of behaviors, such as link asymmetry, that are difficult to understand and are not recreated in current simulators.

There are mainly two goals that this project tries to achieve. The first goal is the construction of a functional indoor static multi-hop ad-hoc wireless testbed [6] and the characterization of its wireless links. The testbed can be a powerful tool in the hands of every researcher because it allows someone to bypass the limitations that simulators impose and test his ideas in practice, thus pushing research by the means of practical real world experimentation. However, in order to understand the results or the performance of the system that one designs, one must have complete (to the extend that this is possible for wireless networks) knowledge of the wireless behavior of the testbed. Therefore, it is of significant importance to characterize to a great extend the wireless behavior of the network through extensive link level measurements.

The second goal is the evaluation of the minimum hop-count metric and the ETX routing metric proposed by MIT [7]. I will try to recreate the results mentioned in [7] based on experimentation in TikNet wireless testbed and understand the advantages and mainly the disadvantages of them. This evaluation along with the wireless characterization of the testbed could provide a basis for understanding how routing metrics affect the performance of the routing protocols in practice and thus could lead to the design of new routing metrics that would make the mesh networking paradigm a viable realizable solution.

The remainder of this technical report is organized as follows. Section 2 describes the procedure that was followed and the hardware and software tools that were used in implementing the TikNet testbed. In Section 3 I present all the link level measurements and try to characterize the behavior of the wireless testbed from a physical and link level point of view. In Section 4 the different routing metrics and their performance evaluation are presented. Finally, in Section 5, I summarize and conclude about the results and propose possible future work.

2. TikNet testbed setup

In this section I present all the various components of the currently 20-node TikNet testbed - from the topology and the hardware till the software tools - that make this testbed an available operating platform for experimentation in wireless networking.

2.1 Hardware and topology

TikNet consists of 20 personal computers distributed at random spots throughout the G-floor of the ETZ building (Department of Information Technology and Electrical Engineering – ETH Zurich) covering a physical area of roughly 2250m^2. Figure 1 is a snapshot of the exact position of the 20 nodes. Each numbered circle in Figure 1 serves also as a correspondence of the IP alias of the node, which makes it remotely accessible through the ETH wired network (e.g. node 3 can be reached by the alias "tik-wifi3").

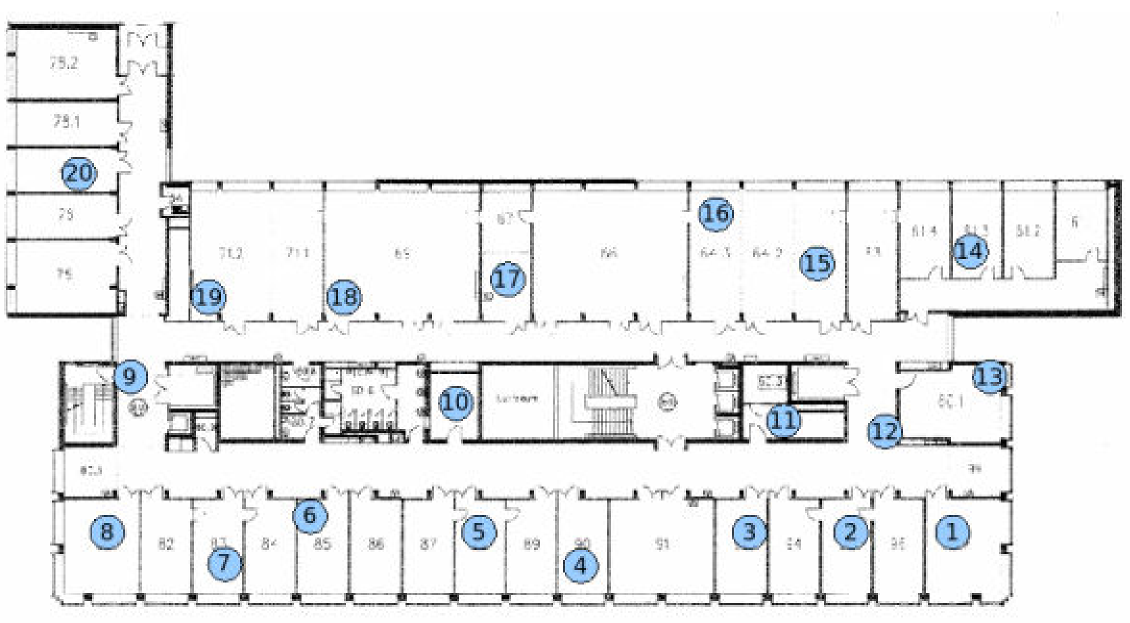

Figure 1 Topology of TikNet wireless testbed

The upper and lower parts of the floor plan consist of multiple office rooms separated by non-concrete walls. However, the middle part consists of very concrete and hard to penetrate walls (mostly concrete cement and metal), thus making the communication in the vertical axis more difficult than in the horizontal axis. In order to overcome any possible unintended partitioning of the network, extensive connectivity measurements were taken place (mostly with the use of the *fping* command) that lead to slight calibrations of the initial position of selected nodes. The final position of the 20 nodes, which is shown in Figure 1, assures no partitioning in the wireless network and provides full functionality to the user of the TikNet testbed.

The 20 PCs used can be roughly separated into two categories: i) Dell PCs with 2GHz processor and 512MB RAM memory, and ii) PCs with 866MHz processor and 512MB RAM memory. All of them are currently equipped with the 802.11b/g D-Link G-520 [8] wireless NIC based on the Atheros AR5212 chipset and an external omni-directional antenna and (unless stated otherwise) operating at 2.422GHz (802.11b channel 3) with the transmission power set to +15dBm (30mW).

2.2 Software

2.2.1 Linux and drivers

The operating system selected is *GNU/Debian Linux* with a Linux kernel version of 2.6.18-4. As already mentioned each node can be accessed remotely through the ETH wired network (e.g. using *ssh*) using its alias (e.g. ssh tik-wifi5.ethz.ch for accessing node 5). In the current implementation, node 4 (tik-wifi4.ethz.ch) plays the role of a “server” in the sense of being able to connect to all other nodes using ssh without the need for a password. This provides the user with the flexibility to execute every script that is used for a certain experimental setup in node 4, avoiding to connect to every node individually. Of course, this is something that should be changed in the near future, by assigning to another PC that is not part of the wireless testbed the role of the “server”. Also implementing management, monitoring and security tools for the TikNet testbed would provide more flexibility to the system and make the experimentation more user-friendly.

Because the wireless NIC installed in every node of the wireless testbed uses Atheros chipset, the decision of using MADWIFI (Multiband Atheros Driver for WiFi) (MADWIFI) was pretty straightforward. In particular, I used the *Madwifi-ng* (next generation) driver, which is a Linux driver for 802.11a/b/g universal NIC cards - Cardbus, PCI, or miniPCI - using Atheros chip sets. Madwifi drivers are vastly used by the research community because they are open-source and allow for versatile and flexible configuration of the wireless card. It is also widely documented (at least in the context of initial experimentation). For more information about Madwifi drivers one should check [9].

2.2.2 Packet capture

For dumping the wireless traffic on TikNet network I used extensively tcpdump [10]. Tcpdump can capture all the packets in the specified interface and allows for printing out the headers of the packets in a human-readable format. For further processing of the dump files I used wireshark [11] and its related utilities.

2.2.3 Click modular router

For routing purposes of the testbed I installed in every node the *Click Modular Router* software [12]. Click Router, presented in Eddie Kohler's Ph.D. thesis [13], is a general purpose software tool for implementing modular router configurations. A general description of the Click Modular Router software is first presented and then follows the specific router configuration I used for importing wireless routing functionality to the testbed.

The motivation and the idea behind Click Modular Router Project is to represent the packet flow through a network router as a sequence of packet processing modules (called elements) connected in a directed graph. The vertices of the directed graph represent the various elements. The edges of the graph represent the connections between elements and are possible paths for packet handoff. The functionality of the elements should be kept as simple as possible (e.g. queuing, decrease of TTL, traffic classification, interfacing with network devices). Designing a router simply means connecting different elements in a directed graph. Therefore, due to the fine-granularity of the elements and the simplicity of building a router configuration, a Click router configuration is modular, flexible, and easy to extend.

