www.ijsrnsc.org

Available online at www.ijsrnsc.org

IJSRNSC

Volume-7, Issue-4, August 2019
Research Paper

Int. J. Sc. Res. in
Network Security
and Communication

E-ISSN:2321-3256

Assessment on different tools used for Simulation of routing for Low power and lossy Networks(RPL)

Manish Mishra[1*], Piyush Shukla[2], Rajeev Pandey[3]

[1] Department of Computer Science & Engineering, University Institute of Technology, RGPV Bhopal, India
[2,3]Department of Computer Science & Engineering, University Institute of Technology, RGPV, Bhopal, India

[*]*Corresponding Author: mpmishra96@gmail.com, Tel.: +91-62631-26692*

Received: 14/Jul/2019, Accepted: 21/Aug/2019, Published: 31/Aug/2019

***Abstract*—** RPL is the IPv6 routing protocol for low-power and lossy networks, standardized by IETF in 2012 as RFC6550. Specifically, RPL is designed to be a simple and inter-operable networking protocol for resource-constrained devices in indus-trial, home, and urban environments, intended to support the vision of the Internet of Things with thousands of devices interconnected through multihop mesh networks. More than four-years have passed since the standardization of RPL, and we believe that it is time to examine and understand its current state. In this paper, we review the history of research efforts in RPL; what aspects have been (and have not been) inves-tigated and evaluated, how they have been studied, what was (and was not) implemented, and what remains for future inves-tigation. We reviewed over 97 [41] RPL-related academic research papers published by major academic publishers and present a topic-oriented survey for these research efforts. Our survey shows that only 40.2% of the papers evaluate RPL through experiments using implementations on real embedded devices, ContikiOS and TinyOS are the two most popular implementations (92.3%), and TelosB was the most frequently used hardware platform (69%) on testbeds that have average and median size of 49.4 and 30.5 nodes, respectively. Furthermore, unfortunately, despite it being approximately four years since its initial standardization, we are yet to see wide adoption of RPL as part of real-world systems and applications. We present our observations on the reasons behind this and suggest directions on which RPL should evolve.

Keywords— RPL, IPv6, routing protocol, Internet of Things (IoT), low-power and lossy networks (LLN), Cooja.

I. INTRODUCTION

RPL, THE IPv6 routing protocol for low-power and lossy networks (LLNs), was designed to be suitable for resource-constrained devices in industrial, home, and urban environments [1]. The main goal of RPL is to provide IPv6 connectivity to a large number of battery-operated embedded wireless devices that use low-power radios to communicate and deliver their data over multiple hops. From the initial design phase, RPL builds upon widely-used routing protocols and research prototypes in the wireless sensor network (WSN) domain such as the collection tree protocol (CTP) [2] and Hydro [3], but is extended and re-designed to be part of, and ready for, IPv6. Specifically, RPL was designed to meet the requirements of several applications in the WSN and Internet of Things (IoT) domain [4]–[7], and is considered a critical component that links the low-power network connectivity to application layers in the IETF protocol suite for LLNs.

More than seven-years have passed since the standardization of RPL as RFC6550, and we believe that it is time to look back to examine how researchers are utilizing RPL as part of their system implementations. With its importance and interest, over the past few years there has been considerable amount of effort to characterize, evaluate, and propose enhancements to RPL. These studies range from the domain of optimal parameter selection for target applications to interoperability and performance testing among different implementations. Using open implementations of RPL, some work focuses on evaluating the performance of RPL in testbeds and deployments, while many studies utilize simulated environments to explore and validate the flexibility provided in the RPL standard. We notice that the two most widely used open-sourced RPL implementations are ContikiRPL [8] and TinyRPL [9] within ContikiOS and TinyOS, respectively, and these implementa-tions have been used in almost all RPL research activities that involve real experiments. Given that the RPL standardization process took multiple years, we notice that some work took place prior to the standardization, but most work with RPL occurred after its official standardization in 2012.

Given that the research community and industrial leaders have emphasized the attractiveness of IoT applications in various domains, we started this work with the hope to see RPL be applied in many real-world applications. To understand the current state of RPL, in this work, we review the his-tory of research efforts in RPL; what aspects have been (and have not been) investigated and evaluated, how they have been studied, what was and what was not implemented in open implementations, and what remains for future investigation. Specifically, we have reviewed 97 academic research papers published by major academic publishers with the key-word "RPL" and present a topic-oriented categorization for these research efforts. Based on these observations, we discuss the challenges that RPL (yet) faces today four years after its standardization, and propose points of revision to RFC6550.

