Effective link quality estimation as a means to improved end-to-end packet delivery in high traffic mobile ad hoc networks

Digital Communications and Networks 3 (2017) 150–163

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.elsevier.com/locate/dcan

Effective link quality estimation as a means to improved end-to-end packet delivery in high traffic mobile ad hoc networks☆

Syed Rehan Afzal[a,*], Sander Stuijk[a], Majid Nabi[a], Twan Basten[a,b]

[a] Eindhoven University of Technology, Den Dolech 2, 5612AZ Eindhoven, The Netherlands
[b] TNO-ESI, Embedded Systems Innovation, De Rondom 1, 5612AP Eindhoven, The Netherlands

ARTICLE INFO

Keywords:
Asymmetric link quality
Link-quality measurement
Wireless ad hoc networks

ABSTRACT

Accurate link quality estimation is a fundamental building block in quality aware multi hop routing. In an inherently lossy, unreliable and dynamic medium such as wireless, the task of accurate estimation becomes very challenging. Over the years ETX has been widely used as a reliable link quality estimation metric. However, more recently it has been established that under heavy traffic loads ETX performance gets significantly worse. We examine the ETX metric's behavior in detail with respect to the MAC layer and UDP data; and identify the causes of its unreliability. Motivated by the observations made in our analysis, we present the design and implementation of our link quality measurement metric xDDR – a variation of ETX. This article extends xDDR to support network mobility. Our experiments show that xDDR substantially outperforms minimum hop count, ETX and HETX in terms of end-to-end packet delivery ratio in static as well as mobile scenarios.

1. Introduction

The ascent in modern day broadband and fiber optic speeds has given birth to a range of communication intensive applications. Handheld devices are predominantly on the source or sink of the content while the rest of the network comprises plethora of wired/wireless relaying and routing infrastructure. Mobile ad hoc networks (MANETs) are different. In MANETs, mobile nodes act as routers and employ the radio channel to discover their neighbors. With a sufficient number of in-range nodes, a multi-hop network emerges, capable of high traffic communication without the need of any base station or centralized authority. Each node can be static or mobile and enters or leaves the network at any time. This ease of use, self-sufficiency and self-sustenance of MANETs makes them an ideal network to use when and where the existing infrastructure is either overloaded or unavailable. However, due to inherently lossy and unpredictable wireless communication medium as well as the fact that the topology is dynamic, each node needs to perform additional control functions in order to utilize the network's full potential.

The goal of quality driven routing is to achieve a more deterministic behavior of the network apparatus. It enables communicating applications to efficiently utilize the network resources resulting in improved performance in terms of delivery ratio, bandwidth, latency, throughput etc. Accurate link quality estimation is pivotal and paramount for the success of quality driven routing. A number of Link Quality Estimators (LQEs) have been proposed over the years [2] where throughput and packet delivery ratio (PDR) have gained special interest. Among them, the Expected Transmission Count (ETX) [4,7,19,24] metric has been one of the most widely used metrics aimed at improving end-to-end throughput in wireless networks.

We originally employed ETX in our end-to-end quality of service provisioning framework to facilitate applications in selecting routes with higher end-to-end PDR. Our experiments discovered little correlation, even in static scenarios, in ETX link estimates with respect to actual data delivery experienced over the respective links. We observed the causes of this disparity and through experimentation highlighted the fundamental attributes contributing to weak correlation between ETX estimates and actual delivery ratio. We presented these findings along with our link quality estimation metric xDDR (estimated Directional Delivery Ratio) in [1]. xDDR estimates link quality reception of proactive unicast beacons across a fixed length and varied positioned window, where the position of the window is dictated by the flooding traffic. We ran a series of experiments in a static environment (no mobility) and found xDDR to score highest among ETX, HETX [4] and minimum hop count metric in terms of overall yielded end-to-end packet delivery. This article furthers our work and makes the following additions to the prior work.

☆ Peer review under responsibility of Chongqing University of Posts and Telecommunications.

* Corresponding author.
E-mail address: s.r.afzal@tue.nl (S.R. Afzal).

http://dx.doi.org/10.1016/j.dcan.2016.12.001
Received 24 December 2015; Received in revised form 15 September 2016; Accepted 9 December 2016
Available online 10 December 2016

1.1. Mobility support

The scope of xDDR [1] work is limited to static networks only. This is due to the fact that periodic beaconing in xDDR is based on unicast packets, whereas schemes such as ETX and HETX employ broadcast packets. Broadcast based schemes, by virtue of the broadcast, are able to add new nodes automatically to the neighborhood table. Broadcast packet transmission may not be the most accurate means to estimate link quality [1,16] but, as it turns out, it is arguably the best way for in-range neighbor discovery in a dynamic topology. For a unicast based scheme such as xDDR, we therefore devise a very lightweight pulse reception quality based neighbor update and classification scheme that uses a cyclic small burst duration of broadcasts in the case of dynamic mobile topologies. This scheme refreshes the neighborhood tables once every cycle. Our experiments in mobility scenarios demonstrate that this overhead is justified for applications seeking more accurate estimates and higher end-to-end packet delivery in comparison to broadcast based quality estimation schemes.

1.2. Demonstrate metric's impartiality to Routing protocol

xDDR as a metric, in principal, is independent from the routing protocol employed. In order to verify it, besides Dynamic Source Routing (DSR) protocol [14] based implementation, we extend the Ad hoc On-Demand Distance Vector (AODV) routing protocol [23] with xDDR. The destination node of a route discovery packet in DSR receives the entire source route containing IDs of every node traversed on the route. AODV, on the other hand, only provides the information of previous hop node. This also means DSR nodes maintain a neighborhood table containing routes to every node on the network whereas AODV nodes only log the ID of the next hop node needed to forward the packet to the intended destination. Our results show that despite their dissimilar design performance, both schemes were able to yield comparable gains by using xDDR as a quality estimation metric.

The rest of the article is organized as follows. We discuss the different link quality based metrics as our related work in Section 2, while Section 3 formulates the generic network attributes common between quality estimation in mobile ad hoc networks and network communication in general. Section 4 analyzes ETX metric and related MAC layer behavior in detail as motivation for our quality estimation metric. In Section 5 we present a detailed description and functions of our proposed metric xDDR (expected Directional Delivery Ratio). In Section 6 we present the end-to-end quality driven routing framework we devised to test and experiment with the existing and our proposed metric. Section 7 demonstrates the experimental setup and results. We analyze and compare the accuracy of the estimates as well as its impact on end-to-end packet delivery ratio.

2. Related work

Several research efforts have been conducted in the past in the domain of link quality estimation. Proposed schemes vary from single to multi-layer solutions (from the OSI Layer perspective). Authors in [26] categorized software based LQEs into three main categories, namely, Packet Reception Ratio (PRR), Required Number of Packet transmissions (RNP) based and score based schemes.

PRR based scheme use packet reception ratio as the estimation metric which is in essence the ratio of number of received packets to number of sent packets. Besides the original PRR ratio, Window Mean with Exponentially Weighted Moving Average (WMEWMA) [27] is another popular receiver-side LQE based on a passive monitoring scheme. It applies Exponentially Weighted Moving Average (EWMA) and a Kalman filter on PRR to smooth it. Results have shown the scheme to be fairly responsive to major link quality changes thereby providing a metric that is resistant to the transient fluctuation of PRRs. Furthermore, Some LQEs provide a link estimate that does not refer to physical phenomenon (like packet reception or packet retransmission); rather, they provide a score or a label that is defined within a certain range. Most of these schemes employ training and classification, regression and fuzzy logic to classify a link with a score or labels. However similar to PRR based schemes they also perform passive estimation. In other words, they employ control information to data packets in the form of in-band signaling. Unlike wireless sensor networks, mobile ad hoc network topology is more predictable and independent. We are therefore interested in active or proactive estimation schemes which provide on-demand link estimates.

This brings us to RNP [28] based schemes that categorize sender-side proactive LQEs that estimate the average number of packet transmissions (or retransmissions) required to result in successful reception. Notable schemes in this domain include Four-bit [29], ETX (Expected Transmission count) [7] and ETT (Expected Transmission Time) [13] and HETX [12]. Among these Four-bit presents a cross-layer solution whereas the rest are Network Layer solutions. In terms of link latency estimation, RTT [10] calculates Round-Trip Time between neighboring nodes by sending beacon packets at periodical times. Receiving nodes acknowledge the packet by a reply encapsulating the timestamp of reception. The calculation of RTT thus accounts for queuing delay, channel contention and lossy links. In Packet Pair delay (PktPair) [11], each node uses two back-to-back packets (one small, and one large) to its neighbors. The receiving node in return calculates the delay between the two packets and reports it back to the sender. The sender node then uses an exponentially weighted moving average of these delays with respect to each neighbor. The routing algorithm then chooses the path with the minimum delay. The advantage of PktPair over RTT is that it eliminates the queuing delay problem that exists in RTT [5].

Four-bit [29] is devised to be used by routing protocols and provides four bits of information that are contributed by different layers. The white bit is from the physical layer. The ack bit is from the link layer and indicates whether an acknowledgment is received for a sent packet. Finally, the pin bit is contributed by the network layer and is used for the neighbor table replacement policy. It combines two metrics *(i) estETXup*, as the quality of the unidirectional link from sender to receiver, and *(ii) estETXdown*, as the quality of the unidirectional link from receiver to sender. Although this scheme is shown to perform significantly better in an end-to-end environment, we believe such a cross-layer solution would require tweaking in the physical link and network layer making it very difficult to implement in ad hoc networks applications.

The ETX metric is a routing metric particularly for finding high end-to-end throughput paths in multi-hop wireless networks by measuring link-level packet loss rates. ETX has been extensively shown to outperform other routing metrics in yielding higher throughputs in static wireless multi-hop networks [15–17]. Draves et al. [5] compared minimum hop count, RTT, PktPair, and ETX for setups with varying bandwidths on the DSR [14] routing protocol. They established that ETX outperforms the other three metrics in a static wireless network.

