A context-aware tool-set for routing-targeted mutual configuration and optimization of LLNs through bridging virtual and physical worlds

Y. Chen, J.P. Chanet, K.M. Hou, P. Zhou

A context-aware tool-set for routing-targeted mutual configuration and optimization of LLNs through bridging virtual and physical worlds

Yi-bo CHEN [#*1], Jean-Pierre CHANET [#2], Kun-Mean HOU [*3], Peng ZHOU [*4]

[#] *UR TSCF, IRSTEA*
Aubière, France
[1] yibo.chen@irstea.fr
[2] jean-pierre.chanet@irstea.fr
[*] *LIMOS Laboratory UMR 6158 CNRS, Blaise Pascal University Clermont-Ferrand II*
Aubière, France
[3] kun-mean.hou@isima.fr
[4] peng.zhou@isima.fr

Abstract—**In the history of the WSN, unexpected routing behaviors is always a main issue of the large scale WSN deployments. Considering the high cost of building a real deployment, network simulators are often used in this domain. However, the original problem is still not solved although the era of IoT has been coming. A new concept of LLN is emerging. We realize that, no matter how wonderful the results from the simulation scenarios and thorough testing, the problems, such as bad performance or even severe system failures due to suboptimal routing path, would still happen in the real-world system. Our context-aware tool-set can help to build the simulation topology which is more close to the real network through mapping a serial routing metrics defined by IEFT ROLL working group and the link situation of the actual network. We believe our suggestion of bridging virtual and physical worlds reflected on our proposed tool-set could conduct more precise routing-targeted simulations. Moreover, by close-loop method, the knowledge and analyzed simulation results can lead us to improve the routing topology of the deployed LLN.**

I. INTRODUCTION

For a long period of time, the researchers and developers have to suffer the routing issues of Wireless Sensor Networks (WSNs). Although this domain has been continued to research for more than a decade, deploying sensor nodes is surely not an easy take as expected because the wireless communication may cause lots of unpredicted problems. Especially for the distribution of nodes in the wild, the troubles of routing will become more due to environmental condition changes and often mentioned remote firmware updates [1]. As we know, it is not enough to keep using a star topology or single-hop network for most of larger WSNs deployments since the radio range is limited and the area to cover is large. Thus, a well optimized and configured routing path searching strategy will bring significant effects on such multi-hop wireless-mesh based network.

As the coming of Internet of things (IoT) technologies, we have numerous reasons to believe that the quantity of sensor nodes will grow rapidly to huge number in the near future [2]. Unfortunately, the important problem, which is how to provide proper routing functions, is still not settled down. Thanks to the emerging concept of low power and lossy network (LLN) [3] and its routing framework of the IPv6 Routing Protocol for LLN (RPL) proposed by the IETF ROLL working group, the standardization effort has been made for the nodes which are typically operated with constrains on processing power, memory, power consumption, and lifetime. Furthermore, communication links of LLN are characterized by high loss rate, low data rate, low transmission power, and short communication range. These features are similar to traditional WSN so the LLN can be seen as an extension. Based on the reasons chartered by ROLL working group [4], the conventional routing protocols in mobile ad-hoc networks, such as AODV, OSPF, OLSR and DSR, are not suitable for routing in LLNs because of high routing overhead [3].

RPL has been designed to operate in low power and lossy networks with thousands of nodes. Its applicability in the Advanced Metering Infrastructures (AMIs), home automation, building control and industrial networks has been proposed and recognized [3]. RFC 6550 has been published in March 2012, and most of the structure and parameters of RPL were clearly defined. Destination Oriented Directed Acyclic Graph (DODAG) is basic logic structure in RPL. DAG topology is used to establish bidirectional routes for LLNs. RPL routes are optimized for various most common traffics, such as multipoint-to-point (MP2P), point-to-multipoint (P2MP) and point-to-point (P2P), based on the measurements of routing metrics and constraints, the rank calculation and preferred parent selection strategy defined by Objective Function (OF).

