Experimental Evaluation of BMX6 Routing Metrics in a 802.11an Wireless-Community Mesh Network

Llorenç Cerdà-Alabern and Axel Neumann
Univ. Politècnica de Catalunya, Computer Architecture Dep.
Barcelona, Spain
Email: llorenc@ac.upc.edu, neumann@cgws.de

Leonardo Maccari
DISI, University of Trento
Trento, Italy
Email: leonardo.maccari@unitn.it

***Abstract*—Modeling the performance of Wireless Mesh Networks (WMN) is a task that has received a lot of attention and has produced a large body of related literature. Most of the times, such literature is based on analytic assumptions or computer simulations, and results not always match reality. In this paper we use data collected over a one-month period in a 50-nodes wireless community network in Barcelona to compare the experimental throughput we measured over multi-hop paths with the capacity estimated using the well known conflict graph model introduced in [1]. Our experiments show that the model generally overestimates the available capacity, despite the availability of precise information on the underlying network graph. We also use the data to test the performance of the BMX6 routing protocol that runs on the network nodes and show that in the large majority of cases the path chosen by BMX6 corresponds to the path with the highest available capacity, which is a key feature to guarantee the maximum exploitation of the network resources.**

***Index Terms*—routing; 802.11n; community mesh networks; routing metrics; path capacity; conflict graph**

I. Introduction

After years of research devoted to Wireless Mesh Networks (WMN), today, this technology is mature enough to proliferate. One very enlightening example is the number of Wireless Community Networks (WCN) that are blooming in many countries. WCNs are networks built by a community of people that install wireless antennas on top of their houses and use them to access the Internet and to provide internal services managed by the community. WCNs started as a last-mile replacement for underserved (mostly rural) areas but today they are present in many cities around the world. Since WCNs are run by open communities of enthusiasts they are a perfect scenario to design and test new protocols and applications.

In this paper we exploit the full access to the wireless routers of the WCN in the Sants neighborhood in the city of Barcelona, a 50-nodes production network equipped with IEEE 802.11n radios. We thus achieve two goals: The first is to compare the experimental measures of throughput on multi-hop paths that we perform on the network with the expected capacity estimation on the same paths derived using the well known conflict-graph model introduced in [1]. Our experiments show that even with an accurate knowledge of the network parameters the conflict-graph model introduces an overestimation of the available capacity. We discuss the possible causes for this error and propose a correction. The second goal is to test the capability of the BMX6 routing protocol used in the WCN to choose the path that can guarantee the highest throughput. We show that BMX6 is able to choose the best path in the large majority of the cases, which is a key feature for any routing protocol, enabled by the right combination of the protocol internals and the metric used for link and path quality estimation.

II. Related Work

The experimental evaluation of production-state wireless mesh network has been done only in a small number of papers in literature compared to the enormous amount of works that use simulations, a review of the experimental research papers can be found in [2]. Some of the works use a similar approach to this paper for the extraction of real measurement data [3], [4], but most of the networks analyzed are single-channel networks using omnidirectional antennas. In our case the use of multi-channel and directional antennas makes the analysis more challenging, since we neither assume interference between every couple of neighbor links (like in [4]) nor its absence and thus have to rely on a complex model for capacity estimation [1]. Moreover, to our best knowledge, this is the first empirical evaluation of a real IEEE 802.11n-based community network. Other empirical works use controlled scenarios [5] to compare routing metrics (like ETX [6] and ETT [7] metrics). Finally, some other works use off-line evaluation of available data to estimate various network properties [8], including routing performance [9] but can not really be compared to on-field experimentation.

