IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 8, 2019, accepted August 9, 2019, date of publication August 19, 2019, date of current version September 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936191

Novel ETX-Based Metrics for Overhead Reduction in Dynamic Ad Hoc Networks

NENAD J. JEVTIC, (Member, IEEE), AND MARIJA Z. MALNAR
Faculty of Transport and Traffic Engineering, University of Belgrade, 11000 Belgrade, Serbia

Corresponding author: Marija Z. Malnar (m.malnar@sf.bg.ac.rs)

This work was supported by the Serbian Ministry of Education, Science and Technological Development under Projects TR32025 and TR36047.

ABSTRACT In recent years most of wireless ad hoc networks usually include a larger number of mobile nodes. To take this into account, routing protocols, initially proposed for static networks, need to be upgraded. Higher network performance can be achieved if a suitable routing metric is included, such as frequently used Expected Transmission Count (ETX). However, with ETX metric routing overhead is increased making it unusable for networks with a large number of mobile nodes. Therefore, we proposed ETX-based routing metrics called Light ETX (L-ETX), Light Reverse ETX (LR-ETX), and Power Light Reverse ETX (PLR-ETX) that reduce overhead and improve other network performances. In order to test the impact of proposed metrics on the network performance, all three metrics are implemented in the Network Simulator 3 (NS-3) within Ad hoc On Distance Vector (AODV) routing protocol. Since the NS-3 lacks standardized tools for network performance analysis, we developed Network Performance Analysis Framework (NPAF) and use it to compare the proposed metrics. NPAF provides support for network analysis based on key performance indicators such as throughput, packet delivery ratio (PDR), useful traffic ratio (UTR), end-to-end (E2E) delay, jitter, etc. NS-3 source code extensions are publicly available.

INDEX TERMS Ad hoc on distance vector (AODV) routing protocol, expected transmission count (ETX) metric, key performance indicators (KPIs), network simulator 3 (NS-3), wireless ad hoc networks (WANETs).

I. INTRODUCTION

Modern communications cannot be imagined without permanent usage of different types of wireless networks. Wide spectrum of everyday mobile applications increases the need for Wireless Ad hoc NETworks (WANETs) that combine wireless communication with a high degree of node mobility. Dynamic WANET architectures are usually implemented in the form of Mobile Ad hoc NETworks (MANETs) and Vehicle Ad hoc NETworks (VANETs). Since topology changes frequently in these networks, it is very important to find the optimal route between source and destination node. Therefore, the right choice of routing protocol and appropriate routing metric can improve the reliability and performance of WANETs.

Most of the popular routing protocols use hop-count as a default routing metric, but this metric often does not choose the optimal route between source and destination node. This problem is more evident in networks with dynamic nodes such as MANETs and VANETs since the quality of the chosen route changes over time, so a number of link brakes increases, and therefore high packet losses are introduced. In order to overcome hop-count shortcomings, especially in throughput and Packet Delivery Ratio (PDR), Expected Transmission Count (ETX) metric has been proposed in [1]. However, additional broadcast control packets, Link Probe Packets (LPP) are needed for ETX calculation, so the ETX metric has greater overhead than hop-count. Overhead is even more increased in ad hoc networks with mobile nodes because in these networks topology changes frequently and a higher number of route discovery procedures are needed. Having in mind all of this, we proposed three novel ETX-based metrics with the aim of reducing routing overhead and improving overall network performance, especially for mobile and vehicle wireless ad hoc networks.

Although many researchers proposed different routing protocol upgrades to improve network performance, the evaluation of proposed improvements often is not an easy task. When possible, measurements, experiments, and theoretical analysis should be included in research.

The associate editor coordinating the review of this article and approving it for publication was Muhammad Alam.

Regardless of network nature, and due to its large-scale, heterogeneity and dynamics, it can be complicated, time-consuming and expensive to provide an experimental evaluation of new protocols. If there is a need for a cost-effective and scalable mechanism to analyze the behavior of the network under different conditions or with different parameters, network simulation is a very popular choice. Network simulations can help in achieving a profound understanding of network behavior, as well as the verification of new solutions [2]. Different types of simulators, both commercial and open source, are available and can be employed for simulation modeling of WANETs. The most commonly used open source simulators are Objective Modular Network Testbed in C++ (OMNET++) [3], Global Mobile Information System Simulator (GloMoSim) [4], Network Simulator 2 (NS-2) [5], Network Simulator 3 (NS-3) [6], [7], etc. Two very popular commercial network simulators are OPtimized Network Engineering Tools OPNET [8] and QualNet [9].

NS-3 is widely used in the research community since it has a simulator core that supports parallel execution, includes a wide range of communications protocols, and it is relatively easy to extend for a particular requirement. Therefore, we have chosen the NS-3 simulator for the evaluation of our work. Although NS-3 supports several network types, routing protocols, Medium Access Control (MAC) interfaces etc., it lacks in standardized method for calculation of basic Key Performance Indicators (KPIs) such as throughput, Packet Delivery Ratio (PDR), End-to-End (E2E) delay, jitter, received packets, routing packets, dropped packets, overhead, etc. That was our motivation to provide NS-3 implementations of proposed metrics and a framework for efficient KPIs calculation.

The main contributions of this paper comprise the following aspects:

1) In order to decrease routing overhead, three novel ETX-based metrics especially suitable for dynamic WANETs called Light ETX (L-ETX), Light Reverse ETX (LR-ETX), and Power Light Reverse ETX (PLR-ETX) are proposed. These three metrics decrease routing overhead compared to ETX and at the same time improve other KPIs comparing with both ETX and hop-count metrics.

2) The impact of proposed metrics on the performance of routing protocols in WANETs is evaluated in the NS-3 simulator. Since neither routing protocol in NS-3 supports other metrics than hop-count, we have implemented models for ETX, L-ETX, LR-ETX, and PLR-ETX metrics within Ad hoc On Distance Vector (AODV) [10] protocol.

3) In order to efficiently analyze WANET networks, a NS-3 based Network Performance Analysis Framework (NPAF) for gathering simulation data, performing necessary statistical processing and calculating of basic KPIs, is developed.

4) Although proposed metrics can be used in any type of WANET, we chose a highly dynamic use-case of urban VANET to evaluate their performances. A comprehensive analysis of the proposed metrics has been conducted using NS-3 and NPA framework. The results of obtained KPIs are given together with the relevant comments and conclusions.

The remainder of this article is organized as follows. The state of the art of AODV-based routing protocols, modifications of ETX metric and protocol implementations in the NS-3 simulator are summarized in Section II. The proposed NPA framework for efficient simulation of WANETs is described in Section III. In Section IV proposed metrics that improve dynamic WANETs performances are given. Implementation details of proposed metrics in the NS-3 simulator are provided in Section V. Simulation results and discussions are given in Section VI. The final section gives conclusions and ideas for future work.

II. RELATED WORK

Although proposed back in 1999, AODV protocol is still one of the most used routing protocols for wireless ad hoc networks. One of the interesting research papers that analyze the use of AODV protocol in various types of MANETs is given in [11]. Priority AODV (P-AODV) [12] created for MANETs presents a new route maintenance mechanism that repairs a route by selecting a neighbor based on the traffic being handled by it. Fitness Function Ad hoc On demand Multipath Distance Vector (FF-AOMDV) [13] protocol is specially designed to reduce energy consumption in MANETs.

Further research results indicate that AODV protocol can also be used in different wireless networks such as Wireless Sensor Networks (WSNs). To make AODV protocol suitable for wireless sensor networks, a ZigBee AODV (Z-AODV) [14] protocol is proposed. Z-AODV has been further improved to be even more efficient through ZigBee Associated Gateways (ZAG) [15] and Directional Broadcasting algorithm in Routing Discovery (DBRD) [16] protocols.

Improvements of AODV protocol specially focused on VANETs can also be found in the literature. For instance, in Urban-AODV (U-AODV) [17] protocol various environmental parameters are proposed that when applied to AODV make it more suitable for VANETs. An improvement of AODV which modifies basic AODV routing parameters to provide a better quality of service is given in [18].

All previous modifications of AODV protocol are based on improvements in route discovery or route maintenance procedures. Another possible approach to improve routing protocols is to add a metric other than hop-count into route cost. One commonly used metric in WANETs is ETX, which introduces the dependence of the number of successfully received and sent packets into the link cost. Although the ETX metric was first published back in 2005, it is widely used and many researchers are still proposing its improvements for different routing protocols and network types.

One of the first improvements of ETX is Expected Transmission Time (ETT) [19] metric that includes both size of transmitted packet and bandwidth of a link into link cost. Based on ETT, Weighted Cumulative ETT (WCETT) metric that counts both inter and intra flow interference is proposed by the same authors. Results show that WCETT has better throughput than ETX metric. A set of routing metrics that follow dynamic link changes by introducing the parameter that depends on the received signal power is introduced in [20]. These metrics called powerETX, powerWCETT and powerMIC have shown better throughput and smaller E2E delay compared to ETX for mesh networks in indoor scenario. Source-Based Routing (SBR) [21] uses a novel routing metric as a combination of packet losses (ETX-based parameter), intra-flow and inter-flow interferences, and load at gateways to improve the performance of wireless mesh networks. Simulation results show that SBR improves throughput, E2E delay and loss ratio compared to ETX metric.