The element graph is then implemented in C++ using classes as elements and function calls as packet transfer mechanisms, combined with clever macros. Click routers run as a userlevel program or as a Linux kernel module. An evaluation of Click's performance as a kernel module in IP routing shows that it can achieve very fast operation [14]. Apart from userlevel program or kernel module, Click router is implemented for ns-2. This turns out to be very important for research since the transition from simulations to real world testbed experimentation can happen very fast: almost nothing has to be changed in the router configuration file.

Based on Click router as a general software tool described above and apart from the Roofnet project, MIT also developed router configurations that support various routing protocols for wireless ad-hoc networking. This was done as a part of the Grid project [15] and it is what I also use to provide wireless ad-hoc routing functionality to the testbed.

The *Grid* software implements the DSDV (Highly dynamic Destination-Sequenced Distance-Vector routing) and DSR (Dynamic Source Routing) protocols for routing in ad hoc mobile wireless networks. DSDV is described in [16] and is a pro-active routing protocol. That means that entire network topology is known to all nodes and maintained in routing tables with the aid of periodic routing advertisement messages sent by all nodes. One the other hand, DSR belongs to the category of reactive routing protocols; it is described in [17]. In a reactive protocol a route is discovered only on-demand. The Grid DSR implementation follows revision 9 of the IETF Internet-Draft specification, following the requirements for networks which require bidirectional links to send unicast data. Both the DSDV and DSR implementations include modifications for better performance, which also allow the protocols to work properly with routing metrics other than minimum hop-count; these changes are described in [7].

The Grid code, as a part of the Click Router software is a set of Click elements that can be put together in various ways to run DSDV, DSR. All the routing protocol state (e.g.

routing tables, link tables, ARP caches, etc.) is maintained in Click elements, independently of the OS. Protocol state is accessed using through Click.

Because Click can run at userlevel on many different operating systems and machine architectures, Grid also runs on many operating systems and machines. In addition, Grid can run at kernel level in Linux, using the Click Linux module. Both userlevel and kernel module are practically the same protocol code. The only things that change are the Grid interfaces to the network devices and to the operating system [15].

Elaborate details of setting up the Grid Click configuration for a wireless node are provided in [15].

3. Link characterization of the wireless testbed

3.1 Motivation

As already mentioned, experimentation using a wireless testbed has some clear advantages against simulation-based experimentations. Due to the fact that simulations often rely on simplifying rules about the wireless signal propagation, which have not been validated using today's wireless radios, real hardware-based experimentation fills in the gap between conventional assumptions and possible misconceptions (depending strongly on the accuracy of the models used and the protocol's code implementation) and real world actual recorded behavior. This however does not come without a cost. Setting up a testbed experiment involves taking into account a lot of parameters and it is only by the means of a concrete and properly structured experimentation methodology that someone could understand and interpret in a correct way the measured results.

It is exactly in this general context that understanding the wireless behavior of TikNet testbed becomes essential before moving on to analyzing results about higher layers mechanisms (such as routing and transport protocols or routing metrics – presented in Section 4). Therefore, I have conducted a set of experiments and processed the collected data trying to accurately understand the link-level behavior of the TikNet testbed. The results and the methodology used are presented in the following paragraphs.

3.2 Experimental setup – methodology

The technical characteristics of the wireless NICs used in the testbed can be found in section 2.1. These cards operate at 802.11b mode and can be configured to transmit at 1, 2, 5.5, or 11Mbit/s; throughput the entire set of experiments in this section rate adaptation mechanism was disabled.

It is mentioned that the wireless cards manufacturers have often not implemented in full detail the ad-hoc operation mode. This can lead to functionality problems after some time of operation, such as network partitioning. In order to avoid unintended partitioning of the ad-hoc network and profiting from the experience of previous testbed experimentation I set the cards in "Ah-demo" mode. "Ah-demo" mode is a simplified version of the ad-hoc mode, since absolutely no beacons or other control packets are sent.

It is the same mode as "pseudo IBSS" mode that is supported by other 802.11 chipset such as Prism 2.5 [4].

The data presented in section 3 is derived from two main sets of experiments. In the first set of experiments, each node in turn sends 3000 broadcast packets uniformly transmitted within a time interval of 30secs, while the rest of the nodes are passively listen. The same experiment is repeated but for different possible configurations (transmission power: 0, 15dBm – packet size: 180, 1080bytes – transmission rate: 1, 2, 5.5, 11Mbps). The experiment uses 802.11 broadcast packets because they involve no link-level acknowledgements or retransmissions and effectively capture the actual packet loss rate of a link between two nodes. The second set of experiments simply consists of all the nodes been set in monitor mode capturing all the packets that are in the air for a total period of 5min.

3.3 Results

As already mentioned, in order to measure the loss ratio for all the possible direct links (19x20 = 380 in total), each node took a turn transmitting 3000 broadcast packets uniformly spread in an interval of 30 seconds. Within these 30 seconds every other node recorded the number of packets received. Each experiment lasted 600 seconds and provided us with the pair-wise link delivery ratios in the following simple way; the delivery ratio from node A to node B is calculated by dividing the packets B received from A to the total number of packets that A sent. The corresponding results are presented in the figures included in section 3.3.1[1]. Experiments were conducted for different wireless configurations.

We must also note that using broadcast packets allows us to capture the actual packet losses because 802.11b broadcast packets do not involve acknowledgements or retransmissions. On the other hand, if 802.11b unicast packets are lost, the MAC layer retransmits them. The retransmissions hide from the upper layers a great deal of the actual packet losses (according to the maximum number of retransmission allowed) making the wireless communication possible under a perceived lower loss ratio.

3.3.1 Link asymmetry

In figure 2 and 3 (where small packets with 100-bytes payload were used) each vertical bar corresponds to both ways of a direct radio link between a pair of nodes; the two ends of each vertical line represent the two delivery ratios of each direct link (one in each direction). The above figures reveal a fundamental characteristic of real-world wireless networks; the asymmetry between the two directions of a single direct link.

---

[1] Figures shown only for 1Mbit/s and 11Mbit/s. Experiments were conducted also for 2Mbit/s and 5,5Mbit/s but are not shown here. In order to check the validity of these results some of the experiments were repeated. It was found that due to high dynamics of the wireless channel the re-measurements were not strongly correlated with the initial measurements when one looked at specific node-pairs and their corresponding link delivery ratios. Although link asymmetry was well pronounced for every iteration of the experiment, a more careful evaluation based on statistical analysis would be beneficial. Finally, all the experiments were conducted throughout the second half of the semester during late night hours in order to avoid any dramatic changes due to interference or channel change.

