An efficient RSSI-aware metric for Wireless Mesh Networks

Ebenezer Amusa, Osei Adjei, Jie Zhang, Ali Mansour
Institute for Research in Applicable Computing
University of Bedfordshire, Park Square, Luton, LU1 3JU, UK
Email: eoamusa@ieee.org, {osei.adjei, jie.zhang, ali.mansour}@beds.ac.uk

Antonio Capone
Department of Electronics and Information
Politecnico di Milano, Italy
Email: capone@elet.polimi.it

***Abstract*—This paper proposes a RSSI-aware (*rETT*) routing metric for WMN. The embedded RSSI information from the mesh nodes are extracted, processed, transformed and incorporated into the routing process. The performance of the *rETT* metric is then optimised and the results compared with the fundamental hop count metric and the link quality metric by Expected Transmission Count (ETX).**

The implementation results on OLSR show that *rETT* improvement on network throughput is more than double (120%) compared to hop count and a 21% improvement compared to ETX. Also, an improvement of 33% was achieved in network delay compared to hop count and 28% better than ETX.

This work provides a valuable insight into adapting RSSI information in mesh network routing and the implementation method ensures that the results are realistic.

I. Introduction

Wireless Mesh Network (WMN) is an emerging and promising technology for the next generation of wireless networks. Currently, there are commercial interests in adapting WMN in the area community networking, high speed metropolitan area networking, broadband home networking, enterprise networking and building automation [1]. These have motivated interest in optimising or re-inventing some of the existing protocols for WMN.

Routing in WMN is significant in maximising network performance. The classical routing approach is by hop count which has been observed to be inadequate [2]. Recently, some cross-layer routing solutions have been investigated mainly through simulations [3], [4] some by theoretical approaches [5], [6] but few by empirical methods [7]. Although, there are works using the implementation approach, most methods introduce extra probes to achieve the cross-layer effect to adapt the routing process. For example, the work by Carrera *et al.* [8] used three extra probes to estimate the Expected Transmission Time (ETT) for the WMN. Routing decisions should be fast and lightweight and this is one of the objectives of this work.

This paper is organised as follows: Section II gives a brief overview of routing in WMN. The design and methodology of this novel approach is highlighted in Section III. Section IV describes the test-bed and implementation on Optimized Link State Routing Protocol (OLSR). The results from the implementation are discussed in Section V while Section VI concludes the paper.

II. Routing in WMN

Routing in WMN occurs on the backbone routers for the purpose of multi-hop communication. This is similar to Ad-hoc network routing except that mesh routers are typically stationary. The hop count metric is appropriate for Ad-hoc networks because of user mobility and the need to find new routes quickly. Conversely, the stationary nature of mesh nodes facilitates proactive estimation of link-quality metric for the routing process. Hence, it can be concluded that the stationary topology of WMN benefits quality-aware routing metrics [9].

A. WMN Link Quality Routing Metrics

The digression from hop count metric has opened up the link-quality routing approach. This approach aimed at improving the routing process in WMN by estimating and incorporating the changing link conditions into the routing process. Effective routing decisions help in the network topology formation that will ensure optimal network performance. Consequently, link quality routing metrics for WMN have been receiving some attention lately [9], [10], [11], [12], [13], [14], [15], starting with the Expected Transmission Count (ETX) [16]. ETX is based on loss probability and is estimated by evaluating the *forward delivery ratio* (d_f) and the *reverse delivery ratio* (d_r). Therefore, the ETX of a link between two nodes is expressed as $ETX = 1/(d_f * d_r)$.

ETX metric uses `HELLO` packets to estimate the link quality. These are relatively small in size (64-134 bytes) compared to the actual data packets. Usually, `HELLO` packets will be acknowledged even when the link is not good. ETT improved on the ETX approach by introducing link capacity into the link metric. There are currently two main techniques for achieving ETT. One is the *packet-pair technique* [10] and the second approach whose metric is termed *ScrRR* is reported in [17]. In the *packet-pair technique*, ETT is achieved by $(ETX * t)$, where t is defined as (S/B). The authors defined S as a fixed data-packet size and the estimated bandwidth as B. The packet-pair technique then estimates B by transmitting a sequence of two back-to-back packets. These are unicast in sequence, a small packet followed by a large one. Each neighbour measures the inter-arrival time between the two packets and reports back to the sender. Hence B is calculated as the size of the large packets divided by the minimum delay received for the link.