As our scheme is similar to ETX, we will discuss it in more detail in the following section. The ETX of a link is the estimated number of retransmissions required to send a packet over that link. ETX of a link is written as ETX(link)=$1/đ_f \times đ_r$. DeliveryForward ($đ_f$) is the probability of successful packet delivery in the forward direction while the opposite direction is DeliveryReverse ($đ_r$). To calculate $đ_f$, each node broadcasts beacon messages with period ETXTimeInterval (ε) to its neighbor nodes. $đ_f$ then is the ratio between number of beacons sent and number of beacons received within a fixed duration window *etxWindowSize* ($\hat{e}$) by the neighbor. Similarly, $đ_r$ is the rate of success of beacon messages in the opposite direction. ETX of a route is the sum of ETX of all the links comprising that particular route. Among routes competing for the same stream, the route with the minimum ETX sum is recognized to concede the highest throughput. Although, in our observations regarding ETX's throughput performance in mobile ad

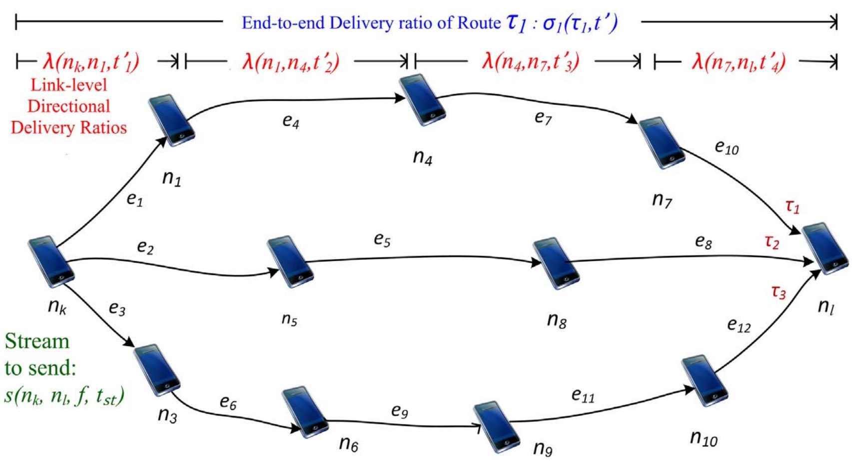


Fig. 1. Network topology.

hoc networks, we found that ETX is not always the best in estimating link quality. Similar observations sometimes discussed in different contexts have also been reported in the past [19].

Waharte et al. [18] revealed that the throughput from ETX is similar, and in some cases even worse than using a hop-count metric. They conclude that in case of increased traffic load, the interference avoidance strategy of ETX does not lead to a better end-to-end throughput. It was inferred that this is because it selects longer paths which add more self-interference and may lead to flow starvation [5]. Furthermore [19] and more recently [4] found that during high traffic scenarios throughput of ETX deteriorates considerably, mainly due to collisions between route discovery broadcast (RREQ) packets and ETX beacons. Tran et al. [4] proposed HETX which circumvents this effect by calculating ETX over a previous window if the current *ETXwindow* overlaps with RREQ discovery times.

Hardware based LQE such as Received signal strength indicator (RSSI) and Signal to Noise Ratio (SNR) let a given node calculate the signal strength or signal to noise ratio of packets within the node's communication range. Hong et al. [33] propose a link quality prediction model (LQP) to sense the link state by analyzing the Signal to Noise ratio of the neighbor links. LQP relies on the route handoff delay which represents the link quality change sensitivity, and use SNR to predict it. Since most modeling predictions require larger data samples, LQP addresses the small sample size problem using Grey Theory [34].

Authors of [31] proposed a reactive LQE real-time End-to-End Delay Estimation Metric (EEDEM) that uses delay as a metric to achieve high throughput end to end routes. It takes into account queuing delay, transmission delay and layer delays within an individual node. Delay accounting is accomplished by placing 9 timers at parts of code in the application, network, MAC and PHY layers from where the data packets pass through. Inter-Network-Layer-Delay (INLyD) [16] is a reactive LQE. Using two timers INLyD nodes measure aggregate of all the delays incurred by a data packet and classifies link quality based on aggregate average delay a link incurs to transmit data. End-to-end INLyD of a route is the sum of INLyDs of all links included. The destination node picks the route with the minimum end-to-end INLyD.

3. Generic network attributes

This section enlists network architecture and related attributes which are generic regardless to the estimation metric being employed. Our network topology is modelled as a directed graph comprising nodes and edges (N, E) transmitting connectionless UDP streams between each other where $N=\{n_1, n_2, n_3,...\}$ and $E=\{e_1, e_2, e_3,...\}$. An edge represented as $<n_1, n_2>$ is a directional link from *rootNode* $n_1 \in N$ to *childNode* $n_2 \in N$ ($<n_1, n_2> \neq <n_2, n_1>$) when node n_2 is within the transmission range of n_1 (1-hop neighbor). We use $\mathbb{N}$ (not to be confused with the set of nodes N) to represent the set of natural numbers, $\mathbb{R}$ represents the set of real numbers and $\mathbb{R}^{\geq 0}$ to represent the set of non-negative real numbers. Between a pair of in-range nodes such as n_1 and n_2, we compute the directional delivery ratio estimate represented as $\lambda(n_1, n_2, t') \in \mathbb{R}^{\geq 0}$ with t' the current timestamp. Moreover $\lambda(n_1, n_2, t')$ will often not be equal to $\lambda(n_2, n_1, t')$ due to the asymmetry in link delivery ratios widely observed in wireless mobile communication [3,25] and $0 \leq \lambda \leq 1$.

For a given node n_m, we have $E_m \subseteq E$ consisting of directional edges with the first-hop neighbors of n_m where $m \in \mathbb{N}$. We have a set of connectionless UDP streams S, to transmit from a source to a specific destination node. A given stream $s \in S$ is described as follows:

$$\forall\ s \in S = s(n_k,\ n_l,\ f,\ t_{st}) \in N \times N \times \mathbb{R}^{\geq 0} \times \mathbb{R}^{\geq 0}$$

where, n_k is the stream source, n_l is the stream destination, f is the data rate of the transmission and t_{st} represents the start time. For a given stream $s \in S$, $\mu(s,t')$ denotes the total number of packets transmitted till timestamp t'. Similarly $\delta(s,t')$ represents the total number of packets successfully received by destination node n_l till time t' where $\delta \leq \mu$. Consequently, for a given stream the end-to-end packet delivery ratio (PDR) is:

$$PDR(s,\ t') = \frac{\delta(s,\ t')}{\mu(s,\ t')} \times 100\% \tag{1}$$

In our network, the link-layer feedback is active and sender's MAC layer attempts to retry packets that are unsuccessful on first attempt for (r-1) times where r represents the retry limit (including the first attempt). A stream transmission can only be initiated if there exists one or more possible routes represented as set $s^{\tau}(n_k, n_l, f, t_{st}, t'_1)=\{\tau_1, \tau_2, \tau_3, ...\}$ to reach from the stream source n_k to destination n_l. t'_1 represents a given timestamp and therefore $s^{\tau}(n_k, n_l, f, t_{st}, t'_1)$ may or may not be equal to $s^{\tau}(n_k, n_l, f, t_{st}, t'_2)$. Since we employ source routing, each route entails a list *nodeList* comprising an ordered, disjoint and loop free set of nodes that act as relay nodes between the stream source and destination. In Fig. 1, the three route options $\{\tau_1, \tau_2, \tau_3\}$ for transmission from n_k to n_l includes *nodeList* $\{n_1, n_4, n_7\}$, $\{n_5, n_8\}$ and $\{n_3, n_6, n_9, n_{10}\}$ respectively. A route τ therefore can be decomposed into *{source, nodeList, destination}*. In other words, the route $\tau_1=\{n_k, n_1, n_4, n_7, n_l\}$ where the expression $n_1 \in \tau_1$ indicates that node n_1 is included in the route τ_1. Moreover, $\sigma(\tau_1, t')$ is the end-to-end directional delivery ratio estimate for a given route τ_1 at timestamp t', calculated from the link-level delivery ratios $\lambda(n_k, n_1, t'_1)$, $\lambda(n_1, n_4, t'_2)$, $\lambda(n_4, n_7, t'_3)$ and $\lambda(n_7, n_l, t'_4)$ corresponding to all the edges present in the route τ_1 (further discussed in Section 4). Finally, we have a set τ'_s that comprises the route ids of available routes that are available by the stream s for transmission.

In summary, our goal is to accurately estimate the end-to-end directional delivery ratio $\sigma(\tau, t')$ for multi-hop routes. This facilitates the destination of route discovery to select the route with the highest expected PDR among the contending route options.

4. Problems with ETX and broadcast based LQEs

From the related work in Section 2 we infer that ETX has been shown to outperform minimum hop-count, RTT and PktPair [15–17] while some works outlined certain network and environmental conditions where ETX is shown to underperform even the minimum hop count metric [4,18,19]. Nevertheless, ETX is still widely used as a standard LQE for high throughput routing. Consequently, for our end-to-end QoS provisioning framework we originally employed ETX to estimate end-to-end PDR in multi-hop ad hoc mesh networks. Our experiments even in the most simplistic low traffic static network conditions recorded very little to no correlation between ETX link estimates and actual data delivery experienced over those respective routes. Although network topology or traffic plays a significant role in ETX's performance (and can be dealt with accordingly, e.g. heavy traffic handling extension of ETX in [4]), we observe that the main cause of this disparity of ETX-based schemes is far more fundamental in nature. Broadcast packets are fundamentally different compared to unicast packets, and the nodes, the physical channel as well as the network treat them differently accordingly. Most applications communicate in terms of unicast packets that have varying packet sizes, data rates and PHY modulation when compared to broadcast packets etc. Unlike broadcasts, those unicast packets are transmitted after successful channel sense and RTS/CTS handshake. Furthermore, transmission failure of unicast packets results in MAC layer retransmission of the failed packet (if the MAC retry limit is not reached). In this section, we see in detail how two factors, namely the dissimilar nature and heavy traffic, result in wrong estimations.

4.1. Dissimilar treatment by MAC layer

To understand what impact the MAC layer has on the behavior of ETX, we look at the correlation between PDR estimates calculated using ETX broadcast and actual data transmission with respect to varying MAC retransmissions. We used the INETMANET [9] extension running on the OMNeT++ [8] simulator for our experiments. We only use the deliveryForward (d_f) which is the probability of packet delivery from one node to another [4,7] and compare its relationship with the actual packet delivery ratio. Algorithm 1 highlights the steps comprising link delivery ratio estimation and its respective translation to end-to-end delivery ratio estimates.

Algorithm 1. End-to-end route directional delivery ratio estimate using ETX.

1. Periodic ETX broadcast packets at rate ε by n_q are sent to first hop neighbors E_q
2. A given neighbor n_r computes d_f delivery ratio $n_q \rightarrow n_r$ at time t_1: $\lambda(n_q, n_r, t_r) = count(t_r - \hat{e}, t_r)/(\hat{e}/\varepsilon)$; where $n_q, n_r \in N$, $q, r \in \mathbb{R}$ and $q \neq r$, $count(t_r - \hat{e}, t_r)$ is the number of ETX packets received during the window $\hat{e}$, and $\hat{e}/\varepsilon$ is the number of probes that should have been received. t_r represents ETX packet's arrival time at n_r
3. Stream to send $s(n_k, n_l, f, t_{st})$ with start time t_{st} and data rate f results in route discovery by n_k.
4. Each node along discovery route τ_x updates cumulative PDR field σ using λ of respective upstream node: $\sigma(\tau_x, t_p) = \prod_{n_o, n_p \in \tau_x} (\lambda(n_o, n_p, t_p))$, where $o \neq p$ and t_2 denotes RREQ packet's arrival time at n_p
5. RREQ over route τ_x arrives at destination node n_l with end-to-end delivery ratio $\sigma(\tau_x, t_l)$
6. Destination node n_l waits for duration t_Δ to receive subsequent route options for the given stream
7. All route options for stream s received during t_Δ comprise a set of routes $s^\tau(n_k, n_l, f, t_{st})$
8. Stream $s(n_k, n_l, f, t_{st})$ selects a route τ_x if and only if: $Max_{\tau_u \in s^\tau}[\sigma(\tau_u, t_l + t_\Delta)] = \sigma(\tau_x, t_l + t_\Delta)$

A selected route τ_x continues to serve the stream until one or more links becomes too lossy and results in a RouteError (RERR) message, eventually causing a new discovery. However, we want to study the correlation between our delivery ratio estimates and the actual delivery ratio achieved by the network. For this purpose, throughout the simulation run of 200 s, we superimpose a new route discovery every 10 s by erasing all routing tables in the network. The data rate is 500 packets per sec (*CBR-500*), the *etxWindowSize* $\hat{e}$=10 s and the *etxTimeInterval* ε =0.1 s. The network comprises 30 nodes on a 400 m×400 m terrain with varying MAC retransmission limits (*r: 1–6*). The sender and the destination of the streams are placed on the diagonals at coordinates. For experimental correctness we ran 20 trials for each of the six retransmission limits where each trial entails a different placement of all the nodes except the stream's sender and destination.

