

F-ETX: An Enhancement of ETX Metric for Wireless Mobile Networks

Sebastien Bindel^{1,*}, Serge Chaumette¹, and Benoit Hilt²

¹ LaBRI, University of Bordeaux, France

² MIPS, University of Haute Alsace, France

Abstract. The accuracy and the reactivity of link quality estimators are key concepts in Mobile Ad hoc networks (MANETs) and especially in Vehicular Ad hoc Networks (VANETs), since routing process uses link quality related metrics based on neighbour information for making routing decisions. The Expected Transmission Count (ETX) is the most salient metric that assesses the link quality for wireless ad hoc and mesh networks. Its estimation varies according to the link changes like disruption and bidirectional to unidirectional switching. Such changes occur more often in highly mobile networks like VANETs. In order to be efficient routing protocols have to deal with these events. It has been observed that current implementation of ETX leads to a limited accuracy and cannot be efficient in VANETs. In this paper is presented an algorithm which makes the ETX metric more adapted for VANETs. Our contribution significantly improves the reactivity and the accuracy of the metric. In addition, our metric is not limited to the link quality estimation but also brings a link stability information. This results a new metric called Fast-ETX (F-ETX). This metric has been tested under realistic physical layer and mobility patterns to evaluate its reactivity, accuracy and stability properties.

Keywords: VANET, ETX, local link quality, routing metric, dynamic window.

1 Introduction

A wireless ad hoc network results from the cooperation of independent entities to form a decentralized network without Access Points (AP). MANETs are a subset of ad hoc networks consisting of mobile nodes moving arbitrarily. RFC 2501 [4] describes the MANET features and reveals four salient characteristics: dynamic topology, variable link capacity (bandwidth, jitter, error rate, ...), energy constraints and limited physical security. Among the set of current challenges for MANETs, routing is still an important one wherein a dynamic topology implies many rapid path changes. MANET routing protocols attempt to hide the effects of the mobility to higher layers by determining virtual stable paths to transport data. For this, routing processes use values called metric to select the best route.

* This research was supported by a DGA-MRIS scholarship.

Historically only the number of hops was taking into account, without considering the link quality. That is why in wireless environments, metrics like ETX based on link observation give better performances, especially about the packet delivery ratio [5].

Some routing protocols for VANETs (subset of MANETs) such as OLSR (Optimized Link State Routing Protocol) [9] maintain a global topology view to compute the best path. This strategy becomes inefficient if the environment changes rapidly. Other routing protocols such as BATMAN (Better Approach To Mobile Ad hoc Networking) [8] have another philosophy. They maintain both a local and global vision of the network. The local view is based on the local links quality with direct neighbours and the path selection is made from a global point of view. For example, BATMAN uses the ETX metric to estimate the local link quality and determine an end to end path by a computation taking into account the succession of the local link quality estimations through the route.

De Couto et al. proposed the ETX metric in [5] and implemented it in DSDV (Destination Sequenced Distance Vector) and DSR (Dynamic Source Routing) to compare its performances with the "classical" hop count metric. Javaid et al. [6] studied ETX and its variants. They observed that most of metrics were designed for Wireless Mesh Networks (WMNs) and use some techniques to improve performances (i.e. the channel selection to have less interferences). Liu et al. [7] made a proposition to adapt the ETX metric to MANETs. The contribution is based on a novel computation method but requires also a modification of the routing messages. This work is the only one to adapt the metric for MANETs but any propositions explored its reactivity when events occur (e.g. links disruption and links becoming unidirectional). Nodes in VANETs should be quickly aware about link state changes. This is a key issue for a better efficiency in VANET routing protocols and applications [2]. To this end we suggest in this paper a new algorithm with two features. The first one improves the metric reactivity when specific events induced by mobility occur (i.e. link disruptions and bidirectional link becoming unidirectional). The second one deals with the link stability.

The remainder of this paper is divided into four parts. Section 2 reviews the ETX metric and presents an analysis on its behaviour when specific events occur. Section 3 presents the algorithm of the F-ETX and a performance analysis. In section 4 the enhanced metric is tested under realistic physical layers in the ns-3 simulator. Section 5 concludes on the paper and open future works.