III. Experimental Network

The network on which we have carried out our experimental measurements is deployed in a neighborhood of the city of Barcelona (Spain) called Sants, as part of the *Quick Mesh Project* (QMP) [10]. The network was started in 2009 and in 2012 nodes from *Universitat Politècnica de Catalunya* (UPC) joined the network supported by the EU CONFINE project [11]: from now on we will refer to this network with the *QMPSU* acronym (from Quick Mesh Project at Sants-UPC). QMPSU is part of a larger Community Network started in 2004 which has more than 28.000 operative nodes deployed all over Spain called Guifi.net [12]. At the time of writing QMPSU has around 50 nodes, 16 at UPC and 34 at Sants and it is a network in production state, accessed every day

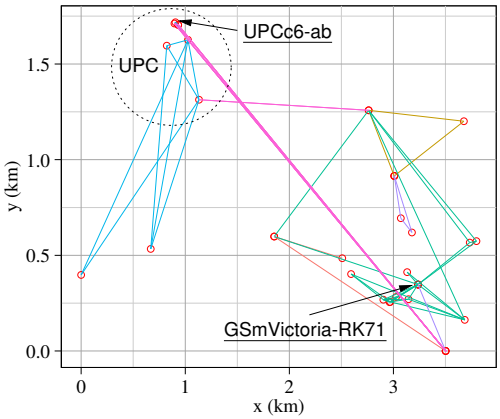

Figure 1. QMPSU topology. Gateways are underlined.

by tens/hundreds of users. Figure 1 shows the geographic location of the nodes and the active links, using distinct colors to represent wireless links configured with different channels. In QMPSU there are 2 gateways, one in the UPC Campus and another in Sants, that connect QMPSU to the rest of Guifi.net. The selection of the gateway is done by the routing protocol, although the user can manually select a preferred one. A detailed description of QMPSU can be found in [13], and a live monitoring page updated hourly can be accessed on-line [14].

The typical configuration of a QMPSU node is made of an outdoor router (OR) placed on the roof and equipped with an IEEE 802.11n radio. The OR is connected through an Ethernet cable to an indoor AP, that is used as a gateway in the local network of the user. The most common OR in QMPSU is the Ubiquiti NanoStation M5, which integrates a full router with a 802.11n radio and a sectorial antenna with about 40 degrees of horizontal coverage. In some strategic locations multiple ORs are connected to the same AP in order to provide a wider horizontal coverage angle. Some of the links instead are realized with parabolic antennas (Ubiquiti NanoBridge) to achieve long point-to-point connections. ORs in QMPSU are flashed with the linux distribution that has been developed inside QMP project. This distribution is a branch of OpenWRT [15] and uses BMX6 as mesh routing protocol [16].

Measurements have been obtained capturing data via remote shell (ssh) hourly to each QMPSU OR and running basic system commands. In total, 715 captures have been done during the whole month of April, 2014. Each capture consists of throughput from the node to its gateway; throughput from the node to each of its neighbor nodes reported by BMX6; channels of the wireless links and routing tables. Throughput is measured using TCP connections by means of the netperf tool.

IV. BMX Routing Protocol and Metrics

BMX6 is a destination-sequenced, proactive, distance-vector like, routing protocol for IP networks using UDP broadcast messages to exchange link, node, and path-discovery messages between neighboring nodes and, by re-broadcasting requested messages on demand, propagating global information to all network nodes.

A node (router) in BMX6 is, unlike to traditional routing protocols, not identified by its primary IP address(es), but by two ID values of global scope, being (i) a permanent ID that identifies a particular router at any time and (ii) a description ID that is generated by the current set of configuration parameters of a router at given time. This set of configuration parameters contains, among others, its global permanent ID, a sequence number, the announced address ranges reachable via this router, and the specification and parametrization of a metric function that defines how forwarding routes (next hops) towards the announced address ranges should be selected and how the path metric propagated via routing updates should be calculated by other nodes of the network. The description ID is given simply by the SHA1 hash of each's router configuration.

A link in BMX6 is detected by the reception of a hello message from a neighboring node and is identified by the link-local IPv6 address of the broadcasting node (given by the source address of the IPv6 header) and the link-local address of the receiving interface. Hello messages are broadcasted periodically at a fixed interval ($0.5s$ by default) and contain a sequence number. The receive-link quality r_l to a hello-sending node represents the fraction of recently received hello messages in the range [0..1] and is computed by comparing received and non-received sequence numbers in a sliding link window (with a default size of 100) and whose upper boundary is given by the most recently received sequence number. Upon reception of each hello message, this value is broadcasted back to the hello-sending node with a hello-reply message that also references the two link-identifying link-local IPv6 addresses[1]. Upon reception of a hello-reply message, which references one of a node's own link-local interface addresses, each node can record the contained link-quality as its transmit quality t_l for this link. Interruptions in the supposedly continuous reception of link hello or reply messages that exceed the double of the hello interval are used to penalize the most recently recorded RX and TX quality of a link.