978-1-61284-824-2/11/$26.00 ©2011 IEEE

The approach in [17] estimates loss probability by considering data and ACK frames. The data frames are estimated by broadcasting same size packets, one packet for each of the data rates defined by IEEE 802.11. Furthermore, the ACK frames are also estimated by broadcasting at basic rate, a sequence of small packets of the same size for ACKs. ETT is calculated as the inverse of the product between the best throughput achievable (r_t) and the delivery probability of ACK packets in the reverse direction p_{ACK}. Therefore, $ETT = 1/(r_t * p_{ACK})$. It should be noted that both methods employ the use of extra probes to estimate the link quality metric.

It is imperative to note that control and processing overheads are required to be kept to the minimum in order to maintain optimum network performance. Control overheads as a result of extra probes from designs which are included in the control messages consume bandwidth. Also, computationally complex algorithms demands significant cycles in wireless devices. Thus, the more complex the algorithm, the more energy or battery power that is consumed. The expected transmission time for the metric defined by Draves *et al.* [10] will incur an overhead which is a function of the number of nodes on the network. For example, in a network of a-nodes and each node with b-neighbours, the number of probes required for the ETT metric is a function of ab [18]. Likewise, the extra probes required for the estimation of transmission time by $ScrRR$ metric by Aguayo *et al.* [17] constitutes an additional overhead which will compete with the limited network resources and eventually hamper network performance.

Hence, this work focuses on a more efficient approach by proposing a RSSI-aware metric. The already embedded RSSI information from the mesh nodes is extracted, processed, transformed and then used in driving the metric of OLSR routing protocol. The next section gives the details of the metric design and methodology.

III. Methodology and Design

A. Experimental study of RSSI

Received Signal Strength Indicator (RSSI) is measured by a simple channel power measurement that a wireless device performs. They are primarily used in 802.11 by wireless transceivers to classify the status of the assigned channel as busy or free and to make decisions based on the channel access protocol to transmit or not. Other common applications include rate adaptation, transmission power control, ADC input pre-conditioning, electromagnetic field monitoring etc. A typical RSSI sample oscillates round a mean value depending on the amount of signal strength received. The main reasons for this are the channel multi-path propagation as well as fading and shadowing of the Radio Frequency channel. Furthermore, there are manufacturer design parameters that could influence its variability and these include antenna orientation, transmitter variability and receiver variability.

Due to the variable nature of RSSI, a preliminary study was carried out to establish the randomness of the RSSI samples. The study is also used to establish that this parameter is suitable for estimating link quality for routing purposes in WMN. Statistical study of this parameter is documented in [19]. Some of the results are highlighted in Figure 1. Figure 1(b) shows the auto-correlation of the measured RSSI samples where $\rho(\Delta t)$ denotes the auto-correlation co-efficient and Δt the time-lag. This is primarily to determine the randomness of the samples under different environmental conditions. The Probability Distribution Function (PDF) was calculated to study the distribution of the samples and determine if the samples are normally distributed.

It was concluded from the study that the auto-correlation for all measured samples are 1 when the time lag is 0. This confirms the non-randomness of the RSSI samples. Furthermore, the RSSI signal level correlates with the link conditions and the variability further deteriorated as the link condition decline. The effect of noise and transmission power levels are minimal on the received signal strength. The PDF shows some inconsistency in distribution. Some samples are normally distributed while some appear like Rayleigh distributions and hence, this requires further investigation. Meanwhile, for the purpose of routing metric estimation, the characteristics of the observed samples shows that with further processing, RSSI could be well qualified to estimate link capacity in WMN.

