Reproducing Measured MANET Radio Performances Using the EMANE Framework

Alexandre Nikodemski, Jean-Frédéric Wagen, François Buntschu, Christophe Gisler, and Gérôme Bovet

Simulation or emulation of mobile ad hoc networks (MANETs) is used to predict or analyze the performance of MANETs under various scenarios. One challenge is to realistically emulate the MANET's radio performance. Running the Extendable Mobile Ad Hoc Network Emulator (EMANE) framework, the authors show how to reproduce measured characteristics.

Abstract

Simulation or emulation of mobile ad hoc networks (MANET) is used to predict or analyze the performance of MANETs under various scenarios. One challenge is to emulate realistically the MANET's radio performance. Running the Extendable Mobile Ad Hoc Network Emulator (EMANE) framework, we show how to reproduce measured characteristics, namely throughput and round-trip time, of real tactical radios using wideband or narrowband TDMA-based waveforms. Additionally, a solution to simulate rate adaptation is proposed. An introduction to EMANE and the EMANE radio model plugins is also provided.

Introduction

In this work we aim to reproduce real mobile ad hoc networks' (MANETs') performance using an emulated environment: the Extendable Mobile Ad Hoc Network Emulator (EMANE) [1], promoted by the NATO IST-124 task group. We rely on the technical description and reference measurements performed under lab conditions to assess the behavior of the waveforms. This information allows us to define medium-fidelity models of the real tactical radios.

Using the EMANE framework [1], the nodes forming a MANET are represented by Linux containers (LXCs) similar to the computers embedded in real tactical radios. The radio transmission at the physical layer and the medium access control (MAC) protocols are simulated by the EMANE plugins.

The EMANE framework offers components focused *"on real-time modeling of links and physical layer connectivity so that network protocol and application software can be experimentally subjected to the same conditions that are expected to occur in real-world mobile, wireless network systems. The EMANE architecture provides for Network Emulation Modules (NEMs) that can be associated with computer system (real or virtualized) network stacks as interfaces. The EMANE framework further provides an event-driven control bus and logging facilities."* as quoted from [1].

The next section introduces EMANE and its usage for the emulation of MANETs. We describe the topology and the basic radio characteristics. We describe the implementation of the time-division multiple access (TDMA) MAC protocol. The performance results, measured and simulated, are compared in later sections. The last section concludes with the outcome of this work.

EMANE Implementation

The EMANE environment is created following the guidelines described in the EMANE tutorial [2].

A MANET node is created by running a NEM inside a Linux virtual container (LXC), as presented in Fig. 1. An LXC is created in the Ubuntu 14.04 operating system. Each node is therefore a totally isolated environment, running the EMANE emulator components along with other programs needed for the simulations.

A single machine is usually sufficient for running a certain number of nodes (e.g., 10). We have set up a cluster of virtual machines using VMware in order to be ready to scale up by an order of magnitude. When using a cluster of machines (virtual or not), a tight time synchronization must be enabled across all the NEMs to allow for the simulation of TDMA scheduling.

Each node or LXC container instance has two, possibly virtual, network interfaces, one for control and one for (radio) data exchange.

The control interface is used for the node control traffic (e.g., ssh on the node) and the simulation events (e.g., node location, path losses). Each node runs a secured shell (SSH) server to allow manual and script interactions.

The radio or data interface is used to exchange the radio packets, also called over-the-air (OTA) packets or simply data. Four NEM components — OTA manager, physical layer, radio model, and transport plugins (Fig. 1) — simulate in real time the major impairments due to the radio transmission. With EMANE (v1.2.1) and for the results presented here, the impairments are simulated based on packet timing, packet overhead, radio data rate, and packet losses based on signal-to-interference-plus-noise ratio (SINR) calculations.

A routing protocol was used to set up the routing tables for each node in the network. In our scenario, each node runs an instance of OLSRd2 [3]: an open source implementation of the MANET routing protocol OLSR v2 [4]. The OLSRd2 daemon communicates through the radio interface (*emane0* in EMANE). OLSRd1 was also implemented and used when instabilities due to OLSRdv2 were encountered (e.g., at very low data rates).

More details are provided in the following sections.