2 ETX Metric

The hop count metric is the historical technique used by routing protocols to select the best path to route end-to-end traffic. It has the benefit to be simple and requires low computation. But if a destination can be reached via two paths with the same hop count value, the routing process chooses one arbitrarily. It is therefore essential to take into account the link or the path quality, especially when nodes are mobile and the environment disturbs the wave propagation like in urban setting.

2.1 Basic ETX

The ETX of the link aims to predict the number of retransmissions required to send a packet over this link. ETX is an additive metric. It means that the ETX of a route is the sum of the ETX of each link composing the route. Its computation is based on the delivery ratio d_f which is the probability of a data to be delivered. And the reverse delivery ratio d_r represents the probability that the ACK packet is successfully received back. The expected probability that a message is successfully received and acquitted is $d_f \times d_r$. If we consider a packet transmission as a Bernoulli trial (success or fail), ETX is calculated as following:

$$ETX = \frac{1}{d_f \times d_r} \quad (1)$$

Accordingly, the expected loss estimation is:

$$ETX_l = \frac{1}{(1 - d_f) \times (1 - d_r)} \quad (2)$$

ETX provides information on the link loss ratio in each direction, on the link quality and the presence of asymmetric links when d_r is null.

2.2 d_f and d_r Computation

These ratios are computed by sending broadcast probe packets at an average period τ . A jittered parameter is added up to $\pm 0.1\tau$ per probe to avoid synchronization. A node receiving a probe must rebroadcast it and allows for the originator of a probe to compute the d_r probability. This feature is native for wireless mesh nodes but must be implemented for ad hoc nodes. Besides, a mechanism to avoid a broadcast storm due to the rebroadcasting of probe packets is mandatory. Finally, nodes are using a windowing mechanism to compute the d_f and d_r probabilities. Two types of window are currently used.

- **Temporal windows:** De Couto and all. [5] suggested the use of timers to count probe packets. To compute d_r each node remembers the packet probe sent and triggers a timer. If this probe is received back within a period (i.e. the window size) it is considered as received, otherwise it is considered as lost. The d_f is computed by counting incoming probes native from neighbours every $\pm 0.1\tau$ time. The main benefit of this method is the feedback about the delay set with timers maintained after each probe packet that has been sent, until its reception. However, a mechanism (e.g. a garbage collector) must be implemented to delete too old timers in order to not overload nodes with expired timers.
- **Sequence Number (SN) windows:** Another method for computing d_r and d_f is based on SN. In order to determine the freshness of the carried information, current routing protocols include a SN field. The last version of BATMAN algorithm (i.e. BATMAN IV) uses this information to maintain windows for computing delivery and reverse delivery ratios [8]. Each node

sends periodical probe packets that contain a SN representing its freshness and marks the packet received in the current window according to its SN. After each sending, if the maximum size is not reached, the window is increased. Otherwise it is shifted by one to maintain a fresh window. Algorithm 1 describes the basic method to fill up a window when a probe packet is received.

Algorithm 1. Management algorithm for a SN window

INPUT: SN value extracted from the probe packet

```

1: function MARK_RECEIVED_PACKET(window, SN value, MAX_WINDOW_SIZE)
2:   if SN value < window.first_value then
3:     drop information
4:   else
5:     if SN value < window.at(window.size-1) then
6:       window.at(SN value-window.at(0)) ← marked as received
7:     else
8:       counter ← (SN value-1)-window.at(window.size-1)
9:       while counter > 0 do
10:        slide window
11:        --counter
12:      end while
13:      if window.size < MAX_WINDOW_SIZE then
14:        window ← increase by one
15:      else
16:        window ← slide by one
17:      end if
18:      window.at(window.size-1) ← marked as received
19:    end if
20:  end if
21: end function

```

Nodes compute ETX by maintaining a window for each probability (d_f and d_r). Each probability is computed by counting all packets marked as received in the associated window F having the window's size ς . In the following equation, F_i is the value stored at the i^{th} cell of F .

$$probability = \frac{\sum_{i=1}^{\varsigma} F_i}{\varsigma} \quad (3)$$

Temporal windows have the advantage to provide information on delays. However, in order to determine if an incoming packet is expected, it must be identified (commonly by a sequence number). A node must extract from the packet the sender's address, the packet ID and its transmission time. Because only one timer is required to maintain the window freshness, methods based on sequence numbers are less complex than the temporal solutions. To determine if an incoming

packet is expected, a node inspects only the sender's address and the sequence number fields in the packet. As a result, the computation task is reduced, increasing the node's capabilities and reducing its energy consumption.