II. BACKGROUND - RPL

We provide a brief background of RPL, the IPv6 routing protocol for LLNs, standardized by IETF in March 2012.

A. Vision and Efforts of IETF RoLL Working Group

IETF chartered the routing over low-power and lossy networks (RoLL) working group in 2008 to standardize a practical IPv6 routing protocol for LLNs (RPL). RoLL expected that with the help of RPL standardization, vari-ous useful applications would be realized through LLN. The main characteristics of LLN are described in RFC6550 as follows [1]:

- LLN comprises thousands of constrained nodes that have limited processing power, memory, and sometimes energy (when they are battery operated).
- These constrained nodes are interconnected by lossy links that are usually unstable and typically support only low data rates.
- LLN supports various traffic patterns, not primarily point-to-point (P2P), but in many cases multipoint-to-point (MP2P) or point-to-multipoint (P2MP).

With this vision, RoLL first published several documents during 2009~2010 that describe unique routing requirements in LLN by taking four representative types of applications as examples: urban applications in [4], industrial applications in [5], home automation in [6], and building automation in [7]. These requirements can be summarized as follows:

- Traffic support: A routing protocol for LLNs must be able to provide bi-directional connectivity between arbitrary two nodes in the network, and support unicast, multicast, and anycast service.
- Resource constraint: It should be implementable in resource constrained devices (e.g., 8-bit devices with no more than 128kB (host) or 256kB (router) of memory [7]). For battery-powered nodes, it should provide no more than 1% of duty-cycle [6] and/or at least five years of lifetime [5], [7].
- Path diversity: It must be able to provide alternative routes for reliable packet delivery (>99.9% packet deliv-ery ratio with no more than three retransmissions [7]) over lossy links.
- Convergence time: It must converge after the addition of a new node within a few minutes [5], after re-establishment of a node or losing connectivity within tens of seconds [5] or 4 seconds [6], and within 0.5 seconds [6] if no nodes have moved.
- Node property awareness: It must take into account node characteristics, such as power budget, memory and sleep interval, for routing. It should route via mains-powered nodes if possible [6].
- Heterogeneous routing: It must be able to generate dif-ferent routes with different characteristics for different flows to assure that mission-critical applications cannot be deferred while less critical applications access the network.
- Security: It must support message integrity to prevent attackers and/or unauthenticated nodes from manipulating routing functions or participating in the routing decision process.

After additional 3 years efforts, in 2012, RoLL finalized RPL standardization to fulfill the aforementioned require-ments. RPL standard is described in RFC6550 [1], its routing metrics in RFC6551 [10], timer algorithm in [11], and its objective functions (OFs) for route calculation are described in [12] and [13].

III. RESEARCH ANALYSIS - STATISTICS AND SUMMARY

For experiments, TinyRPL in TinyOS and ContikiRPL in ContikiOS were the two most widely used software implementations. These implementations were popular not only because they are open-source, but due to the popularity of their respective operating systems in the WSN/LLN commu-nity. Specifically, TinyRPL was used in 14 unique papers and ContikiRPL in 26 papers, with 4 papers using both, adding up to 35 unique papers using these two implementations (92.3%) out of 39 papers that conducted experiments. The only three other implementations were NanoQplus in [14], RIOT [20]–[23], and FreeRTOS in [24].

An interesting point to be noted here is that, no experiment-based evaluation was performed for multi-instance, LOAD(ng), and security subtopics. Also, only a small fraction of multi-sink and mobility-related work have done experiments. Again, this is disappointing, and demands more real-experiment based research in those categories.