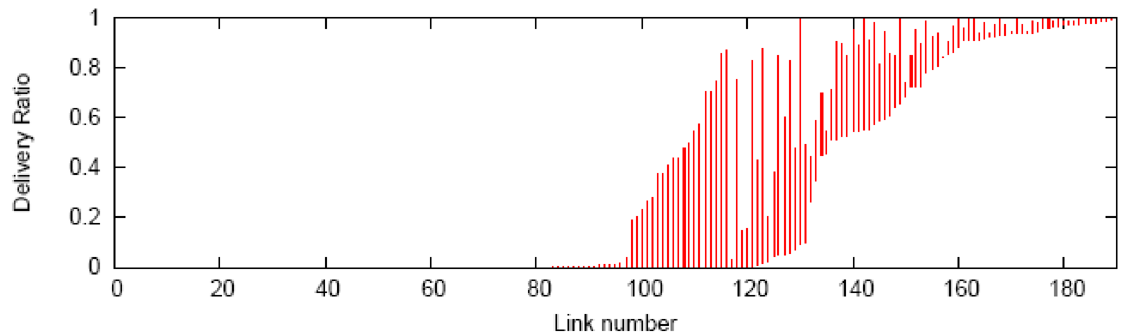

Figure 2 Pair-wise delivery ratios at 30mW, 1Mbit/s, small packet

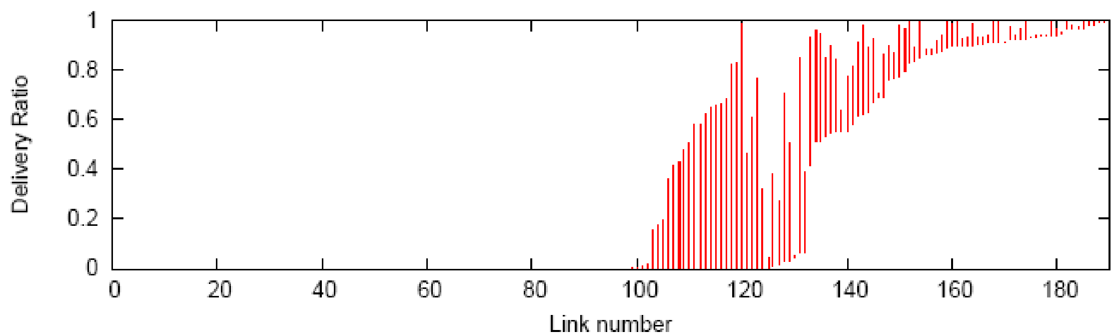

Figure 3 Pair-wise delivery ratios at 30mW, 11Mbit/s, small packet

Of the 190 node pairs in figure 2 (1Mbit/s), there are 121 pairs out of 380 that delivered packets in at least one direction. Of those links, 37 are asymmetric with forward and reverse delivery ratios that differ by at least 25%. That is around 30% of the active links that have quite emphasized delivery ratio asymmetry. The respective results for figure 3 (11Mbit/s) are 111 out of 380 active links and 35 (around 31%) asymmetric links. We can see that the asymmetry remains around the same percentage as for 1Mbit/s. This is something that we expected since for the only the payload is transmitted with 11Mbit/s (the MAC header is transmitted with 1Mbit/s).

We repeated the experiments using large packet size (1000-bytes payload) and lower transmit power (1mW). The results are presented in the figures 4, 5. For large packet size we observed that the asymmetric links are fewer percentage-wise compared to the previous results with small packet size (around 26% for figure 4 and 17% for figure 5). It seems that for large packet size – where the 11Mbit/s could make a substantial difference – the asymmetry of the links is less pronounced. Also the transition from low values of delivery ratios to high values of delivery ratios is sharper.

Commenting on the results for 1mW (presented in figures 6, 7) we can see clearly that the number of node pairs that have packets delivered in at least one direction has dropped. That causes the connectivity graph to be sparse and increases the average number of hops that a packet needs to transverse in the network. Finally, it is also interesting to note that the highest percentage value of asymmetric links (around 33%) occurs for low transmission power (1mW) and small packet size.

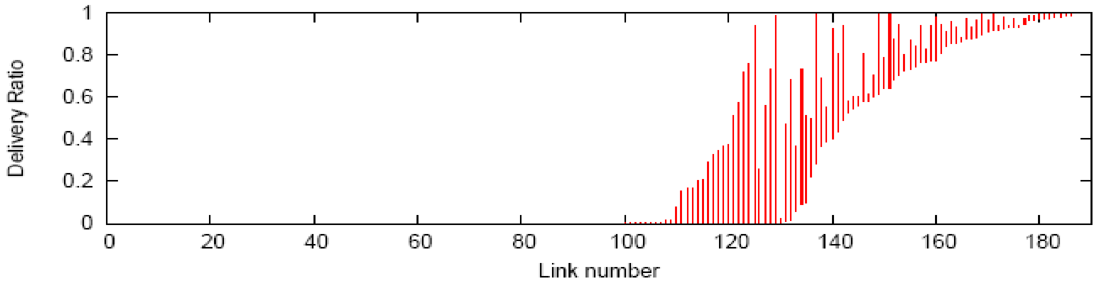

Figure 4 Pair-wise delivery ratios at 30mW, 1Mbit/s, large packet

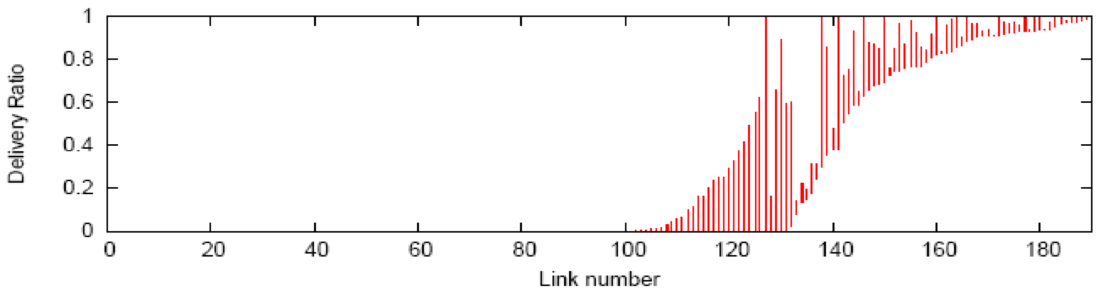

Figure 5 Pair-wise delivery ratios at 30mW, 11Mbit/s, large packet

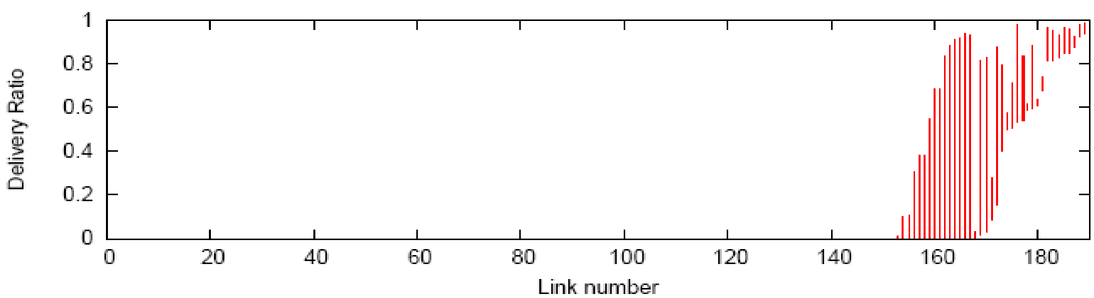

Figure 6 Pair-wise delivery ratios at 1mW, 1Mbit/s, small packet

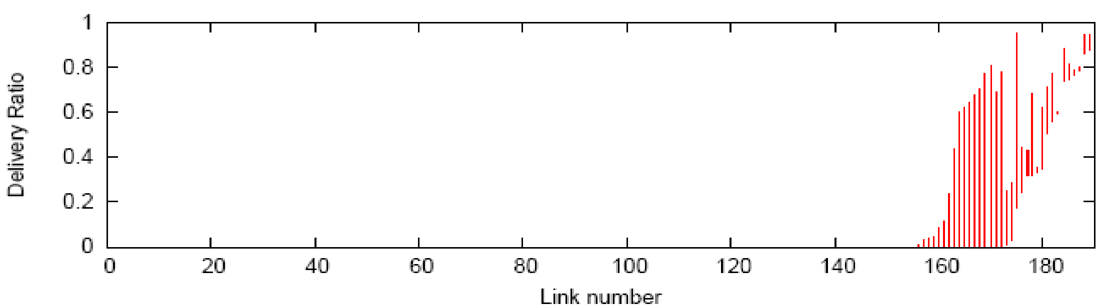

Figure 7 Pair-wise delivery ratios at 1mW, 1Mbit/s, large packet

Link asymmetry in wireless networks has been also reported from several independent research groups ([4], [5]). Since 802.11 uses link-level ACKs to confirm delivery, both directions of a link must work well in order to avoid retransmissions. Therefore, link asymmetry becomes an essential design factor for routing protocols of 802.11 wireless networks. Focusing on the results presented in figures 2-7, we can easily distinguish links with extreme asymmetry in the directions of the link, even zero delivery ratios in one direction and near 80% in the other direction. In such an extreme case and keeping in mind that 802.11 uses MAC-level ACKs, direct communication between two nodes becomes nearly impossible.