Currently, the researchers and developers find that it is a big challenge to adapt RPL framework for their deployments because of its complexity and flexibility. For example, in most of cases, energy consumption is a critical factor related to the expected lifetime of an LLN system, and RPL should be able to provide a set of corresponding routing mechanisms in order to minimize energy use and prolong system lifetime by defining a number of energy-aware routing metrics and assigning suitable parameters of built-in energy-saving mechanisms like trickle algorithm [5]. However, a deployment may have various specific requirements on different aspects such as Quality of Service (QoS), Therefore, simulating and testing our LLN system are necessary before its deployment in the real world.

Apparently, a good way of testing is to set up a test-bed network in an indoor environment. However, toggling LED cannot reflect enough information on what is going on in our

RPL model. So if each node in the network can run a probe thread and export the data can map to the changes of nodes routing behaviors, the testing can be more reliable. Unfortunately, it is not easy to execute any debugging on the test-bed hardware. A WSN-simulator can be helpful not only on the network debugging, but also on setting up arbitrary network topologies in different scripted scenarios or test cases to optimize the WSN lifetime.

In this article, we propose a context-aware tool-set for configuring and optimizing the routing algorithms running on various LLN deployments. The approach we would adopt is to integrate the above previous discussed methods, namely, bridging virtual and physical worlds. The rest of this paper is organized as follows. Section II provides a brief problem statement to explain the reason why we need this routing-targeted tool for LLN. In Section III we present the architecture of the two main components in this tool, which are named as Map4LLNSim and LLNRun4NS. The Section IV shows some related work and the last section shows our current development state and future work.

II. Problem statement

Test-bed and simulation are the most common used methods to test protocol models even a whole system before real deployment. As we have discussed in the previous section, unpredicted routing issue is a significant factor for the WSN deployment. Numerous routing protocol models have been built in various WSN-simulators. However, in the real WSN system, the routing algorithm seldom shows the same behavior as in the tests. One main reason is that neither test-bed nor simulation can be tuned exactly environment of the real deployment, because the radio frequency propagation model is environment dependent. Therefore we suggest adopting a context-aware tool which allows a routing-targeted LLN development. As there is no alternative candidate of routing protocol, adjusting parameters and OF code of RPL for specific deployment requirements are our final target.

Our context-aware tool-set can provide more correct and accurate configurations for a WSN-simulator, so it can rebuild a more reasonable good resemblance of a hardware test-bed even a real deployment in the simulated environment. Most of the current WSN simulators are able to support these complex network configurations. Using such well configured simulated scenario, it can help us to understand why algorithms in RPL framework behave in the unexpected way, and allows detecting the RPL routers with exceptional issues, such as resource insufficiency, congestion, cycle paths, etc. We hope that our solution is possible to provide the automated optimization and configuration on a whole LLN network. But, it is not a good idea to start from the consideration of all the components and network layers in the LLNs, like Operating System (OS), services application, MAC layer and other working consequence algorithms. As an initial work of exhaustive iLive-CLAS system, this context-aware tool-set is only functional for providing the means to configure a WNS-simulator and its RPL protocol model. However, the essential concept of iLive-CLAS system, based on a continuous close-loop method which enables to map bi-directionally the simulation results and the real world ones, will be tested in this tool-set.

As we have discussed, a simulator solo cannot make accurate assumptions about the network topology [6]. Thus, the detailed context knowledge about the deployed LLN is the core information in this approach. But for simplifying the current implementation, it is more practical to limit the selected mapping data to a basic level because the redundant data will surely make our work laborious. It should contain the description of the adjustments (i.e. RSSI or LQI) for a radio medium model of WSN-simulator to get reasonable predictions. Furthermore, information about RPL routing metrics is necessary for this simulator-supported routing targeted LLN development. After the simulator has been configured accordingly, the extracted information is expected to improve the RPL routing behaviors of the deployed LLNs. This procedure opens three main problems for this research work: First, how to automatically and efficiently map the real network to the simulator? Second, how the simulation results can be used efficiently to the target LLN? The third problem is: how to measure the quality and performance of the new updated LLN topologies, namely the continuing evolutions of RPL DODAG, after our context-aware tool-set provides the feedbacks of simulation to the real world LLN system.