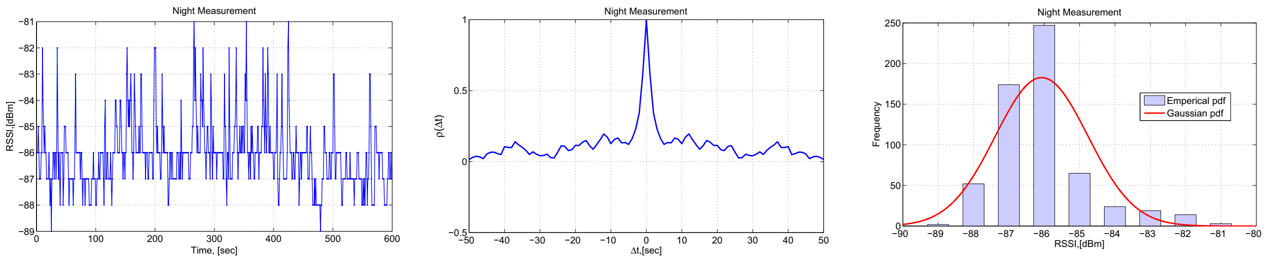
B. Processing of the RSSI samples.

*1) **Extracting the RSSI samples from the mesh nodes**:* Algorithm 1 briefly describes the processes involved in this operation. The station information which also comprises of the RSSI values and the MAC addresses of the neighbouring nodes on the mesh network is extracted. This is achieved by IOCTL calls by MadWiFi driver for Atheros chipsets [20]. IOCTL calls are Input/Output ConTroL calls to provide user-to-kernel interface to the hardware of the wireless cards.

Algorithm 1 - Extract RSSI from the mesh nodes

neighbour_stations = empty list
THREAD *update_average(t)*
WHILE True
station_list = get_station_info_from_card
FOR s **IN** *station_list*
*update_station(s.macAddress, s.*RSSI)
ENDFOR
filter_old_stations
SLEEP t
ENDWHILE
ENDTHREAD

*2) **Filtering the measured RSSI samples**:* Based on the observation of the PDF during the experimental stage of the research, it became necessary to filter the measured sample of the RSSI. In order to follow the signal trend in the samples, an Exponentially Weighted Moving Average (EWMA) filter was proposed and tested. This is a tunable filter with α as filter constant whose values range from 0 to 1. In summary, when the α value is low, the output of the filter follows the value of the last averaged value. Conversely, a high α value gives



(a) Measured RSSI samples (b) Auto-correlation of measured RSSI samples (c) PDF of measured RSSI samples

Fig. 1. Statistical analysis of measured RSSI samples

an output that is sensitive to the present measured value of RSSI. The expression in equation 1 describes how the moving average filter was implemented on the RSSI samples.

$$RSSIi(avg) = RSSIi\alpha + (1-\alpha)RSSIi - 1(avg) \quad (1)$$

where $RSSIi(avg)$ = present average value of the RSSI
$RSSIi$ = present sampled value of the RSSI
$RSSIi - 1$ = last average value of the RSSI

*3) **Conversion and transformation of the RSSI values**:* This process is highlighted in algorithm 2. It should be noted that RSSI measurements are not yet standardized. Therefore there are few variations on how different chipset manufacturers report this value. For example, Cisco uses 100 as RSSI_Max while Symbol chipset uses RSSI_Max of 31 [21]. For this investigation, RSSI values for Atheros chipsets range from 0 - 60 (RSSI_Max) are converted to dBm range (after the filtering process). Hence, the dBm values are then transformed to rates based on the sensitivity limits of the chipsets. It should be noted that algorithm 2 can be adapted to use other chipsets depending on the RSSI_Max of the specified chipset. Also, the conversion to dBm range and sensitivity limit information are provided by the manufacturers. Therefore, this framework can be adapted to any wireless card irrespective of the chipset. Figure 2 describes the sensitivity limits of Atheros chipset AR5212 used on the implementation test-beds.

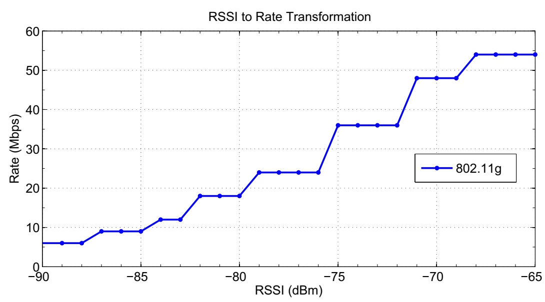


Fig. 2. Received Sensitivity limits for AR5212 chipset

Algorithm 2 - Convert and Transform averaged RSSI values

FUNCTION RSSI_*to_rate*(RSSI)
RSSI_*dBm* = RSSI−95
Transform RSSI_*dBm* to *rate* based on the sensitivity limit of the chipset
ENDFUNCTION