Although asymmetric signal propagation is impossible due to the fundamental reciprocity theorem (i.e. if the role of the transmitter and the receiver changes, the instantaneous signal transfer function between the two remains the same), there are a variety of factors that can contribute to link asymmetry. These factors are listed in a comprehensive way in [18]. For example, transmit power differences exist not only between different models of wireless cards, but also among wireless cards of the same model. Likewise, the receiver sensitivity (i.e. the capability of the receiver to decode incoming packets as a function of the received signal strength) can differ significantly between different wireless NICs. Receiver noise variation in the means of interference variations at the receiver or even the performance of the LNA amplifier can also contribute to link asymmetry. Finally, when there is wireless traffic in the network the spatial distribution of the MAC collisions (due to ineffectiveness of the carrier sense mechanism) can contribute as well to link asymmetry.

3.3.2 Distribution of delivery probabilities

Many routing and link-layer protocols assume that a communication between two nodes in a wireless network is either possible with a high delivery ratio or rather impossible. This is called “neighbor” abstraction in wireless networks and it is based on the assumption that the transition as the SNR changes from essentially zero bit error rate (BER) to a BER of a high enough value to make communication impossible is rapid. “Neighbor” abstraction is borrowed from the wired networks, where it is true, and has driven the design of many graph-theoretic routing algorithms for wireless networks. It also justifies the design of conservative rate adaptation algorithms that reduce the bit rate when only a few packets are lost.

An analysis of the link delivery probability distribution will reveal if the “neighbor” abstraction is valid or not for the measurements on TikNet. Figures 8, 9 show the distribution of inter-node delivery probabilities on TikNet at the rates of 1Mbit/s and 11Mbit/s. Measurements were also performed for 2Mbit/s and 5,5Mbit/s but they don’t differ significantly from the plotted curves and were omitted for convenience in interpreting the graphs.

In the following paragraphs I will compare my results with the results presented in [4] (see also figure 10). There are two main differences. The distribution of the delivery probabilities does not change with the modulation scheme used. This is true for both small packet size and large packet size. The plotted lines for 1, 2, 5,5 and 11Mbit/s (2 and 5,5Mbit/s are not shown) are very close to each other, nearly overlapping whereas in figure 10 there is a clear distinction between the lines with the 11Mbit/s line being significantly steeper than the other ones.

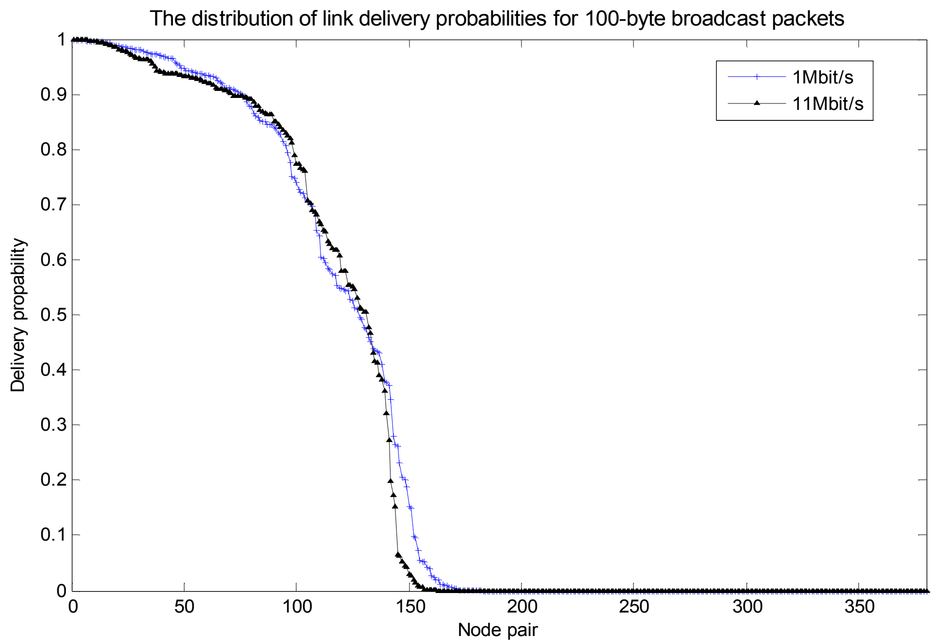

Figure 8 Distribution of the link delivery probabilities for 100-bytes payload broadcast packets

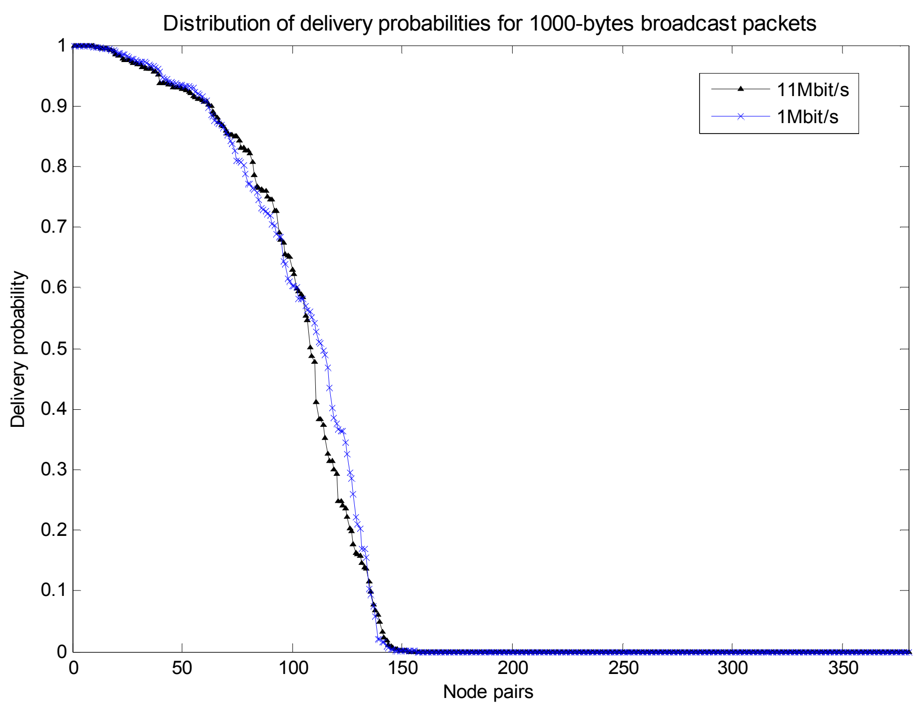

Figure 9 Distribution of the link delivery probabilities for 1000-bytes payload broadcast packets

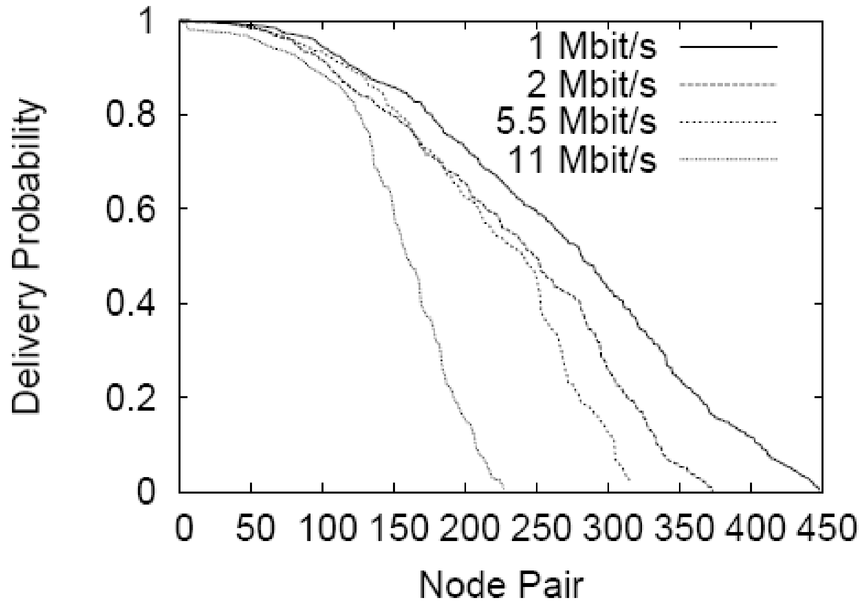

Figure 10 Distribution of the link delivery probabilities for 100-bytes payload broadcast packets based on measurements on Roofnet [4]

In order to have a better and more exact understanding about the distribution of the delivery probabilities I also plotted the empirical cumulative distribution function only for node pairs with non-zero delivery ratio. Again because the results we quite similar for the different rates and also for large or small packet size, the CDF for 1Mbit/s is only presented. The CDF is shown in figure 11.