III. Map4LLNSim/ LLNRun4NS tool-set architecture

In this article, we develop a tool-set enables to solve the questions stated in the previous section. This context-aware tool-set includes:

1) Map4LLNSim: A tool for real-time mapping Real/Physical world (i.e. RPL nodes metrics information, parameters and network connectivity situations) into simulation scenarios.

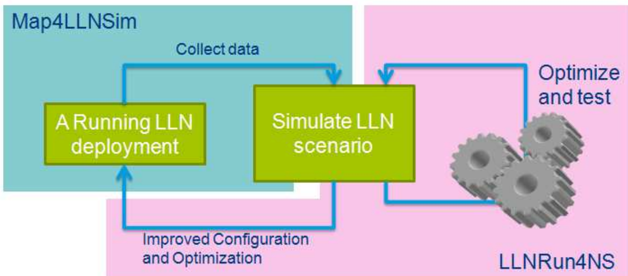

Fig. 1. MAP4LLNSim/ LLNRun4NS TOOL-SET ARCHITECTURE

2) LLNRun4NS: A tool for using the information from Map4LLNSim to configure our RPL model in simulator and help to analyze, visualize and at last optimize the targeted LLN deployment by the conclusions drawn from the simulation results for improving RPL OF algorithm and trickle timer configurations used by this physical LLN system (Fig. 1).

MAP4LLNSim is divided into an application running on the RPL routers that transmit relevant information through upward routing path of RPL, a corresponding application running on the RPL sink node that collect these information, and a program with support of SLIP [7] on a PC that can connect the RPL sink node, collect the data periodically and translate it into a script that can configure the targeted simulator. LLNRun4NS is a set of programs that is able to use the well configured simulator by MAP4LLNSim to draw improvement instruction and recommended parameter settings. These tools can be adopted by different tests, such as non-functional properties, parameters and visualized prediction of RPL routing behaviors.

As already illustrated, it is difficult to make accurate predictions about the connectivity of a LLN. But with the actual measurements from MAP4LLNSim, this main factor that influences a node's behavior can be analyzed and processed by a specific propagation model in the simulation. Our first phase work concentrates on the LLN with the mobility feature at minimal level and thus the low rate change can allow to make reasonable predictions or to track the changes with minimal load or overhead to RPL routers. If captured data can be recorded over a sufficient amount of time, most of the temporary disturbances in connectivity of target LLN are considered representative in the simulator. MAP4LLNSim is also capable to log the unexpected events in OS level or hardware level, such as system reboots, sudden failures, and errors derived from our experimental LLN deployment.

When our proposal commences to be implemented, the function of mapping the link quality between the different nodes is in our first phase plan. The link quality is often used for routing protocol to organize the network topology dynamically. Furthermore, in RPL routing framework, the IETF ROLL working group defined its usage as RPL routing metrics (M) or constraints (C) [8] which are shown in Table I.

TABLE I
RPL ROUTING METRIC AND CONSTRAINT OBJECTS

Routing Metric /Constraint objects	Description
Node state and attribute	CPU, Memory, congestion situation
Node Energy	Power node, estimated remaining lifetime and, self-built power metrics
Hop Count	Number of hops
Link Throughput	(M&C) Maximum or minimum value
Link Latency	(M&C) sum of all latencies, pruning links higher than certain threshold
Link Reliability	Packet reception ratio, BER, mean time between failures... LQL; ETX
Link color	10-bit encoded color to links, avoid or attract specific links -> traffic types