Digital Object Identifier: 10.1109/MCOM.2018.1800294

Alexandre Nikodemski, Jean-Frédéric Wagen, François Buntschu, and Christophe Gisler are with the University of Applied Sciences and Arts of Western Switzerland; Gérôme Bovet is with armasuisse Science+Technology.

Topology and Waveforms

The network topology used to measure the real performance in the lab is a static inline topology containing 2, 3, or 4 nodes (Fig. 2). The nodes are numbered from 1 to 4 and the hops from 1 to 3. We aim to reproduce this topology in EMANE and to obtain similar performance results.

Using EMANE, the topology is structured inside a so-called emulation event log (EEL) file that contains EEL-events. An EEL-event is a timestamp, a geographical position for a node, or a path loss in dB between a pair of nodes. Unspecified path loss values mean no connection. Similarly, many physical layer characteristics (e.g., Tx power, noise figure, antenna gain) can be set on the NEM physical layer plugin. Most of these features were left with their default values for our comparison. A few more details are provided later showing how to reproduce an adaptive multi-rate functionality. In the following section, concerning the simulation of a TDMA radio, the main parameters of interest are the data rate (called slot data rate) and the slot duration.

The radios under test have been measured using two different waveforms : wideband (WB) with a modem bit rate specified to be equal to 1.25 Mb/s and narrowband (NB) with 22 kb/s modem bit rate. The measured UDP throughput between 2 nodes provides the target slot data rate to be used in our EMANE setup: the measured data rates are (WB) 375 kb/s and (NB) 6.5 kb/s. The resulting slot data rate to be used in the EMANE simulation is found to be 880 kb/s for WB and 15 kb/s for NB, assuming the TDMA schedule described below. The resulting protocol efficiencies, defined here as the specified radio bit rate over the slot data, that is, 64 percent (WB) and 68 percent (NB), can be explained due to the expected complexity of the encapsulation and protection of the user bit for transmission over the WB and NB waveforms.

TDMA Scheduling in EMANE

EMANE offers the possibility to implement a TDMA radio model. This model will use:

1. The locations and radio characteristics of the nodes
2. The TDMA schedule to compute the SINR and resulting packet loss for each packet

A TDMA schedule describes the organization in time slots during which a node can transmit or receive (or be idle) at a given data rate, namely the slot data rate. The time organization of the EMANE TDMA model is described by three variables: slot, frame, and multiframe. A slot is the smallest unit of time (in microseconds in EMANE). A frame is defined by its number of slots per frame. A multiframe is defined by its number of frames per multiframe. The multiframe is repeated during a simulation unless a scheduling event changes the TDMA schedule. The following characteristics must be the same for the entire TDMA structure (i.e., for all the slots in the multiframe): number of slots per frame, slot overhead, slot duration, and bandwidth. The carrier frequency, Tx power, transmission class, and data rate can be set for each slot.

As for real radios, a slot overhead should be set to the maximum propagation delay expected between two nodes. In our simulation, the slot overhead was set to 30 μs, a tiny value compared to a slot duration over 30 ms. It is pointed out that the slot overhead should be increased to account for distances over 9 km and/or to account for large delay spread in mountainous regions. Due to the nature of the TDMA scheme, non-zero slot overheads and accurate time synchronization between nodes are mandatory. Because our simulations are running on the same physical machine, all the nodes (LXCs) are perfectly synchronized. Simulating TDMA radios using EMANE on a cluster of virtual machines requires synchronization: the native synchronization capability of the used cluster might be sufficient [5]; otherwise, PTP or NTP is recommended in EMANE documentation [1].

Figure 3 shows the TDMA structure used by the radios under test: a multiframe made of one frame for management followed by seven frames for user data. We named this structure M1D7. Using separate frames for control data and information data is usual in TDMA scheduling for MANETs (e.g., [6, 7]). Each frame has two or more slots dedicated to transmission according to a predefined schedule (e.g., round-robin between four nodes as shown in Fig. 3). The radios under test follow a 2 s long multiframe. The slot duration was therefore configured to reflect this implementation design. Without access to the exact scheduling used by the real radio under test, we based the scheduling on simple round-robin schedule accounting for empty slots as needed to reproduce the measured throughputs. All the transmis-