For each route discovery, the destination node suggests a route with the highest cumulative PDR estimate σ to carry on the transmission. We compute the average $Avg[\sigma(\tau'_s)]$ for all the routes τ'_s employed during the course of each simulation as well as the actual PDR yielded from selecting those routes. For each MAC retransmission limit r, $Avg[\sigma(\tau'_s)]$ and PDR are further averaged for the 20 trials and then compared in Fig. 2. We can observe that our actual data delivery ratio shows a linear increase as we increase the MAC retransmission limit and starts to saturate at retry limit 4–6 (since, at some stage the overhead of retransmitted packets starts to cause more link contention than gain in the data delivery ratio [21]). On the other hand, the ETX-based delivery ratio estimate behaves very differently. While our unicast UDP traffic is getting higher delivery as a result of an increase in MAC retransmissions, the ETX broadcast packets are suffering more losses resulting in lower delivery ratio estimates. Statistical analysis shows that the Pearson correlation coefficient of Fig. 2 is −0.24 between the delivery ratio estimate and the actual data delivery ratio. This implies weak/low correlation indicating ETX isn't a good estimator of data delivery ratio. This somewhat random behavior in link estimation is also visible in other setups that we ran with different traffic rates or node densities but is more evident in high traffic scenarios.

We believe this insufficient correlation between ETX and actual packet delivery ratio ensues due to a number of reasons. (1) First and foremost is the intrinsic difference in how the physical layer and MAC

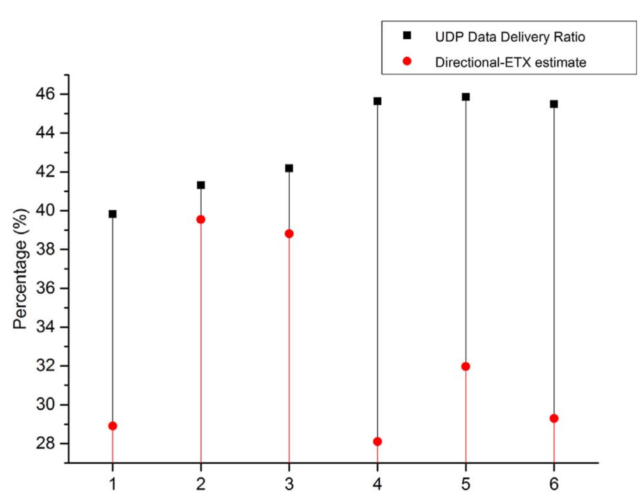


Fig. 2. Directional ETX estimated vs actual Data Delivery Ratio yielded.

layer treat broadcast packets as compared to unicast. The broadcast probe packets used for ETX are small and are sent at the lowest possible data rate (1Mbps for 802.11b/g, 6Mbps in case of 802.11a) [6]; they may not experience the same loss rate as data packets sent at higher rates. (2) When the buffer gets full MAC layer gives lower priority to the in-buffer unicast packets than broadcast and starts discarding unicast packets first. (3) The metric does not directly account for link load or data rate. A heavily loaded link may have very low loss rate, and two links with different data rates may have the same loss rate. (4) Furthermore, wireless channels rely heavily on the underlying link level feedback and RTS/CTS mechanism provided by the retransmissions [20]; a feature which is only applicable to unicast packets.

4.2. ETX under heavy traffic load

In wireless environments heavy communication traffic loads result in more packet drops, packet buffer overflows as well as link breakages. In MANETs such service denials cause routing errors, thereby initiating route discovery via broadcast flooding. The authors of [4,9] studied the behavior of ETX under such route request packets as well as heavy traffic in general. They observed that higher traffic loads entail more frequent link route discoveries and eventually a greater probability of incorrect ETX estimation. To study this behavior, we ran high traffic simulation with 5 senders sending heavy UDP traffic at a data rate of 1000 packets per seconds (*CBR*-1000) towards 5 distinct destinations. The *etxWindowSize* $\hat{e}$=10 s and the *etxTimeInterval* ε =0.1 s in a network comprising 70 nodes on a 500 m×500 m terrain with the MAC retransmission limit r set to 6. Fig. 3 illustrates the ETX estimate variation of 9 randomly picked links over a duration of 200 s. The green colored vertical drop lines indicate the times that route discoveries are initiated. The figure shows that Link-5 retains an ETX value of approximately 1 until 120 s indicating that nearly all ETX packets transmitted over this link have been successful. Furthermore, we see that some links exhibit much higher fluctuations in ETX value distribution as compared to others. Link-1 and Link-2 experience an ETX increase as high as 5200% and 3000% respectively.

At such a constant high data rate, the routes discovered are not stable and recurrently broken; resulting in numerous new consecutive route discoveries and consequently increasing the traffic load. This continuous flooding further increases the probability of ETX broadcast packet failures, as compared to unicast data packets, leading to wrong estimates. This behavior can cause serious degradation in the network PDR. When link quality measurements are affected by flooding of request discoveries, nodes may choose low quality paths that may easily be broken. This continuous selection of bad paths may continue repeatedly and thereby adversely affect the overall network's performance.

5. xDDR – expected Directional Delivery Ratio

In this section, we detail our link quality estimation metric xDDR (expected Directional Delivery Ratio). In the previous section we highlighted two main characteristics affecting ETX link quality estimates: (1) MAC layer's disparity in handling broadcast versus unicast, and (2) the effect of route broadcast flooding on link quality estimates. These observations inspired the design of the xDDR metric. xDDR simply is a selective unicast beaconing mechanism to estimate link's PDR over a fixed-length sliding-window while excluding the link's performance during the route discovery flooding phase. xDDR achieves end-to-end route delivery ratio estimation by employing three main processes, namely: (1) pulse quality based neighbor selection and classification, (2) selective proactive unicast beaconing, and (3) a reactive quality-driven routing module as explained previously in Section 4. Processes (1) and (2) are specific to link-level delivery ratio estimation while (3) translates link-level quality estimates to end-to-end route estimates, as well as providing sub-modules for route selection and reply.

5.1. Pulse quality based neighbor classification and update

xDDR utilizes proactive unicasts for the estimation mechanism. Unicast packets require a specific address as a destination which is furnished as IP or MAC address depending on the layer originating the packet. It is therefore imperative to identify each node's first hop neighbors, amongst whom few would be designated as unicast recipients. These unicast recipient nodes will be addressed in separate individual unicast packets. In a similar scenario, ETX uses one potentially receivable broadcast not only to all unicast recipients but all its first hop neighbors. Furthermore, in a clustered situation a node can have disproportionately large number of first hop neighbors, most of which could be entirely substitutable. We therefore limit the number of designated unicast recipients ($|E'_m|$ =4 in our experiments).

In addition to that, mobile ad hoc networks nodes are subject to

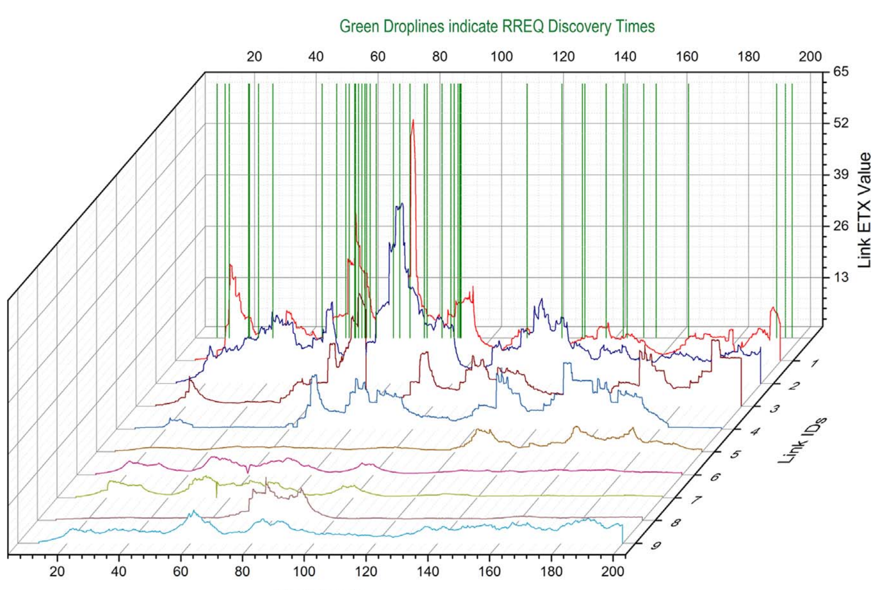


Fig. 3. Link ETX estimate variation and route discoveries.

mobility and the topology is hence dynamic. This means the unicast recipients, the first hop neighbors and the node itself can move anytime and autonomously. Our approach needs to be swift in reacting to such changes. We therefore extend our legacy First Come First Serve (FCFS) where nodes designate unicast recipients chronologically from the received broadcast beacons to incorporate more information related to the reception from neighbors.

