

Performance Analysis of an IP based Protocol Stack for WSNs

Sumeet Thombre¹, Raihan Ul Islam², Karl Andersson³, and Mohammad Shahadat Hossain⁴

^{1,2,3}Pervasive and Mobile Computing Laboratory, Luleå University of Technology, SE-931 87 Skellefteå, Sweden

⁴Department of Computer Science and Engineering, University of Chittagong-4331, Bangladesh

¹sumtho-5@student.ltu.se, ²raihan.ul.islam@ltu.se, ³karl.andersson@ltu.se, ⁴hossain_ms@cu.ac.bd

Abstract—Wireless sensor networks (WSNs) are the key enablers of the internet of things (IoT) paradigm. Traditionally, sensor network research has been to be unlike the internet, motivated by power and device constraints. The IETF 6LoWPAN draft standard changes this, defining how IPv6 packets can be efficiently transmitted over IEEE 802.15.4 radio links. Due to this 6LoWPAN technology, low power, low cost micro-controllers can be connected to the internet forming what is known as the Wireless embedded internet. Another IETF recommendation, CoAP allows these devices to communicate interactively over the internet. The integration of such tiny, ubiquitous electronic devices to the internet enables interesting real-time applications. We evaluate the performance of a stack consisting of CoAP and 6LoWPAN over the IEEE 802.15.4 radio link using the Contiki OS and Cooja simulator, along with the CoAP framework Californium (Cf).

Keywords - 6LoWPAN, CoAP, Contiki, Cooja, Californium (Cf).

I. INTRODUCTION

When thinking about sensor networks the first thing that does not come to mind is the use of the Internet Protocol (IP). IP has always been considered the protocol for Local Area Networks or Wide Area Networks, PCs and Servers. It was not thought appropriate for sensor networks or Personal Area Networks, Premise Area Networks, Perimeter Area Networks (PANs) and the sensors themselves because of the perception that it was too heavy weight for these applications. Recently there has been a large community starting to rethink many of the misconceptions about IP and have embarked on the development and standardization of 6LoWPAN with the question ‘Why invent a new protocol when we already have IP?’ 6LoWPAN defines how to layer IP version 6 (IPv6) over low data rate, low power, small footprint radio networks (LoWPAN) as typified by the IEEE 802.15.4 radio [1].

Secondly, the most common way to communicate with sensors and other wireless low capacity devices has been by the implementation of proprietary protocols over IEEE 802.15.4 radio links. However, proprietary protocols present interoperability problems when trying to transfer packets to devices outside the IEEE 802.15.4 network, and thus, IP connectivity is desired [2]. Sensor networks today are a sensing, computing and communication infrastructure that are able to observe and respond to phenomena in the natural environment and in our physical and cyber infrastructure. One such application is the real-time application of flood assessment, which is a pilot test case for our proposed

stack. This work attempts to evaluate the quality of service parameters namely, the throughput, the end to end delay and the packet loss by means of simulations.

The rest of the paper is organized as follows. Section II discusses the related work and the background. Section III presents a pilot test case for the network stack by proposing a system architecture to assess the risk of flooding. Section IV explores the various available network simulators and operating systems for sensor networks. Section V illustrates the simulation scenario and section VI shows the results and analysis. Finally section VII concludes the paper with section VIII indicating the future work.

II. RELATED WORK AND BACKGROUND

In the IoT space, both the industry and the academic research community seem to be converging towards establishing standards. For knowledge formulation, this research included both academic and industrial works. For a more fundamental understanding of wireless sensor networks and associated technologies like 802.15.4, Akyildiz, Ian et al. [3] is a good foundation. But, undeniably the biggest source of comprehension for this research were the IETF recommendations [4][5][6]. Shelby, Zach et al. [7] provides a detailed insight into the different operating systems and stacks available in sensor networks. [8] provides specific details about the aspects of proto-threads and the Cooja simulation environment, while providing full IP networking under the hood as an open source software. Some industrial white papers like [9] provide comparative analysis between technologies. Other major industrial alliances like the thread group and Zigbee alliance provide stack details, fact sheets and benchmarks [10]. The eclipse foundation provides a publicly available CoAP framework Kovatsch, Matthias et al. [11], which provides reference libraries and functions to generate interesting use cases and applications. Mehmood, Waiser et al. [12] describe the performance evaluation of 6LoWPAN using Contiki OS and the Cooja simulator.