Routing updates in BMX6 contain 3 fields. A path-metric value, a sequence number, and, instead of a destination address or network, a node-configuration reference such as the description ID[2]. Each router maintains a table with relations between description IDs and corresponding descriptions. A router receiving a routing update with an unknown ID inquires the desired description by sending a description request containing the corresponding hash value to the neighboring node via which the routing update has been received.

Based on the routing update messages originated from node n and received via link l, the contained path-metric value $m_{n,l}$, and the metric function F_n described for n an updated end-

[1]Stateful compression is used here to reduce the repetitive overhead of the two 128 bit IPv6 addresses

[2]Via stateful compression, the used implementation reduces protocol overhead caused by frequent exchange of routing updates between neighboring nodes by substituting the 20-bytes (SHA1) description ID with a 16-bit integer value

to-end path metric M is calculated as $M_{l,n} = F_n(r_l, t_l, m_{n,l})$ with r_l and t_l being the locally measured receive and transmit qualities for link l. This way, once the description, referenced by a routing update and received via one or several links (neighbors), is resolved, the best next hop (route) for forwarding data packets to node n is given by the link with the best updated path metric value. This value is also used as metric when re-broadcasting the received routing update for further end-to-end path propagation.

The path metric value used for representing the quality of an end-to-end path in routing updates and internally is given by an exponentially encoded value with 5-bit exponent and 5-bit mantissa, allowing to express dimension-less values at 3% steps in the range of $[36..128\ 10^9]$. The meaning of this value is exclusively defined by the metric function defined by each's node description with only two general requirements being that (i) a greater value represents a better end-to-end metric and, (ii), its value *must* decrease with each application of metric function F. This way, the metric value of an initial routing update sent by an originating node (typically set to the maximum possible metric value M^{max}) continuously decreases which each hop it gets further propagated.

A number of different metric functions based on broadcast-link probing are supported by BMX6. For example the behavior of a simple Hop Count (HC) or Shortest Path First (SPF) metric is implemented as $F^{HC} = 0.97\, m_{n,l}$, doing nothing else than reducing the updated metric value by the minimal-possible constant factor of 3% with each iteration, thus every additional hop.

The equivalent to the Expected Transmit Count (ETX) metric, used by many community mesh networks based on OLSRd, is implemented as $F^{ETX} = \min(F^{HC}(m_{n,l}), \frac{M^{max}}{M^{max}/m_{n,l}+1/q_l})$ with $q_l = r_l\, t_l$. This function, when assuming $M^{max} = 1$, in fact equals the inverse of the original ETX function as defined in [17] as $ETX_n^{path} = \sum_{l_j \in L(P)} ETX_l^{link}$ with $ETX_l^{link} = 1/(r_l\, t_l)$ and which is used to minimizes the cost in terms of totally required transmissions for sending a packet along a path. Here, and in the following function, the term F^{HC} is considered to ensure that output decreases even in case of perfect (zero-loss) links.

Both, HC and ETX, can be seen as additive metrics as total path cost is calculated as the sum of it's sub paths. In contrast, the Transmit Quality (TQ) metric function used by the BATMAN protocol can be considered multiplicative as it aims to maximize the "the goodness" of a path by building the product of its link qualities. Related behavior is implemented in BMX6 as $F^{TQ} = \min(F^{HC}(m_{n,l}), (m_{n,l}\, r_l^a\, t_l^b))$. With its defaults $a = 0$ and $b = 1$, the impact of measured link-receive quality is neglected and the metric output value is driven mainly by the transmit qualities of links along a path. Also different parametrizations like a=1 and b=2 are possible to not entirely neglect the receive quality but still putting greater weight on the transmit qualities of links in the weighted end-to-end forwarding path.