Each node in xDDR maintains a neighbor selection table *neigh_tbl* (represented as E_m). In order to identify the 1-hop neighbors, during the initialization phase each node proactively broadcasts ψ number of beacon messages at the same send interval rate as proactive unicast beacon messages. These broadcast messages act as a pulse for their first hop neighbors. This phase yields each node n_m the list of its first hop neighbors, represented as set of edges $E_m = \{e_{m1}, e_{m2}, e_{m3}, \ldots\}$ where $m1, m2, m3 \in \mathbb{R}$ indicating pulse messages received from nodes n_{m1}, n_{m2} and n_{m3} etc. Node n_m then measures the pulse since it knows the maximum number of expected pulses ψ as well as the duration of the measurement i.e. no more than $t'_{arr} + (\psi \times \varepsilon)$ seconds after the reception of the first pulse (t'_{arr} indicates the arrival time of the first successful pulse). Given t' as the current time stamp, n_m measures the pulse reception quality ∂ from first hop neighbor n_{m1} as:

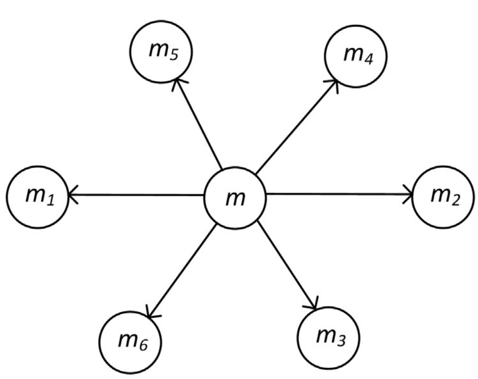
$$\partial(n_{m1}, n_m, t') = \frac{count(t' - (\psi \times \varepsilon), t')}{\psi} \tag{2}$$

Now our neighbor selection table E_m possesses the list of neighbors represented with their respective edges as well as their corresponding quality ∂. Pulse reception quality ∂ therefore becomes the first classification criteria to assess the quality of first hop neighbors. Node n_m then selects $|E'_m|$ number of nodes where $E'_m \subseteq E_m$, to be designated as its recipients of proactive unicast beacon messages (explained in the next sub-section). The 1st hop neighbor with the highest pulse quality is picked first, and so on. Since the topology is dynamic and nodes are subject to change, the pulse reception quality based neighbor update and classification process therefore is recursive. The frequency of the recursion depends on the mobility dynamics of the environment, e.g. mobility in a music festival, art exhibition etc. In our experiments we emulated movement in festivals with individual nodes with random speeds (ranging between 0.5mps to 0.8mps) and directions (every 100 s, change of angle between 0 and 90 degrees) and kept the refresh interval for pulse quality based neighbor update and classification at $t'_{\Delta} = 50$ s. We didn't see gains in further lowering the update interval indicating it to be optimal for our mobility context. Fig. 4 illustrates the neighbor table view of node n_m where it selects nodes n_{m2}, n_{m3}, n_{m1}, *and* n_{m6}, as unicast recipients based on their respective pulse reception quality.

We compare the pulse based classification with the legacy FCFS method that generates the same amount of control packets. Our results are later shown in Fig. 13, Section 7.3 signifying the value in neighbor classification with pulse reception quality over our legacy FCFS scheme. In terms of overhead, each node transmits $Ŧ \times \psi / t'_{\Delta}$ number of broadcast packets during the network lifetime $Ŧ$. An increase in node mobility requires smaller refresh intervals t'_{Δ} and thereby more frequent broadcast traffic. This scheme is therefore most suitable for networks with slow to medium mobility. In high mobility scenarios hardware based LQEs are shown to most responsive but their accuracy is still an open subject [33].

5.2. Selective proactive unicast beaconing

In the previous section, we established that unlike regular data packets, broadcast packets do not benefit from the MAC layer RTS/CTS mechanism and are transparent to the MAC retransmission limit. This results in a weak correlation between link quality estimates and delivery ratio of the actual traffic that is almost entirely based on unicast transmissions. As shown earlier, in xDDR each node n_m maintains a list of its first hop neighbors represented as set $E_m = \{e_{m1}, e_{m2}, e_{m3}, \ldots\}$ where $m1, m2, m3 \in \mathbb{R}$. Now in order to estimate link-level PDR, each node n_m separately transmits proactive unicast beacon messages addressed specifically to its 1st hop neighbors. These packets have an empty payload and time to live field $TTL = 1$. In comparison, proactive beaconing messages in ETX additionally carry the respective node's 1st hop neighborhood table listing all active and inactive nodes with their respective deliveryForward ($đ_f$) values. This is used by the receiving node as its deliveryReverse ($đ_r$) value. Since xDDR only uses directional PDR, sharing one's neighborhood table is not required. On the downside, reaching first hop neighbors via unicast requires more packet transmissions as opposed to broadcast due to a couple of reasons. *(a)* The unicast beacon requires a particular IP address as a destination and *(b)* due to collision each network layer beacon packet may entail multiple physical layer transmissions (depending on the retransmission limit $ɍ$ and successful delivery). In order to contain this overhead, each node in xDDR limits the number of unicast beacon recipients. Node n_m therefore uses pulse quality-based neighbor classification and selects a subset from E_m as its unicast beacon recipients (represented as E'_m) to transmit packets at rate etxTimeInterval ε. To quantify the overhead in terms of the number of packets, in ETX each node transmits $Ŧ/\varepsilon$ number of packets where $Ŧ$ represents the network lifetime. In case of xDDR, the number of packets transmitted physically varies from node to node as a result of additional factors such as E'_m and MAC retransmission limit $ɍ$. The best case therefore is $(Ŧ \times |E'_m|)/\varepsilon$ and $(Ŧ \times |E'_m| \times ɍ)/\varepsilon$ is the worst case. In our experiments, we limit each node to have at most 4 unicast beacon recipients. There are two main behavioral benefits of using unicasts instead of broadcasts towards making the estimation module more analogous to the actual data transmission. *Firstly*, the unicast beacon messages adhere to the RTS/CTS mechanism just like regular data



neigh_tbl E_m for node m

Node	∂	Unicast?	t'_{arr}
m2	0.9	true	51sec
m4	0.6	*false*	50sec
m3	0.9	true	50sec
m1	0.7	true	51sec
m6	0.7	true	50sec
m5	0.5	*false*	51sec

Unicast recipient removal condition

```
if  t' - t'arr > t'Δ than
    delete node;
```

Fig. 4. Pulse quality based classification and update.

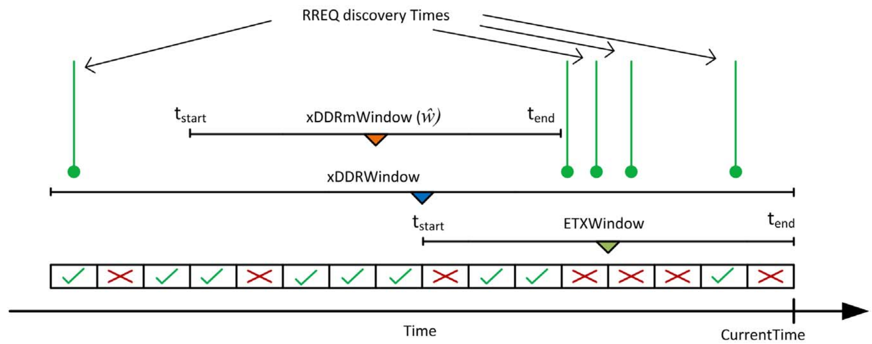


Fig. 5. xDDR measurement window w.r.t ETXwindow.

packets. Secondly, it estimates data delivery ratios while accounting for the underneath MAC retransmission limit.

Each of the recipients from E'_m maintains a window xDDRWindow logging timestamps of the packets received from n_m. This window is used to derive the xDDR measurement window represented as ŵ which in essence is the window used to compute link-level directed delivery ratio λ. The xDDRWindow, in practice, is twice the size of the ETXWindow. We have to be careful with the size of ETX and xDDRWindow size since it directly affects the freshness of the link quality estimate.

Fig. 5 illustrates how instead of measuring the ratio from the entire last slot as in ETX; xDDR derives the xDDR measurement window (xDDRmWindow) $\hat{w}$ from the xDDRWindow thereby effectively excluding behavior during RREQ flooding phases. As shown, the xDDRWindow is twice the size of the ETXWindow. $\hat{w}$ covers the freshest possible window set from the xDDRWindow that excludes the discovery times. In other words, xDDR trade-offs some freshness where there is a possibility of noisy values. In case a full ETXWindow size cannot be extracted outside the noise zones, xDDR picks the same entry slots as ETX would have picked. Given the xDDR measurement window $\widehat{w}_{m\to o}$ (where $m\to o$ indicates the direction of unicast beacons i.e. from n_m to n_o) and the unicast sending interval ε, a node n_o calculates the directed delivery ratio $\lambda(n_m, n_o, t')$ as:

$$\lambda(n_m, n_o, t') = \frac{count(t' - \widehat{w}_{m\to o}, t')}{\widehat{w}_{m\to o}/\varepsilon} \tag{3}$$

$count(t' - \hat{w}, t')$ counts the number of successful proactive unicast beacons received (with help of retransmissions, where necessary) between time $(t' - \hat{w})$ up until current time t' i.e. the duration of the xDDR measurement window $\hat{w}$. $\widehat{w}/\varepsilon$ results in the total number of unicast packets sent within this duration. Algorithm 2 presents the pseudo-code for dynamically selecting the xDDRmWindow by specifying the window's start t_{start} and end t_{end} at a given time *currentTime*. And how this information is used to calculate the number of packet received eventually resulting in the link delivery ratio estimate λ for the given link.

Algorithm 2. xDDR Window Selection.

1. T→ list of received packet timestamps in reverse chronological order
2. rreqTimes→ list of timestamps of rreq packets observed
3. xDDR Window Size: = 2 × ê
4. **if**! ETX Window Conflict **then**
5. t_{end}:= currentTime
6. **else if** Conflict Free Window Available
7. t_{end}:= time Of Last Discovery
8. **else**
9. t_{end}:= current Time
10. **end if**
11. t_{start}:= t_{end} − (ê × ε)
12. **for each** t ∈ T
13. **if** $t_{start} \le t \le t_{end}$
14. count++
15. **end if**
16. **end for**
17. λ: = count / (ê / ε)

This way each node in the network measures the delivery ratio of its respective upstream neighbors. Algorithm 3 pencils in the steps on how xDDR works together with our quality driven routing framework in order to provide end-to-end route estimates based on unicast beaconing.

Algorithm 3. End-to-end route directional delivery ratio estimate using xDDR.