Some researchers proposed several improvements of ETX metric for WSN. An Energy Balanced Reliable Metric (EBRM) [22] is an improvement of ETX metric that can not only ensure reliable communication, but also balance the network load, and prolong the network lifetime. From the result of simulation, EBRM metric outperforms both ETX and hop-count metrics in PDR, network lifetime and energy consumption. An Energy-Efficient ETX (E^3TX) routing metric that combines the residual node energy with the ETX to get an optimal decision strategy is proposed in [23]. Results show that E^3TX has better performances in reliability, E2E delay, and energy consumption, but has shown smaller throughput compared to other metrics. Authors in [24] proposed SIGMA-ETX metric as a combination of the minimum hop-count metric and ETX, where the best route for the routing is calculated with the standard deviation of the ETX values between each node. SIGMA-ETX metric has proven better than ETX in terms of latency, PDR and energy consumption. Worst-case ETX (W-ETX) [25] is a routing metric that selects good quality links to route data in converge cast scenarios. W-ETX is compared to ETX and results show that W-ETX has better performances in terms of the number of transmission attempts and PDR due to queue overflow.

ETX metric can also be used for specific routing protocols in VANETs. For instance, Link State aware Geographic Opportunistic (LSGO) [26] routing metric takes a combination of geographic location and the link state information through ETX metric. The simulation results showed that LSGO metric improves PDR and throughput but has larger overhead compared to ETX. A Collision Aware Routing (SCAOR) [27] is a hybrid metric for VANETs that includes three parameters: link quality based on ETX, packet advancement, and node density. SCAOR improves the efficiency of packet delivery, E2E delay, throughput, and PDR compared to ETX and LSGO metrics. Fast ETX (F-ETX) [28] metric introduces a novel link quality assessment by using four link state estimators. It has shown more accurate and reliable link estimation results than ETX.

For testing of new protocols and metrics either hardware or software evaluation tools are needed. Most of the proposals from the previous papers are verified using open source network simulators such as OMNET++, NS-2 or NS-3. In recent years, the NS-3 simulator is widely used for the analysis of different networks: some researchers propose implementations of new models for devices and channels, while the others use NS-3 for testing and evaluation of new protocols. Most of the NS-3 models are developed for different types of wireless networks, for instance: model of multi-radio multi-channel 802.11-based mesh network is given in [29], performance evaluation of WirelessHART network model is proposed in [30], and implementation of ETX metric within AODV protocol for WANETs is given in our previous work [31], [32]. Development and performance evaluation of a mobile IPv6 model for NS-3 is given in [33], while in [34] a framework for IPv6 based implementation of Lightweight On-demand Ad hoc Distance-vector next generation (LOADng) protocol is presented. Besides wireless networks, some authors developed models in NS-3 for optical communications. For instance, a visible light communication module is given in [35], while the implementation of a 10-gigabit-capable passive optical network module is proposed in [36].

TABLE 1. Overhead analysis of ETX-based metrics.

Metric	Overhead analysis included in the paper	Overhead compared to ETX
WCETT [19]	✗	Expected to be worse
powerETX [20]	✗	Expected to be worse
SBR [21]	✗	Expected to be the same
EBRM [22]	✗	Expected to be the same
E^3TX [23]	✗	Expected to be worse
SIGMA-ETX [24]	✗	Expected to be the same
W-ETX [25]	✗	Expected to be the same
LSGO [26]	✓	Worse than ETX
SCAOR [27]	✗	Expected to be worse
F-ETX [28]	✗	Expected to be the same

Based on the previous analysis of the available research articles, it can be concluded that ETX metric is still in the focus of the researchers and that its modifications can be used to improve the performance of different types of networks. Although many authors tried to enhance ETX metric, they usually focus on the improvement of throughput, E2E delay, energy consumption, PDR, etc, but only a few of them considered routing overhead. As can be seen from Table 1, only one of the proposed ETX-modifications includes results of the overhead analysis, but it has shown worse results than ETX metric. Other metrics did not consider overhead, but in Table 1 we give our expectations of their overhead compared to the ETX.

Increased overhead is especially problematic for modern dynamic WANETs in which changes in topology occur frequently. Numerous route search requests and a large number of link probe packets lead to oversized routing overhead, so the performance of networks, with the increasing number of nodes and their mobility, is significantly degraded. Therefore, we proposed novel ETX-based metrics that are more suitable for dynamic network environments and provide not only smaller overhead compared to ETX, but also a significant improvement in other KPIs, as well. The efficiency of the proposed metrics is demonstrated in the case of a VANET network in urban conditions, but this solution can be used for any other wireless ad hoc network. For performance evaluation, we selected the NS-3 simulator because it is open source and widely used in the research community, and therefore a better verification of published results is possible. However, the NS-3 does not have the standard environment for the performance evaluation of the network. Therefore, in addition to the implementation of the proposed metrics, we have developed a framework for evaluating the performance of the network and determining the basic KPIs.

III. EXTENSION OF NS-3 SIMULATOR FOR NETWORK PERFORMANCE ANALYSIS

One of the most important aspects of each network simulation is to evaluate selected KPIs for the simulated scenario using reliable statistics. The selection of relevant KPIs for the simulation is especially significant because it is usually not possible that the proposed network improvements have the best results in all KPIs. The NS-3 simulator currently does not have an efficient and complete solution for evaluating at least the basic network performance parameters such as throughput, PDR, E2E delay, jitter, overhead, etc. However, this simulator provides several methods for collecting the simulation results:

- ASCII and PCAP (Packet CAPture) tracing,
- Flow Monitor,
- Trace sources & trace sinks,
- Data Collection Framework.

The ASCII tracing system is a good tool, but it is not suitable for extensive simulations since it creates relatively large data files that require subsequent data processing to get relevant KPIs. PCAP tracing relies on capturing network packets and also requires subsequent data processing with external packet analysis software, such as WireShark [37].

Flow Monitor (FM) [38] is a special NS-3 module that can be used to calculate basic KPIs such as throughput, E2E delay, and jitter. But FM calculates KPIs based on packets on the network layer instead of the application layer. It can be used only for unicast IP streams over TCP/UDP, while broadcast packets are not supported. Another important disadvantage is that FM does not support all routing protocols for example, Dynamic Source Routing (DSR) [39].

But the NS-3 provides high quality and efficient general tracing system for exporting data from the simulator, based on trace sources and trace sinks. Trace sources are embedded into NS-3 models and represent the source of internal data from a model that is otherwise difficult to access. These data are used by trace sink functions, which are created by the NS-3 users to calculate the corresponding KPIs based on the data obtained from the trace sources.

NS-3 also provides Data Collection Framework (DCF) module that can be used for statistical processing of data obtained from trace sources. DCF consists of three elements: probes, collectors, and aggregators. Probes format the data obtained from trace sources and forward them to collectors. Collectors use data from one or more probes and combine them to calculate network key performance indicators. The resulting KPIs collectors are forwarded to aggregators, which are used to store data in files or generate graphics. The DCF module provides good concepts for statistical analysis, but it lacks a large number of different types of probes and collectors, as well as the flexibility in software design for storing data in user defined format. Therefore DCF is not easy to use either for gathering data and calculation of network KPIs or for storing them into files.

Apparently, the NS-3 does not have the appropriate framework for data collection and automatic calculation of basic network KPIs. Although general principals are well implemented, final solutions seem to be incomplete and very inefficient, especially if multiple simulation iterations are needed to obtain statistically significant results. Therefore, users most often create their KPI calculators based on trace sources and trace sinks. This leads to inconsistent and unreliable solutions that differ among researchers, and which are often insufficiently documented and verified.

Therefore, we have developed Network Performance Analysis Framework (NPAF) for calculating the basic network KPIs that is publicly available [40] and open to use by the research community. The NPAF provides statistically processed results for KPIs that include: throughput, PDR, number of sent and received packets, Useful Traffic Ratio (UPR) and E2E delay statistics (minimum, maximum, average, median, histogram and jitter).

The NPAF is based on the modification of the NS-3 simulator at the application layer and consists of two parts. The first part includes a pair of test applications, one for generating and the other for receiving user data traffic. Installation of source and sink applications into NS-3 nodes follows the standard application installation procedure, as illustrated in Fig. 1. Both of these applications contain trace sources that are triggered whenever an application packet is sent or received. The second part of NPAF consists of matching trace sink functions (connected to application trace sources) that are used for application packet collection, statistical processing and storage of processed data into files.

To facilitate the use of NPAF to the potential users, implementation is described in more detail in the following Subsections. First, source and sink applications are described, and then statistical data processing and calculation of supported KPIs are provided. Also, one example of NPAF usage for a generation of statistically significant simulation results for selected KPI is given.

A. SOURCE AND SINK APPLICATIONS

For the generation of network traffic, two general purpose test applications are created, one at the source side used for generation of user data packets, called `StatsPacketSource`, and the other, at the destination side used for reception of data packets, called `StatsPacketSink`. Both of these applications are inherited from the NS-3 base class `Application` (Fig. 2), so they can be installed at the NS-3 nodes.

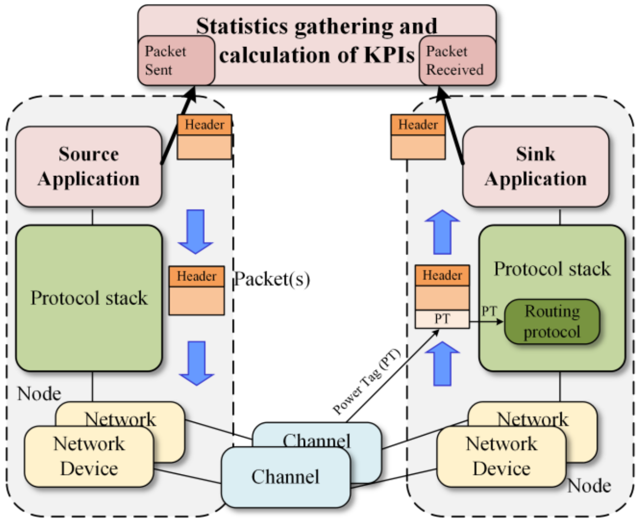


FIGURE 1. Installation of test applications in NS-3 nodes.