C. RSSI-aware Metric

The RSSI-aware metric proposed is an improvement on the ETX metric but with a different and efficient technique of estimating the expected transmission time. The driving parameter of this metric is R, which is the output of the filtered RSSI-to-rate transformation. Hence, the estimation of the transmission time on a link between two nodes is expressed as t = (S/R). It should be noted that S is a measure of packet size while R is the measure of amount of packet transmission per time. Therefore, the RSSI-aware metric termed $rETT$ for a link between two nodes is expressed as:

$$rETTi = ETX * \frac{S}{R} \quad (2)$$

where S = size of the packet, R = RSSI-to-rate transformation. The R value is based on the received sensitivity limits which are expressions of the link capacity in this metric.

The routing algorithm prefers the link with lesser $rETTi$ in forming the topology for the network. Also, the metric of the path between the sending and the destination nodes in a mesh network is the summation of the respective metrics of the links between the nodes on the transmission path. So the path metric $rETTp$ is expressed as:

$$rETTp = \sum_{i=1}^{n} rETTi \quad (3)$$

D. Dynamics of rETT metric

The benefits of this metric are best appreciated when the link condition is poor which is when hop count algorithm will be inadequate. However, this metric is carefully designed and adaptive to function equally when the link quality is good or improves from a bad condition. This dynamics are illustrated in three distinct cases.

Case 1: When loss probability is variable & RSSI dBm is $<$ -68dBm: This is the default state of the metric and the it will estimate the quality of the link by the expression in equation 2. The transformed RSSI-to-rate can increase or decrease the rETT value which the routing protocol uses in estimating how the topology is formed. The lower the rate, the higher the rETT value of the metric.

Case 2: When loss probability is variable & RSSI dBm is $>$ -68dBm: The RSSI values are generally above the threshold to effect any change in the transformed rate value. Hence, the rate will return 54Mbps for the metric calculation. Here the metric assumes the signal strength of the link is at maximum. Therefore, S/R will remain constant and the only changes in the rETT metric will be as a result of the loss probability which is calculated through ETX estimation. rETT can then be expressed as:

$$rETT_i = ETX_i * k \tag{4}$$

where k=(S/R) and R is the maximum bit rate for IEEE 802.11g (which is 54Mbps).

Case 3: When loss probability is maximum & RSSI dBm is $>$ -68dBm: This is an extreme condition with no packet loss. Every HELLO packet sent is received and acknowledged. When the loss probability is at maximum, the ETX value is 1.0, i.e. the probability of sending and acknowledging the test packets is 1. $ETX = \frac{1}{d_f * d_r}$ and if there is no packet drop in both direction, then $ETX = 1$. Also, the signal strength is at the theoretically maximum value. Therefore, the rETT is constant for the link. Hence the routing process will adopt the classical hop count mode for the metric calculation (OLSR RFC 3626). Therefore, *rETT* will be expressed as:

$$rETT_i = k \tag{5}$$

where k is a constant which is S/R as described in case 2.

The three cases mentioned above can occur at any time in a WMN due to the transient nature of the wireless communication channel. Therefore, it is important for the routing metric to be able to handle varying conditions adaptively as described in subsection III-D.

IV. Experimentation Setup and Implementation

A. Hardware Components

The RSSI-aware metric was implemented on a 5-node WMN. The description of the hardware components for the test-bed are highglighted in Table I.

TABLE I
OVERVIEW OF THE TESTBED

Component	Description
PC	RM Desktop, Intel 2.0GHz, 1 GB RAM
Wireless Card	D-Link DWL-G650, TP-Link TL-WN650
Card Chipset	Atheros AR 5212
Card Driver	MadWiFi 0.9.4
Operating System	Ubuntu 8.04, OpenWRT

B. Software Components

The metric was implemented on OLSR which is a proactive routing protocol. Because of the stationary nature of WMN, OLSR seems to work well on them. OLSR version 0.4.9 was used because this metric is a novel approach and a very stable version was required to avoid unnecessary bugs which may affect experimental results. The configuration file (*olsrd.conf*) was configured with the parameters in Table II. The same interface configuration is required on all the nodes on the WMN. It is important to set the *LinkQualityLevel* to 2 on all interfaces for the OLSR to use the link quality metric for the topology formation.