1. Each node n_q selects E'_q from E_q where $E_q \in E$, using pulse quality based classification
2. Periodic unicasts to subset E'_q to be received by 1st hop neighbor(s) n_r, r∈$\mathbb{R}$
3. A given neighbor n_r computes λ delivery ratio $n_q \to n_r$ at time t_r over $\hat{w}$ preventing (where possible) link estimation during RREQ discovery phase:

$\lambda(n_q, n_r, t_r) = \frac{count(t_r - \widehat{w}_{q\to r}, t_r)}{\widehat{w}_{q\to r}/\varepsilon}$, $n_q, n_r \in N \& q \ne r$ $\widehat{w}_{q\to r}$ denotes the xDDRwindow over unicasts from $n_q \to n_r$ & t_r is the unicast beacon's arrival time at n_r

4. Stream to send $s(n_k, n_l, f, t_{st})$ with start time t_{st} & data rate f results in route discovery by n_k for n_l.
5. Each node along the discovery route τ_x updates the cumulative PDR field σ using λ of the respective upstream node:

$\sigma(\tau_x, t_p) = \prod_{n_o, n_p \in \tau_x} (\lambda(n_o, n_p, t_p))$; *where*, $o \ne p$ and (t_p denotes RREQ packet's arrival time at node n_p)

6. RREQ over route τ_x arrives at destination node n_l with end-to-end delivery ratio $\sigma(\tau_x, t_l)$
7. Destination node waits for duration t_Δ to receive subsequent route options for the given stream
8. All route options for stream $s(n_k, n_l, f, t_{st})$ received during t_Δ add to set of routes $s^\tau(n_k, n_l, f, t_{st})$
9. Stream $s(n_k, n_l, f, t_{st})$ selects a route τ_x if and only if: $Max_{\tau_u \in s^\tau}[\sigma(\tau_u, t_l + t_\Delta)] = \sigma(\tau_x, t_l + t_\Delta)$
10. n_l unicasts RREP back to n_k encapsulating the route with highest estimated σ
11. n_k employs the suggested route for stream transmission

6. Quality-Driven routing framework

In order to assess the accuracy as well as effectiveness of a routing metric for multi-hop MANETs, we devised a multi-hop quality-driven routing framework that supports route selection based on link delivery ratio estimation/calculation. Our framework can choose between the implemented PDR estimation matrices, and as a result provides end-to-end route delivery ratio estimates. In this regard, we have two performance indicators to assess the quality estimation schemes. *(a)* How accurately the metric indicates the packet delivery ratio of the end-to-end route? *(b)* Secondly, which metric eventually yields the highest end-to-end packet delivery ratio for a given stream? It's important to note that having most accurate estimation doesn't necessarily guarantee the highest yield. For instance, a highly accurate estimation metric may result in excessive control packets or buffer overhead etc., consequently resulting in lower end-to-end data packet delivery than a less accurate scheme. Our goal therefore is to find which metric is sufficiently accurate in estimates as well as results in the highest end-to-end packet delivery.

We extended DSR [14] and AODV [23], two widely used MANET routing protocols. DSR is a source routing based protocol where the Destination node of a route discovery packet (RREQ) in DSR receives the entire route containing IDs of every node traversed on the route. AODV on the other hand only provides the information of the previous hop node. This means the DSR nodes maintain a neighborhood table containing routes to every node in the network whereas AODV nodes only log the ID of the particular next hop node needed to forward the packet to the intended destination. Both schemes have their own advantages and disadvantages. For example, AODV uses destination sequence numbers to ensure loop free routes whereas DSR nodes by virtue of source based routing are able to eliminate routing loops during route discovery. As mentioned earlier, xDDR being a link-quality estimation metric is independent of the multi-hop routing protocol that uses it. In other words, xDDR estimates the link-quality independently and it's the job of the quality driven routing framework to deliver the product of link-quality-estimates of the nodes traversed in whichever way is fitting to the protocol. We now closely examine how this is achieved in the case of DSR and AODV and in the next section we compare their respective performance.

6.1. Dynamic Source Routing Protocol Extension (DSRxDDR)

We extend DSR to work associated with whichever packet delivery ratio estimation metric is employed. Fig. 6a demonstrates how the xDDR estimation metric is used with the multi-hop end-to-end quality driven routing module. When a node n_k in the network needs to send a stream to a destination node n_l, it initiates a route discovery broadcasting a new RREQ packet with a unique *c_id*. $n_k \rightarrow *$ represents a broadcast to all first hop neighbors. The broadcast RREQ packet from n_k reaches the first hop neighbor(s) who will first use the *c_id* to verify whether it is a new request for a new stream. Once this freshness is established, the specific node let's say n_5 adds/appends itself to the RREQ. In other words, the empty *nodeList* and Cumulative PDR (*CumPDR*) field is initiated with node id $\{n_5\}$ and the estimated link delivery ratio $\lambda(n_k, n_5, t'_1)$ respectively of its upstream link (the link between the current node and its previous hop i.e. n_k) e_2. n_5 then rebroadcasts the updated RREQ packet to be received by its respective first hop neighbors. Upon RREQ packets arrival node n_8 checks whether it is the intended destination. Otherwise, it updates the *nodeList* tuple with its ID $\{n_5, n_8\}$ and updates the *CumPDR* as $\lambda(n_k, n_5, t'_1) \times \lambda(n_5, n_8, t'_2)$ and rebroadcasts the updated RREQ. This broadcast eventually reaches the destination node. The destination node multiplies the received *CumPDR* with $\lambda(n_8, n_l, t'_3)$ yielding the end-to-end route estimate for route option τ_2 i.e. $\lambda(n_k, n_5, t'_1) \times \lambda(n_5, n_8, t'_2) \times \lambda(n_8, n_l, t'_3)$. It then waits for time t_Δ expecting to receive alternate options τ_1, τ_3 etc. If more RREQ reach the destination node n_l within t_Δ, it compares to see which route implies the highest estimate for end-to-end packet delivery for instance $\sigma(\tau_2, t'_3)$. Based on this, the destination node prepares a unicast reply (RREP) back to the source node n_k which includes the *nodeList* with the highest estimated end-to-end PDR. The RREP traverses back to the source node via the nodes indicated in the *nodeList* of the selected route, eventually reaching back to the source node n_k. Once the source node receives the RREP it initiates the data transmission along the suggested route.

6.2. Ad hoc On-Demand Distance Vector Routing Protocol Extension (AODVxDDR)

AODV [12] is essentially a combination of both DSR and DSDV [32]. It borrows the basic on-demand mechanism of Route Discovery (RREQ) from DSR, whereas sequence numbers and periodic beacons are inspired by DSDV. It uses destination sequence numbers to provide loop free routes. Unlike DSR RREQ which forwards the entire nodelist of a route, AODV nodes maintain route tables for a given destination. A typical route entry tuple includes <*DestinationID, NextHop, Hopcount, DestinationSequenceNo.*>. Each intermediate node uses the routing table information to forward the packet along the selected route. Fig. 6b shows the working of AODVxDDR. When a node n_k in the network has a stream $s(n_k, n_l, f, t_{st})$ to send to destination node n_l, it initiates a new RREQ packet with a broadcast id *c_id*. Since the destination sequence number is unknown at this time, the sequence number field is empty. Similar to DSR, the *CumPDR* is also empty. The broadcast RREQ packet from n_k reaches the first hop neighbor(s) who will first use the *c_id* to verify whether it is a new request for a new stream. Once this freshness is established, the specific node let's say n_5 creates a route entry for RREQ source n_k. n_5 then increments the hopcount by 1 and updates the *CumPDR* field with estimated link delivery ratio $\lambda(n_k, n_5, t'_1)$ of its upstream link. n_5 than rebroadcasts the updated RREQ packet to be received by its respective first hop

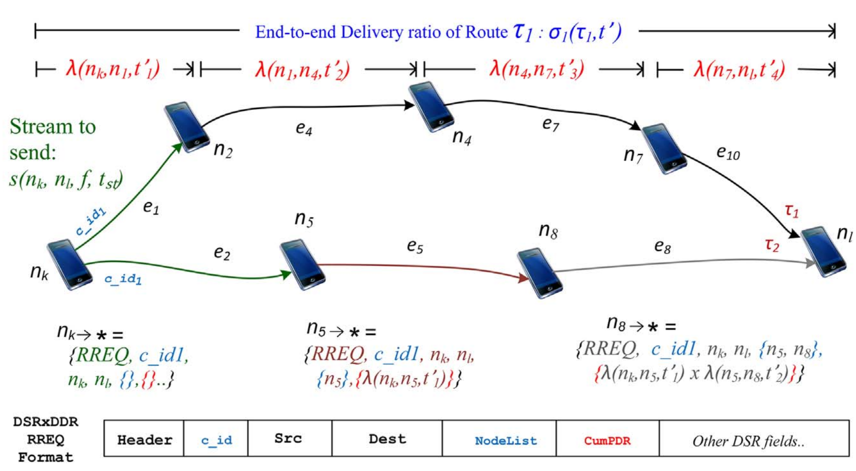


Fig. 6.(a). DSRxDDR quality driven route discovery.

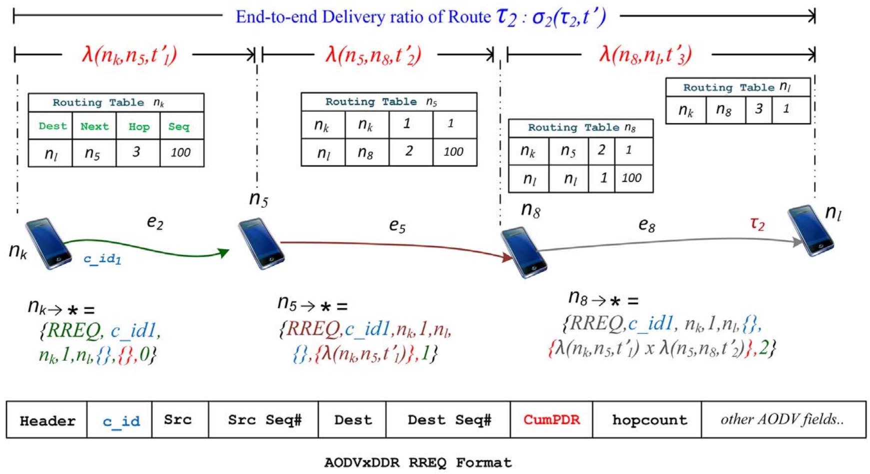


Fig. 6.(b). AODVxDDR quality driven route discovery.

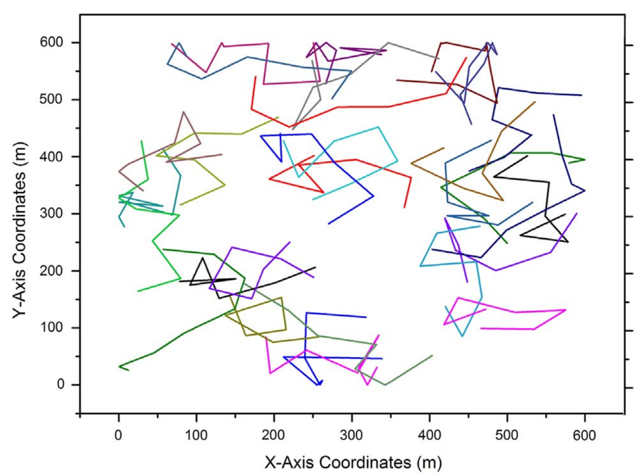


Fig. 7. Mass mobility movement trajectory of 30 of nodes in a 600 m×600 m terrain during a run of 600 s selecting new angle and speed every 100 s.

neighbors. Upon RREQ packets arrival, node n_8 checks whether it is the intended destination. Otherwise, it also adds a new route entry, updates the hopcount and the *CumPDR* as $\lambda(n_k, n_5, t'_1) \times \lambda(n_5, n_8, t'_2)$ and rebroadcasts the updated RREQ. This broadcast eventually reaches the destination node n_l which waits for time t_Δ expecting to receive alternate options τ_1, τ_3 etc. If more RREQ reach the destination node n_l within t_Δ, it compares to see which route implies the highest estimate for end-to-end packet delivery for instance $\sigma(\tau_2, t'_3)$. n_l then generates a unicast reply back to the neighboring node that possessed the highest end-to-end PDR estimate. Unicast RREP $n_l \rightarrow n_8$ will be as follows:

$n_l \rightarrow n_8$ (RREP) = {RREP, n_l, *100*, n_k, $\{\lambda(n_k, n_5, t'_1)\ x\ \lambda(n_5, n_8, t'_2))\ x\ \lambda(n_8, n_l, t'_3), 0\}$.