As shown in Fig. 1, [41] the major hardware platform used for RPL experiments was the ‘TelosB’ (‘Tmote sky’ is equiva-lent) platform with MSP430 microprocessor and CC2420 radio developed in 2004. It was used in 27 unique papers out of 39 papers that conducted experiments, a 69%. Other platforms used were ‘WSN430 open node’, ‘M3 open node’, ‘JN5168’, ‘MSB-A2’, ‘Zolertia Z1’, ‘PowerNet’, ‘WPCDevKit’, ‘PLC G3’, and ‘BCM4356’, each appearing only once or twice in our list of papers. In the perspective of physical and RF layers, ‘WPCDevKit’ and ‘PLC G3’ have a PLC transceiver, only ‘BCM4356’ has a BLE transceiver, and all other plat-forms have an IEEE 802.15.4 transceiver. An interesting point is that, despite continuous advancement of IoT platforms, ‘TelosB’ [25], one of the classic WSN platforms, is still frequently used for research of RPL which was standardized in *2012* (after 8 years). Given that MCU of ‘TelosB’ has memory of 48kB ROM and 10kB RAM, which is much smaller than recent platforms such as Firestorm [26], RPL implementations considering the use of ‘TelosB’ may not (and does not) provide the full functionality of RPL.

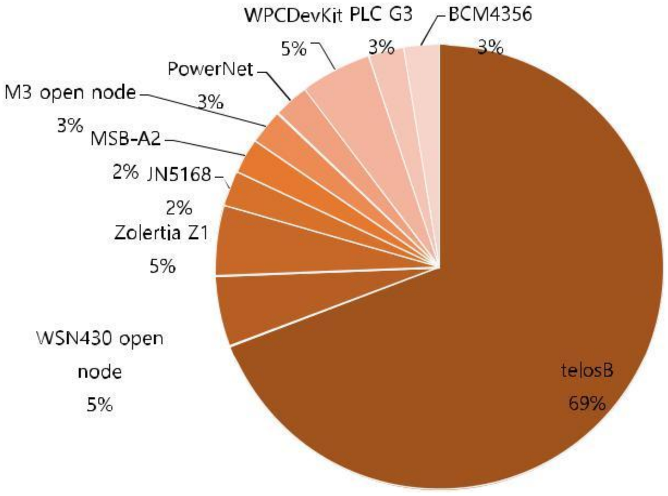

FIG. 1 DISTRIBUTION OF HARDWARE PLATFORM

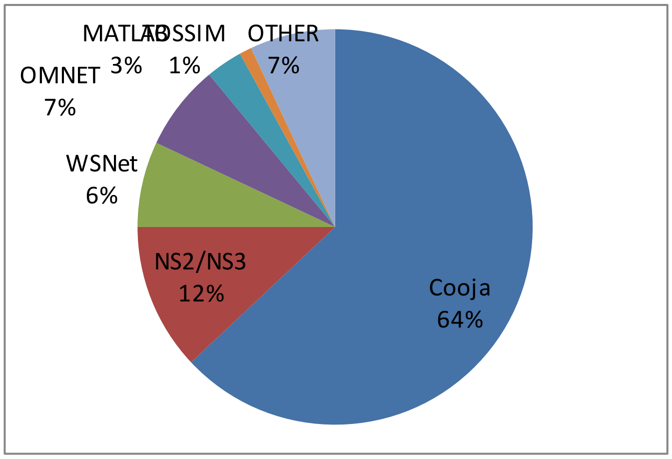

Fig. 2. Distribution of simulation method

For simulations studies, COOJA simulator [27] using ContikiOS/ContikiRPL implementation was the dominant method – 62.9% [41]of all the simulation studies used COOJA simulator, as shown in Fig. 2. Runners-up were the NS-2/NS-3 [28], WSNet and OMNET++ simulators with 11.4%, 7.1% and 7.1% respectively. Other simulators used were MATLAB, OPNET, Qualnet, Python, TOSSIM, etc. It is interesting that the TOSSIM simulator for TinyOS was used only once in the 97 publications[41]. It turns out that, TOSSIM simulator only supports Mica2/MicaZ platforms for simula-tions, but these platforms did not have enough RAM to run the BLIP/TinyRPL stack for IPv6 and RPL in TinyOS. The only one paper that used TOSSIM investigated the DODAG root failure detection problem, but implemented the proposed mechanism not on RPL but CTP, due to lack of memory [29].