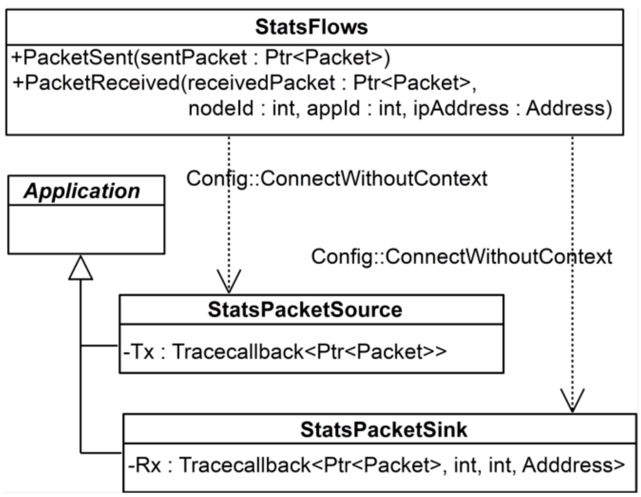


FIGURE 2. NPAF software organization.

StatsPacketSource is a modification of the frequently used OnOffApplication which is an integral part of the NS-3 simulator. This application provides generation of Constant Bit Rate (CBR) traffic that can be periodically enabled and disabled. The basic parameters of this application, packet size, and data rate, are controlled using the NS-3 attributes PacketSize and DataRate, respectively. Modifications made in OnOffApplication refer to providing a trace source for packet transmission and using a specific header in all application packets.

The first modification involves adding the trace source (named Tx in Fig. 2) that provides a copy of every generated application packet for trace sink functions. This trace source is used in statistical processing by the PacketSent trace sink function for counting all sent packets. The difference between all sent and all received packet counts is used to determine PDR.

The second modification made in OnOffApplication consists in adding a header to the application packets.

Seq ID (4B)	Time Stamp (8B)	Source Node ID (4B)	Source App ID (4B)
IPv4/IPv6 (1B)	IPv4 Address (4B)		Port (2B)

(a)

Seq ID (4B)	Time Stamp (8B)	Source Node ID (4B)	Source App ID (4B)
IPv4/IPv6 (1B)	IPv6 Address (16B)		Port (2B)

(b)

FIGURE 3. Structure of packet header, IPv4 (a), IPv6 (b).

TABLE 2. Modifications of source application.

Step	Description
1	Create new StatsHeader header
2	Increment sequence ID and add it to the header
3	Fill time stamp field with current simulation time
4	Add source node and application IDs to the header
5	Optionally add the IP address and port to the header
6	Create an application packet and add the header to it
7	Send the packet to the destination node and notify all trace sink functions connected to the Tx trace source

Packet header is defined by the class StatsHeader that is inherited from the NS-3 abstract class Header. Structure of the application packet header is shown in Fig. 3 and contains four mandatory fields: packet sequence number (Seq ID), the time when the packet is sent (Time Stamp), identifiers of the source node (Source Node ID), and source application (Source App ID). These data are used in statistical processing to provide unique identification of the received packet, as well as for calculation of KPIs. Applications can choose either IPv4 or IPv6 address type depending on the protocols used in the network layer. If a user needs information about the IP address of the node, three more fields should be added in the header: address type (IPv4/IPv6), IP address filed and port, Fig. 3.

Modification of StatsPacketSource application for sending of data packets is given by the algorithm in Table 2.

To calculate throughput, E2E delay and PDR, additional StatsPacketSink application, intended for installation in the receiving node, is implemented. The StatsPacketSink application is a modification of the existing PacketSink receiving application. In this case, modification consists only in adding the trace source (named Rx in Fig. 2) that provides a copy of the received packet and the information about the recipient node and application to the trace sink functions. Whenever StatsPacketSink application receives a packet, it uses this trace source to notify connected trace sink function PacketReceived. This function extracts information from the packet header to carry out statistical processing and calculation of KPIs.

B. STATISTICAL PROCESSING OF PACKET DATA

Previously described source and sink applications provide trace sources, but in order to have statistically processed data,

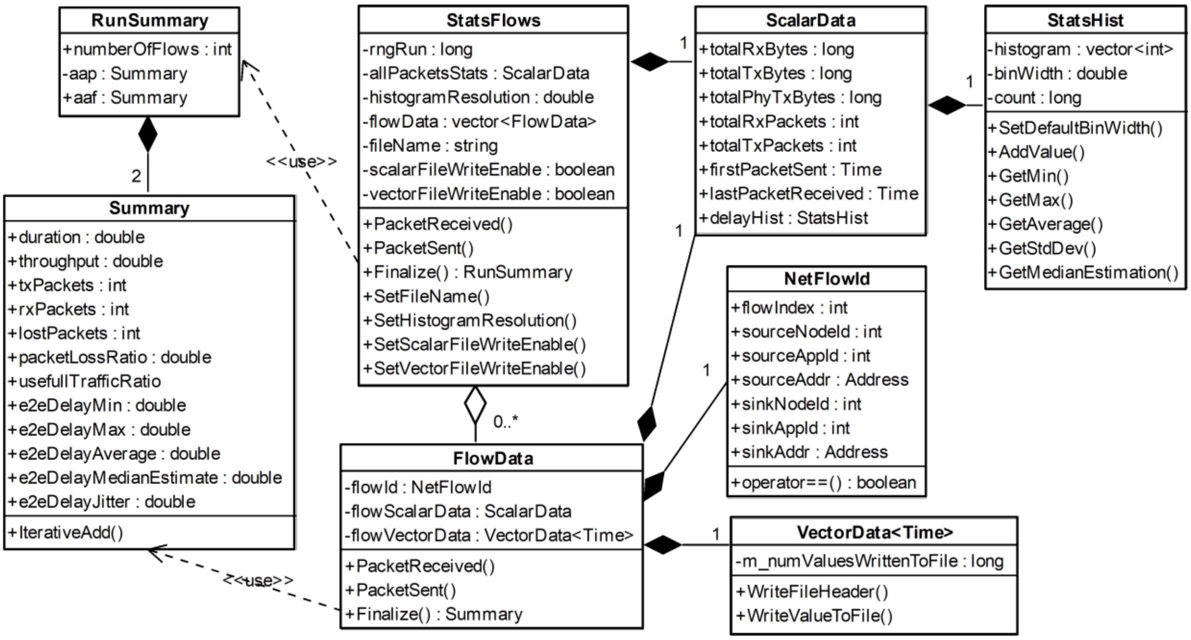


FIGURE 4. Class diagram of NS-3 extension software for statistical processing of data packets.

the development of matching trace sink functions is needed. Implementation of trace sink functions is done within the data collector class, named `StatsFlows`, by providing two functions, `PacketSent` for processing sent packets, and `PacketReceived` for processing received packets.

The connection between application trace sources (named `Tx` for transmitting and `Rx` for receiving packets) and trace sink functions is done automatically using the NS-3 `Config` system. Class diagram of NS-3 extension software for statistical processing of data packets is shown in Fig. 4.

Simulation scenario often requires that more than one node generate traffic. Therefore, it can be useful to classify packets according to network traffic flows. Network traffic flow (class `NetFlowId`) is uniquely defined with 4-tuple: source node ID, source application ID, sink node ID and sink application ID. Alternatively, IP addresses can be used instead of node IDs. For instance, if three nodes generate and receive traffic, the corresponding three network flows will be created. Every packet then can be uniquely identified by its sequence number and previously defined 4-tuple, which are obtained from the header attached to all packets generated by `StatsPacketSource` application. Based on this information, packets can be classified into appropriate network flow and processed accordingly.

Since multiple flows are possible, whenever the trace sink function `StatsFlows::PacketSent` receives a copy of a packet sent by `StatsPacketSource` application, it automatically detects new flows and stores their IDs and corresponding statistical data in the vector of `FlowData` objects by calling `FlowData::PacketSent` function.

When the packet is received by the `StatsPacketSink` application, tracing mechanism is used again to call `StatsFlows::PacketReceived` sink function to process data from the received application packet. `NetFlowId` is detected from the packet header and appropriate `FlowData::PacketReceived` function is called to calculate and store packet statistics.

For every flow, `FlowData` class keeps two types of records: vector and scalar. Vector data are processed in the `VectorData` class. These records are time dependent and are produced whenever a packet is received. Corresponding packet data, including reception time, packet flow ID, sequence number and E2E delay are stored into the output file using `WriteValueToFile` function.

Scalar data are processed in the `ScalarData` class, and include: number of sent and received packets, total amount of sent and received data, time when the first packet is sent, time when the last packet is received, and E2E delay histogram for all received packets. Packet delay histogram, implemented as `StatsHist` class, also includes functions for calculation of average, median, standard deviation, minimum and maximum values of E2E delay.

At the end of one simulation run all data from all flows, which are collected in the `StatsFlows` object, are written into the output file. This is done by calling member function `StatsFlows::Finalize` that calculates and writes to file all relevant network KPIs: throughput, PDR, E2E delay and others.

In this software release, all data are recorded in the text files using a comma as a separator, thus obtaining comma separated value (CSV) files. Since the CSV file format is highly portable, processing and graphical presentation of the results can be done in various software tools such as MS Excel or other spreadsheet programs, Matlab, Python, and many others. Simulation results are stored using three types of output files: vector, scalar, and summary file.

One vector file is produced for each simulation iteration and it contains time series data from `VetorData` class. Since the amount of data generated in this way is potentially very large (proportional to the total number of received packets in the network), vector files are disabled by default. The user can enable this feature by setting the appropriate flag (`vectorFileWriteEnable`) in the `StatsFlows` object.