Hopcount 0 indicates RREP packets distance from sender which in this case is n_l and not n_k. We notice that the end-to-end CumPDR for AODV RREQ is the same as the DSR RREQ in previous subsection. Each intermediate node during the RREP adds new route entry for the destination n_l and from the previous n_k entry looks up the next hop neighbor to whom the unicast packet is sent on its way back to n_k. Once n_k receives the RREP packet, it first creates a route table entry for n_l and then initiates the data stream.

7. Experimental setup and analysis

In this section we present (1) implementation details, environment and experimental setup (2) review comparison of correctness of estimation, (3) effect of estimation in improving end-to-end data traffic and (4) finally comparison of overhead.

7.1. Implementation details and setup

For our implementation and experimentation for a quality driven routing framework and packet delivery estimation metrics we used OMNeT++(Objective Modular Network Test-bed) [8]. OMNET++ is an open-architecture, extensible, modular, component-based C++ discrete event simulation environment with strong GUI support and an embeddable simulation kernel. Due to its extensible nature, it is widely used for developing and testing large scale communication networks. OMNET++ offers a few extensions tailor-made for specific networking paradigms. Most notable among them are the INET [22] and INETMANET [9] framework. For our experimentation we used the INETMANET extension which is specifically dedicated to mobile ad hoc networks and offers a variety of mobility models specifically related to MANET mobility. We furthered the DSR implementation available in the INETMANET extension to accommodate a quality driven routing framework. Our framework can select between ETX, HETX and xDDR as the link level delivery ratio estimation metric, to then provide end-to-end route level delivery ratio estimates.

In mobility scenarios, we aimed to emulate movement of people with smartphones in a festival terrain or outdoor exhibitions. For this purpose, we reviewed a number of mobility models presented in [30], including Random Waypoint mobility, Gauss-Markov mobility, Mass Mobility (MM) and Chiang mobility models. We found MM to be the best fit for the scenario. In MM, mobile nodes move within an area specified in the terrain. Nodes move along a straight line for a specific duration and then make a turn. This duration of movement in a straight line is controlled by the parameter *changeInterval* which takes duration and standard deviation (*changeInterval* =100 s, *SD* =1 s in our setup). The turn angle to dictate the new direction after every change interval is a normally distributed random number with average equal to preceding direction and standard deviation (*SD* =90 degrees in our setup). Similarly, after each *changeInterval*, the node speed is taken as a uniform distribution within a range of speeds. Our setup used 0.5mps (1.8kph) to 0.8mps (2.9kph). Fig. 7 shows the movement pattern of 30 nodes in a 600 m×600 m terrain during 600 s. Since the *changeInterval* is set to 100 s, we can see the 6 instances per node of change in angle of movement.

Fig. 8 shows related network parameters generic to our setups.

7.2. Estimation integrity comparison

In order to compare the estimation integrity of the schemes, we can audit how accurately a given scheme estimates the route's end-to-end

Parameter	Value	Parameter	Value
SimulationTime	300sec-1000sec	RREQ wait time t_Δ	10ms
Seeds/experiment	20	MAC	802.11g
Mobility Change Interval	100sec	MAC bitrate	54Mbps
Mobility Change Angle	normal (0deg, 90deg)	Mobility Speed	uniform (0.5mps,0.8mps)
UDPSendInterval	0.01sec; and 0.005sec;	ETX send interval	0.1s
UDPStartTime	11sec	xDDR send interval	0.1sec; and 0.5sec
Trans AntennaGain	-1.4dB	Neighbor refresh interval t'_Δ	50sec
WLAN Radio Sensitivity	-90dBm	Radio Transmit Power	1.0mW

Fig. 8. Generic network simulation parameters OMNET++.

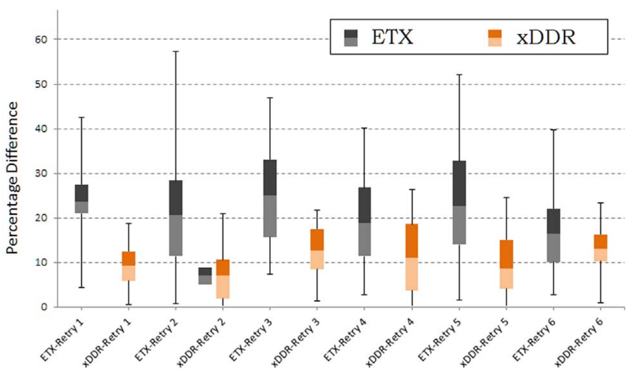


Fig. 9. Difference in PDR estimate and UDP delivery ratio.

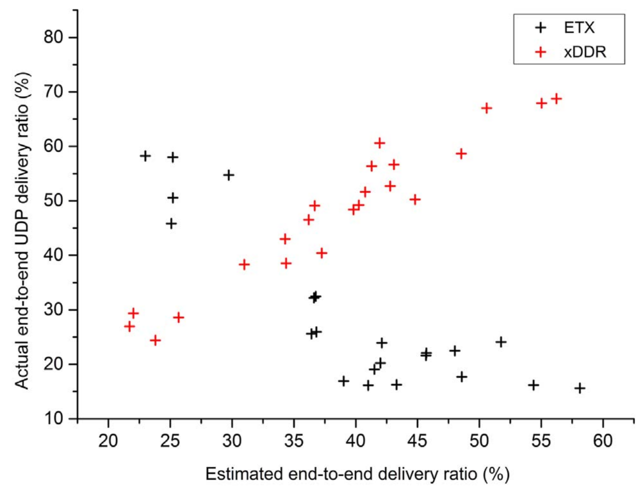


Fig. 10. Best case estimates for xDDR (RetryLimit: 1).

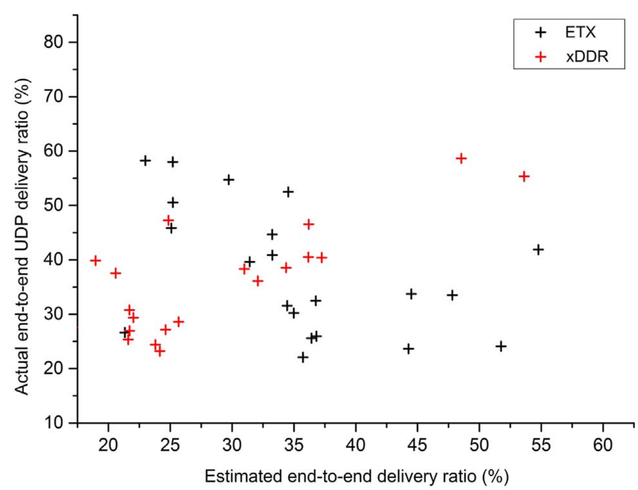


Fig. 11. Best case estimates for ETX (RetryLimit: 4).

delivery potential by comparing it with the actual data delivery ratio experienced. We therefore log the highest end-to-end PDR's selected for the routes after each discovery and compare it with the end-to-end PDR experienced between the communicating nodes. It's important to note that the change in a node's spatial position also influences its delivery ratio. It is reasonable to not include node movement as another factor of deflection from an otherwise accurate estimation or vice versa. Therefore, only within this context of evaluating the estimation integrity, we keep the network as static.

In Fig. 9 we compare ETX and xDDR for PDR estimates with respect to actual packet delivery ratio. For experimental correctness we ran 20 trials for each of the MAC retransmission limits where each trial generates a different placement for all the 30 nodes except the stream's sender and destination. The sender and destination keep the same coordinates between each trial. For each route discovery, the destination node upon receiving multiple RREQs suggests a route with the highest cumulative PDR estimate σ to carry on the transmission. We compute average $Avg[\sigma(\tau'_s)]$ for all the routes τ'_s employed during the course of each simulation. At the end of each simulation run, we compute the actual PDR yielded from selecting those routes. For each MAC retransmission limit, $Avg[\sigma(\tau'_s)]$ and PDR are further averaged for the 20 trials and then compared for correlation. The y-axis represents the percentage difference between the route's end-to-end delivery ratio estimates vs. the actual end-to-end delivery ratio yielded as a result of selecting those routes. A lower difference indicates better correlation and vice versa. The upper whiskers indicate the maximum difference for each set of experiment indicating the maximum difference in estimates are a lot higher in the case of ETX.

We calculated the Pearson correlation coefficient of both metrics across all trials and all retransmission limits. The Pearson correlation coefficient between delivery ratio estimate and the actual data delivery ratio in the case of ETX is −0.14 indicating a weak negative correlation. xDDR on the other hand demonstrated a correlation of 0.63 implying moderately strong correlation between estimated and actual delivery. We further explore the results of the best case estimates of xDDR i.e. retry limit 1 and the best case of estimates for ETX i.e. retry limit 4 in Figs. 10 and 11 respectively.

7.3. Effect of estimation on end-to-end data traffic

To explore the role of accurate estimates in impacting the end-to-end delivery ratio, we ran experiments with high traffic and mobility (Mass Mobility – exhibition scenario). In low traffic scenarios the impact is less visible due to the fact that most links are expendable with near 100% link delivery ratio for data packets, meaning the end-to-end delivery ratio would still be upwards of 90% even with random route selection. High traffic and mobility therefore introduces more varied link and consequently route options. We compare the outcome of

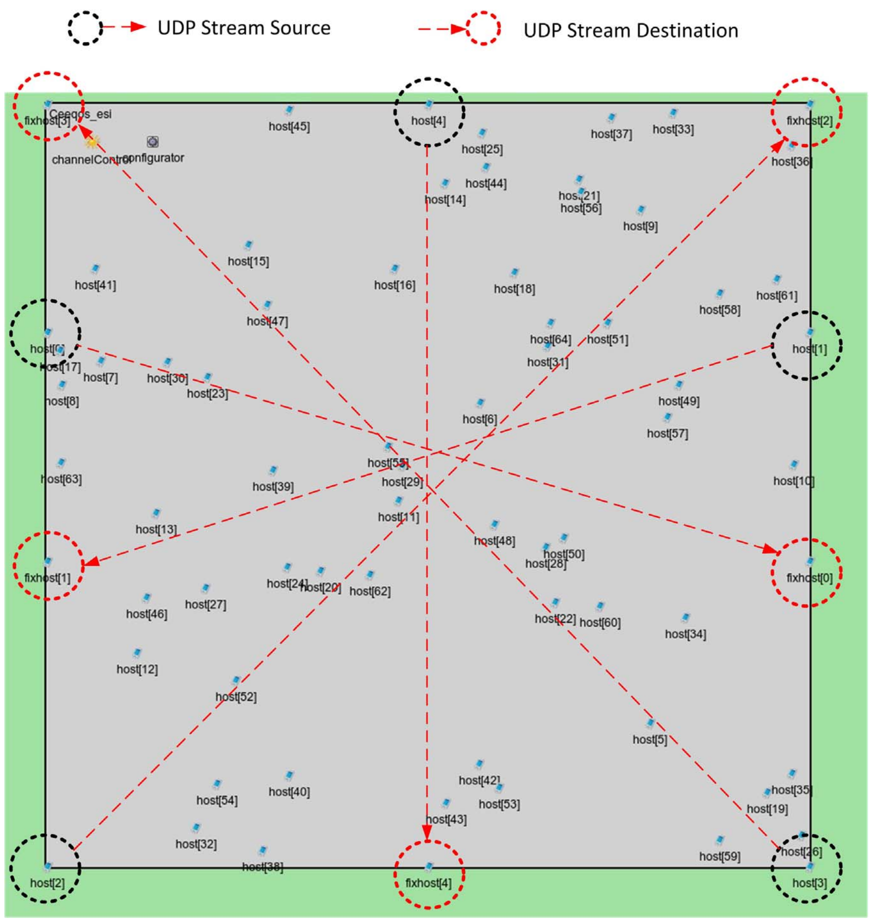


Fig. 12. Network topology enclosed in 600×600 m terrain, containing 70 nodes with overlay annotations representing the senders, destinations and their respective direction of stream flow.