2.3 Analysis of ETX

To improve the time detection of ETX, two approaches are possible. The first one is to reduce the sending interval. But this method leads to increase the number of probe packets per second, overloading the network. In addition, this method is limited by the media access technique and the number of nodes in the network. Another method is to use different window sizes, which is believed as more interesting. The impact of different window sizes for link events like the emergence, the disruption and a bidirectional link becoming unidirectional have been analyzed and tested. For each case, a theoretical analysis is presented with a plot to support our demonstration.

We choose to study the ETX_l for a local link evaluation. Simulations have been performed under JbotSim [3] offering a simulation environment for prototyping distributed algorithms in dynamic networks. It makes a perfect sense here because we test only the behaviour of ETX_l for a link without taking into account any additional information. In our scenarios a periodic interval of one second for sending probe packets has been defined. These packets contain three fields; the current sequence number, the node ID that initiated the message and the previous sender ID. A node originates a probe packet by fixing its current sequence number and populating the second and third fields with its ID. When a neighbour rebroadcasts a probe packet it updates only the previous sender field with his own ID. A node drops a packet if it is not the originator of the incoming rebroadcasted packet. This feature guarantees a local link evaluation only.

Link Emergence A link is discovered as soon as a window marks a packet as received (see algorithm 1). According to the equation 4, a packet sent has a 100% chance to be lost if the value of ETX_l converges to 1. At the opposite, a packet has a 100% chance to be delivered if the value of ETX_l converges to $+\infty$.

$$\lim_{\substack{d_f \rightarrow 0 \\ d_r \rightarrow 0}} ETX_l = 1 \qquad \lim_{\substack{d_f \rightarrow 1 \\ d_r \rightarrow 1}} ETX_l = +\infty \quad (4)$$

The maximum value of ETX is reached when the windows of d_r and d_f get to their maximum size. However, because packets are returned after a RTT (Round Trip Time), the growth of d_r window is slower than the one of d_f window. Therefore results a time offset between the d_f and the d_r window. Whereas all nodes send periodically probe packets over an interval noted P and the RTT (Round Trip Time) of their packet is RTT_b . Since windows have a maximum size n , the maximum ETX value is reached after a $nP + RTT_b$ period. Figure 1(a) shows the impact of the window size on the convergence time to reach the

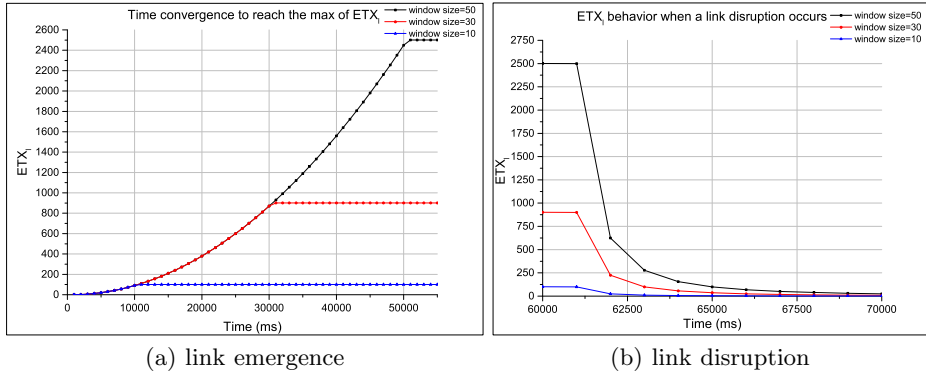


Fig. 1. Emergence and disruption

maximum value of ETX. Note that we limited the max ETX value to the sum of the d_f and the d_r windows size + 1.