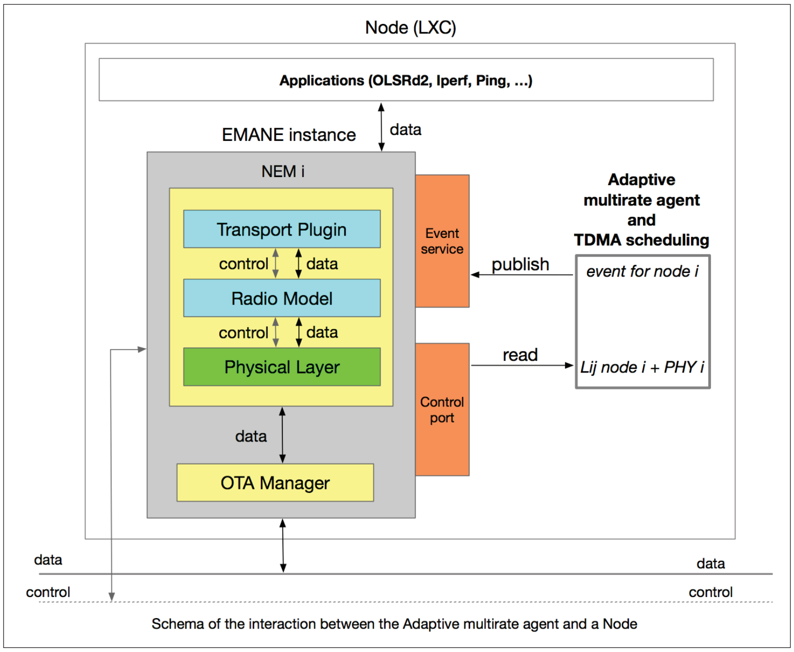

Figure 1. Block diagram of an EMANE node.

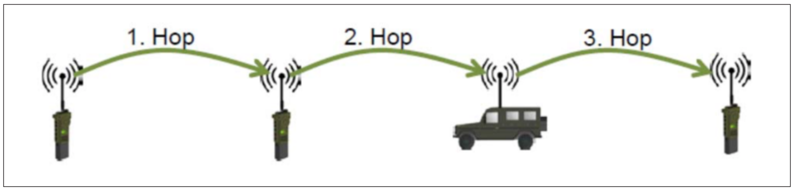

Figure 2. Topology.

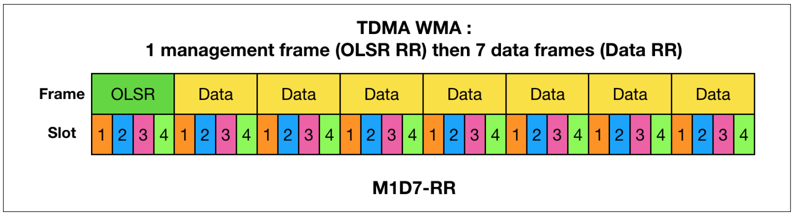

Figure 3. The basic TDMA structure M1D7: a multiframe made of one frame for management followed by seven frames for user data. Each frame has two or more slots dedicated to transmission according to a predefined schedule, such as round-robin, here for four nodes.

sion slots are assigned the same data rate.

The management (M) frames are reserved for the management data (i.e., the OLSR packets). The data (D) frames are dedicated to user data. It is pointed out that the real radios split the D frame into a data portion and a voice portion. This additional complexity does not need to be modeled since only the UDP throughput and round-trip time (RTT) for data packets are of interest here.

The M1D7 multiframe is implemented by creating eight frame elements and indexing them: the M frame takes index 0, and the D frames are indexed with the list 1:7, meaning that the data frame is repeated from index 1 to index 7.

Reserving the M frames for management and the D frames for the user data is achieved using the class and queue parameters.

The TDMA model classifies user traffic into four categories (0 to 3), which are mapped to four queues. Traffic is assigned to a queue based on the downstream packet differentiated services code point (DSCP [8]) used in the differentiated services architecture. All transmit slots have a class parameter assigned. This class represents one of the four traffic queues, and the TDMA model first dequeues the packets from the assigned class matching queue, followed by all other queues from highest to lowest priority order. The parameter *queue.strictdequeue* is a radio plugin parameter that defines if a transmit slot is allowed to dequeue traffic from a class different from the one it is assigned. When the parameter is set to true, only the queue matching the transmit slot class is used to dequeue traffic. Therefore, a transmit slot can be reserved to a certain kind of traffic based on the DSCP. The ability for slots to dequeue traffic from other classes than the one assigned is a radio model parameter. Like most of these parameters, this particular radio model parameter cannot be changed after the NEM has been started.