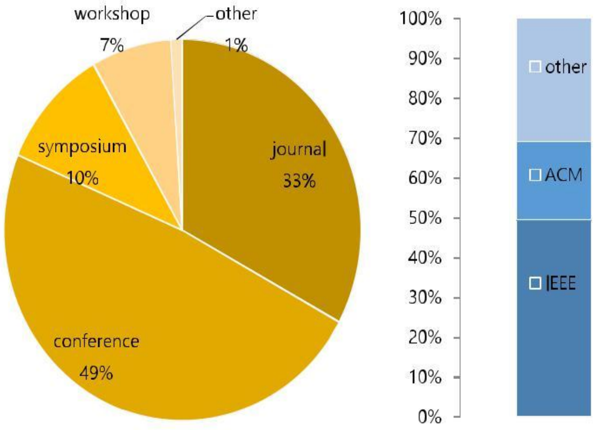

Fig. 3. Distribution of publication venue

For simulations studies, COOJA simulator [27] using ContikiOS/ContikiRPL implementation was the dominant method – 62.9% of all the simulation studies used COOJA simulator, as shown in Fig. 2. Runners-up were the NS-2/NS-3 [28], WSNet and OMNET++ simulators with 11.4%, 7.1% and 7.1% respectively. Other simulators used were MATLAB, OPNET, Qualnet, Python, TOSSIM, etc. It is interesting that the TOSSIM simulator for TinyOS was used only once in the 97 publications. It turns out that, TOSSIM simulator only supports Mica2/MicaZ platforms for simula-tions, but these platforms did not have enough RAM to run the BLIP/TinyRPL stack for IPv6 and RPL in TinyOS. The only one paper that used TOSSIM investigated the DODAG root failure detection problem, but implemented the proposed mechanism not on RPL but CTP, due to lack of memory [29].

One interesting point to note is that, most of the publications from Europe used ContikiRPL implementation for experiments while those from America and Asia mainly used TinyRPL implementation.

By Publication Venue and Demographics: Fig. 3 plots the distribution of venue types for RPL-related publications. 34.4% of the papers were published in international journals, and 64.6% were published in con-ference/symposium/workshop proceedings where IEEE SmartGridComm and ACM SenSys were the most popular venues with 5 and 4 publications respectively. 49.5% of all the papers were published at venues sponsored by IEEE, and 20.4% at ACM sponsored venues. Other publishers include Elsevier, Springer, Inderscience, Hindawi, etc.

IV. LITERATURE SURVEY:

Tsiftes *et al*. [8] first evaluated the performance of ContikiRPL and Ko *et al*. [9], [40] first evaluated that of TinyRPL, which show that both the two representative RPL implementations provide reliable upward packet deliv-ery. Especially in [9] and [30], Ko *et al*. showed that TinyRPL provides upward packet delivery performance that is compa-rable to CTP [2]. Kim *et al*. [28] deployed a TinyRPL-based multihop network in an urban marketplace, which confirmed the reliability of TinyRPL's upward packet delivery.

Ancillotti *et al*. [15] evaluated ContikiRPL's uplink performance using COOJA simulator, which showed that ContikiRPL makes some nodes maintain unreliable routes even though reliable alternative routes exist, resulting in severe performance degradation for those nodes

Some authors designed RPLca+ that includes a fast link quality update of each neigh-bor based on DIS unicasting and priority-based neighbor table management [16], and evaluated its performance through both COOJA simulations and testbed experiments.

Dawans *et al*. [31] evaluate ContikiRPL's performance in a large-scale testbed

Khelifi *et al*. [33] find that even when ContikiRPL succeeds in detecting unreliable links, it requires a long detection time (after experiencing many packet losses) due to RPL's reactive nature

Oliveira and Vazão [17] survey research on RPL-based mobility support. In addition, the authors evaluate the four routing protocols presented in [34]–[36] and [37] through COOJA simulations. The results reveal that a RPL-based mobile routing protocol suffers from severe performance degradation due to large amount of control traffic if it keeps up-to-date routing table. In contrast, less responsive proto-cols with fewer control traffic provide better packet delivery performance.

Vucini˘c´ *et al*. [38] compare the performance of RPL and LOADng through COOJA simulations.

Elyengui *et al*. [39] evaluated RPL and LOADng through COOJA simulations under bi-directional traffic scenarios, which revealed that RPL provides less delay, less overhead, and higher reliability than LOADng.

Mayzaud *et al*. [40] address topological inconsistency attacks, which maliciously trigger local repairs (i.e., fre-quent resets of *TrickleTimer*). They restrict the number of *TrickleTimer* resets per hour by using a threshold and propose an adaptive threshold control scheme, named AT, which reduces both control overhead and energy consump-tion. They evaluate AT through COOJA simulations.