One scalar file is also produced at the end of each simulation iteration and this file contains the data from `ScalarData` class. Scalar results are available for every individual flow but are also averaged for all flows and given at the end of the file. It should be noted that `StatsFlows` object also provides `ScalarData` statistics for all packets regardless of network flows, so the users can choose between statistics based on Averaging All of network Flows (AAF) or based on Averaging All of Packets (AAP). Since scalar data occupy a relatively small amount of memory, writing to scalar files is enabled by default, but also can be disabled using the corresponding `StatsFlows` flag (`scalarFileWriteEnable`).

Simulations are frequently repeated many times to obtain statistically significant results. In this case, it is inconvenient and time-consuming to handle a large number of created files in each simulation iteration. Within NPA framework we have also provided a generic simulation script (`multi-run.cc`) to suggest a simple way for controlling multiple simulation iterations. Every simulation iteration returns both AAP and AAF summary results to the main function of the simulation script using `RunSummary` class. These data are then written as one line into the common output summary file. Therefore, at the end of the simulation series, all results are stored in the single file.

C. CALCULATION OF KEY PERFORMANCE INDICATORS

The basic network KPIs that are available in the NPA framework are as follows.

1) THROUGHPUT

represents the number of successfully received bits at the destination node during active network time. It is expressed in (b/s) and it is calculated as:

$$Throughput\,[b/s] = \frac{8 \cdot totalTxBytes}{T_{activeNet}}, \tag{1}$$

where *totalTxBytes* is amount of received bytes on the application layer stored in `ScalarData` class, and $T_{activeNet}$ is time during which network is active (nodes send and receive application packets). Active network time period is calculated as:

$$T_{activeNet} = T_{lastPacketReceived} - T_{firstPacketSent}, \tag{2}$$

where $T_{lastPacketReceived}$ and $T_{firstPacketSent}$ are times when the last packet is received and the first packet is sent, which are also stored in `ScalarData` class.

2) PACKET DELIVERY RATIO (PDR)

is the ratio of the received data packets by the destination node to the data packets that are generated by the source node. PDR is calculated as:

$$PDR\,[\%] = \frac{totalRxPackets}{totalTxPackets} \cdot 100, \tag{3}$$

where *totalTxPackets* represents the number of packets generated by the source node and *totalRxPackets* number of packets that are received by the destination node, which are both stored in `ScalarData` class. Besides PDR, numbers of transmitted, received, and lost packets are also available in the scalar and summary output files.

3) USEFUL TRAFFIC RATIO (UTR)

is a performance indicator that shows the amount of the overhead introduced by all network layers. It is calculated as the ratio of all transmitted application bytes in source node (*totalTxBytes*) to all transmitted bytes (both data and control) in the same node at the physical layer (*totalPhyTxBytes*):

$$UTR\,[\%] = \frac{totalTxBytes}{totalPhyTxBytes} \cdot 100, \tag{4}$$

where both *totalTxBytes* and *totalPhyTxBytes* are stored in `ScalarData` class.

4) END-TO-END DELAY (E2E)

represents the time taken by a data packet to be transmitted through the network from source to the destination node. E2E delay includes all types of delays in the network: queuing delay, propagation and transfer times, MAC retransmissions delay, buffering during route discovery, etc. E2E delay is calculated as:

$$E2E\ delay\,[ms] = T_{packetReceived} - T_{packetSent}, \tag{5}$$

where $T_{packetSent}$ and $T_{packetReceived}$ represent moments in simulation time when application packet is sent from the source node, and when the same packet is received in the destination node, respectively. Packet sent time is embedded in the packet by the source application using packet header, and the packet received time is measured by the sink application. E2E delay given by (5) represents a delay of a single packet and it is stored in vector file (if enabled) and statistically processed in `StatsHist` class (included in `ScalarData` class). The `StatsHist` class provides various E2E delay statistics including: histogram, minimum and maximum E2E delay, average, median, and standard deviation.

Standard deviation of E2E delay is an important network KPI that is usually called **jitter** and represents the variation of data packets delay. In some cases, even if the delay is small, it can have great variations and therefore jitter has large values. This parameter is important since it influences the quality of service, especially for multimedia traffic (audio and video data).

Simulations usually need to be repeated many times with different random generator settings that provide statistically independent results. Therefore, the NPA framework includes overall statistical data: minimum, maximum, average, median and standard deviation of all KPIs, and store these data at the end of the summary output file (an example is given in the script `multi-run.cc`). Base on these data, users of the NPA framework can determine the optimal number of simulation iterations that should be performed to obtain statistically significant results. Choosing a higher number of iterations gives more reliable results, but also increases simulation time, especially when simulations include a large number of network nodes and a high level of network traffic load. As an illustration of this problem, Fig. 5 provides a graphical presentation of statistical analysis of simulation results for one chosen KPI: average E2E delay. Analysis is conducted for 300 simulation iterations and the results are obtained from the summary file.

Fig. 5 (a) shows that the results converge to their average value as the number of simulation iterations increases. This is an important observation since the users have to determine an optimal number of iterations that should be performed to avoid unreliable simulation results. One of the quantitative methods to evaluate the relevance of the simulation results is to determine the relative error of the results. Fig. 5 (b) represents the relative error of average value of the chosen KPI (E2E delay in this case), calculated as:

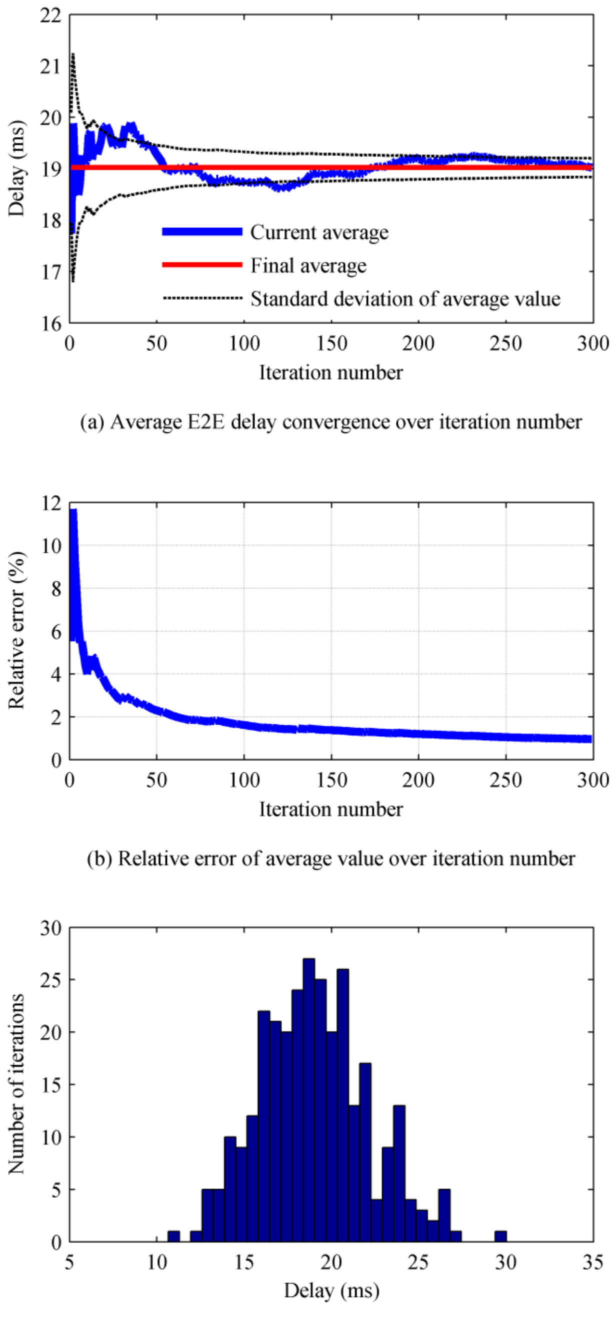
$$\varepsilon_{KPI}\,[\%] = \frac{\sigma_{KPI} \big/ \sqrt{n}}{\langle KPI \rangle}, \tag{6}$$

where $\sigma_{KPI} \big/ \sqrt{n}$ represents the standard deviation of the average value of the KPI, n is the number of iterations, and $\langle KPI \rangle$ is the average value of this KPI.

From Fig. 5 (b) it can be seen that at least 100 simulation iterations should be performed to achieve a relative error of average E2E delay less than 2%, and more than 300 iterations are needed for simulation results with relative error less than 1%. A similar analysis can also be conducted for any other network KPI supported by the NPAF, using data from the summary file. In our work, we have chosen to perform 100 simulations for every simulation scenario, and therefore achieved a relative error of less than 2% for presented simulation results.

IV. PROPOSED ETX-BASED METRICS

In this section three novel ETX-based metrics that have the aim to improve network performance without any investments in network hardware and with small changes in software of each node are described. Implementation details of all metrics in the NS-3 simulator are given in Section V.



(a) Average E2E delay convergence over iteration number

(b) Relative error of average value over iteration number

(c) Average E2E delay histogram

FIGURE 5. Average E2E delay convergence as a function of number of iterations (a), relative error of E2E delay (b) and E2E delay histogram (c).

A. EXPECTED TRANSMISSION COUNT METRIC

The Expected Transmission Time (ETX) metric for a link l, is given by [1]:

$$ETX_l = 1 \big/ \left(p_f \cdot p_r\right), \tag{7}$$

where p_f represents the probability of successful packet transmission and p_r the probability of successfully received ACK packet.

The probabilities p_f and p_r are measured using dedicated Link Probe Packets (LPPs) which are broadcasted every τ seconds [1]. To avoid accidental synchronization, period τ is jittered by up to $\pm 10\%$. Because the probes are broadcast, nodes do not acknowledge or retransmit them, so the calculation of probabilities p_f and p_r requires special technique. Every node remembers the number of received LPPs during the last w seconds allowing it to calculate the probability p_r at any time t as [1]:

$$p_r = count\ (t - w, t) \big/ w/\tau. \tag{8}$$

Count $(t - w, t)$ is the number of LPPs received during the window w, and w/τ is the number of LPPs that should have been received. With this technique, some node X can easily measure p_r by counting successfully received LPPs from its neighbor Y. But due to the lack of acknowledgments, node X cannot determine the probability p_f. Therefore, each LPP sent by node Y contains the number of LPPs received from X during the last w seconds and with this value node X can calculate the p_f. An empirical method is used for choosing w [1]. Experiments found that a value of $w = 10\tau$ performs well. Usually, the value of $\tau = 1$s is used.