Eventually, the function used by default by BMX6 and also in the deployment used for the experiments in this work is called Vector Metric (VM) and is given as $F^{VM} = \min(F^{HC}(m_{n,l}), \frac{M^{max}}{\sqrt{(M^{max}/m_{n,l})^2+(1/q_l)^2}})$. The rationale for this function is to achieve an increased negative impact for additional hops in short paths while relieving the impact of additional hops for long paths where the quality of the links becomes much more important. This characteristic is not provided by the strictly additive hop count or ETX metrics, but relevant when optimizing for throughput in a wireless multihop network [18]. Another interesting characteristic of the VM metric is its penalization of paths with un-balanced links via the square-root product of its links which should avoid bottlenecks along a path.

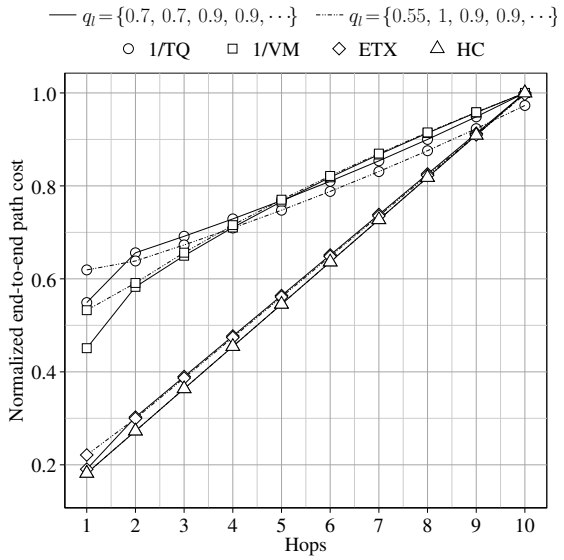

Figure 2. Evolution of end-to-end path cost of metric functions along two exemplary scenarios with differently balanced link qualities.

The growth of path cost at different hop-distance to a destination node when applied on two exemplary scenarios has been numerically calculated and is illustrated in Figure 2. Continuous-line graphs are calculated assuming a better-balanced occurrence of link qualities in the first two hops than those printed with dashed lines. For better comparability, results were normalized regarding their maximum observed value and the *goodness* of calculated TQ and VM metrics has been translated to a *cost* by representing their inverse values.

For the selected paradigms it can be seen that only the VM metric would favor the better-balanced chain of links and that the impact of additional links diminishes for longer path. As expected, path cost calculated for ETX and HC (SPF) metrics show a linear increase while TQ metric even shows a slightly exponential increase. However, it must be noted that the desired effects, although probable, seem rather small.

V. Theoretical Path Capacity

In order to evaluate the performance of the routing protocols we need to estimate the capacity of selected paths. Accurate capacity estimation in wireless is challenging, and the *Protocol Model* proposed in [19] is typically used in 802.11 networks.

With this model any couple of nodes using the same channel and in interference range can not simultaneously transmit. The protocol Model was used to define the concept of *conflict graph* in [1] to estimate the capacity of wireless networks as an LP optimization problem. Afterwards, the conflict graph has been extensively used in the literature to estimate the capacity of wireless networks in resource optimization problems, e.g channel allocation [20], [21], [22]. In the following we will recall the concept of conflict graphs and will use it not to formulate an optimization problem but instead to estimate the capacity of a multi-hop path once the capacity of the single hops has been measured.

Let $G(V, E)$ be a graph in which the set of vertices V corresponds to the set of nodes in the network and the set of edges E corresponds to the set of links. Let $N = |V|$ the number of nodes of the network denoted by $n_i,\ 1 \leq i \leq N$. Take a generic path $P = \{n_1, \cdots n_d\}$ as the ordered set of nodes chosen by the routing protocol to deliver a packet from the source node n_1 to the destination node n_d. Let l_i be the link used to connect each node n_i to the next node n_{i+1} in P, c_i the capacity (in bit per second) of link l_i and $L = \{l_1, \cdots l_{d-1}\}$ the set of links used in P.

Let $G_c(E, C)$ be the conflict graph of G. In G_c vertices correspond to links in G, between two vertices there is an edge if the two links interfere and thus can not transmit simultaneously. Let $G_c(P)$ be the induced sub-graph of G_c, where the vertices are the links L in P, and the edges are the same as those that links L have in G_c.