It will be pity to spurn these well defined metrics. Especially, the Link Quality Level (LQL) can be deduced by Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI). The packet reception ratio can be got from packet loss. Thus, this set of information will be sufficient for both of our propagation model and RPL model in simulation. Moreover, MAP4LLNSim adopts a hybrid solution including passive probing and active probing data packets sending. The target is, to provide a default periodic data collection, and an automatic probing mechanism to cope with temporary changes in the RPL DODAG. In the latter case, the OF of RPL or a data link layer model can active a suspended sub-thread of MAP4LLNSim in the deployed LLN system. But, the changes of routing metrics used by OF are more preferable because the information from the received DODAG information object (DIO) messages can also estimate and reflect disturbances very efficiently [3]. Except the above mentions, MAP4LLNSim will also support an evaluation method to eliminate redundant acquiring properties for reducing transmission overhead from RPL router to the sink node. This mechanism is also efficient for data management when MAP4LLNSim continues to conduct the evolution of simulated DODAGs in an infinite close loop method.

LLNRun4NS is based on the common graphical interfaces of the simulator. The gathered information about DODAG topology from MAP4LLNSim can be easily visualized. The Collect-View tool of COOJA [9] represents the connection quality in the DODAG by using a spring graph layout and locating the nodes manually on a map. Namely, the distance between two nodes shows the path weight/cost calculated from collected data. Once the simulation scenario is well configured and it is visualized similarly to the real DODAG, the simulation is called to start running. LLNRun4NS will acquire the data from simulator and analyze them. However, too much simulation tracing log cannot be beneficial for executing simulation. LLNRun4NS can intercept the data acquisition to select and preprocess the tracing log for further analysis. As our tool-set targeting RPL routing, the preprocessing and the results of analyzing are strongly concentrated on the requirements of configurations and optimizations for the behaviors of RPL. Furthermore, LLNRun4NS can control the RPL simulation model to have a thorough test using different trickle timer configurations, routing metrics compositions [10], and even debugging or plotting the data from simulator can be helpful to detect the unexpected RPL routing behaviors.

How to find the best settings of RPL routing protocol with optimal path is always a problem that should be considered when planning a LLN deployment. Fortunately, in our tool-set, only the routing layer will be tuned based on RPL framework, in particular the RPL routers possess enough context-aware features. For the other network layers like MAC layer [11] or application layer, in-depth knowledge is required to provide reasonable tunings because the nodes might need different configured strategy depending on their location or function within the LLN. The current LLNRun4NS therefore provides a parameter list defined in RPL framework, and supports finding an optimized configuration for each RPL router in the deployed DODAG. At the same time, LLNRun4NS also helps the maintenance of the LLN especially when MAP4LLNSim detects the changes in the RPL DODAG and maps them to the simulator.

After the analyzing a simulation run of the current DODAG and comparing it with an older run, LLNRun4NS can detect the explicit problems and what is going on in the RPL modular of the running RPL routers and sink node. It is not easy to affirm that one revision test against the reference-run can help to detect all the causes of regression or improvement. Thus we suggest that LLNRun4NS might run a several recursive tests to give the exact reasons of changes and misbehaviors of RPL DODAG. Moreover, one cycle of executing this tool-set only means one time evolution for both of the virtual and physical world. To achieve the better results, the close loop needs to be performed continuously. Of course, we also recommend that the drawn results from this tool-set are only applied to the distinct LLN deployment tests because the same parameters may cause unpredictable impact on the behavior of another deployment.

IV. RELATED WORK

Based on our investigations, the classic network simulators like NS-2, NS-3 or OMNet++ have been extended to the WSN domain [12]. They all can meet the needs of our tool-set, but the code must be ported and recompiled to run in both of the simulator and hardware. Thus, it is difficult to keep the coherence of the programs for the two sides especially when the limited resources of the real sensor nodes are considered. Due to the choice we have made for the first phase, the programs implemented for RPL's logic can be same but the delays or bottlenecks caused by the hardware peripherals still cannot be neglected. The COOJA simulator [9] targets exclusively the WSN and IoT relevant simulations. It can connect various plug-in including the emulators like MSPsim or Avrora for working together with different network models.