OLSRd1 and OLSRd2 packets have a high DSCP (56), meaning that they are classified by the TDMA model in the class 3 queue (highest priority). The user traffic DSCP can be configured, but is usually not set; therefore, it is classified in the class 0 queue.

The full details and the exhaustive list of the TDMA radio parameters can be found on the EMANE wiki [9].

Changing the parameters of the TDMA radio model offers sufficient flexibility to reproduce an adaptive TDMA schedule at a rate limited by the multiframe duration.

Two types of TDMA schedule events can be sent during a simulation: full and update. A full TDMA schedule defines the TDMA structure to use along with assigning, per NEM, the transmit, receive, and idle slots. A full schedule is required when, for example, the list of involved NEMs changes. TDMA schedule update concerns only the NEMS listed in a full schedule. An update TDMA schedule changes slot assignment information for a NEM but does not change the TDMA structure. In both full and update cases, the received schedule will take effect at the start of the next multiframe boundary. As described in the next section, changing the slot data rate via scheduling events was used to reproduce the rate adaptation featured by the radios under test.

Emulation Results

This section compares the results obtained from measurements conducted on real tactical radios with the ones conducted on our EMANE platform. As mentioned earlier, the radios can operate two waveforms: wideband (WB: 1,25 Mb/s modem rate) and narrowband (NB: 22 kb/s modem rate). The measurements conducted by armasuisse were undertaken with the 1-, 2-, and 3-hop in-line topologies described in Fig. 2. The measurements were conducted in a laboratory setup with RF coaxial cables used to interconnect the radio. Attenuators were used to ensure sufficient received power without saturating the receivers.

The throughput of correctly received UDP packets and the RTT (using ICMP packets) were measured with a proprietary software. The conventional networking tools Iperf and ping were used on the emulated platform. The same measurement procedures as in the lab conditions were used for the emulation. For the WB simulations, the Iperf default packet size (1470 B) was used, and the client (sender) UDP throughput was adjusted so as to not overload the links. For the NB simulations, the Iperf packet size had to be reduced to about 300 B and adjusted to obtain the desired throughput. The ping packet size was 1000 B for both WB and NB cases.

The measured results were provided to us by armasuisse and are compared with the results using EMANE in Fig. 4 (WB) and Fig. 5 (NB). The figures show that the reproduced performance matches very well with the measured results.

Figure 4 for the wideband cases and Fig. 5 for the narrowband cases present the measurements on real radios (top panel) and the simulation results (lower panel). The comparisons are presented from left to right: the throughput (kilobits per second), the BDP/RTT (kilobits per second), and the RTT (milliseconds or seconds). The BBP is the bandwidth delay product. The BDP/RTT is the expected throughput based on the RTT. The BDP/RTT results allow us to verify the expected inverse proportionality relationship between throughput and RTT [10]. Percentages are provided to easily compare the 1-hop (100 percent) result (leftmost bar) to the 2- and 3-hop results.

The wideband results, Fig. 4, show very good agreement for both the throughput and RTT results aside from the 3-hop RTT measured in the 3-hop case. A 2 s delay was measured using the real radios in the 3-hop case. The measured BDP/RTT results in the 3-hop case are inconsistent with the 1- and 2-hop results: a 5 percent ratio is obtained for the BDP/RTT instead of the 24 percent ratio expected from the throughput results. This very long delay of 2 s was not reproduced in the EMANE simulations. The 3-hop RTT results using the real radio are expected to be improved in future releases. Interestingly, delays close to 2 s were obtained using large ping packet size (> 3500 B).

The wideband results were obtained using a slot data rate of 880 kb/s for the EMANE radios. The M1D7 TDMA schedule explained above (Fig. 3) was used. The multiframe duration was 2 s, close to the real specifications of the measured radio. The resulting frame duration was 250 ms. A frame was divided in five subframes to shorten and match the RTT. Each subframe contains the number of slots required by the TDMA schedule. The 1-, 2-, and 3-hop results follow the measured behavior by using a "one two" ("12"), "123xx," and "1234xxxx" TDMA schedule, respectively. The x means a slot where all nodes are idle. It is claimed that the real radios use these types of schedule as a result of using smart adaptive scheduling algorithms.