The authors of [4] observed that intermediate loss rates (and therefore delivery ratio) dominate the distribution of the delivery probabilities of all node pairs. They concluded that multi-path fading could be the main reason for the large number of links with intermediate loss rates. By arbitrarily defining a loss rate as intermediate if the delivery probability for that link is between 30% and 70%, we can see in figure 11 that only 20% of all the active node pairs have intermediate loss rates. Therefore we can assume that the "neighbor" abstraction is roughly valid for TikNet testbed. Although this comes in contradiction with the results for Roofnet presented in [4] it can be explained if we consider that TikNet is an indoor environment and Roofnet is an outdoor environment. Indoor environments have substantially lower values of delay spread (time difference between first and last path) and therefore the effect of multi-path fading is less pronounced in such an environment.

Some experimental 802.11 measurements suggest that "neighbor" abstraction holds [3, 19] while others do not [4]. In a wireless environment where "neighbor" abstraction could be a reasonable assumption, traditional routing with minimum hop-count as a metric could perform quite well. The reason is that minimum hop-count routing is based on the concept that if the small "HELLO" packets can reach a node, that link has a rather good delivery ratio (otherwise it would have zero delivery ratio). That also means that it would be able to sustain communication with the two nodes exchanging data packets.

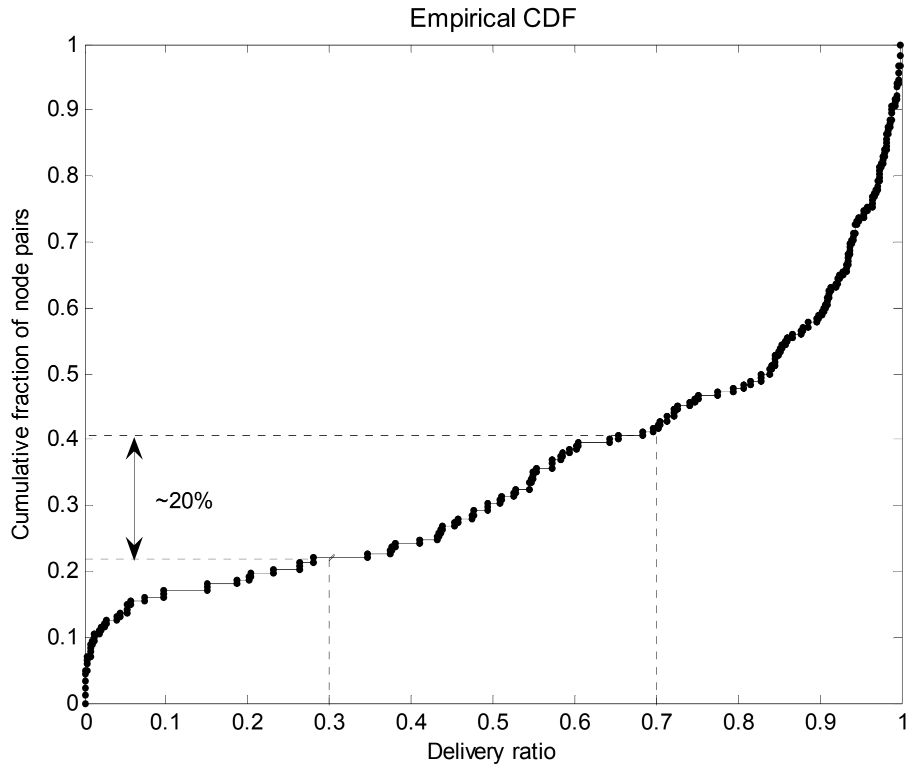

Figure 11 Cumulative distribution function of delivery probability for all node pairs in TikNet (1Mbit/s, small packet size, 30mW)

3.3.3 Interference from other 802.11 sources

For completeness of the wireless characterization of the TikNet testbed I measured the external interference from other 802.11 sources. In the G-floor of the ETZ building (and in the floors above and below) there are about 4 802.11 access points that belong to the ETH public wireless network transmitting at various frequencies. The measurements consider only the external 802.11 traffic in channel 3 (the channel the testbed is operating). In order to be representative for the main experiments, I also measured the interference during late night hours.

The procedure that I followed is the following: All the nodes were set in monitor mode operating at channel 3 (2.422 GHz) and captured the ongoing 802.11 traffic for 300sec. The dump files were afterwards processed offline and I extracted the traffic that is due to beacons and the data traffic. The values are averaged for all TikNet nodes within an interval of 1 second and are tabulated in table 1. The standard deviation is reported in the parentheses.

4 data packets per second and 13 beacon packets per second are not considered an important interference factor that could alter substantially the outcomes of the other

experiments. In [4] we can see that for Roofnet the external 802.11 interference is even higher. Therefore, operating at channel 3 and measuring during late night hours is justified.

Channel	Data	Beacons
3	4 packets/s (±9)	13packets/s (±21)

Table 1 Data and beacons from other 802.11 sources averaged for all TikNet nodes

4. Evaluation of routing metrics on the testbed

Building on the understanding of the wireless behavior of TikNet testbed, this section addresses the issue of routing metrics for wireless networks and their performance on the testbed. More specifically, I focus on two well known metrics, the minimum hop-count metric and the ETX metric, and compare their performance under different scenarios. Another presentation and comparison of different routing metrics is also available in [19].

4.1 Introduction to routing metrics for wireless networks

Before describing the various routing metrics, it seems appropriate to remind that a link metric is simply a weight assigned to each link that characterizes its capability to sustain direct communication between two nodes. On the other hand, a path metric is the combination of the metrics of all the paths that constitute the specific path.

4.1.1 Minimum hop-count metric

The simplest and most commonly used by existing ad hoc routing protocols metric is the minimum hop-count metric. As already mentioned in 3.3.2, routing with minimum hop-count as a metric is based on the assumption that "neighbor" abstraction (links either work well or don't work at all) is valid for wireless networks. This metric does not take into account the loss characteristics of the link or any other information. All links that can exchange routing packets have weight 1 and are equivalent. There is no possible way to distinguish them by means of link delivery probability or exposed interference. The link quality is binary and all paths created out of this metric have minimum number of hops.

An obvious advantage of minimum hop-count metric is its simplicity. It does not require any probing or measurements and it can be easily implemented. On the other hand, minimum hop-count is proven to have many disadvantages as well. Minimizing the hop count maximizes the distance traveled by each hop. The greater the distance, the lower the signal strength is and consequently the higher the loss rate. Even if the best route is that of minimum hops, in a dense network there may be many routes with the same minimum length and the metric simply can not distinguish between them. The selection is random and it is likely to be non-optimal. Finally, minimum hop-count does not take into account the asymmetry between the forward and reverse delivery ratios of a node pair.

4.1.2 Expected Transmission Count (ETX) metric

The ETX metric was proposed in [7]. It stands for Expected Transmission Count and its design goal is to find paths that minimize the expected number of transmissions (including retransmissions) required to deliver a packet from its origin to its destination. It predicts the number of retransmissions based on per-link measurements of packet loss rate in both directions of each wireless link. The ETX of a path is the sum of the link metrics. By minimizing the number of retransmissions it tries to find paths with high throughput by distinguishing between different loss rates.

If the measurement-based probabilities of successful transmissions in the forward and reverse direction of a link are d_f, d_r respectively then the ETX of a link [7] is

$$ETX = \frac{1}{d_f \times d_r}$$

The forward delivery ratio, d_f, is the measured probability that a data packet successfully arrives at the destination; the reverse delivery ratio, d_f, is the measured probability that the ACK arrives at the sender.

EXT measures the delivery ratios between two nodes by probing. Each node broadcasts small probes at a predetermined rate (every 1 sec in Grid Click implementation [7, 12]). Because broadcast packets do not require ACK or retransmission the loss rate of the broadcast probes is the actual loss rate. Each probe also carries information about probes received from other neighbors. Each node keeps track of the probes that it received during the last 10 seconds and computes the probability of successful reverse transmission. The information in each probe packet is used to calculate the probability of successful forward transmission.