The metric of a route from source to the destination node is the sum of the ETX values for each link l in the route [1]:

$$ETX_{route} = \sum_{l \in route} ETX_l. \tag{9}$$

ETX metric is proven to provide better results than hop-count, but one of the main drawbacks of this metric is increased overhead. Since in static networks with a small number of nodes overall performances are improved, generated overhead can be tolerated. However, in dynamic WANETs, with the rise in node density, the number of routing control packets is growing. The generated routing overhead increasingly affects the degradation of other network KPIs, hence the positive effect of introducing ETX metric is significantly reduced. Therefore, in the following Subsections, we described three novel ETX-based metrics that reduce overhead while improving other network KPIs.

B. LIGHT ETX METRIC

The first proposed modification of the ETX metric, Light ETX (L-ETX), is designed to reduce the size of the ETX field in the routing packets. This is done because the original ETX metric, defined by (7), can have values from 1 (when both p_r and p_f are equal to 1) to infinity (when at least one of p_r or p_f is equal to zero) with a highly non-uniform distribution of values, where values around 1 are very dense and large values are rare. Therefore floating point number should be used for ETX representation, which leads to the addition of at least 4 bytes for ETX field in both Route REQuest (RREQ) and Route REPlay (RREP) packets [32].

In order to reduce ETX field to 1 byte, L-ETX metric for a link l is defined by:

$$L - ETX_l = \left[60 \cdot \log\left(ETX_l\right)\right], \tag{10}$$

where ETX_l metric is given by (7).

The logarithmic function is used to increase differences in close values around the $ETX_l = 1$. Metric is then rounded to an unsigned integer value of 1 byte. When using (10), values for *L-ETX*$_l$ can range from 0 to 255, where 0 is reserved for the best links and 255 for the worst links. Reducing ETX field length to 1 byte is very important since in broadcast RREQ packet we can use reserved bits in the packet header. This way size of the RREQ packet will stay the same as with basic AODV protocol, and routing overhead will not increase. For RREP packet additional field must be added for L-ETX metric, but RREP is a unicast packet and this additional field will not significantly influence the overhead.

Coefficient 60 in (10) is chosen experimentally based on a series of simulations to achieve the best network performance. Lower values of this coefficient reduce the boost effect (for small values of ETX_l) that is introduced with logarithmic function and therefore degrade the performance of the L-ETX metric. Larger values of this coefficient can cause saturation problems for low quality multihop links.

C. LIGHT REVERSE ETX METRIC

The ETX overhead can be further reduced by decreasing the length of LPP packets. As already mentioned, a node can calculate reverse probability p_r only by counting the number of LPPs that are received from its neighbors in that node. But, this node cannot easily calculate forward probability p_f of successful reception of LPPs by neighboring nodes due to lack of acknowledgments. In order to calculate probability p_f, each LPP has to include information about the number of LPPs that are successfully received by neighboring nodes in the last w seconds. Hence LPP has to include potentially lots of information since a number of neighbors can be large. As the density of nodes in the network increases, the number of neighbors also increases and the size of LPPs grows. A significant improvement can be achieved for mostly symmetric links where $p_f \approx p_r$, so the calculation of p_f is not necessary. For these cases, we propose Light Reverse ETX (LR-ETX metric) for a link l defined by:

$$LR - ETX_l = \left[60 \cdot \log\left(ETXr_l\right)\right], \tag{11}$$

where $ETXr_l$ is based only on the reverse probability of successful reception of LPP and is given by:

$$ETXr_l = 1 \big/ p_r. \tag{12}$$

As can be seen from (12) only reverse probability is used to calculate $ETXr_l$, so the size of broadcast LPP packets is kept fixed and small, thus reducing the ETX overhead.

Although LR-ETX metric is suitable for networks with symmetric links, it can also be used for networks with asymmetric links with a small modification of the protocol. In this case, two ETX fields should be introduced into RREQ and RREP packets to simultaneously carry the information of ETX value along the route in both directions.

D. POWER LIGHT REVERSE ETX METRIC

ETX metric has one more disadvantage that is significant for highly dynamic networks. As previously stated LPPs are broadcasted every τ seconds and counted during w seconds to measure the quality of the link. These parameters are obtained experimentally and usually found that a value of $w = 10\tau$ performs well, while the value of $\tau = 1$s is mostly used. However, the value of $w = 10$s may be potentially too large, making the metric slow in dynamic environments.

By choosing smaller values for w, the metric can be corrected to have a faster response, but the resolution w/τ is reduced. Choosing smaller values of τ can increase the resolution of the metric but also increases overhead. In previous work [20] it is proven that received signal power can be used to enhance ETX metric for static nodes. For mobile nodes received signal strength can be used to improve the dynamic response of the metric without increasing the overhead. Therefore, we propose Power based Light Reverse ETX (PLR-ETX) metric for a link l defined by:

$$PLR - ETX_l = \left[60 \cdot \log\left(ETXr_l \cdot (1 + 0.005 \cdot X)\right)\right], \quad (13)$$

where $ETXr_l$ is given by (12), and parameter X represents the influence of Receiver Signal Strength Indicator (RSSI) into the metric, and is given by:

$$X = \begin{cases} 0, & RSSI \geq 0\text{dBm} \\ |RSSI|, & -100\text{dBm} < RSSI < 0\text{dBm} \\ 100, & RSSI \leq -100\text{dBm} \end{cases}. \quad (14)$$

It should be noted that similarly to L-ETX and LR-ETX metrics, coefficients 60 and 0.005 have been chosen experimentally by analyzing a series of simulations to achieve optimal network performance. Although in this paper, the optimal coefficient values were selected experimentally, the analysis showed that coefficients values can be chosen from a very wide range of values, with the little impact on the performance of the metric. Metric coefficients should not be chosen well below the suggested values as this significantly reduces the resolution of the metric, and therefore reduces its performance. However, choosing higher values has little effect on the performance of the metric. Tests of the proposed metric have shown that a significant improvement is achieved over the ETX for other simulation scenarios as well, so it can be assumed that those coefficients are a good choice for practical applications in real environments.

Analogous power based metrics can be formulated for ETX and L-ETX metrics as well, but further analysis in this paper includes only modification of LR-ETX since it has the greatest potential in reducing overhead and therefore achieving better overall network performance.

V. IMPLEMENTATION OF THE PROPOSED METRICS IN NS-3 SIMULATOR

The NS-3 simulator has recently become very popular as a research tool for studying the performance of wireless networks. The NS-3 includes all basic routing protocols, but it does not contain the implementation of ETX metric for any of these protocols. In order to test the performance of the proposed ETX-based metrics in AODV protocol, it is necessary to provide appropriate implementations of all metrics in the NS-3 simulator. It should be noted that although only AODV protocol is considered in this paper, all metrics can be included in any other routing protocol in a similar manner.

Type	8b
LPP ID	8b
Originator IP Address	32b
Originator Sequence Number	32b
Number of Neighbors (n)	8b
Neighbor IP Address 1 (32b)	n * 40b
Forward LPP Count (8b)	

FIGURE 6. Structure of LPP packet for ETX and L-ETX metrics.

A. IMPLEMENTATION OF ETX

In order to implement the AODV-ETX protocol, several modifications of the existing implementation of the AODV protocol should be made:

- Implementation of LPP packets,
- Creating a new list of neighbors that can provide counting of received LPPs and calculation of ETX metric for all neighboring nodes,
- Modification of the RREQ and RREP packets to include the ETX metric field,
- Extension of routing tables with ETX field, and
- Modification of the routing algorithm in terms of the response to RREQ packets in order to find routes with the smallest value of ETX.

The first modification to be performed in the original AODV protocol is the realization of LPP packets, which are used to determine the ETX metric of the link between the two adjacent nodes. AODV already has four types of routing control packets, so the fifth packet type had to be introduced using the `LppHeader` class. The structure of the LPP packet is shown in Fig. 6. The LPP packet for ETX and L-ETX metrics has identical structure and consists of seven fields:

- Type (8b) – indicates new LPP type of AODV packet,
- LPP ID (8b) – sequential number of LPP used for identification,
- Originator IP Address (32b) – IPv4 address of the node that generates LPP packet,
- Originator Sequence Number (32b) – sequence number of the node that generates LPP that is used for the route discovery process in AODV protocol,
- Number of Neighbors (8b) – indicates the number of neighbors whose LPPs are received by this node,
- Neighbor IP Address (32b) – IPv4 address of a neighbor form which at least one LPP packet is received in last w seconds,

- Forward LPP Count (8b) – indicates the number of received LPP packets from the neighbor, with the Neighbor IP Address from the previous filed, in the last w seconds.

The last two fields in the LPP packet are repeated for each neighbor. Since for dense networks the number of neighbors can be very big, the length of LPP packets is potentially large.

Another major modification of the AODV module, which is important for the implementation of ETX metric, is the introduction of a new neighbor table that stores data entries on the number of successfully received LPP packets. This is done using the `NeighborEtx` class. Each table entry corresponds to one neighbor. Table entries have three fields:

- Neighbor IP Address, which contains IPv4 address of the neighbor,
- Reverse LPP Count, which stores the number of LPP packets received from the neighbor in last w seconds, and
- Forward LPP Count, which stores the number of LPP packets that the neighbor has received from this node in last w seconds; these data are obtained by reading the contents of the LPP packet.