Now, let $N_i(P),\ i = 1, \cdots d-1$ be the sets formed by each vertex of $G_c(P)$ and its neighbors. Consider two links $l_i, l_j \in N_i(P)$ that require a time $\frac{1}{c_i}$ and $\frac{1}{c_j}$ respectively to send one bit on the link. Note that each set $N_i(P)$ is formed by links that need to schedule their transmissions in different time intervals, so if a bit has to travel over link c_i and then c_j it will require a total time of $\tau = \frac{1}{c_i} + \frac{1}{c_j}$. The capacity of the path l_i, l_j is thus given by $\frac{1}{\tau}$. Generalizing, for each sub-path formed by links belonging to $N_i(P)$ the expected capacity is:

$$C_i(P) = \frac{1}{\sum_{l_j \in N_i(P)} \frac{1}{c_j}},\ i = 1, \cdots d-1. \tag{1}$$

The *theoretical capacity of the path*, $C_t(P)$, is given by the most restrictive sub-path, thus:

$$C_t(P) = \frac{1}{t_b(P)} \tag{2}$$

where

$$t_b(P) = \max_i \sum_{l_j \in N_i(P)} \frac{1}{c_j},\ i = 1, \cdots d-1 \tag{3}$$

Note that $N_{i=b}(P)$ is the set of links of the path P that minimize (1). Thus, we shall call $t_b(P)$ the *bottleneck airtime* of the path.

A. Validation

In order to validate equation (2) we have proceeded as follows: We have experimentally estimated the capacity c_i of each link by measuring the throughput with netperf. The same has been done to estimate the capacity of the path to the gateway for each node. We shall refer to these measurements as the *experimental capacities*, and denote them as $C_e(P)$.

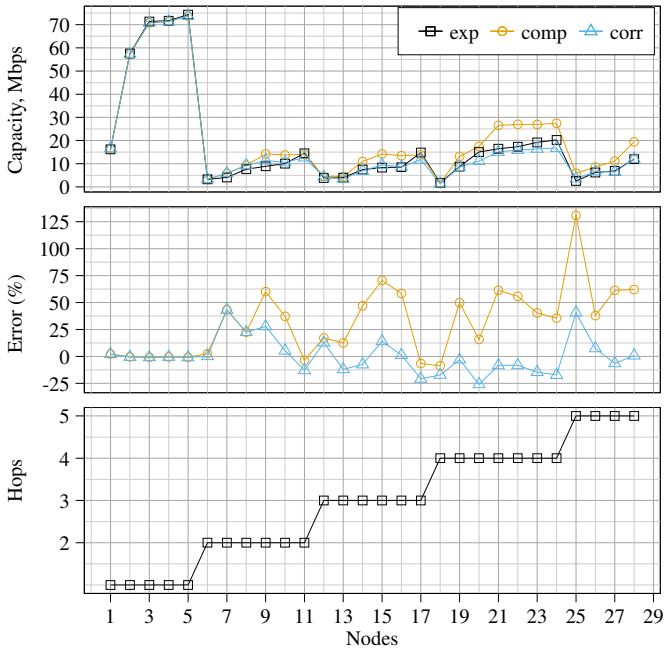

Figure 3. Experimental $C_e(P)$ (exp), theoretical $C_t(P)$ (comp), and fitted $C_f(P)$ (corr) capacities (top); relative error of theoretical $e_t(P)$ and fitted $e_f(P)$ capacities (middle); and number of hops (bottom).

In order to compute the conflict graph $G_c(P)$ we proceeded as follows. First we defined the graph G and we assigned to each link a value for the capacity c_i that equals the measured one. Then we generated G_c, using as vertices the node links, and adding edges between neighbor links using the same channel. Thus, we assume that interference only occurs between WiFi neighbor interfaces using the same channel. For each node we computed the path used to reach the gateway by means of the routing tables and, on that path, we computed the theoretical capacity using (2). We shall refer as *theoretical computed capacity*, $C_t(P)$, to the capacity obtained by (2).