ETX's main advantage is that is takes into account both directions of a direct link and thus it can cope with link asymmetry. It is also immune to self-interference [19] because it does not measure delay. More explicit analysis of the ETX metric is presented after the experimental results of this section.

4.2 Metric performance with DSDV

4.2.1 Experimental setup

The wireless cards for the following measurements were configured to send at 1Mbps, Ahdemo mode and with RTS/CTS turned off. The transmit power is set at 30mW. I use two different packet sizes, a small one of 256bytes (including UDP header) and a large one of 1160bytes.

Each data packet when transmitted consists also of 160 additional bytes (IP header, Grid encapsulation, LLC, 802.11 header, 802.11b physical layer header). An 802.11b ACK packet takes 304 μs to transmit, the inter-frame gap is 60 μs (DIFS+SIFS), and the minimum expected back-off time is 310 μs, resulting in a total of 2,754ms for every small data packet and 11,234ms for every large data packet. The above values give a maximum throughput of 250 small unicast packets per second and 89 large unicast packets per second over a loss-free link.

Each experiment presented in this section lasts about 200min. Two iterations of each experiment, one for minimum hop-count and another one for ETX are called a Run. I have enumerated all the sets of experiments as Run 1, Run 2, etc. depending on the transmit power and packet size configuration of the experiment. Within the 200min of an experiment each node transmits to all other nodes (including itself). All node pairs are covered in turn. The sending node sends UDP data packets to his destination as fast as it can. The destination measures the rate at which the packets arrive.

Before starting each experiment, the routing protocol runs for 3 or 4 minutes, long enough for it to stabilize. Also, as opposed to the procedure followed in [7], we didn't use any snapshot of the routing tables. The routing tables were changing dynamically throughout the whole duration of experiment.

4.2.2 DSDV throughput results

Figure 12 compares the throughput CDFs of paths found by DSDV using minimum hop-count and ETX for all node pairs. These data were collected when the wireless cards were transmitting with 30mW and the packet size was 256bytes (Run 1).

Figure 12 shows that ETX manages to find better throughput paths only for 20% of all the possible pairs. In this small region that ETX outperforms minimum hop-count the throughput is very low. ETX is better for these low throughput paths mainly due to two reasons:

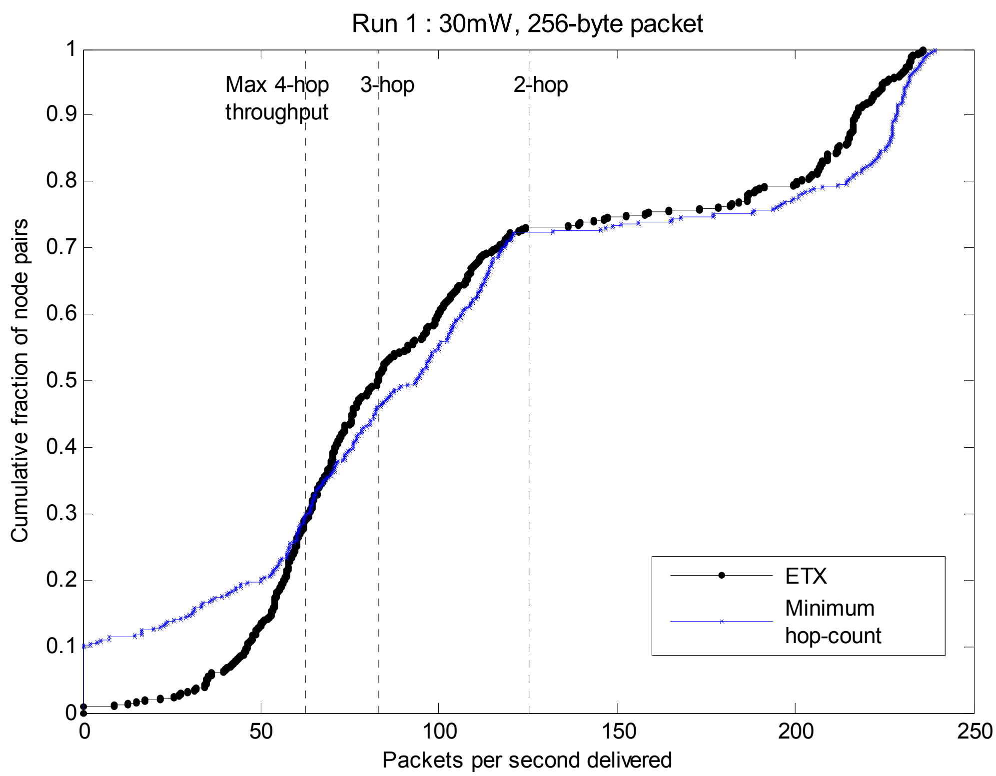

Figure 12 DSDV throughput CDF for ETX and minimum hop-count (30mW, 256-byte packet)

i) ETX avoids links with great asymmetry. The analysis of the delivery ratios of the wireless links of the testbed presented in 3.3.1 showed that there are highly asymmetric links with good delivery ratio in one direction and zero delivery ratio in the other. Minimum hop-count does not take into account the bi-directional delivery ratios and thus can not estimate correctly the quality of a very asymmetric link, resulting in zero throughput for 10% of all possible paths. On the other hand, ETX avoids the highly asymmetric links and find paths with non-zero throughput where minimum hop-count does not succeed providing more connectivity to the network.

ii) ETX also avoids communication grey zones. If two nodes that want to communicate are very far apart that the delivery ratio is very low, but nonetheless they manage to exchange some routing packets, it is said that the destination is in the grey zone of the source. Where minimum hop-count has no way of characterizing the link quality, ETX possibly identifies that these links are of very low quality and avoids them.

However, apart from that small region of 20% of all the nodes, where throughput is quite low, minimum hop-count outperforms ETX. By operating at the usual transmit power of 30mW the links of an indoor wireless testbed have pretty good delivery ratios (50% of all the active links have delivery ratio more than 85% - see figure 11). Also, as already mentioned in 3.3.2 there are not many links with intermediate loss rates. That is of course in favor of minimum hop-count as explained in 3.3.2.

At the upper right corner of the figure one can also see the overhead of the DSDV routing packets and the overhead of the ETX probe packets. DSDV routing packets force the maximum throughput to be less than the theoretically predicted (250 packets /sec). Moreover, the throughput with ETX is slightly smaller than with minimum hop-count due to the overhead of the ETX probe packets.

A more careful analysis of the data shows that the average throughput for all node pairs achieved by ETX and minimum hop-count is almost the same (107,8 packets/sec and 107,4 packets/sec respectively). The main advantage of ETX is that it avoids link asymmetries. However, asymmetry can be addressed to some extent also with the link handshaking scheme proposed in [20][2]. Although the link handshaking scheme can not discriminate between links with varying degrees of asymmetry, it is expected to improve minimum hop-count performance. It is also an improvement in the sense that it is a mechanism that does not introduce overhead through probing. Therefore, it is reasonable to expect that the average throughput achieved by minimum hop-count will increase even above the average throughput of ETX. Alternatively, in a rather illustrative way, one should expect less pair nodes with zero throughput at the lower left corner of figure 12. Using ETX as a metric in such a case becomes questionable.

I repeated the experiment with the transmit power set to 30mW and the packet size set to 1160bytes (Run 2). The results are similar to the results from Run 1 and are shown in figure 13. ETX still outperforms minimum hop-count for extremely asymmetric links and

---

[2] In this scheme, a node A only accepts route updates from a neighboring node B if B is advertising a direct route to A.

large number of hops although the packet size is larger. The mean throughput for ETX is again slightly better than the mean throughput of minimum hop-count metric.