Having these two values of forward and reverse LPP counts for each neighbor, the node can calculate ETX metric for all its neighbors according to (7)-(8) and use it for finding the optimal route with the lowest ETX value.

The inclusion of ETX metric in the AODV protocol requires the insertion of the ETX field in more than one place in the implementation of the protocol. First, each entry in the routing table should be extended with the ETX field. In the AODV protocol, entries in the routing table are classified by destination. If there is more than one route to the destination, the best route is the one with the least number of hops. By adding ETX metrics in AODV protocol, the routes should be classified in accordance with the ETX value of the route, where the best route is with the lowest ETX. If there is more than one route to the destination node with the same ETX value, the best is one with the lowest number of hops.

Also, both RREQ and RREP packets have to be modified to include the ETX field. When the source node creates RREQ, it sets the initial value of ETX field to zero. This field is processed similarly as the number of hops. As the RREQ is forwarded to the destination node, the ETX field is updated in each node. The same principle is used for RREP.

The last modification of the AODV protocol is in the way that network nodes handle RREQ packets. In the basic AODV protocol, the node responds only to the first received RREQ (with the given source and destination addresses and the same ID). However, with the inclusion of ETX in the AODV protocol, such algorithm is not acceptable, since the goal is not to find the shortest route but the route with the lowest ETX value. Therefore, even if the node has already received the RREQ with the same ID, it should check if the ETX of the received RREQ is lower than the previously known ETX metric for that route. If the RREQ packet contains a lower metric value, the node will modify its routing table and respond to that RREQ. In this way, the detection of the route with the lowest ETX metric is ensured, and therefore the selection of the optimal route is guaranteed. A more detailed description of ETX implementation can be found in our previous work [31], [32].

Type	8b
LPP ID	8b
Originator IP Address	32b
Originator Sequence Number	32b

FIGURE 7. Structure of LPP packet for LR-ETX and PLR-ETX metrics.

B. IMPLEMENTATION OF L-ETX

Light ETX implementation follows the same principle as ETX. The only difference is that the calculation of the metric is done according to (10). This also leads to the size reduction of ETX field in the routing control packets, routing table and neighbor table. The size of this field is reduced from 4 bytes, which is necessary for the representation of a floating point number, to 1 byte unsigned integer. This reduction should result in decreasing overhead and performance improvements as previously explained.

C. IMPLEMENTATION OF LR-ETX

In order to further reduce the overhead, the LR-ETX metric calculates ETX value based only on reverse LPP count. Therefore, all data for neighbors can be excluded from LPP, which allows the probe packet to be of the fixed length of only 10 bytes, as shown on Fig. 7. This modification not only reduces overhead but also more accurately measures probability p_r and therefore improves other network KPIs.

Another significant change comparing to ETX and L-ETX metrics is in the implementation of the `NeighborEtx` class. The ETX and L-ETX metrics store data for both forward and reverse probabilities of receiving probe packets. The LR-ETX metric only store data for the reverse LPP reception probability and the metric is calculated according to (11)-(12).

D. IMPLEMENTATION OF PLR-ETX

The implementation of power based metric follows the implementation of previously explained ETX-based metrics with the addition of the RSSI as indicated by (13)-(14). The RSSI value provides information on fast changes of the link state due to movements of network nodes and other changes of propagation conditions and can assist in choosing a better route in dynamic environments.

The value of RSSI is provided to the routing protocol using the `RxPowerTag` class, which is a specialized type of NS-3 `Tag`. The `RxPowerTag` is attached to the received packet in the destination node (Fig. 1) and conveys information of the received signal strength at the moment of packet reception. In our simulation scenario, we used the `YansWifiChannel` model to simulate propagation

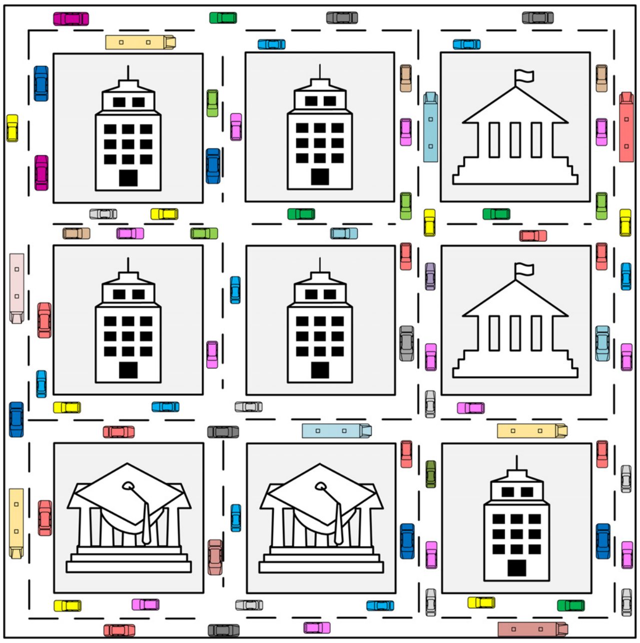


FIGURE 8. City area used for the VANET simulation scenario.

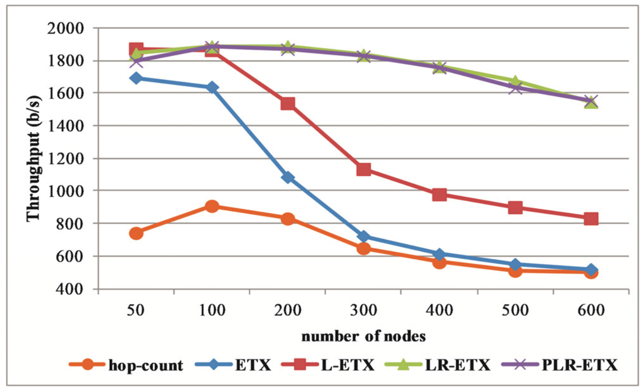


FIGURE 9. Throughput.

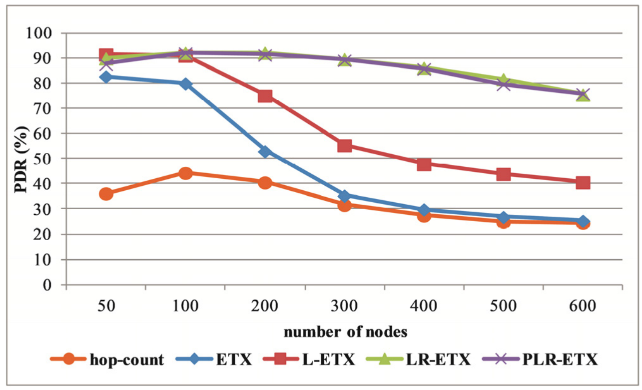


FIGURE 10. Packet Delivery Ratio (PDR).

through a physical medium, so the appropriate changes in the function `YansWifiChannel::Receive` were made to add packet tag with the appropriate value of RSSI. A similar modification can be done for any other NS-3 physical model.

VI. SIMULATION RESULTS

In this paper, we analyzed the performance of the proposed ETX-based metrics in one typical use-case of dynamic WANET – a VANET network in a city scenario. VANET performance evaluation is carried out in the NS-3.29 simulator, based on KPIs obtained using the NPAF.

Vehicles are distributed over the city area of $2000 \times 2000\text{m}^2$, divided into 3×3 blocks as shown in Fig. 8. Since NS-3 does not support the Manhattan grid model, the mobility scenario is generated using BONNMOTION tool [41] with the following parameters. The distance after which a new speed and direction are chosen (update distance) is set to 5 m, probabilities of speed and turn change are both set to 0.5, mean speed is 12 m/s, standard deviation of a speed is 0.6 m/s and probability that vehicle would stop after update distance is set to zero. The mobility model is loaded in NS-3 using NS-2 trace files.

A MAC sub-layer 802.11p with 6 Mb/s throughput and 10 MHz bandwidth is used. Channels are modeled according to `YansWiFiChannel` with the two-ray ground propagation model. Packets are routed using the AODV routing protocol and five routing metrics are used: hop-count, ETX and three novel metrics proposed in this paper L-ETX, LR-ETX, and PLR-ETX. UDP protocol is applied in the transport layer. In all scenarios 10 randomly distributed vehicles simultaneously generate Constant Bit Rate (CBR) traffic with fixed size packets of 64 bytes and bit rate of 2 kb/s.

To analyze the impact of traffic jams on the performance of the proposed routing metrics, several cases are considered, when the network consists of 50, 100, 200, 300, 400, 500, and 600 vehicles. For each case and all 5 metrics 100 simulations are done with different `RngRun` values, having a total of 3500 simulations. The number of 100 iterations is chosen to obtain that average values for all KPIs have a relative error less than 2% (calculated according to (6)).

The performances achieved using the proposed L-ETX, LR-ETX, and PLR-ETX metrics are compared with ETX and hop-count based on several KPIs obtained from NPA framework: throughput, PDR, UTR, maximum, median, average, and jitter of E2E delay. Simulation results are shown in Figs. 9-15.

By comparing the proposed metrics in terms of throughput and PDR (Figs. 9 and 10), it can be noticed that throughput and PDR show the same behavior. The hop-count metric achieves the worst performance with a very small packet delivery ratio that is about 25%. The introduction of the ETX metric leads to finding high-quality routes and therefore significantly increases the PDR. This is the expected result for VANETs with a small number of vehicles. But as the number of vehicles increases, ETX metric introduces large overhead, which leads to network congestion and small PDR. The advantage of ETX in finding high-quality routes is diminished and therefore delivery ratio reaches similar values as with the hop-count metric. Since L-ETX metric reduces

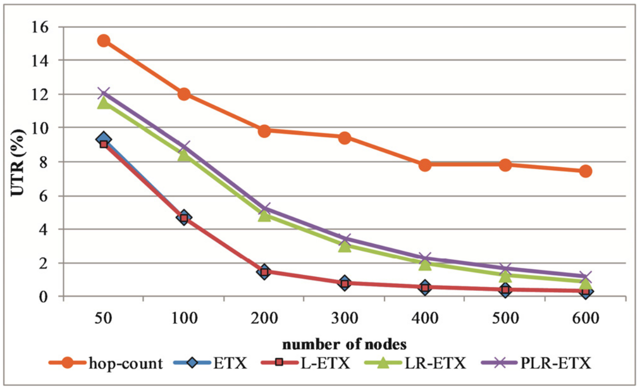


FIGURE 11. Useful Traffic Ratio (UTR).