Our context-aware tool-set is inspired by the RealSim and Dryrun architecture [6] designed for COOJA simulator. These tools can be used by universal WSN deployment and provide a suite kit of deployment-targeted development mainly on the respective, such as system debugging, deployment optimization. Their work and the authors of [13] also present the concept of mapping a real deployment to a simulator. The latter focuses on moving the information about the state of real sensor nodes to and from the simulator, and former's interests is the common network topology. But none of them tries their optimizing strategy for a tuning of routing layer, not even mention to the promising IoT compatible RPL routing framework.

V. CURRENT STATE AND FUTURE WORK

In the current context-aware tool-set, MAP4LLNSim is the core component to provide the context sensing and mapping function. It supports automatic detection of RPL routing metrics and link situation by active probing. This scripted information is automatically used for tuning a simulator. In our implementation, the data about link quality can be used to represent the links by COOJA's Directed Graph Radio Medium (DGRM) and the propagation model designed for our WPAN model in NS-3 [14]. While LLNRun4NS is a JAVA program based on the COOJA's Collect-view tool, we added necessary additional interfaces to control the simulations of COOJA and NS-3. A plug-in of COOJA named Cooja-dbus is used to control its simulation no matter it is running or not. But for NS-3, LLNRun4NS is just able to modify or reorganize its main simulation scripts. Currently, we still use the file input and output operations to access and analyze as much tracing log as possible, but we are planning to adopt an efficient approach to do this task and save all the helpful data to a database. For a better description of our current implementation with our RPL model in NS-3 [15], a simplified data flow chart is shown in Fig. 2.

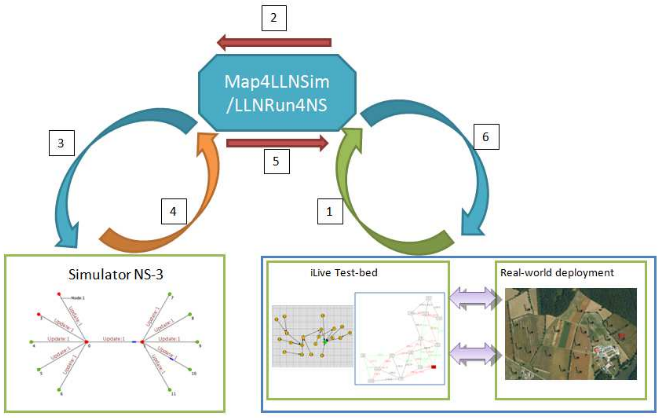

Fig. 2. SCHEMATIC FLOW DIAGRAM OF MAP4LLNSim/ LLNRun4NS TOOL-SET

1) Real-world link quality properties and RPL routing metrics probing are exported for a new cycle: A hybrid solution for link quality monitoring: Passive probing (by default or manually) and Active probing (automatically); Sensor reading is also part of acquiring properties; the data will be exported to the MAP4LLNSim through a RPL sink node in a certain format.

2) Saving, acquiring and preprocessing the data message in a script for analysis: The data will be saved and acquired in structures based on different requirements; LLNRun4NS will provide an analysis module GUI for graphing the network topology and data display.

3) Generating, importing acquired data to NS-3 simulator: LLNRun4NS will call and execute NS-3 simulation in the background after NS-3 has been configured to resemble the real RPL network; NS-3 module will import the real-world data into its simulation and output the tracing file of all the RPL nodes.

4) Validation of NS-3 tracing files and transferring them to LLNRun4NS: NS-3's Pcap tracing file can be tested and analyzed in the Wireshark tool; NS-3's RPL tracing file is organized to store the important information of the entire simulated RPL message.