V. WHAT HAS NOT BEEN STUDIED?

RFC6550 [1] is the core document for the RPL standard, and RFC6551 [10] is the companion standard that defines and describes the routing metrics used for path calculation in RPL.

First of all, none of RPL's own security mechanisms are implemented in either TinyRPL or ContikiRPL. There are many other features in the standard that are not implemented in both TinyRPL and ContikiRPL. Below are a few:

Regarding the downward routing operation, both TinyRPL and ContikiRPL[1] implemented only the 'storing-mode', although some prior work have implemented their own version of the 'non-storing-mode' [14], [29]. However, no prior work seems to have implemented the 'path control' feature, which allows nodes to request for or allow multiple downward routes, described in Section 9.9 of the standard. Furthermore, from RFC6551, features such as 'Node State and Attribute Object' , 'Node Energy Object' , 'Throughput/Latency' and the 'Link Color Object' could not be found in any prototype implementations of any prior work that we have found. Even before the RPL standardization, Hui and Culler [32] propose to forward data traffic through alternative routes tem-porarily to probe link qualities of a diverse set of nodes, which is not implemented in TinyRPL nor ContikiRPL

It is true that the aforementioned un-implemented features are defined as 'optional' in the RPL standard. Furthermore, our survey of open-source prototype implementations and prior academic research publications might not be exhaus-tive enough to state that these features were 'never' imple-mented. However, our position is that *there are too many 'optional' features in RPL*. We acknowledge that this was intended to provide flexibility in the RPL design, but at the same time, it increases the complexity of the standard doc-uments, and hinders more open-source implementations

ACKNOWLEDGMENT

Our work presents a survey of how the RPL routing proto-col has been used and evaluated by examining 98 papers[41] that study RPL with a focus on those that utilize open-source RPL implementations. Among the publications that provide experi-mental evaluation, ContikiOS and TinyOS were the two most popular implementations (92.3%), TelosB was the most frequently used hardware platform on testbeds (69%), and the testbeds comprise average and median size of 49.4 and 30.5 nodes respectively. Through our studies, we were also able to notice that many of the optional RPL functionalities were not well supported (nor needed) in many scenarios in which RPL was targeted to be applied. Furthermore, despite approx-imately four years since its initial standardization, we are yet to see wide adoption of RPL as part of real-world systems and applications[41].

[1] Non-storing mode implementation has been added to ContikiRPL recently (Feb. 2 2016) in the latest head of the Contiki GitHub reposi-tory. However, the latest official release of Contiki-OS (which is Contiki 3.0 released on Aug. 25 2015) still does not have a non-storing mode implemen-tation. This is probably the reason why there is not yet any published research paper that uses the non-storing mode on ContikiRPL.

REFERENCES

[1] T. Winter *et al.*, "RPL: IPv6 routing protocol for low-power and lossy networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 6550, Mar. 2012.

[2] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. ACM Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, Berkeley, CA, USA, 2009, pp. 1–14.

[3] S. Dawson-Haggerty, A. Tavakoli, and D. Culler, "Hydro: A hybrid routing protocol for low-power and lossy networks," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Gaithersburg, MD, USA, Oct. 2010, pp. 268–273.

[4] T. Watteyne, Ed., *et al.*, "Routing requirements for urban low-power and lossy networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 5548, May 2009.

[5] E. K. Pister, E. P. Thubert, S. Dwars, and T. Phinney, "Industrial routing requirements in low-power and lossy networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 5673, Oct. 2009.

[6] A. Brandt, J. Buron, and G. Porcu, "Home automation routing require-ments in low-power and lossy networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 5826, Apr. 2010.

[7] E. J. Martocci, P. D. Mil, N. Riou, and W. Vermeylen, "Building automation routing requirements in low-power and lossy networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 5867, Jun. 2010.

[8] N. Tsiftes, J. Eriksson, and A. Dunkels, "Low-power wireless IPv6 routing with ContikiRPL," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Stockholm, Sweden, 2010, pp. 406–407.

[9] J. Ko *et al.*, "ContikiRPL and TinyRPL: Happy together," in *Proc. Workshop Extending Internet Low Power Lossy Netw. (IP+SN)*, Apr. 2011.

[10] J. P. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing metrics used for path calculation in low-power and lossy networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 6551, Mar. 2012.