Link Disruption. All packets lost or dropped are marked by a null value in the window. Packet losses under JBotSim have been simulated through link disruptions because it does not support error probability models. A link is declared as disrupted by ETX when windows have marked all expected packets as lost (i.e. as null). This behaviour is different from the emergence case because a disruption can occur when a node did not receive either or both its own and neighbour's probe packet. The time to detect a link disruption varies from the previous case. As a result, the time to declare a link as disrupted is in the interval $[nP, nP + P]$. We have simulated several window sizes and triggered a link disruption even when the maximum value of ETX has been reached. The result is shown in figure 1(b).

Unidirectional Link Detection. The reverse delivery probability (d_r) measures the capability of neighbours to send back a probe packet. If a link is unidirectional, a node only receives probe packets originated by its neighbours (d_f) but does not receive back its own probe packet from its neighbours (d_r is null). Therefore the detection of bidirectional links becoming unidirectional depends on the window size maintained for d_r . We have simulated a bidirectional link becoming unidirectional and traced the behaviour of d_r with different window sizes in figure 2.

Discussion. It is observed that the window size has an impact on the convergence time and the granularity of ETX values. With a small window size, the ETX reaches its maximum and minimum quickly because the windows are filled up faster. However, the convergence with larger windows is significantly slower. The resulting granularity is improved (i.e. more different ETX values are available). For example, a window with a size of 5 is efficient to declare quickly a

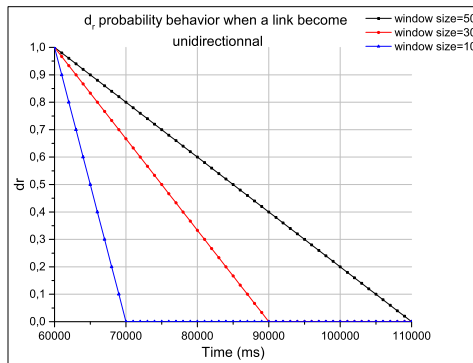


Fig. 2. unidirectional link

disruption. This is especially adapted to evaluate link quality in highly mobile networks. But it provides a poor granularity to estimate the link quality (only 5 values). A window with a size of 60 has a better granularity (i.e. 60 values) but has a slower convergence time to declare for example a link disrupted. This type of window is more adapted for estimating the quality of static or semi-static links. Therefore, the classical ETX version implies a trade-off between the convergence time and granularity and it becomes difficult to give an efficient link estimation for both volatile and static links.

3 Computation of F-ETX

We have profited from the experience of TCP congestion-avoidance algorithm to design an algorithm which improves the reactivity and the accuracy of ETX. Our goal here is to find a trade-off between a fast convergence and a good accuracy. We suggest here a new algorithm with two features; one dealing with the convergence time and another one, with the metric accuracy. This algorithm has been tested with the JbotSim simulator where we have simulated a direct communication between two nodes. We present each improvement independently from each other although they are gathered into the same algorithm.

3.1 Convergence Time Improvement

The moving average is an analysis technique used in statistics to calculate an average over a certain period of time. Unlike the simple average method, where the calculation is made with the full data set. Its moving version calculates the average over a last period of time. This approach has the main benefit to give more importance to the newest values in the data set ensuring a higher freshness of the analysis. Even if this method is used in the current implementation of ETX, its limitations have been demonstrated in the previous section. To deal with this problem the use of the weighted moving average method is suggested.

We choose to weight data according to the number of detected losses. This strategy penalizes probabilities when the last packet expected is lost and increases as the number of losses grows. Let the binary weight called w^{nth} weighting n -day values x_n, \dots, x_1 (i.e consecutive expected SN), α is the number of consecutive packets marked as lost in the window and F_n the window size at the n day moment. The number of data weighted with a non null weight is determined by the equation 5.

$$\theta = \frac{F_n}{2^\alpha} \quad (5)$$

Until losses are detected, probabilities are computed following the equation 6 wherein each data in the window is weighted with a binary value determined by the equation 5.

$$probability = \frac{w^n x_n + w^{n-1} x_{n-1} + \dots + w^1 x_1}{\sum_{i=1}^n w^i} \quad (6)$$

This process maintains the same weight when θ is one and takes over as soon as the last expected packet is received. This approach can be presented in a different way to provide a simpler implementation for nodes. Our weighting method puts a positive weight on data according to the half of window size and leaves the others with a null weight. In doing so, our algorithm can be also seen as successive reductions. For each expected packet detected as lost, the window size is halved to keep the newest values. The algorithm has been tested both when a disruption occurs and when a bidirectional link becomes unidirectional. Table 1 sets out the convergence time of the classical ETX and our enhanced version.