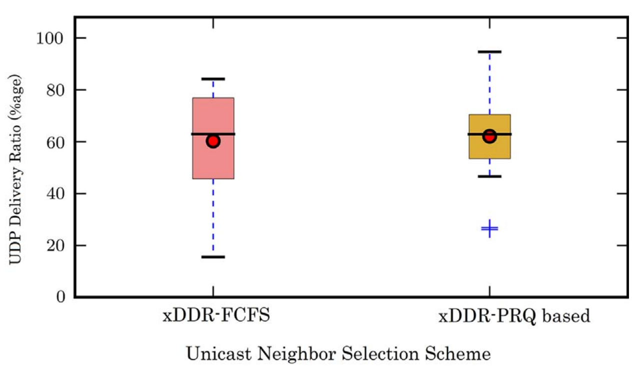


Fig. 13. Unicast neighbor selection scheme comparison.

employing different quality estimation metrics on the actual delivery ratio of end-to-end UDP streams. We placed 5 stream senders and 5 destination nodes placed at the cross-diagonal boundaries to each other on a 600 m×600 m terrain. HETX and ETX have their beaconing interval set to 0.01 s whereas xDDR has a beaconing interval of 0.1 s. The UDP stream's transmission rate is 100 CBR sending 100 packets per seconds. The underlying MAC retransmission limit is 6. The sender and destination nodes stay static at the boundaries whereas all other nodes move in Mass Mobility. Fig. 12 shows the network topology for one of the trials from the experiment set involving 70 nodes.

We first document the benefits of selecting unicast neighbors by employing Pulse Reception Quality (PRQ) presented in Section 5.1 over the simple first come first serve (FCFS) scheme. In the FCFS scheme a

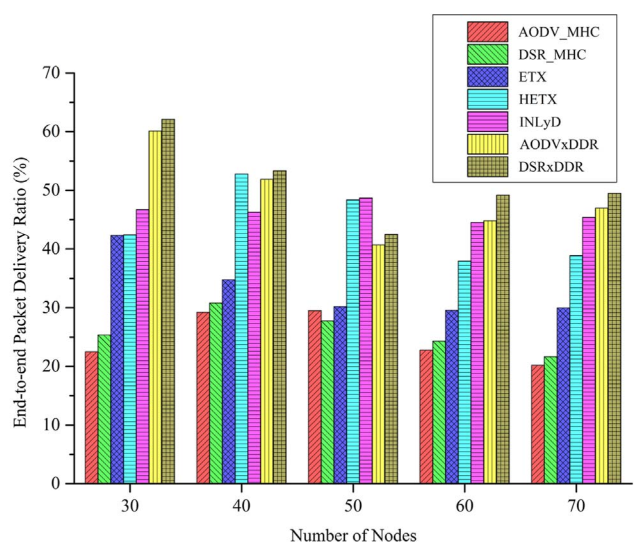


Fig. 14. End-to-end UDP Data Delivery Ratio in mobile scenario with mass mobility movement.

node designates unicast recipients chronologically from the received broadcast beacons whereas the PRQ scheme maintains the broadcast reception ratio and designates unicast recipients in reverse-numerical order of the recorded ratios (highest pulse reception ratio first). We ran 20 trials containing 30 mobile nodes (5 senders, 5 receivers and 20 intermediate nodes) with each trial resulting in a different network topology of the intermediate nodes. Fig. 13 shows the comparison of both schemes.

Fig. 14 compares the end to end UDP data delivery ratio yielded as a result of route selection based on estimates provided by a distinct link level metric. We compare the end-to-end packet delivery ratio achieved by the communicating source and destination nodes aiming to find the best routes with the help of a given link-level metric namely DSR with the Minimum Hop Count metric (DSR_MHC), AODV with Minimum Hop Count metric (AODV_MHC), ETX, HETX, DSR with xDDR (DSRxDDR) and AODV with xDDR (AODVxDDR). DSRxDDR and AODVxDDR both have *gratuitous_reply* option disable i.e. only the destination node sends the RREP message back to sender. We can first observe that both AODV and DSR protocols are able to double the end-to-end delivery ratio by replacing the minimum hop count metric with xDDR link estimation. Compared to other schemes this implies a 2.1 times gain over AODV_MHC, 2 times gain over DSR_MHC, a 1.6 times gain over ETX and a 1.3 times gain over HETX in terms of actual packet delivery yielded.

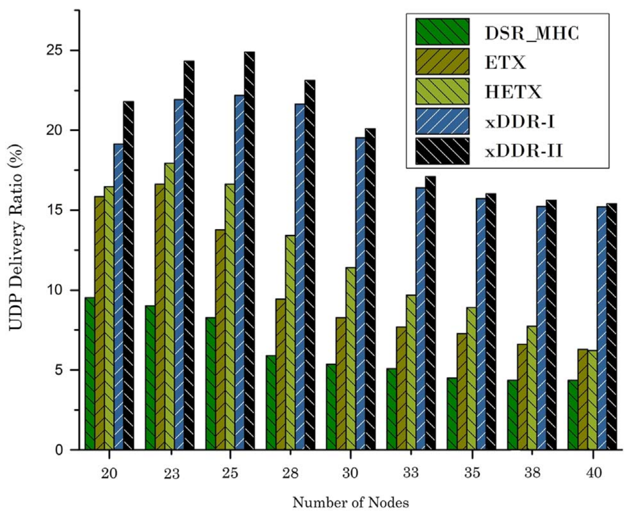


Fig. 15. End-to-end UDP Data Delivery Ratio comparison in a static environment with no mobility.

With the addition of our pulse based neighbor refresh mechanism we have been able to achieve the same level of freshness (w.r.t. first hop neighbors) as any broadcast based scheme, while at the same time benefitting from the accuracy of unicast beaconing. Our results in the mobility context are consistent with our previously published result in the static environment in [1]. For reference, Fig. 15 shows the end-to-end UDP delivery ratio comparison in a static environment. The UDP stream's transmission rate is 100 CBR sending 100 packets per seconds whereas the MAC retransmission limit is 6. xDDR-I represents the xDDR link estimation mode that employs only the selective unicast beaconing (and same window size as ETXWindow), while xDDR-II uses selective unicast beaconing as well as xDDRWindow.

7.4. Overhead comparison w.r.t. end-to-end quality driven route selections

Picking a lossy route wrongly as best performing, in terms of PDR, results in data being transmitted through links that may require more retransmissions per successful packet delivery than a healthier route. On the other hand, xDDR's quality estimation does require additional control packets due to the network penetration difference between unicast and broadcast communication in general. The unicast packets will require more retransmission as opposed to broadcast traffic. We compared the overhead of the quality estimation schemes by further

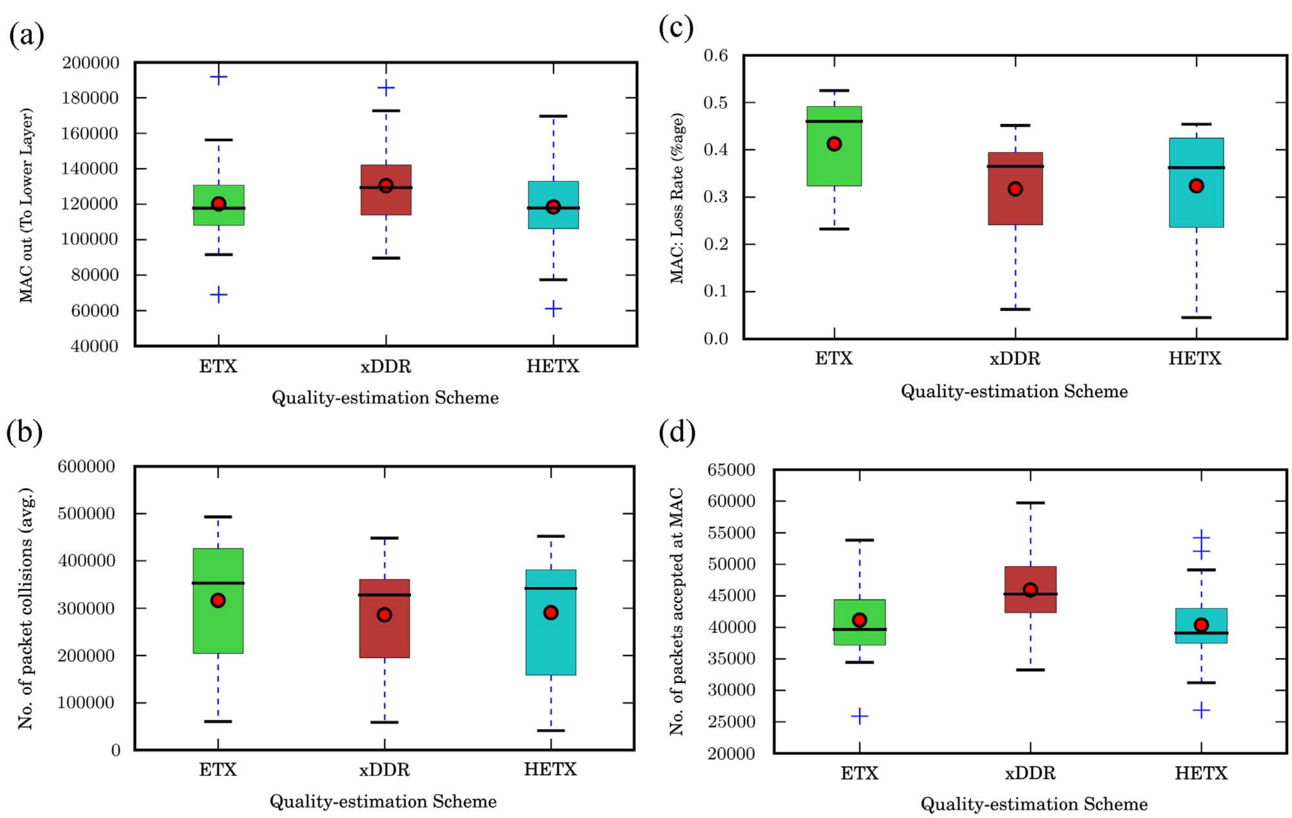


Fig. 16. (a) Total number of packets passed down by the MAC Layer (averaged). (b) Total number of collisions (averaged). (c) Loss rate at the MAC Layer (averaged). (d) Total number of packets accepted by the MAC Layer (averaged).

inspecting the effects on individual nodes particularly from the MAC layer's perspective. We averaged the results over all runs i.e. from network of 30 till network of 70 nodes with each network comprising 20 runs resulting in 20 distinct topologies in terms of node placement.