5) Processing NS-3 tracing files in the LLNRun4NS: LLNRun4NS will provide a GUI to show the simulation results; Forecasting the network condition, RPL DODAG changes and Optimizing best configured parameter (OF, Metrics, Position and function) for each single RPL router.

6) Transmitting the feed-back optimizing configuration to the real-world LLN and waiting for the next cycle: First phase of this project is to test the tool-set on the iLive test-bed using ContikiRPL: 1. Cooja-based; 2. Development suite-based; Next step is for the real LLN deployment on the experimental field.

As we have discussed in previous section, MAP4LLNSim/ LLNRun4NS tool-set is targeting a system specifically for a LLNs system equipped by I/WoT protocol stack. Its context-aware features, including RPL routing metrics, location and link quality, will help to detect the behaviors of RPL routing protocol, and assist to avoid some troubles, and find problem caused by suboptimal routing path (delay or unexpected battery energy consumption) in advance. Hopefully, it can also provide the estimation and forecast functions for LLNs in the near future.

VI. Conclusion

In the research domain of RPL protocol, we can find various mechanisms to improve its performance from different perspectives. Most of their work is based on the tailored RPL model in simulation. Few of them adopt experimental test-bed for performance tests of their proposals. However, our idea presented in this article might help to get a better understanding of the RPL's misbehaviors, and to avoid the unexpected failures of LLN deployments. Essentially, through comparing and evolving the real DODAG topology and the simulated one, we achieve their mutual configurations, optimizations and continuing evolutions by bridging virtual and physical world. Furthermore, we also believe that this context-aware optimization strategy can be carried out in the use cases of RPL like building automation, home automation, industry, smart city and agricultural application scenarios.

Acknowledgment

The authors would like to thank the "Conseil Régional d'Auvergne" and the European Regional Development Funds (ERDF) which have financial supports for this research.

References

[1] X. Liu, et al., "Efficient and portable reprogramming method for high resource-constraint wireless sensor nodes," in Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on, 2011, pp. 455-460.

[2] J. Vasseur, et al., "RPL: The IP routing protocol designed for low power and lossy networks," Internet Protocol for Smart Objects (IPSO) Alliance, 2011.

[3] T. Winter, et al., "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks, RFC 6550," IETF ROLL WG, Tech. Rep, 2012.

[4] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," 2012.

[5] P. Levis, et al., "The trickle algorithm," IETF ROLL working group,[RFC 6206], 2011.

[6] M. Strübe, "RealSim and DryRun: Tools for a deployment-targeted development of WSNs," 2011.

[7] J. Romkey, "Nonstandard for transmission of IP datagrams over serial lines: SLIP," 1988.

[8] J. Vasseur, et al., "Routing metrics used for path calculation in low power and lossy networks," draft-ietf-roll-routing-metrics-19 (work in progress), 2011.

[9] F. Österlind, et al., "Demo abstract: Cross-level simulation in cooja," in Proceedings of the First IEEE International Workshop on Practical Issues in Building Sensor Network Applications, 2006.

[10] P. Trakadas and T. Zahariadis, "Design Guidelines for Routing Metrics Composition in LLN," 2012.

[11] P. Di Marco, et al., "MAC-aware routing metrics for low power and lossy networks," in INFOCOM, 2013 Proceedings IEEE, 2013, pp. 13-14.

[12] (2011, NS-3 Web page. Available: http://www.nsnam.org/

[13] F. Österlind, et al., "Sensornet checkpointing: Enabling repeatability in testbeds and realism in simulations," in Wireless Sensor Networks, ed: Springer, 2009, pp. 343-357.

[14] K. El Ghomali, et al., "A new WPAN Model for NS-3 simulator," in NICST'2103 New Information Communication Science and Technology for Sustainable Development: France-China International Workshop, 2013.

[15] Y.-b. Chen, et al., "A RPL based Adaptive and Scalable Data-collection Protocol module for NS-3 simulation platform," in NICST 2103 New Information Communication Science and Technology for Sustainable Development: France-China International Workshop, 2013.