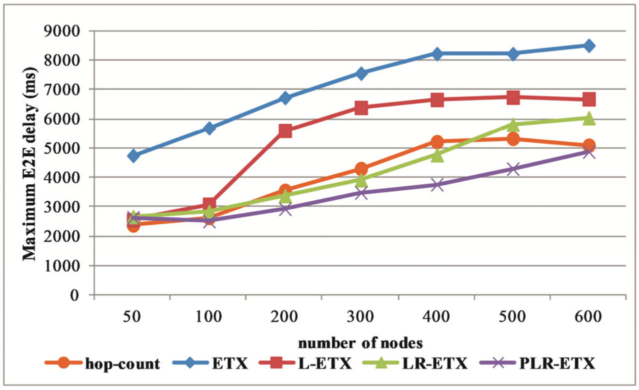


FIGURE 12. Maximum End-to-End delay.

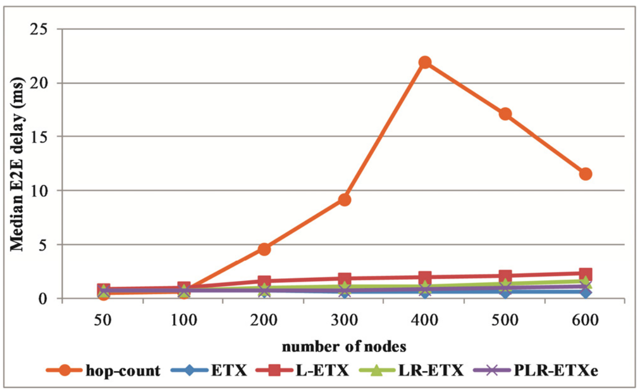


FIGURE 13. Median End-to-End delay.

overhead compared to ETX, its performance in terms of PDR is, as expected, better than with ETX. However, with the increasing number of vehicles L-ETX metric shows similar degradation in performance as ETX, but PDR is still 10-15% better. LR-ETX and PLR-ETX both use only reverse LPP count in calculating ETX and therefore introduced overhead is significantly lower than with L-ETX and especially ETX metric. Both of these metrics show huge improvements in PDR of about 50%. Results show that reverse LPP count is sufficient to produce reliable values for link metric providing low overhead and therefore retain the good performance of ETX even for a large number of vehicles in the VANET.

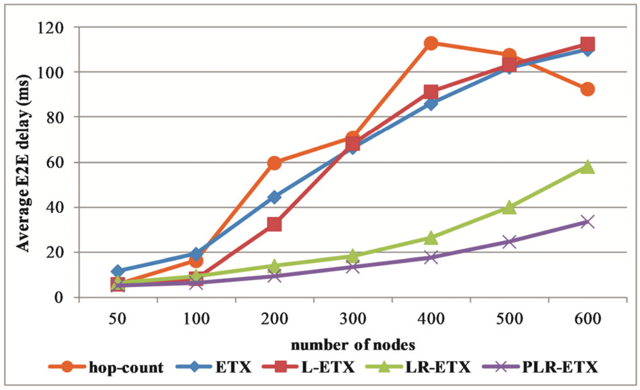


FIGURE 14. Average End-to-End delay.

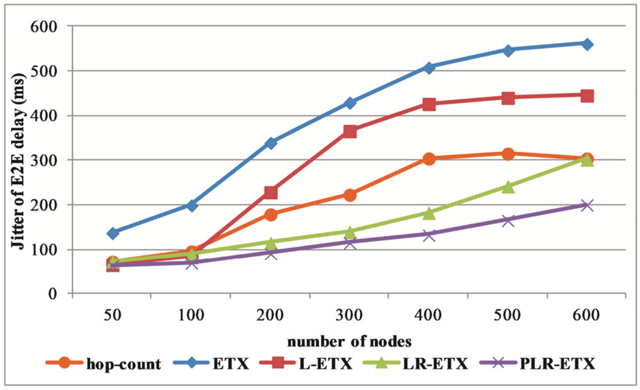


FIGURE 15. Jitter of End-to-End delay.

When compared to UTR (Fig. 11), it can be noticed that regardless of used metric routing protocols introduce overhead that increases with the number of vehicles. It should be noticed that UTR is calculated according to (4) and includes complete overhead from all network layers. Hop-count shows the best performance since it does not introduce any overhead due to metric calculation. ETX and L-ETX metrics behave almost the same because the reduction of overhead in L-ETX is negligible compared to overhead from all network layers, as shown in Fig. 11. For a small number of vehicles LR-ETX and PLR-ETX significantly outperform ETX and L-ETX metrics, but for a higher intensity of the road traffic, performance of all ETX-based metrics tend to very low values, although PLR-ETX always shows the best result. This could be explained using the fact that PLR-ETX always chooses the most reliable routes, thus reducing the routing overhead due to link brakes (that generate a large number of route error and route discovery packets).

Comparative results of E2E delay for the chosen VANET scenario are shown in Figs. 12-15. In order to emphasize the differences in proposed metrics, figures show maximum, median, average, and jitter of E2E delay, respectively. All of these delay parameters are included in the evaluation, because the AODV protocol introduces large initial packet delay before the route from source to destination is found, while in steady state E2E delay shows significantly lower values.

The initial delay is easily analyzed using maximum values of E2E delay (Fig. 12) and steady state delay is represented using median values of E2E delay (Fig. 13). ETX metric shows the worst performance in the initial delay, but the best performance in steady state. This is expected since large overhead slows down the route discovery process, but the routes found have the smallest delay. Hop-count shows great oscillations for the results of steady state delay, but despite these large fluctuations, in each test case and for each number of vehicles, the proposed metrics produce much better results than hop-count. All three novel ETX-based metrics show better performance in initial delay then ETX, but the PLR-ETX metric shows the best behavior, the smallest initial delay and almost the same median delay as ETX metric.

Average and jitter values (Figs. 14 and 15) provide an overall insight into the impact of proposed metrics on the E2E delay since they include both initial and steady state behavior. LR-ETX and especially PLR-ETX metrics show convincingly better results in overall performance of E2E delay in the analyzed scenario.

Bearing in mind all the previous results, it is clear that the AODV protocol with the default hop-count metric is not good enough for use in VANETs because it shows instability both in terms of delays and packet loss. Simulation results showed that the usage of ETX-based metrics can significantly improve network performance. The ETX metric, however, does not produce good results when the number of vehicles is increased due to large overhead. Reduction of overhead further enhances other network KPIs, especially if LPP packets are made sufficiently small by the elimination of all redundant data from the packet. Eliminating forward packet counts from the packet significantly reduces the size of the LPP packet, which clearly increases the efficiency of the protocol and improves the network performance. But in this way, the resolution of ETX metric is also reduced, and the modification of the metric by introducing an additional factor that depends on RSSI provides higher resolution and faster response in the dynamic conditions that are present in VANETs. It can, therefore, be concluded that the use of power based metric with only LPP reverse count (PLR-ETX) gives the best overall results.

VII. CONCLUSION

In this paper, we proposed three novel ETX-based routing metrics L-ETX, LR-ETX, and PLR-ETX that improve the performance of dynamic WANETs. Metrics performances are compared using the NS-3.29 network simulator. Since available routing protocols in the current version of NS-3 support only hop-count, we provided implementations of ETX and the three proposed metrics within the AODV routing protocol. It is important to note that all metrics can also be applied in other routing protocols for ad hoc networks in a similar manner.

In order to compare proposed metrics in the NS-3 simulator, the NPA framework is implemented, and the source code is made publicly available. It is developed to provide a standardized and efficient tool for basic KPIs calculation such as throughput, packet delivery ratio, useful traffic ratio, maximum, average, and median E2E delay, jitter, etc. Although we tested NPAF on AODV protocol in a wireless network, it can be used for any other routing protocol, and, with small modifications, in other networks as well. We hope that NPAF can be useful to the research community for easier comparison of achieved results, and for allowing researchers to focus on the development of new models instead of developing their own KPIs calculation tools.

Based on KPIs obtained from NPAF, we showed that the introduction of novel ETX-based metrics in the AODV protocol can significantly improve the performance of VANETs in the city scenario. This result is not limited only to VANETs and can be used for any other type of WANET. All proposed metrics reduce overhead compared to ETX, but also significantly improve other network performance indicators as well. Throughput and packet delivery ratio are much better than with both ETX and hop-count, especially for the higher number of nodes. End-to-end delay indicators also show improvements, particularly for PLR-ETX metric. But additional work can still be done in order to further reduce routing overhead.

REFERENCES

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

[2] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 4, pp. 392–403, Aug. 2001.

[3] *OMNet++*. Accessed: Aug. 21, 2019. [Online]. Available: http://www.omnetpp.org/

[4] *GloMoSim*. Accessed: May 1, 2013. [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim/

[5] *NS-2*. Accessed: Aug. 21, 2019. [Online]. Available: https://www.isi.edu/nsnam/ns/

[6] G. F. Riley and T. R. Henderson, "The *ns*-3 network simulator," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross, Eds. Berlin, Germany: Springer, 2010, pp. 15–34.