The implemented metric on OLSR is similar to the plug-in approach documented in [22]. Instead of sending probes to estimate the expected transmission time, the plug-in program processed the RSSI data extracted from the nodes on the WMN. Then the output of the plug-in, *RSSI-to-rate* is passed to *olsrd* every second to complete the process of calculating the metric.

For the throughput test, *iperf/jperf* [23] was used for the measurements.

TABLE II
OLSR CONFIG FILE PARAMETERS

Parameters	Values
UseHysteresis	no
LinkQualityLevel	2
LinkQualityWinSize	10
TCRedundancy	2
MprCoverage	3
HELLO Interval	2.0 s
HELLO Validity Time	8.0 s
TC Interval	2.0 s
TC Validity Time	15.0 s

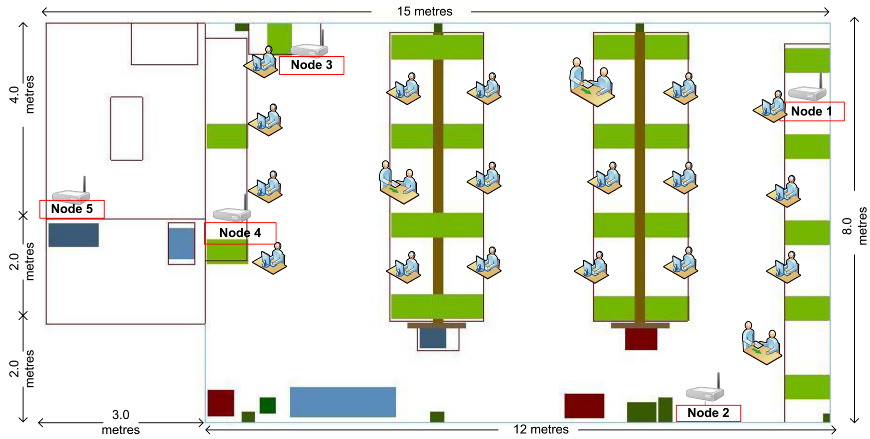
C. Implementation Environment

The implementation environment for this investigation is shown in Figure 3 . This is an open office with frequent human movement; node 5 is located in an annex office separated by a partially glass partition wall. The majority of the investigations were carried out in the afternoon when the office is most occupied and busy. The nodes are distributed in the office as shown. Optimisation, throughput and delay tests were carried out between node 1 (source) and node 5 (sink) while nodes 2, 3 and 4 acted as intermediate nodes.

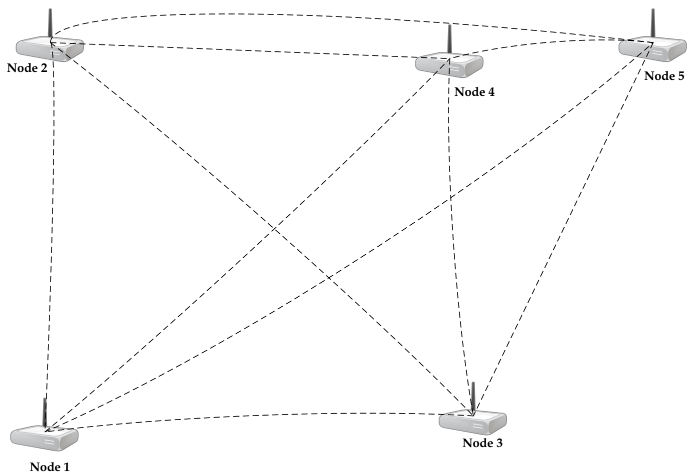
V. Results

A. Filter-constant & Performance Optimization

The first objective was to observe how the value of α can influence the routing process. The transformed *RSSI-to-rate* (R) which is the driving parameter of the metric is compared to the sampled RSSI value and the averaged RSSI value. This was to investigate the effect of α value on the routing process. Due to the variability of the RSSI, the raw value of the samples cannot be used directly to estimate this because every change in its value will trigger a re-computation of the topology table.



(a) Floor diagram



(b) Topology Diagram

Fig. 3. Implementation Environment

This action makes routing unstable and also consumes network resources. Hence, it is important to choose the filter level which will reflect the fluctuations in the link condition and also be stable enough to maintain good network performance.