Figure 3 (top) shows the mean experimental ($C_e(P)$), and theoretical ($C_t(P)$) capacities to the gateway of each node. These are measured only for the most frequent route of each node.The means were obtained averaging more than 100 points in all cases. The resulting confidence intervals were rather small, less than 5% in most cases. In the same figure is shown a third curve ($C_f(P)$), which is a better estimation than $C_t(P)$ and will be explained in next section. Figure 3, middle, shows the relative error of $C_t(P)$ and $C_f(P)$ capacities with respect to the experimental ones, $C_e(P)$, computed as:

$$e_i(P) = \frac{C_i(P) - C_e(P)}{C_e(P)},\ i = \{t,\ f\}. \tag{4}$$

Finally, Figure 3 bottom shows the number of hops of each route. Note that paths are sorted in increasing order of hops, and capacity.

Figure 3 shows that the theoretical capacity overestimates significantly the experimental one. Indeed, the absolute relative error has an average around 34%. This result concerns the usage of the conflict graph as an accurate tool to estimate the capacity of a wireless network. In the following section we discuss this mismatch and propose a better fit of equation (2) to the experimental path capacity.

B. Path Capacity correction

Given the results of Sect. V-A, we can say that the definition we use for the conflict graphs leads to an overestimation of the available capacity. To build the correct conflict graph we need to know all the links that interfere with each other, and can not transmit simultaneously. In G_c we set an edge between two links only when the two links are in the same channel, and are separated by no more than one hop, we say that this approach describes only "*direct interference*". This assumption is reasonable considering that the majority of the radios use directive antennas, but is probably optimistic, since there are a number of factor that produce what we call "*collateral interference*". First we do not consider interference at a higher distance than one-hop, which instead can happen. The number of hops between two nodes depends on the way the radio are configured, and on the decision that the routing protocol takes. Two nodes can be close to each other, but configured with an incompatible MAC layer mode (for instance, both configured to be client of a third node) that prevents them to be direct neighbors. Second, neighbor-channel interference can happen when two radios are placed nearby [23] and even when directive antennas are used [24]. We can not capture this phenomenon with our abstraction so it is reasonable that this contributes to the overestimation of the available capacity.

Since it's impossible to perfectly model a network operating in real conditions with an analytic approach we chose to apply an empirical approach using the experimental data we have.

Thus, we propose to modify equation (2) to estimate the collateral interference in the QMPSU network, as:

$$C_e(P) \approx C_f(P) = \frac{1}{t_b(P) + f(P)} \tag{5}$$

We shall call *airtime bloat* the term $f(P)$, which represents the increment on the bottleneck airtime induced by the interference that we can not precisely model over path P. The formulation that better captures the concept of collateral interference is the following one, in which we assume that the bottleneck is probably made worse by the interference generated by other links, that we introduced with a scaling factor θ.

$$f(P) = \theta \sum_{l_j \notin N_{i=b}(P)} \frac{1}{c_j},\ 0 \le \theta \le 1 \tag{6}$$

In order to estimate θ we used the available experimental data to compute the the mean square relative error of the mean