[11] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle algorithm," Internet Eng. Task Force, Fremont, CA, USA, RFC 6206, Mar. 2011.

[12] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (RPL)," Internet Eng. Task Force, Fremont, CA, USA, RFC 6552, Mar. 2012.

[13] O. Gnawali and P. Levis, "The minimum rank with hysteresis objective function," Internet Eng. Task Force, Fremont, CA, USA, RFC 6719, Sep. 2012.

[14] J. Ko *et al.*, "DualMOP-RPL: Supporting multiple modes of downward routing in a single RPL network," *ACM Trans. Sensor Netw.*, vol. 11, no. 2, pp. 1–20, Mar. 2015.

[15] E. Ancillotti, R. Bruno, and M. Conti, "The role of the RPL rout-ing protocol for smart grid communications," *IEEE Commun. Mag.*, vol. 51, no. 1, pp. 75–83, Jan. 2013.

[16] E. Ancillotti, R. Bruno, and M. Conti, "Reliable data delivery with the IETF routing protocol for low-power and lossy networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1864–1877, Aug. 2014.

A. Oliveira and T. Vazão, "Low-power and lossy networks under mobility: A survey," *Comput. Netw.*, vol. 107, pp. 339–352, Oct. 2016.

[17] H.-S. Kim *et al.*, "MarketNet: An asymmetric transmission power-based wireless system for managing e-price tags in markets," in *Proc. ACM Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, Seoul, South Korea, Nov. 2015, pp. 281–294.

[18] *Connected Grid Networks for Smart Grid—Field Area Network*, Cisco, San Jose, CA, USA, accessed: May 2017. [Online]. Available: http://www.cisco.com/c/en/us/solutions/industries/energy/external-utilities-smart-grid/field-area-network.html

[19] *RIOT OS*. Accessed: May 2017. [Online]. Available: http://riot-os.org/api/group__net__gnrc__rpl.html

[20] *RIOT RPL*. Accessed: May 2017. [Online]. Available: https://github.com/RIOT-iOS/RIOT/tree/master/sys/net/gnrc/routing/rpl

[21] H. Perrey, M. Landsmann, O. Ugus, M. Wählisch, and T. C. Schmidt, "TRAIL: Topology authentication in RPL," in *Proc. Eur. Workshop Wireless Sensor Netw. (EWSN)*, Graz, Austria, 2016, pp. 59–64.

[22] C. Gündogan, C. Adjih, O. Hahm, and E. Baccelli, "Let healthy links bloom: Scalable link checks in low-power wireless networks for smart health," in *Proc. ACM Int. Workshop Pervasive Wireless Healthcare (MobileHealth)*, Paderborn, Germany, Jul. 2016, pp. 11–16.

[23] O. Balmau *et al.*, "Evaluation of RPL for medium voltage power line communication," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Venice, Italy, Nov. 2014, pp. 446–451.

[24] Moteiv Corporation. *Tmote Sky*. Accessed: May 2017. [Online]. Available: http://www.moteiv.com/products/tmotesky.php

[25] M. P. Andersen, G. Fierro, and D. E. Culler, "System design for synergistic, low power mote/BLE embedded platform," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Vienna, Austria, Apr. 2016, pp. 1–12.

[26] N. Tsiftes *et al.*, "A framework for low-power IPv6 routing simulation, experimentation, and evaluation," in *Proc. ACM Conf. Appl. Technol. Archit. Protocols Comput. Commun. (SIGCOMM)*, New Delhi, India, Sep. 2010, pp. 479–480.

[27] L. Bartolozzi, T. Pecorella, and R. Fantacci, "Ns-3 RPL module: IPv6 routing protocol for low power and lossy networks," in *Proc. Int. ICST Conf. Simulat. Tools Tech. (SIMUTOOLS)*, Mar. 2012, pp. 359–366.

[28] K. Iwanicki, "RNFD: Routing-layer detection of DODAG (root) node failures in low-power wireless networks," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Vienna, Austria, Apr. 2016, Art. no. 13.

[29] J. Ko, S. Dawson-Haggerty, O. Gnawali, D. Culler, and A. Terzis, "Evaluating the performance of RPL and 6LoWPAN in TinyOS," in *Proc. Workshop Extending Internet Low Power Lossy Netw. (IP+SN)*, Apr. 2011.