Figure 4 shows a few of the values of α analysed in this investigation. It can be observed that as the value of α increases, the $rate$ information responds to the changes in the sampled RSSI. However, it is important to find an optimized balance between sensitivity of the rate value and stability of the routing process.

To determine the value of α that will yield an optimised network performance for this scenario, an end-to-end throughput test was carried out. The experiment tested 10 values of α starting from 0.025 to 0.25. Measurement for each value of α was taken in turn and this was repeated ten times. Each test was run for 30 seconds using *iperf* with TCP window size of 0.02Mbytes. The average value of the throughput response for each values of α was plotted in Figure 5.

For this scenario, under the operating condition, the best throughput was achieved when α was 0.125. Although, 0.125 produces a good performance, it is recommended to fine tune this value under different condition to achieve an optimized performance.

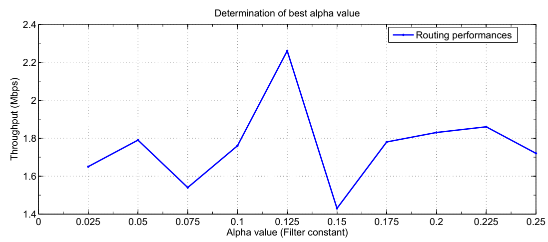
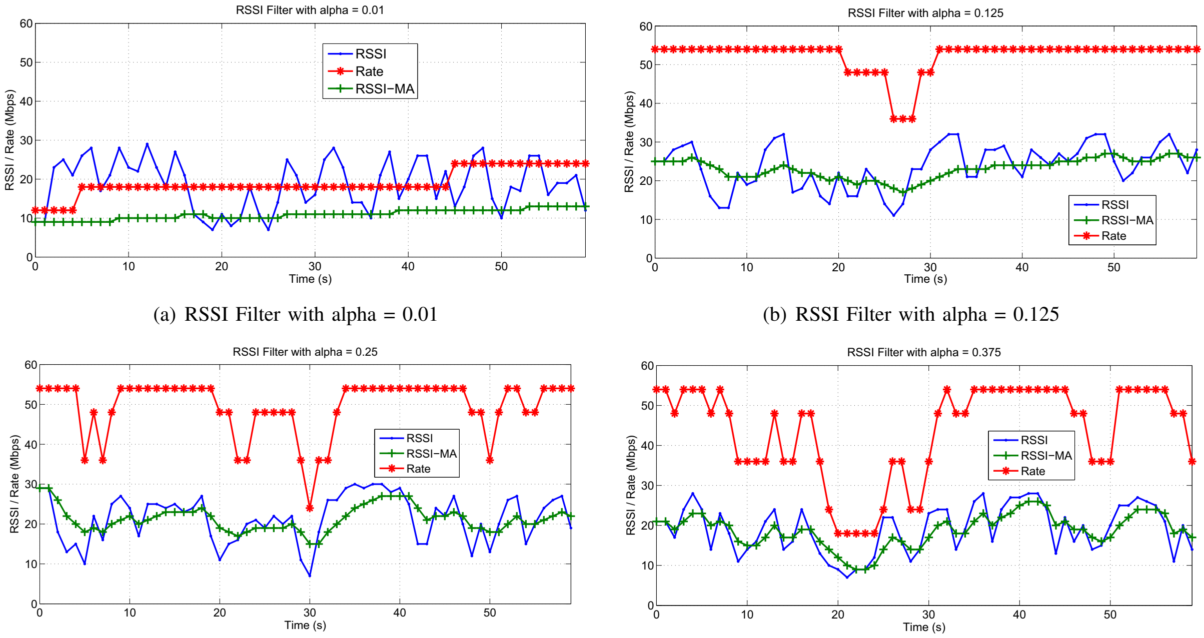


Fig. 5. Performance response with different alpha values

B. Network throughput and Delay

Based on the observation from the throughput test, the value of 0.125 obtained from the previous experiment was configured on the final compilation of the *olsrd*. The performance of RSSI-aware metric was compared with classical hop count metric and the ETX link quality metric.