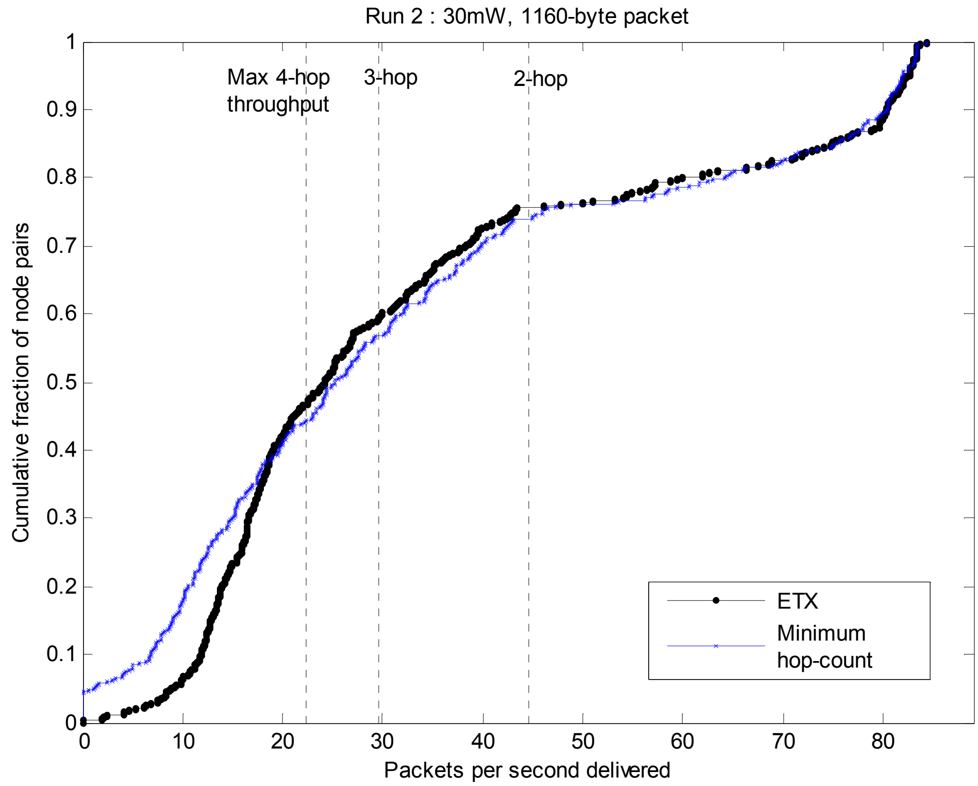

Figure 13 DSDV throughput CDF for ETX and minimum hop-count (30mW, 1160-byte packet)

All the experiments in this section were repeated more than one times and the validity of the results was verified.

4.2.3 Mobility scenario

In the scenarios that I have considered so far, all the nodes were stationary. Although this would be true for community networks scenarios, in ad hoc wireless networks users are mobile. Therefore, I investigated the performance of DSDV when one node is mobile and compared the performance of minimum hop-count and ETX metrics.

I moved in constant slow walking speed the mobile node (extra node) around the periphery of the G-floor through the main corridors, starting from a point near to node 8 (see figure 1). During the whole experiment node 8 was sending UDP data to the mobile node at full rate (the data packet size was small - 256bytes payload). The experiment was performed only once and the results are presented in table 2.

	Minimum hop-count	ETX
Throughput	127,7 packets/sec	115 packets/sec

Table 2 ETX and minimum hop-count performance in a mobility scenario

The throughput under minimum hop-count metric is 11% higher than the throughput under ETX metric. Also in [21] they observed that minimum hop-count metric performs better than ETX metric under mobility scenarios.

As the receiver moves around the network, the ETX metric does not react sufficiently quickly to track the changes in the link quality. This must be a more general problem for all the approaches that try to measure a link quality, since they require some time to come up with a stable estimate for the link quality. ETX's sliding window average mechanism makes the estimation of the link quality less responsive in time. This may lead ETX to report incorrect routing paths. On the other hand, minimum hop-count does not face such problems. It simply uses new links when they are discovered.

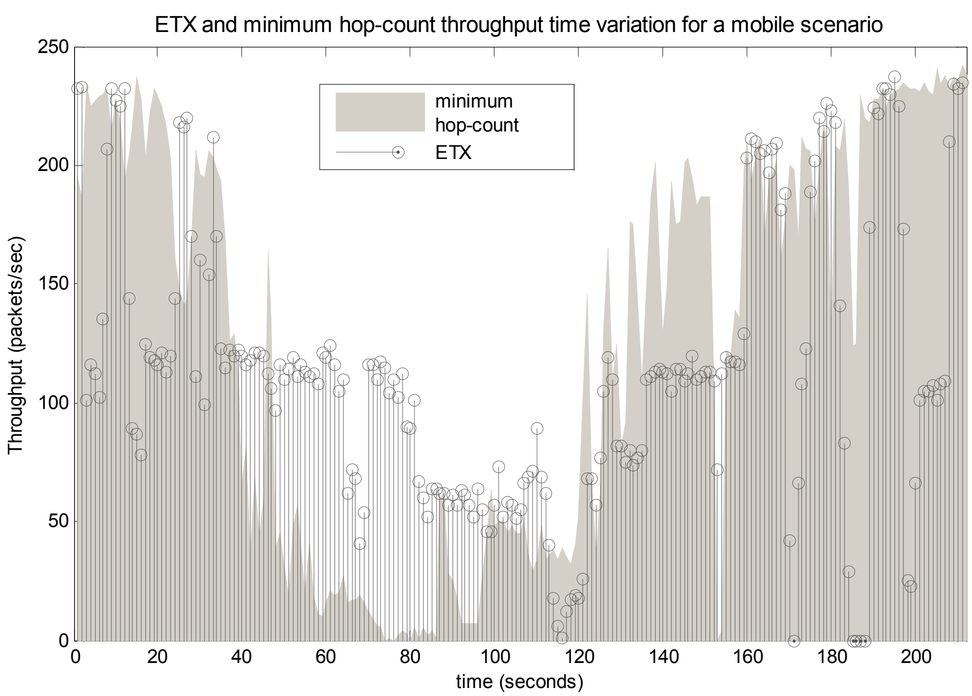

Figure 14 ETX and minimum hop-count performance for a mobile scenario

Finally, analyzing the throughput time variation (see figure 14) I observed that although minimum hop-count performance reaches zero at some points where ETX manages to preserve the connectivity, its big advantage comes from the fact that it becomes aware of a good path pretty quickly (like a direct path), while ETX takes some time to discover it.

This is especially pronounced in the first 20 seconds and the last 50 seconds of the experiment where minimum hop-count finds a good path (the direct path in this case since the movement pattern starts and ends near the sender – node 8) very quickly, in contrast with ETX that can not quickly adapt and find the good (direct in this case) path.

4.2.4 Accuracy of the transmissions predicted by ETX

As a last set of experiments I wanted to test the accuracy of the predictions made by ETX for the transmissions required by one packet to reach it destination. Experiments were conducted only for links (1-hop) and not paths.

During this experiment[3] (DSDV is always the routing protocol in use), the sender was transmitting packets at the maximum rate to the receiver. The receiver captured in monitor mode all the incoming packets and their 802.11 headers. In that way I could capture the exact number of retransmissions and through this value I calculate the actual number of transmissions averaged for every second. I also “sampled” the routing table of the sender every 1 second and extracted the ETX metric value for each second. Remember that ETX can potentially change value every 1 second - the inter-probe time.

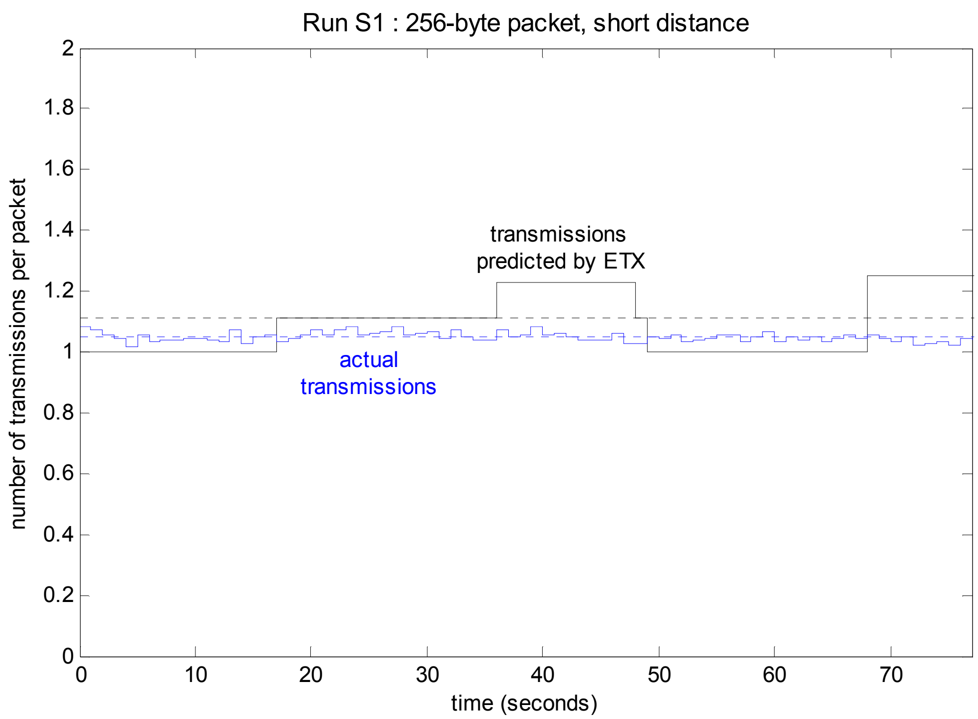

Figure 15 Actual transmissions vs transmissions predicted by ETX (256-byte packet, short distance)