[7] *NS-3*. Accessed: Aug. 21, 2019. [Online]. Available: http://www.nsnam.org/

[8] X. Chang, "Network simulations with OPNET," in *Proc. IEEE Simulation Conf.*, Phoenix, AZ, USA, vol. 1, Dec. 1999, pp. 307–314.

[9] *Qualnet*. Accessed: Aug. 21, 2019. [Online]. Available: https://www.scalable-networks.com/qualnet-network-simulation

[10] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc on Demand Distance Vector (AODV) Routing*, document RFC 3561, IETF, Oct. 2003.

[11] Z. Ismail and R. Hassan, "A performance study of various mobility speed on AODV routing protocol in homogeneous and heterogeneous MANET," in *Proc. 17th Asia Pacific Conf. Commun.*, Sabah, Malaysia, Oct. 2011, pp. 637–642.

[12] D. K. Sharma, A. N. Patra, and C. Kumar, "P-AODV: *A priority* based route maintenance process in mobile ad hoc networks," *Wireless Pers. Commun.*, vol. 95, no. 4, pp. 4381–4402, Aug. 2017.

[13] A. Taha, R. Alsaqour, M. Uddin, M. Abdelhaq, and T. Saba, "Energy efficient multipath routing protocol for mobile ad-hoc network using the fitness function," *IEEE Access*, vol. 5, pp. 10369–10381, 2017.

[14] T. de Almeida Oliveira and E. P. Godoy, "ZigBee wireless dynamic sensor networks: Feasibility analysis and implementation guide," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4614–4621, Jun. 2016.

[15] J. Mu, "An improved AODV routing for the ZigBee heterogeneous networks in 5G environment," *Ad Hoc Netw.*, vol. 58, pp. 13–24, Apr. 2017.

[16] J. Mu, "A directional broadcasting algorithm for routing discovery in ZigBee networks," *EURASIP J. Wireless Commun. Netw.*, vol. 94, p. 94, May 2014. Accessed: Aug. 6, 2019. doi: 10.1186/1687-1499-2014-94.

[17] F. S. Mubarek, S. A. Aliesawi, K. M. A. Alheeti, and N. M. Alfahad, "Urban-AODV: An improved AODV protocol for vehicular ad-hoc networks in urban environment," *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 3030–3036, 2018. Accessed: Aug. 21, 2019. [Online]. Available: https://www.sciencepubco.com/index.php/ijet/article/view/21535
[18] S. Goyal, "An improvement of AODV routing protocol for vehicular ad-hoc networks (VANETs)," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 4, no. 6, pp. 2533–2538, Jun. 2015. [Online]. Available: http://ijarcet.org/wp-content/uploads/IJARCET-VOL-4-ISSUE-6-2533-2538.pdf
[19] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, Philadelphia, PA, USA, 2004, pp. 114–128.
[20] M. Malnar, N. Neskovic, and A. Neskovic, "Novel power-based routing metrics for multi-channel multi-interface wireless mesh networks," *Wireless Netw.*, vol. 20, no. 1, pp. 41–51, Jan. 2014.
[21] M. Boushaba, A. Hafid, and M. Gendreau, "Source-based routing in wireless mesh networks," *IEEE Syst. J.*, vol. 10, no. 1, pp. 262–270, Mar. 2016.
[22] J. Zhu, H. Zhao, and J. Xu, "An energy balanced reliable routing metric in WSNs," *Wireless Sensor Netw.*, vol. 1, no. 1, pp. 22–26, Apr. 2009.
[23] R. Ye, A. Boukerche, H. Wang, X. Zhou, and B. Yan, "E^3TX: An energy-efficient expected transmission count routing decision strategy for wireless sensor networks," *Wireless Netw.*, vol. 24, no. 7, pp. 2483–2496, Mar. 2018.
[24] P. Sanmartin, A. Rojas, L. Fernandez, K. Avila, D. Jabba, and S. Valle, "Sigma routing metric for RPL protocol," *Sensors*, vol. 18, no. 4, p. 1277, Apr. 2018. Accessed: Aug. 6, 2019. doi: 10.3390/s18041277.
[25] H. Tall, G. Chalhoub, and M. Misson, "W-ETX: Worst-case expected transmission count routing protocol for convergecast WSNs," in *Proc. IFIP/IEEE Int. Conf. Perform. Eval. Modeling Wired Wireless Netw.*, Toulouse, France, Sep. 2018, pp. 1–6.
[26] X. Cai, Y. He, C. Zhao, L. Zhu, and C. Li, "LSGO: Link state aware geographic opportunistic routing protocol for VANETs," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, Dec. 2014, Art. no. 96. Accessed: Aug. 8, 2019. doi: 10.1186/1687-1499-2014-96.
[27] V. Sadatpour, F. Zargari, and M. Ghanbari, "A collision aware opportunistic routing protocol for VANETs in highways," *Wireless Pers. Commun.*, pp. 1–14, May 2019. Accessed: Aug. 6, 2019. doi: 10.1007/s11277-019-06557-x.
[28] S. Bindel, S. Chaumette, and B. Hilt, "F-ETX: A predictive link state estimator for mobile networks," *ICST Trans. Mobile Commun. Appl.*, vol. 2, no. 7, pp. 1–16, Jun. 2016.
[29] M. Amiri-Nezhad, M. Guerrero-Zapata, B. Bellalta, and L. Cerdà-Alabern, "Simulation of multi-radio multi-channel 802.11-based mesh networks in ns-3," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, Jul. 2014, Art. no. 118. Accessed: Aug. 6, 2019. doi: 10.1186/1687-1499-2014-118.
[30] M. Nobre, I. Silva, and L. Guedes, "Performance evaluation of WirelessHART networks using a new network simulator 3 module," *Comput. Elect. Eng.*, vol. 41, pp. 325–341, Jan. 2015.
[31] N. J. Jevtić and M. Z. Malnar, "Implementation of ETX metric within the AODV protocol in the NS-3 simulator," *Telfor J.*, vol. 10, no. 1, pp. 20–25, 2018. Accessed: Aug. 6, 2019. doi: 10.5937/telfor1801020J.
[32] N. J. Jevtić and M. Z. Malnar, "The NS-3 simulator implementation of ETX metric within AODV protocol," in *Proc. 25th Telecommun. Forum*, Belgrade, Serbia, Nov. 2017, pp. 1–4.
[33] M. K. Rana, B. Sardar, S. Mandal, and D. Saha, "Implementation and performance evaluation of a mobile IPv6 (MIPv6) simulation model for ns-3," *Simul. Model. Pract. Theory*, vol. 72, pp. 1–22, Mar. 2017.
[34] D. Sasidharan and L. Jacob, "A framework for the IPv6 based implementation of a reactive routing protocol in ns-3: Case study using LOADng," *Simul. Model. Pract. Theory*, vol. 82, pp. 32–54, Mar. 2018.
[35] A. Aldalbahi, M. Rahaim, A. Khreishah, M. Ayyash, and T. D. C. Little, "Visible light communication module: An open source extension to the ns3 network simulator with real system validation," *IEEE Access*, vol. 5, pp. 22144–22158, 2017.
[36] J. A. Arokkiam, P. Alvarez, X. Wu, K. N. Brown, C. J. Sreenan, M. Ruffini, N. Marchetti, L. Doyle, and D. Payne, "Design, implementation, and evaluation of an XG-PON module for the ns-3 network simulator," *Simulation*, vol. 93, no. 5, pp. 409–426, Jan. 2017.
[37] *Wireshark*. Accessed: Aug. 21, 2019. [Online]. Available: https://www.wireshark.org/
[38] G. Carneiro, P. Fortuna, and M. Ricardo, "FlowMonitor: A network monitoring framework for the network simulator 3 (NS-3)," in *Proc. 4th Int. ICST Conf. Perform. Eval. Methodol. Tools*, Pisa, Italy, Oct. 2009, Art. no. 1.
[39] D. Johnson, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks for IPv4*, document RFC 4728, IETF, Feb. 2007.
[40] *Source Code of Proposed Metrics and NPA Framework*. Accessed: Aug. 21, 2019. [Online]. Available: https://github.com/neje/IeeeAccess2019
[41] University of Bonn. *BONNMOTION: A Mobility Scenario Generation and Analysis Tool*. Accessed: Aug. 21, 2019. [Online]. Available: http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/

NENAD J. JEVTIC (M'12) was born in Jagodina, Serbia, in 1975. He received the Dipl.Eng., M.Sc., and Ph.D. degrees from the School of Electrical Engineering, University of Belgrade, in 2002, 2010, and July 2015, respectively. His Ph.D. thesis was focused on the automatic configuration of smart sensors and sensor networks.

Since 2003, he has been with the Faculty of Transport and Traffic Engineering, University of Belgrade, where he is currently an Assistant Professor with the Department of Telecommunication Traffic and Networks. During the last 16 years, he has been involved in several projects funded by the Serbian Ministry of Science and Technological Development. His research interests include the area of wireless communications, especially wireless sensor networks, smart sensors, and the Internet of Things.

MARIJA Z. MALNAR was born in Belgrade, Serbia, in 1983. She received the Dipl.Eng. and Ph.D. degrees from the School of Electrical Engineering, University of Belgrade, in June 2007 and May 2015, respectively. Her Ph.D. thesis was focused on routing in wireless mesh networks and propagation in indoor environments.

Since March 2008, she has been with the Faculty of Transport and Traffic Engineering, University of Belgrade, where she is currently an Assistant Professor with the Department of Telecommunication Traffic and Networks. During the last ten years, she has been involved in several projects funded by the Serbian Ministry of Science and Technological Development. Her research interests include the area of wireless communications, especially routing in wireless networks and indoor propagation modeling.

Dr. Malnar was a recipient of the Blazo Mircevski Award for the best paper of a young author in the IEEE Conference Telecommunications Forum TELFOR 2010.