The narrowband simulation results (Fig. 5) were obtained using a slot data rate of 15 kb/s for the EMANE radios. As for WB, the M1D7 TDMA schedule explained in Fig. 3 and multiframe durations of 2 s and 250 ms frames were used. The slot duration was simply set as the frame duration divided by the number of nodes. The schedules used for the NB were classical round-robin schedules: "12", "123,", and "1234" for 1-, 2-, and 3-hop, respectively. The throughput results are closely reproduced (Fig. 5). The UDP packet size had to be reduced to less than 300 B and adjusted for two reasons:

1. Due to a limitation of Iperf to simulate low data rate
2. Since the results are sensitive to the packet fragmentation over the slot structure

The RTT results are reasonably well reproduced considering the very long delays: 2, 17, and 28 s in the EMANE-based simulation compared to the measured 3, 14, and 30 s.

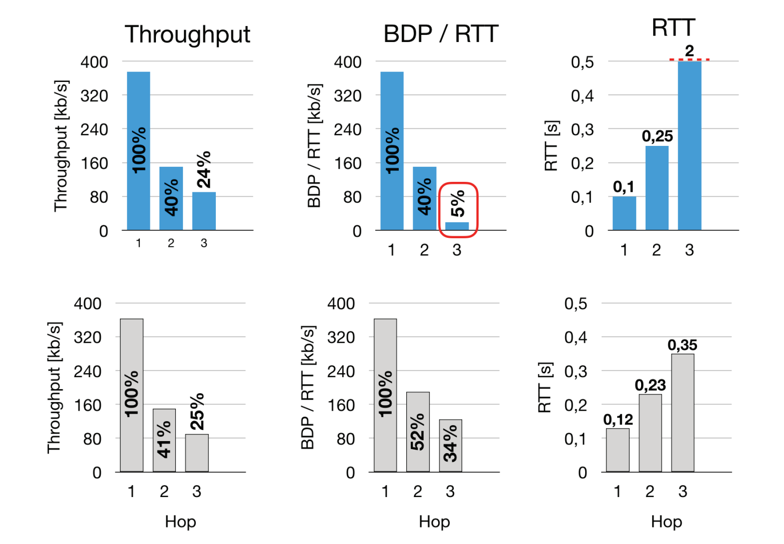

Figure 4. Wideband results: the measurements (top panel) are closely reproduced by the simulations (lower panel). The 2 s measured RTT is not consistent with the other results and could be reproduced only by using large ping packets.

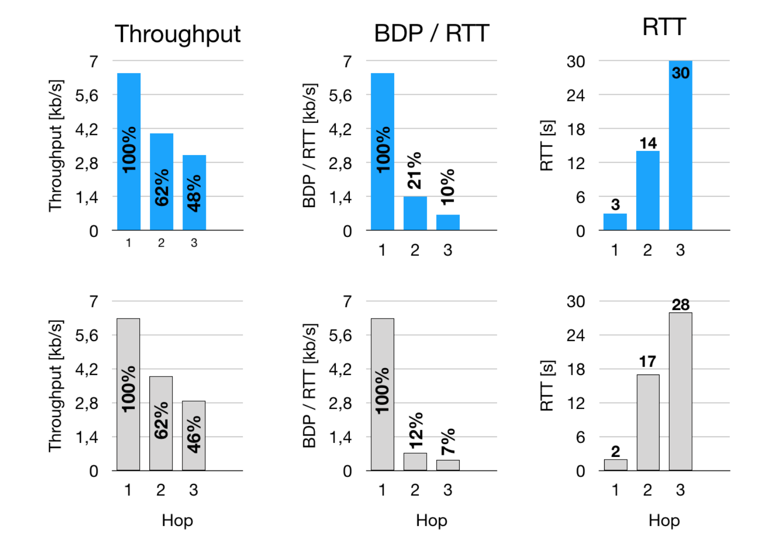

Figure 5. Narrowband results: the throughput measurements (top panel) are closely reproduced by the simulations (lower panel). The measured RTTs are very large (2 to 30 s) and reasonably well reproduced.