Table 1. Detection time comparison

Maximum window's size	Time for detection (classical ETX)	Time for detection (Enhanced ETX)
50	50s	5s
30	30s	4s
10	10s	3s

Theses results show a improvement of the detection time with our algorithm.

3.2 Accuracy Improvement and Link Stability Estimation

As indicated earlier, as soon as an expected packet is lost, the size of the window is halved. Sporadic losses have a low impact on the window size but, a link disruption which implies continuous losses, reduces significantly its size. We suggest an algorithm that uses the window size to adjust the link quality estimation, also

called the link stability evaluation. Figure 3 shows the decision logic scheme of this algorithm. It uses two variables; the $\$Threshold$ that contains the window size when a loss occurs and the $\$Counter$ that counts received packets when the current window size is equal or higher than $\$Threshold$.

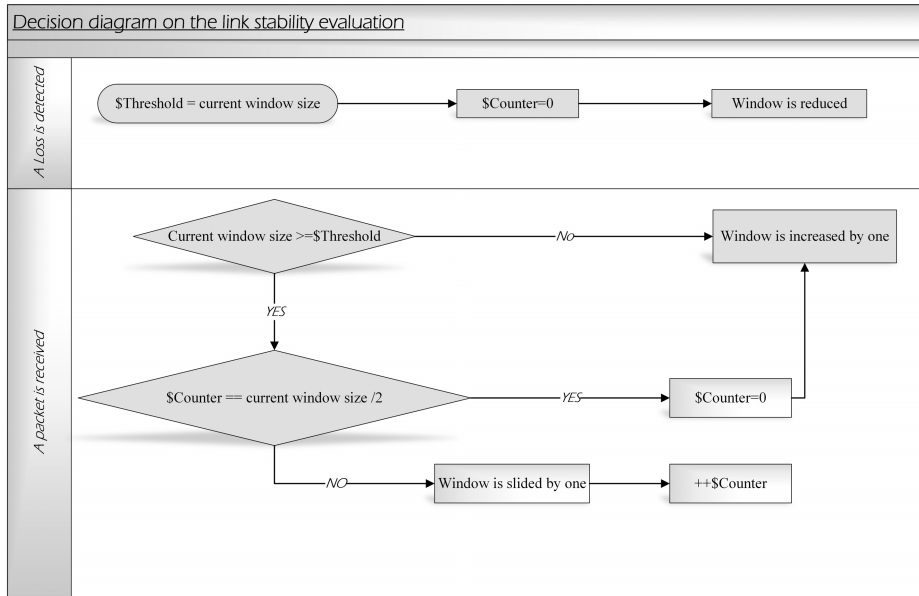


Fig. 3. Decision diagram of the link stability computation

When a loss appears, the algorithm saves the current window size (also called the threshold) before being halved. Until the current window size is lower than the threshold, for each new received packet the window size is incremented by one. If the window size reaches the threshold, the algorithm switches to another strategy. For each new incoming packet, the window is shifted by one and a counter ($\$Counter$) is incremented. If counter value reaches the half of the current window size, the window is incremented by one and the counter is reset. The algorithm restarts this process until the maximum window size is reached. As a result, the growth of the window follows a stepwise according to the reception of successive packets. This achievement results from a link stabilization and a correlation between the window size and the link stability (bigger is the window size, higher is the stability) can be observed.