[3] One has to keep in mind that this analysis is not complete and was done only to acquire an intuition about the conditions under which ETX can have a wrong estimation. More experiments that would also include multiple hops are a next step towards the statistical characterization of the ETX prediction accuracy.

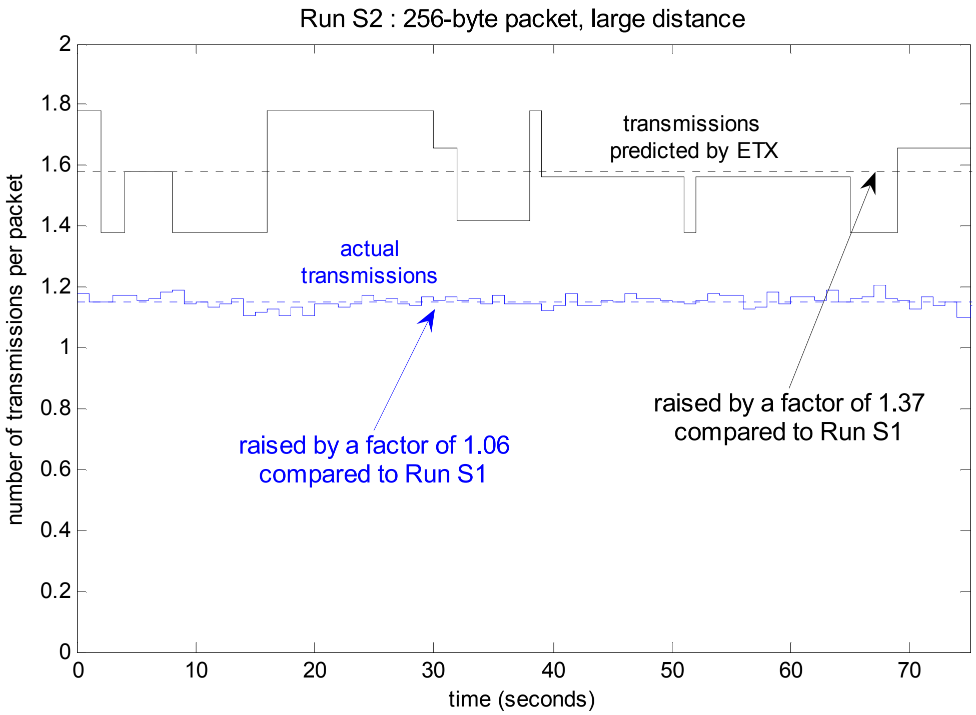

Figure 16 Actual transmissions vs transmissions predicted by ETX (256-byte packet, large distance)

The experiment was conducted one time for short and one time for large distance between the sender and the receiver (figure 15, 16). The communication is direct though in both cases. In that way we wanted to introduce more actual retransmissions into the link and check if ETX is able to correctly predict them of follow the raise.

It is shown that in this special case although the link is pretty stable and with a good delivery ratio, when we increase the distance between the two nodes, ETX incorrectly predicts more transmissions than the actual ones. The raise in the average for the actual retransmissions is only 1.06 whereas for the transmissions predicted by ETX is 1.37. The exact results of these experiments are summarized in table 3.

	Transmissions predicted by ETX	Actual transmissions
Short distance	1.11 ± 0.1	1.05 ± 0.01
Large distance	1.58 ± 0.02	1.15 ± 0.02

Table 3 Actual transmissions and transmissions predicted by ETX

These findings do not represent a general rule or trend but they do show that there are cases where ETX is not able to predict the link quality with a high degree of accuracy. From a physical layer point of view, the dynamics of the wireless channel make the channel stochastic and very unpredictable and by sampling with a rate of 1 packet per second one can not have a good estimator of the link quality. From an engineering point of view, ETX might be able to make some quality differentiation between links that is accurate for e.g. very asymmetric links but can only make roughly estimates about the quality of other links.

5. Conclusion

Experimenting with real-world hardware might be a challenge in terms of involving too many parameters that have to be considered but it also reveals behaviors that are sometimes surpising and are not taken into account in the simulations; like link delivery ratio asymmetry or intermediate loss rates dominating the distribution of the link delivery ratios. One designing new routing metrics or new routing protocols should consider coping with link asymmetry and intermediate delivery ratios (if they exist). For example, if "neighbor" abstraction is not a valid assumption for ad hoc wireless netwroks, then one should avoid designing routing protocols through a graph-theoretic approach. ETX design manages to overcome some of the already mentioned problems that affect minimum hop-count metric at a great deal. One the other hand ETX is based on a probing mechanism and remains a non accuarate estimator of the link quality. I believe that a more clever design based also on the advantages of ETX (e.g. avoiding link asymmetry or grey zones) is possible. Finally, coping with mobility through any probing mechanism seems not enough.

References

[1] http://pdos.lcs.mit.edu/roofnet

[2] S. Kurkowski, T. Camp, and M. Colagrosso. "Manet simulation studies: The incredibles." *Mobile Computing and Communications Review* pp. 50–61, October 2005.

[3] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. "Experimental evaluation of wireless simulation assumptions." *Proc. Of MSWiM 2004*, October 2004.

[4] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. "Link-level Measurements from an 802.11b Mesh Network." *Proc. of SIGCOMM 2004*, August 2004.

[5] K. Papagiannaki, M. Yarvis, and W. Conner. "Experimental characterization of home wireless networks and design implications." *Proc. Of Infocom 2006*, April 2006.

[6] http://tiknet.ee.ethz.ethz

[7] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. "A high-throughput path metric for multi-hop wireless routing." *Proc. Of ACM MobiCom Conference,* September 2003.

[8] www.dlink.com

[9] http://madwifi.org

[10] www.tcpdump.org

[11] http://www.wireshark.org/

[12] http://www.read.cs.ucla.edu/click/

[13] Eddie Kohler. "The Click Modular Router". *Ph.D. thesis, MIT*, November 2000

[14] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. "The Click Modular Router". *ACM Transactions on Computer* Systems *18(3)*, August 2000, pages 263-297.

[15] http://pdos.csail.mit.edu/grid/

[16] C.E. Perkins and P. Bhagwat. "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers". *Proc. of SIGCOMM 1994*, pages 234-244. ACM Press.

[17] David B. Johnson, David A. Maltz, and Josh Broch. "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks". *In Ad Hoc Networking*, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001

[18] Glenn Judd. "Using Physical Layer Emulation to Understand and Improve Wireless Networks", Technical Report 06-164, CS Department, Carnegie Mellon University

[19] R. Draves, J. Padhye, and B. Zill. "Comparison of routing metrics for static multi-hop wireless networks". *In ACM SIGCOMM 2004*

[20] H. Lundgren, E. Nordström, and C. Thschudin. "Coping with communication grey zones in IEEE 802.11b based ad hoc networks". *In 5th ACM International Workshop on Wireless Mobile Multimedia (WoWMoM 2002)*, September 2002

[21] Richard Draves, Jitendra Padhye, Brian Zill, "Comparison of Routing Metrics for static Multi-Hop Wireless Networks" *SIGCOMM 2004*, Aug 2004