Rate Adaptation

Like most modern radios, the measured radios adapt their data rate to the transmission conditions. Typically, a radio has a multi-rate adaptation function that automatically switches the modulation and coding scheme (MCS) between two nodes based on an estimation of the SINR on the link.

The result is the change of the measured throughput, shown in Fig. 6 (top panel) as a function of the received power level in dBm. The adaptive multirate behavior was reproduced on the EMANE platform. The lower panel in Fig. 6 shows the data rate measured using Wireshark (IO Graph) as a result of changing with time the path loss value fed to the NEM. This demonstrates the possibility to reproduce the measured rate adaptation with the EMANE framework.

The EMANE radio models do not implement any multirate adaptation feature: the data rate is set based on the parameters found in the NEM configuration (i.e., at startup) or in a TDMA schedule and is never changed unless an outside interaction with the NEM happens. A rate adaptation feature was implemented using a Python agent external to the NEM (Fig. 1). Each node runs an instance of the rate adaptation agent. The agent fetches the *PHY config* information using the Python control port client module. The PHY config of a NEM contains, among other information, the TX power, the antenna gain, and the bandwidth. The path losses to the other nodes are fetched by using the *emanesh* Python module [1]. The path loss information is received by the NEMs through events specified in the EEL file. The sending of the path loss events is centralized and can be user-controlled. It is therefore possible to know about the path loss changes even before the multi-rate adaptive agent discovers them in the NEM table. However, our solution is only based on local information and requires no communication outside the node. The rate adaptation agent regularly fetches the NEM information and configuration, and then calcules an SINR between the

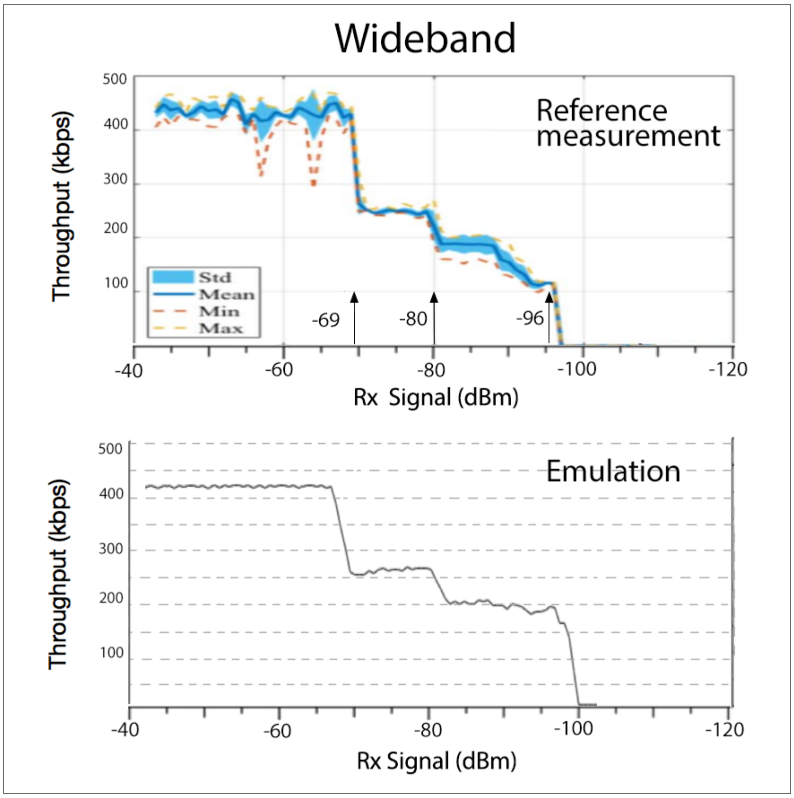

Figure 6. The rate adaptation vs. received power (top). Automatic rate adaptation results (bottom) simulated by changing the path loss with time as if the two nodes are moving away from each other.

nodes to detect if rate adaptation is needed. The slots' data rates of a NEM are changed by sending TDMA schedule update events. The change will take effect at the start of the next multiframe.

Conclusion and Outcome

The challenge considered here is to emulate realistically a MANET's radio performance. The open source EMANE framework was used to reproduce the behavior of a real tactical radio benchmarked in a laboratory. The EMANE framework, the EMANE radio model plugins, and their usage are briefly described. All contributions to EMANE and OSLRd1 required to obtained the presented results are available in a gitlab repository [5].