To show its efficiency the algorithm has been simulated under JBotSim in two situations. The first one is the disruption of a link and the second one is when a bidirectional link becomes unidirectional. Our study is based on the behavior of a static window with a maximum size of 30 SN (classical ETX case) and a dynamic window (its maximum size is also fixed at 30 SN) managed by our suggested algorithm. Results shown in figure 4 represent the values of ETX_l normalized

to $[0, 255]$. Red rectangles indicate the occurrences of link disruption. A better detection of the link disruptions with a metric based on the dynamic window size (step falling edge) is observed. It is also observed that the metric takes 4 seconds to declare the link as disrupted. Besides the classical ETX metric is not enough reactive to adapt its estimation to the dynamic of the link (see figure 4(a)). The figure 4(b) shows the second situation where the bidirectional link becomes unidirectional as indicated in the purple rectangle. Unlike disruptions, this event is not detected by the static window used by the classical ETX. Despite the d_r window marks expected packets as lost, the ETX_l formula (see equation 2) needs to have significant losses in windows for d_f and d_r to change its estimation. But with the dynamic window the changing to an unidirectional link is detected. Like the disruption case the detection takes 4 seconds.

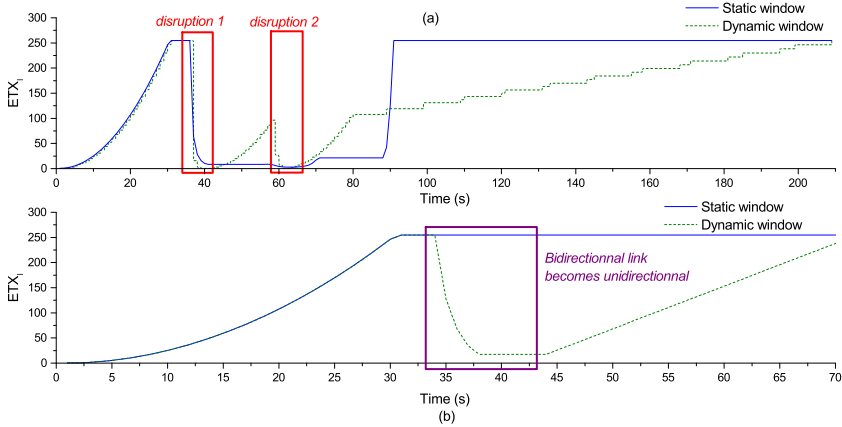


Fig. 4. Classical and enhanced ETX_l comparison

Our enhanced metric based on dynamic window size is clearly more reactive and sensitive. It is explained by the reduction process that highly regresses the metric value by removing old values in the windows. Unlike the static window size has to wait to be filled up with a null value, representing the loss of all expected packets. If the link reappears the dynamic window has the best performance with a link evaluation process adapted faster to a link upcoming. However a static window keeps for a long time too old expected packets. Therefore a disrupted link is considered as a persistent during all the time of convergence. It can be also observed the stair effect introduced by our algorithm when the windows (for d_r and d_f) reach their thresholds (right part of figure 4). This is the result of the link stability process that gradually increases the metric value.

To conclude, our algorithm improves the behaviour of the classical ETX metric by making it more reactive, sensitive and gives information about the link stability. These new features make the metric more responsive to link volatility like in VANETs. Therefore, we call this enhanced metric F-ETX for Fast-ETX.

4 F-ETX Experimentation Under Realistic Condition

This experimentation aims to make a statistical evaluation on the granularity of F-ETX and classical ETX_l with three static window sizes (5, 25 and 64 packets). We have implemented these metrics into the BATMAN routing protocol that we have added into ns-3 [1]. Our evaluation scenario places 20 mobile nodes on a 2D grid. A realistic physical propagation model is used. It is based on a Three-LogDistance attenuation model ($d1=39m$, $d2=120m$, $exp0=2$, $exp1=2,076$) and a Rayleigh fading model (Nakagami model with $m=1$). Communications use the 802.11p standard. The gain of the antennas are set to zero meanwhile the Tx power level is set to 1dBm and the Tx power start and the Tx power end values are set to zero. Figure 5 summarizes with the box-and-whiskers the ability of metrics to estimate the link quality. The results below show the estimations given on all links during the simulation by the ETX_l metric, including three static window sizes fixed at 64, 30 and 5 SN. And the F-ETX (dynamic window) with a maximum window size fixed at 30 SN.