(a) RSSI Filter with alpha = 0.01

(b) RSSI Filter with alpha = 0.125

(c) RSSI Filter with alpha = 0.25

(d) RSSI Filter with alpha = 0.375

Fig. 4. Figures showing the effects of the filter constants on the transformed Rates

*1) **Throughput Tests**:* Throughtput and delay are two key network performance parameters and it was deemed necessary for the proposed metric to undergo these tests. The throughput test was similar to the one in the determination of α test. OLSR version 0.4.9 can be configured to route packets using shortest path metric as well as ETX. This is achieved by changing the LinkQualityLevel value on *olsr.conf*. ETX will run at level 2 while *RFC3626* mode which is hop count metric will run at level 0. The *end-to-end* throughput test was carried out using hop count metric and then using ETX metric. Maintaining similar link conditions, the test was repeated for the $rETT$ metric. The runs were repeated in turns 5 times for each metric.

The results of the throughput test are displayed in Figure 6. Hop count returned the least average throughput of 0.59Mbps, ETX came next with an average of 1.07Mbps while $rETT$ returned the best of the three with an average of 1.29Mbps. It was observed from the debug output during the routing process that with hop count, all nodes were neighbours irrespective of the link quality. ETX considered the lost probability in the estimation of its link quality but this is only based on `HELLO` packets. $rETT$ estimated its metric not only on lost probability but also considered the link capacity through RSSI measurement to estimate the link quality.

*2) **Delay Tests**:* A similar procedure was followed in terms of the implementation of the OLSR metrics. For the delay test, *ping* statistics were used. This includes the total transmission time for 60 rounds of packets. Also, the minimum, average, maximum and standard deviation of the *echo* transmission time were obtained. However, the average delays for the network under different metrics were recorded. Hop count have the highest delays with 73.04 ms, followed by ETX with 67.17 ms and the least average delay was achieved while running on $rETT$ metric with 49.61 ms. The results are also highlighted in Figure 7.

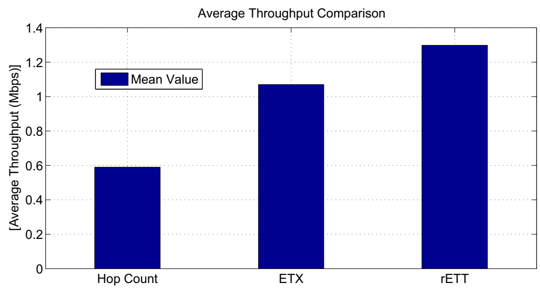


Fig. 6. Average throughput of hop count, ETX and rETT metrics

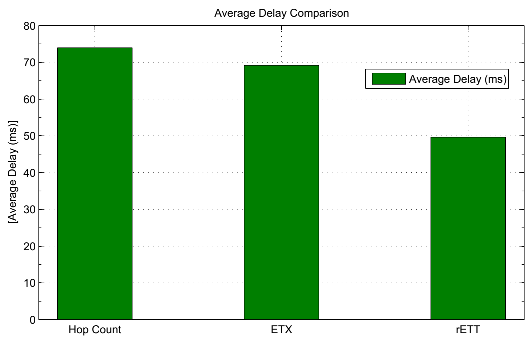


Fig. 7. Average Delay of hop count, ETX and rETT metrics

VI. Conclusion & Future Work

An RSSI-aware metric for WMN has been proposed in this paper. The estimation of the metric has been carried out with an efficient method with the objective of keeping the extra probes at a minimal level. Unlike previous work on the estimation of ETT metrics, this approach uses embedded RSSI information on the wireless card instead of broadcasting additional probes to estimate the metric. The variability of the RSSI samples was addressed and processed for adaptability in the routing process.

The experimental results have shown that the proposed metric performed better than hop count metric and ETX metric under network throughput and delay tests. $rETT$ metric improved the network throughput by 120% when compared with the classical hop count metric and an improvement of 21% when compared with the ETX link quality metric. In terms of delay, the improvement on hop count is 33% while the $rETT$ improved network delay by 28% when compared with ETX metric.

This work can be enhanced by investigating the impact of the incorporation of antenna diversity at all nodes on the network. Also, the possibility of increasing the number of nodes on the WMN test-bed and implementing the metric under different environmental conditions is currently being considered. Other areas of future work include testing *rETT* on a linear topology and comparing the performances with other bandwidth measurement techniques. By implementing this proposed novel routing approach, an interesting technique has been identified for further studies in WMN technologies.

Acknowledgment

This research work was partly supported by GAWIND project (Grant number: MTKD-CT-2006-042783). The authors would like to thank the MobiMESH and ANTLab team for their support in providing the infrastructure for setting up the initial test-bed at ANTLab, Milano.