Fig. 16(a) demonstrates the total number of packets passed down via the MAC Layer per node averaged over all runs. Packets passed down via the MAC layer include routing control packets, ARP, UDP packets as well as control packets for the estimation schemes (xDDR/ETX/HETX) etc. HETX despite having higher average end-to-end PDR than ETX, recorded a smaller number of packets passed down via the MAC layer on average (average represented with scatter plot in red circle). This is because on average the chosen links required lesser retransmissions than ETX. xDDR though shares the same advantage as HETX generates additional packets due to unicast beaconing and their respective retransmissions. We believe this much overhead can be acceptable for applications seeking higher accuracy and end-to-end packet delivery.

It's worthwhile to mention that we expected a higher MAC output than recorded. To elaborate, in Section 6.2 we mentioned that ETX (and HETX) would generate $\mathcal{T}/\varepsilon$ control packets during the simulation duration $\mathcal{T}$ whereas xDDR would generate between $(\mathcal{T} \times |E'_m|)/\varepsilon$ and $(\mathcal{T} \times |E'_m| \times r)/\varepsilon$ depending on r. With unicast recipients limited to 4 ($|E'_m|=4$) and assuming on average 3–4 retries per unicast transmission for a duration of 600 s, we expected xDDR to generate approximately 15% more packets than the broadcast based schemes, as opposed to the 9% measured. Our hypothesis is that accurate link quality estimation results in comparably more uniform distribution of traffic across the network thereby resulting in less congestion and packet loss. This will consequently result in less MAC transmissions per data packet. Fig. 16(b) presents the average packet collision recorded per node averaged over all runs. xDDR nodes on average experienced lesser packet collisions as compared to both ETX and HETX. xDDR nodes on average experienced 286,102 packet collisions as compared to 316,578 and 290,771 in ETX and HETX respectively.

We further analyzed the loss rate registered at the MAC layer for each of the schemes averaged over all runs. Fig. 16(c) shows the comparison of average loss rate registered at the MAC layer. We see the effects of packet collisions on the loss rate with nodes running ETX experienced 41.25% loss as compared to 31.71% in xDDR and 32.42% in HETX.

Fig. 16(d) compares the number of packets accepted at the MAC layer. These are the packets that are either destined for the MAC layer or to be passed up to the higher layers therefore including estimation metric control packets, UDP traffic, ARP as well as routing packets. We see that xDDR nodes on average accept more packets than ETX and HETX which is due to two main factors i.e. more successful UDP data packets received and xDDR unicast beacon packets.

8. Conclusions

In this article we present xDDR, a link quality directional packet delivery ratio estimation metric devised after analyzing the behavior of the ETX metric. xDDR employs a pulse quality based neighbor selection and classification scheme in order to identify and classify neighbors. It selectively unicasts beacon packets to estimate directional delivery ratios of links and avoids the RREQ flooding phase problem by omitting estimation during the conflicting slots. By doing so we are able to achieve improved accuracy in link-level delivery ratio estimation. xDDR coupled with the end-to-end quality aware routing module provides far more accurate end-to-end route delivery ratio estimates. We compared xDDR with a minimum hop count, HETX [4] and ETX [7] in static as well as mobile environments and demonstrated the overall improvement. In a mobile environment, xDDR on average improves absolute packet delivery ratio percentage by 25%, 19% and 11% as compared to Minimum Hop Count, ETX and HETX respectively. This implies a 2 times gain over Minimum Hop Count, a 1.6 times gain over ETX and a 1.3 times gain over HETX in terms of actual packet delivery yielded as a result.

Acknowledgements

This work was supported by the SenSafety project in the Dutch Commit program, www.sensafety.nl.

References

[1] S.R.Afzal, M.Nabi, S.Stuijk, T.Basten, Improving end-to-end packet delivery in high traffic multi-hop wireless ad hoc networks, in: Proceedings of the 8th International Conference on Mobile Multimedia Communications MobiMedia, 2015.
[2] N.Baccour, A.Koubaa, L.Mottola, M.Zuniga, H.Youssef, C.Boano, M.Alves, Radio link quality estimation in wireless sensor networks: A survey. ACM Transaction on Sensor Networks, Sept, 2012.
[3] S.M.Das, H.Pucha, K.Papagiannaki, Y.C.Hu, Studying wireless routing link metric dynamics, in Proceedings of ACM SIGCOMMIMC '07, Octoberpp 0.327–332, 2007.
[4] A.T.Tran, M.K.Kim, Characteristics of ETX Link Quality Estimator Under High Traffic Load in Wireless Networks, in: Proceedings of the 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing HPCC_EUC 2013, pp.611,618, 13-15 Nov. 2013.
[5] R.Draves, J.Padhye, B.Zill, Comparison of routing metrics for static multi-hop wireless networks. InProc. ACM SIGCOMM Conference, September, 2004.
[6] R.Riggio, N.Scalabrino, D.Miorandi, G.Iacovoni, "Wireless mesh networks: state of the art and research directions". Create-Net Technical Report N. 200600006. April 21, 2006.
[7] D. Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing,, Wirel. Netw. 11 (4) (2005) 419–434.
[8] OMNeT++ Network Simulation Framework, ⟨http://www.omnetpp.org/⟩ (accessed November 2015).
[9] INETMANET Extension for OMNET++, ⟨https://github.com/inetmanet/inetmanet⟩ (accessed November 2015).
[10] A.Adya, P.Bahl, J.Padhye, A.Wolman, L.Zhou, A multi-radio unification protocol for IEEE 802.11 wireless networks, BroadNets '04.
[11] S.Keshav, A control-theoretic approach to flow control, in proceedings of SIGCOMM 1991. ACM Press, 1991.
[12] J.D.Padhye, R.P.Draves Jr., B.D.Zill, System and method for link quality routing using a weighted cumulative expected transmission time metric, US Patent, 2009.
[13] R.Draves, J.Padhye, B.Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: proceedings of MobiCompp. 114–128, 2004.
[14] D.Johnson, D.Maltz, Dynamic source routing in ad-hoc wireless networks, in: proceedings of ACM SIGCOMM '96, Aug, 1996.
[15] D. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, MobiCom 03 (2003) 134–146.
[16] S.R.Afzal, M.Nabi, S.Stuijk, T.Basten, "INLyD: Inter-Network-Layer Delay as a Low-cost Quality Metric for Multi-hop Routing in Wireless Mobile Networks," ACM PE-WASUN in: Proceedings of the 13th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Network*(in Press)*, 2016.
[17] D.L.Johnson, G.P.Hancke, "Comparison of two routing metrics in OLSR on a grid based mesh network," Ad hoc Networksavailable online 4 May, 2008.
[18] S.Waharte, B.Ishibashi, R.Boulaba, D.Meddour, Performance study of wireless mesh networks routing metrics, in: Proceedings of the IEEE/ACS AICCSA1100–1106, 2008.
[19] X. Ni, Performance evaluation of ETX on grid based wireless mesh networks, MPhil Thesis Report EET-UNSW, University of New South Wales, Australia, 2008.
[20] P.D.Marco, C.Fischione, G.Athanasiou, P.-V.Mekikis, MACaware routing metrics for low power and lossy networks, in: Proceedings of the IEEE INFOCOM Workshop, Aprpp. 79–80, 2013.
[21] A.Chan, S.-J.Lee, X.Cheng, S.Banerjee, P.Mohapatra, The impact of link-layer retransmissions on video streaming in wireless mesh networks, in: Proceedings of International Wireless Internet Conference. ACM, 2008.
[22] INET Extension for OMNET++ framework, ⟨https://inet.omnetpp.org/Introduction.html⟩ (accessed November 2015).
[23] C. Perkins, E. Belding-Royer, S. Das, Ad hoc on-demand distance vector (AODV) routing. No. RFC 3561. 2003.
[24] E.Presentation slides, people.ee.duke.edu/~romit/courses/s09/material/network-etx.ppt (accessed December 2015).
[25] J.Zhao, R.Govindan, Understanding packet delivery performance in dense wireless sensor networks. in: Proceedings of the First ACM Sensys Conference, November, 2003.
[26] N. Baccour, A. Koubaa, L. Mottola, M.A. Zuniga, H. Youssef, C.A. Boano, M. Alves, Radio link quality estimation in wireless sensor networks: a survey 8, ACM Transactions on Sensor Networks (TOSN), 2012, p. 34.
[27] A. Woo, D. Culler, Evaluation of efficient link reliability estimators for low-power wireless networks. Tech. Rep. UCB/CSD-03-1270, EECS Department, University of California, Berkeley, 2003 ⟨http://www.eecs.berkeley.edu/Pubs/TechRpts/2003/6239.html⟩.
[28] A.Cerpa, J.L.Wong, M.Potkonjak, D.Estrin, Temporal properties of low power wireless links: Modeling and implications on multi-hop routing. in: Proceedings of the 6th international symposium on mobile ad hoc networking and computing (MobiHoc '05). ACM, pp 414–425, 2005.
[29] R.Fonseca, O.Gnawali, K.Jamieson, P.Levis, Four bit wireless link estimation. in: Proceedings of the 6th international workshop on hot topics in networks (HotNets VI). ACM SIGCOMM, 2007.
[30] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research., Wirel. Commun. Mob. Comput. 2. 5 (2002) 483–502.
[31] P. Pinto, A. Pinto, M. Ricardo, End-to-end delay estimation using RPL metrics in WSN., IFIP Wirel. Days (WD) (2013) 1–6.

[32] C.E.Perkins, P.Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. in: Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications, pages 234-244, August, 1994.
[33] L. Hong, X. Liu, L. Zhang, et al., Towards sensitive link quality prediction in ad hoc routing protocol based on grey theory, Wirel. Netw. 21 (2015) 111. http://dx.doi.org/10.1007/s11276-015-0918-z.
[34] S. Liu, Y. Lin, J.Y.L. Forrest, Grey systems: theory and applications, Springer, Berlin, 2010.