Using a simple 1-, 2-, and 3-hop in-line topology, UDP throughput and RTT results are measured. The wideband and narrowband measured performance is reproduced very closely using the same 2 s multiframe (M1D7) TDMA structure as the radios under test. Simple round-robin schedules enabled us to reproduce the measured throughput. Adapting the slot data rate proved to be necessary to reproduce the measured delays. The rate adaptation featured by the real radios could also be reproduced by changing the slot data rate using a Python agent external to EMANE.

It is observed that the measured radios could have produced better performance by changing their TDMA schedules. Future work will focus on evaluating improvements proposed to enhance the performance of MANETs.

Acknowledgment

Work at the HEIA-FR is funded under projects ARAMIS-No. 041-14 and 041-23.

References

[1] EMANE website; https://adjacentlink.com/emulation.html
[2] S. Galgano, "Github Emane Tutorial Wiki"; https://github.com/adjacentlink/emane-tutorial/wiki.
[3] olsr.org, "Olsrd2 wiki"; http://www.olsr.org/mediawiki/index.php/OLSR.org Network Framework.
[4] P. J. T. Clausen, C. Dearlove, and U. Herberg, "The Optimized Link State Routing Protocol Version 2," IETF RFC 7181, 2014.
[5] Predictake Repository; https://gitlab.forge.hefr.ch/predictake.
[6] J. B. Shin and B. H. Roh, "TDMA Frame Format Analysis for Applying Policy Based Management to TDL System," *Proc. 5th Int'l. Conf. Ubiquitous Information Technologies and Applications*, 2010.
[7] P. Djukic and S. Valaee, "Distributed Link Scheduling for TDMA Mesh Networks," *IEEE ICC*, 2007, pp. 3823–28.
[8] K. Nichols *et al.*, "Definition of the Differentiated Services Field (DS Field) in the ipv4 and ipv6 Headers," RFC Editor, RFC 2474, Dec. 1998.
[9] S. Galgano, "EMANE TDMA Model"; https://github.com/adjacentlink/emane/wiki/TDMA-Model.
[10] M. Mathis *et al.*, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *SIGCOMM Comp. Commun. Rev.*, vol. 27, no. 3, July 1997, pp. 67–82.

Biographies

Alexandre Nikodemski received his Master's degree in computer science from the University of Fribourg/Bern/Neufchâtel, Switzerland, in 2017. He works at the HES-SO University of Applied Sciences and Arts — Western Switzerland as a research collaborator. His main research focuses on AI, security, networking, and machine learning.

Jean-Frédéric Wagen [M'18] is a professor at the HES-SO University of Applied Sciences and Arts — Western Switzerland in Fribourg since 2000. He received his Ph.D.EE from the University of Illinois at Urbana-Champaign in 1988. He worked at GTE Labs (now Verizon) and at Swisscom Innovations. He has published over 10 journal papers and over 40 conference papers on radio propagation and wireless communications. He holds one patent.

François Buntschu has been a professor at the HES-SO University of Applied Sciences and Arts — Western Switzerland in Fribourg since 2001. He received his BS.C. in computer sciences in 1992 and his Cisco Certified Internetwork Expert in 1998. He worked at Global One (international telecom operator founded by Sprint, France Telecom, and Deutsche Telecom) as a network expert for the Internet deployment in Switzerland. His main research focuses on networking and virtualization.

Christophe Gisler received his Ph.D. degree in computer science from the University of Fribourg, Switzerland, in 2017. He worked at the University of Fribourg and at Swisscom Innovations. Since 2010, he has been working as a research collaborator in the Institute of Complex Systems at the School of Engineering and Architecture of Fribourg, a member of the HES-SO University of Applied Sciences and Arts — Western Switzerland. His main research focuses on machine learning applications.

Gérôme Bovet (gerome.bovet@armasuisse.ch) is a scientific project manager for armasuisse science and technology within the Swiss DoD since 2015. He received his Ph.D. in networks and systems from Telecom ParisTech, France, in 2015. He focuses on tactical communications from a networking point of view, striving to enhance existing protocols with more intelligence by applying machine-learning techniques. He is also involved in projects related to cybersecurity in 4G and 5G.