References

[1] I. F. Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9):S23–S30, 2005.

[2] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, and Robert Morris. Performance of multihop wireless networks: shortest path is not enough. *SIGCOMM Comput. Commun. Rev.*, 33(1):83–88, 2003.

[3] Wenming Song, Shengming Jiang, and Gang Wei. A congestion-aware multipath routing with cross layer design for wireless mesh networks. In *15th Asia-Pacific Conference on Communications, (APCC 2009).*, pages 656–660, 2009.

[4] Xida Shang, Yaoguo Wang, Qiguo Wang, and Linfeng Yuan. Cross layer design of aodv and spatial-tdma protocol in mobile ad hoc networks. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, pages 1–4, 2009.

[5] Dalei Wu, Song Ci, Haiyan Luo, Haohong Wang, and A. K. Katsaggelos. Application-centric routing for video streaming over multi-hop wireless networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, pages 1–9, 2009.

[6] Jia Liu, Tae Yoon Park, Y. T. Hou, Yi Shi, and H. D. Sherali. Cross-layer optimization of mimo-based mesh networks under orthogonal channels. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 49–54, 2007.

[7] M. E. M. Campista, P. M. Esposito, I. M. Moraes, L. H. M. Costa, O. C. M. Duarte, D. G. Passos, C. V. N. de Albuquerque, D. C. M. Saade, and M. G. Rubinstein. Routing metrics and protocols for wireless mesh networks. *IEEE Network*, 22(1):6–12, 2008.

[8] M. Carrera, H. Lundgren, T. Salonidis, and C. Diot. Correlating wireless link cost metrics to capacity. In *Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on*, pages 223–230, 2009.

[9] C. E. Koksal and H. Balakrishnan. Quality-aware routing metrics for time-varying wireless mesh networks. *Selected Areas in Communications, IEEE Journal on*, 24(11):1984–1994, 2006.

[10] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom '04)*, pages 114–128, Philadelphia, PA, USA, 2004. ACM.

[11] Yaling Yang, Jun Wang, and Robin Kravets. Designing routing metrics for mesh networks. In *IEEE Workshop on Wireless Mesh Networks, WiMesh, 2005*, 2005.

[12] A. P. Subramanian, M. M. Buddhikot, and S. Miller. Interference aware routing in multi-radio wireless mesh networks. In *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, pages 55–63, 2006.

[13] Weirong Jiang, Shuping Liu, Yun Zhu, and Zhiming Zhang. Optimizing routing metrics for large-scale multi-radio mesh networks. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 1550–1553, 2007.

[14] E. Alotaibi and S. Roy. A location-aware routing metric (ALARM) for multi-hop, multi-channel wireless mesh networks. In *IEEE Wireless Communications and Networking Conference, 2008. WCNC 2008.*, pages 2081–2086–, 2008.

[15] Tehuang Liu and Wanjiun Liao. On routing in multichannel wireless mesh networks: Challenges and solutions. *IEEE Network*, 22(1):13–18, 2008.

[16] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high throughput path metric for multihop wireless routing. In *ACM Proceedings of MobiCom 03*, 2003.

[17] Daniel Aguayo, John Bicket, and Robert Morris. ScrRR: A high throughput routing protocol for 802.11 mesh networks (DRAFT). Technical report, MIT, 2005.

[18] P. M. Esposito, M. Campista, I. M. Moraes, L. Costa, O. Duarte, and M. G. Rubinstein. Implementing the expected transmission time metric for olsr wireless mesh networks. In *1st IFIP Wireless Days, 2008. WD '08.*, pages 1–5, 2008.

[19] Ebenezer Amusa. *An enhanced Cross-Layer Routing Protocol Extension for Wireless Mesh Networks based on Received Signal Strength*. PhD thesis, University of Bedfordshire, Institute for Research in Applicable Computing, Luton, 2010.

[20] "MadWiFi Project". Available at http://madwifi-project.org/, 2010.

[21] Joshua Bardwell. The truth about 802.11 signal and noise metrics. Technical report, 2004.

[22] "GTA". Available at http://www.gta.ufrj.br/wmn/, 2010.

[23] "Iperf Project". Available at http://sourceforge.net/projects/iperf/, 2010.