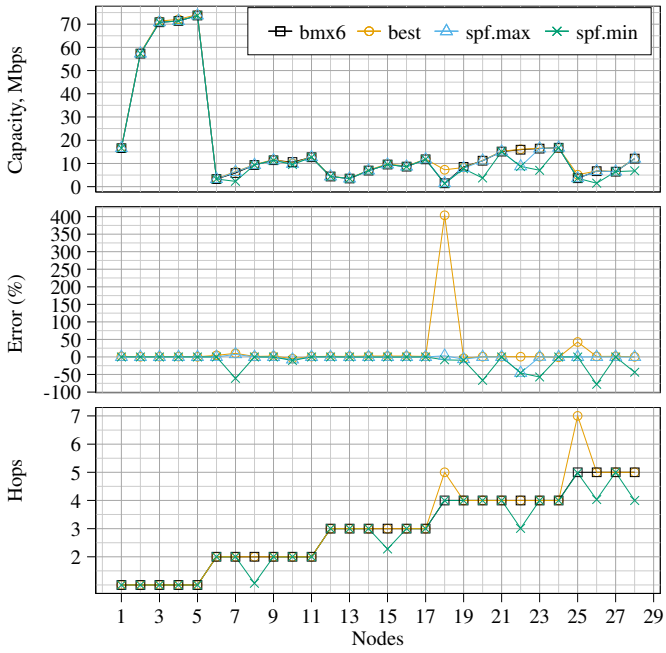

Figure 4. Fitted (corr), best (best), and worst and best SPF paths (spf.min, spf.max) capacities (top); relative error with respect to corrected capacities (middle); and number of hops of fitted, best and SPF paths (bottom).

capacities, i.e. by minimizing the cost function[3]:

$$J(\theta) = \sum_{P} \left(\frac{C_f(P) - C_e(P)}{C_e(P)} \right)^2 \tag{7}$$

From which we obtained that the most suitable value to approximate our data set is given by $\theta \approx 0.5$.

Figure 3, top, compares the experimental (*exp*) and computed capacities using equation (5) (*corr*). Figure 3, middle, reports the relative error. It can be observed that the capacity estimation is significantly improved. In fact, the absolute relative error has an average around 12%, which is almost 3 times smaller than the 34% error obtained with equation (2). Note that the value of θ is a characteristic of the QMPSU network, so it can not be simply re-used in other networks. Nevertheless, giving a reasonable good estimate for QMPSU, as discussed above, equation (5) will be used as reference to investigate the performance of BMX6 carried out in next section.

VI. BMX6 Performance

In this section we compare the paths chosen by BMX6 with the best paths (having the highest capacity). For the sake of comparison we also use the paths obtained using the Shortest Path First (SPF) algorithm. Note that SPF correspond to hopcount metric. All capacities shown in this section are computed using the corrected equation (5). The best path to the gateway has been computed using an algorithm not provided here, for the sake of space. Basically, the algorithm performs a recursive search estimating the capacity of each path. In order

[3]We have used the BFGS algorithm provided by the numerical tool R.

to avoid a costly exhaustive search, it is first guessed a best path using a weighted SPF, with link airtimes as costs. Then, recursion is performed, stopping over paths that give worst bandwidth than the current best path estimate.

Figure 4 compares the capacity of the path chosen by BMX6 (*bmx6*); the best path (*best*); and paths yielding the maximum and minimum capacities using SPF (*spf.max*, and *spf.min*, respectively). Note that the points corresponding to *bmx6* are the same than those marked as *corr* in Figure 3. For the same number of hops, there might be different paths, having different capacities. As in the previous section, these capacities are computed averaging over the most frequent paths chosen by BMX6, and the best and SPF paths obtained in the same captures. Figure 4, middle, reports the relative error of best and SPF paths with respect to BMX6 (see equation (4)). Thus, positive error means better paths than BMX6, and negative error means worst. Finally, Figure 4, bottom, shows the number of hops to the gateway for the paths chosen by BMX6, best and SPF.

Figure 4 shows that BMX6 Vector Metric behaves indeed very well: In most cases the best paths only give a slightly better capacity than BMX6. Only in 2 cases there exists a significantly better path (with relative increases of 400% and 40%, respectively), but having a larger number of hops. Regarding SPF, it was obtained that for the best choice (spf.max), only in 2 points SPF was slightly better, but less than 10%. While spf.min was always worse or equal than BMX6. Indeed, spf.min yielded 6 points (26% of the paths having more than 1 hop) with a relative reduction higher that 40% than BMX6.

VII. Conclusions and Future Works

In this paper we used experimental evidence to analyze the performance of the BMX6 routing protocol. In particular we focused on the capacity of BMX6 Vector Metric to select the route that can achieve the highest throughput and we verified that the combination of metric and protocol internals used by BMX6 is very efficient in selecting a path that is very close to the optimal one. To achieve this goal we performed experiments on the QMPSU network that showed that the model proposed in [1] with simple assumptions on the interference among links produces an overestimation of the achievable throughput.

Next steps of this research direction can be the evaluation of other protocols with the same methodology, and a better and more generic estimation of the conflict graph on a running network based on local measures of throughput.