[30] N. Khelifi, S. Oteafy, H. Hassanein, and H. Youssef, "Proactive maintenance in RPL for 6LowPAN," in *Proc. Int. Conf. Wireless Commun. Mobile Comput. (IWCMC)*, Dubrovnik, Croatia, Aug. 2015, pp. 993–999.

[31] J. W. Hui and D. E. Culler, "IP is dead, long live IP for wireless sensor networks," in *Proc. ACM Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, Raleigh, NC, USA, Nov. 2008, pp. 15–28.

[32] S. Dawans, S. Duquennoy, and O. Bonaventure, "On link estimation in dense RPL deployments," in *Proc. IEEE Conf. Local Comput. Netw. Workshops*, Clearwater, FL, USA, Oct. 2012, pp. 952–955.

[33] I. E. Korbi, M. B. Brahim, C. Adjih, and L. A. Saidane, "Mobility enhanced RPL for wireless sensor networks," in *Proc. 3rd Int. Conf. Netw. Future (NOF)*, Gammarth, Tunisia, Nov. 2012, pp. 1–8.

[34] C. Cobârzan, J. Montavont, and T. Noël, "Analysis and performance evaluation of RPL under mobility," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Funchal, Portugal, Jun. 2014, pp. 1–6.

[35] K. C. Lee *et al.*, "A comprehensive evaluation of RPL under mobility," *Int. J. Veh. Technol.*, vol. 2012, Mar. 2012, Art. no. 904308.

[36] O. Gaddour *et al.*, "Co-RPL: RPL routing for mobile low power wire-less sensor networks using corona mechanism," in *Proc. IEEE Int. Symp. Ind. Embedded Syst. (SIES)*, Pisa, Italy, Jun. 2014, pp. 200–209.

[37] M. Vucini˘c,´ B. Tourancheau, and A. Duda, "Performance compari-son of the RPL and LOADng routing protocols in a home automation scenario," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Shanghai, China, Apr. 2013, pp. 1974–1979.

[38] S. Elyengui, R. Bouhouchi, and T. Ezzedine, "LOADng routing proto-col evaluation for bidirectional data flow in AMI mesh networks," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 6, pp. 37–43, 2015.

[39] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "Mitigation of topological inconsistency attacks in RPL-based low-power lossy networks," *Int. J. Netw. Manag.*, vol. 25, no. 5, pp. 320–339, 2015.

[40] Hyung-Sin Kim, Jeonggil Ko, David E. Culler, "Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 19, NO. 4, FOURTH QUARTER 2017

Authors Profile

Mr. Manish Mishra pursed Bachelor of Engineering from Rajiv Gandhi Proudyogiki Vishvavidyalaya Bhopal in 2015. He is currently pursuing Master of Engineering from same university in computer science and Engineering. He is a member of IEEE & IEEE computer society since 2019. His main research work focuses on IoT, Network Security, Cloud Security and Privacy, Big Data Analytics, Data Mining and Computational Intelligence based education.

Dr. Piyush Shukla received his Bachelor's degree in Electronics & Communication Engineering, LNCT, Bhopal in 2001, M. Tech (Computer Science & Engineering) in 2005 from SATI, Vidisha and Ph.D. (Computer Science & Engineering) in 2013 from RGPV, Bhopal. M.P. India. Currently he is working as an Assistant Professor in Department of Computer Science & Engineering, UIT-RGPV Bhopal. He is a member of IEEE & IEEE computer society since 2019. He has published more than 40 research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences including IEEE and it's also available online. His main research work focuses on Cryptography Algorithms, Network Security, Computer Networking, Data Mining, IoT and Computational Intelligence based education. He has 12 years of teaching experience and 17 years of Research Experience.

Dr. Rajeev Pandey received his Bachelor's Degree in Computer Science and Engineering from IET ,DR. B.R.A. University, Agra (U.P.), M.Tech (Computer Science & Engineering) in 2004 & Ph.D in 2010 from DR. B.R.A. University, Agra (U.P.), India Currently he is working as an Assistant Professor in department of Computer Science & Engineering, UIT-RGPV Bhopal. He is a member of IEEE & IEEE computer society since 2019. He has published more than 20 research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences including IEEE and it's also available online. His main research work focuses on Cryptography Algorithms, Network Security, Computer Networking, Data Mining, IoT and Computational Intelligence based education. He has 12 years of teaching experience and 17 years of Research Experience.