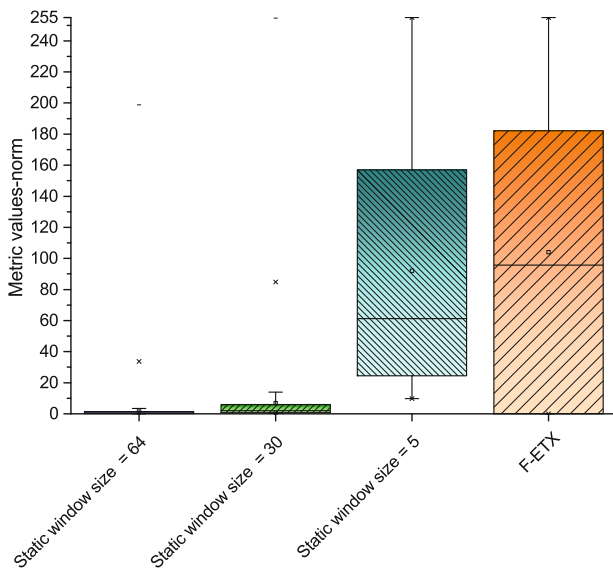


Fig. 5. Ability of ETX_l and F-ETX to estimate link qualities

It is observed on the static window sizes, the influence of window size on the distribution of values taken by ETX_l . With a larger window size, the values taken by ETX_l are more present in the minimum bound. This observation is explained by the fading and the disruption effects which drop a lot of expected packets conducting to low metric values. Besides, a shortest window size shows a better distribution on the metric values (cf. fig 5). The F-ETX metric appears

as the most relevant. Firstly, the box-and-whisker of the F-ETX metric covers more values than other metrics. Secondly, F-ETX has an higher median value than the best static window size (cf. 5 SN). This means that the fading has a lower impact on the F-ETX metric values than done by ETX_l with static window sizes. In addition F-ETX has the best distribution set with a median value close to the mean. This is the result of the use of a dynamic window whose size is self adapted by the link stability process described in section 3.2.

5 Conclusion and Future Works

In this paper we have presented the F-ETX metric, an enhancement of the classical ETX metric adapted to the fast changing environment like in VANETs. The metric uses dynamic windows and a process based on the link stability. The results from tests under JBotSim and ns-3 have supported our approach. The F-ETX metric appears to be more reactive, accurate as the classical ETX metric. This makes F-ETX a very good candidate for a routing protocol metric. Future works will lead to integrate F-ETX in routing protocols in order to make intensive tests. This should confirm that F-ETX is a new efficient routing metric adapted to MANETs and VANETs.

References

1. The ns-3 network simulator, <http://www.nsnam.org/>
2. Belyaev, E., Vinel, A., Egiazarian, K., Koucheryavy, Y.: Power control in see-through overtaking assistance system. *IEEE Communication Letters* 17(3), 612–615 (2013)
3. Casteigts, A.: The jbotssim library. CoRR abs/1001.1435 (2013), <http://arxiv.org/abs/1001.1435>
4. Corson, S., Macker, J.: Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. RFC 2501 (January 1999), <https://www.ietf.org/rfc/rfc2501.txt>
5. De Couto, D.S.J., Aguayo, D., Bicket, J., Morris, R.: A high-throughput path metric for multi-hop wireless routing. In: *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pp. 134–146. ACM, New York (2003), <http://doi.acm.org/10.1145/938985.939000>
6. Javaid, N., Javaid, A., Khan, I.A., Djouani, K.: Performance study of etx based wireless routing metrics. In: *2nd International Conference on Computer, Control and Communication, IC4 2009*, pp. 1–7 (February 2009)
7. Liu, Y., Mi, Z.-C., Zhang, J.-F., Qu, X.: Improvement of etx metric base on olsr. In: *2010 International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–4 (October 2010)
8. Quartulli, A., Cigno, R.: Client announcement and fast roaming in a layer-2 mesh network. Tech. rep., University of Trento (October 2011), <http://eprints.biblio.unitn.it/2269/1/report.pdf>
9. Clausen, T., Jacquet, E.P., Optimized, E.: link state routing protocol (olsr). RFC 3626 (October 2003), <https://www.ietf.org/rfc/rfc3626.txt>