Acknowledgments

This work was supported by the European project CONFINE, FP7-288535 http://confine-project.eu, and Spanish grants TIN2013-47245-C2-1-R and 2014-SGR-881.

References

[1] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless networks*, vol. 11, no. 4, pp. 471–487, 2005.

[2] S. Vural, D. Wei, and K. Moessner, "Survey of Experimental Evaluation Studies for Wireless Mesh Network Deployments in Urban Areas Towards Ubiquitous Internet," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 223–239, 2013.

[3] J. Camp, J. Robinson, C. Steger, and E. Knightly, "Measurement driven deployment of a two-tier urban mesh access network," in *4th international conference on Mobile systems, applications and services*. ACM, 2006, pp. 96–109.

[4] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *11th International conference on Mobile computing and networking*. ACM, 2005, pp. 31–42.

[5] C. Houaidia, A. Van Den Bossche, H. Idoudi, T. Val, and L. A. Saidane, "Experimental performance analysis of routing metrics in wireless mesh networks," in *Wireless Comm. and Mobile Computing Conference (IWCMC), 2013 9th International*. IEEE, 2013, pp. 1011–1016.

[6] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[7] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 114–128.

[8] D. Vega, L. Cerda-Alabern, L. Navarro, and R. Meseguer, "Topology patterns of a community network: Guifi.net," in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2012, pp. 612–619.

[9] L. Maccari and R. Lo Cigno, "A week in the life of three large Wireless Community Networks," *Ad Hoc Networks*, vol. 24, Part B, pp. 175–190, Jan. 2015.

[10] "Quick Mesh Project," http://qmp.cat.

[11] "Community Networks Testbed for the Future Internet, CONFINE," http://confine-project.eu/, FP7 European Project 288535.

[12] "Open, Free and Neutral Network Internet for everybody," http://guifi.net/en.

[13] L. Cerdà-Alabern, A. Neumann, and P. Escrich, "Experimental evaluation of a wireless community mesh network," in *The 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM'13*. Barcelona, Spain: ACM, Nov. 3–8, 2013.

[14] "qMp Sants-UPC monitoring web page," http://dsg.ac.upc.edu/qmpsu.

[15] "OpenWrt Linux distro. for embedded devices," https://openwrt.org.

[16] A. Neumann, E. López, and L. Navarro, "An evaluation of bmx6 for community wireless networks," in *1st International Workshop on Community Networks and Bottom-up-Broadband (CNBuB'2012)*, Barcelona, Spain, Oct. 2012, pp. 651–658.

[17] M. E. M. Campista, P. M. Esposito, I. M. Moraes, L. H. M. K. Costa, O. C. M. B. Duarte, D. G. Passos, C. N. de Albuquerque, D. C. M. Saade, and M. G. Rubinstein, "Routing metrics and protocols for wireless mesh networks," *Network, IEEE*, vol. 22, no. 1, pp. 6–12, 2008.

[18] J. Friedrich, S. Frohn, S. Gubner, and C. Lindemann, "Understanding IEEE 802.11n multi-hop communication in wireless networks," in *2011 International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, May 2011, pp. 321–326.

[19] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, 2000.

[20] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *11th annual international conference on Mobile computing and networking*. ACM, 2005, pp. 73–87.

[21] Y. Ding and L. Xiao, "Channel allocation in multi-channel wireless mesh networks," *Computer Communications*, vol. 34, no. 7, pp. 803–815, 2011.

[22] D. Yang, X. Fang, and G. Xue, "Channel allocation in non-cooperative multi-radio multi-channel wireless networks," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 882–890.

[23] A. Zubow and R. Sombrutzki, "Adjacent channel interference in IEEE 802.11n," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2012, pp. 1163–1168.

[24] S. Lakshmanan, J. Lee, R. Etkin, S.-J. Lee, and R. Sivakumar, "Realizing high performance multi-radio 802.11n wireless networks," in *Sensor, Mesh and Ad Hoc Comm. and Networks (SECON), 2011 8th IEEE Communications Society Conference on*. IEEE, 2011, pp. 242–250.