Robles et al. [13] proposed a smart water management model based on the OPC UA (Object Linking and Embedding for Process Control Unified Architecture) platform combining Internet of Things technologies with business processes coordination and decision support systems. They provided an architecture for sub-system interaction and a detailed description of the physical scenario. Picric et al. [14]

describe an approach to integration of physical and digital worlds through aggregation of Internet of Things (IoT) service with the Augmented Reality (AR) platform, AR Genie. The IoT service is provided by ekoNET platform which provides environmental data through AR based applications. By extending the AR Genie platform with the ekoNET IoT service the authors were able to demonstrate usage of a real-time environmental data within AR mobile applications as well as to enable a new, more engaging way of IoT data visualization utilizing serious gaming and AR technologies.

ZigBee is a networking layer built on top of IEEE standard 802.15.4 MAC. Arguably, ZigBee is the most popular low-cost, low-power wireless mesh networking standard available today. Currently, a good amount of companies are throwing their weight behind the Zigbee alliance. After the IETF recommendations, 6LoWPAN has entered the fray as Zigbee's competition as it can utilize 802.15.4 but even better, it can run on other physical layers, and it allows for seamless integration with other IP-based systems [9].

The essential metrics while choosing a wireless protocol are interoperability, scalability, complexity, re-usability, security, reliability, energy efficiency and availability. 6LoWPAN trumps Zigbee on most of these criteria as it derives most of the advantages from the prowess of IP. Then, there are performance metrics to be studied such as stack size/packet overhead, throughput, end to end delay, latency, packet loss etc. This paper attempts to evaluate these metrics using simulations, which will determine the feasibility of the proposed IETF stack to real-time applications. The next section focuses on the system and network architecture of an application scenario, a flood detection system.

III. PILOT CASE - SYSTEM ARCHITECTURE FOR FLOOD ASSESSMENT AND INVOLVED PROTOCOLS

Various types of wireless sensors can be employed to collect the quantitative data and the heterogeneous WSN architecture is as shown below,

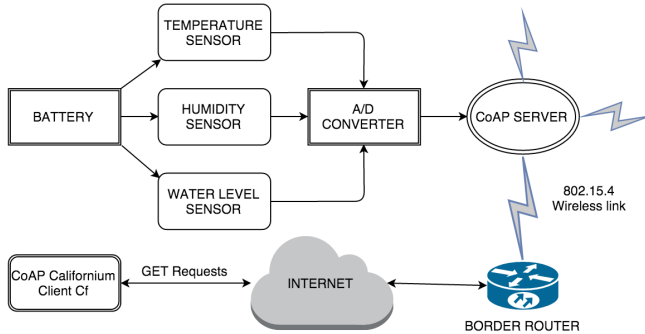


Fig. 1. Proposed network architecture for flood assessment.

Andersson, K et. al. [15] [16] proposed a framework for flood detection which considers meteorological, geological, river characteristics, topographical and human activities. 17 input data are required to feed the system, which are both quantitative and qualitative. Examples of quantitative data are

rainfall, velocity, soil infiltration rate, etc. and the examples of qualitative data are settlement on the flood prone area.

The network stack architecture is in accordance with the IETF recommended stack and the IP for smart objects alliance (IPSO). A brief description of the protocol layers is as given below,

A. 802.15.4

The IEEE 802.15.4 standard has been widely accepted as the de facto standard for Wireless Sensor Networks (WSNs) at the physical and MAC layer, since it provides ultra-low complexity, cost and energy consumption for low-data rate wireless connectivity.

The main characteristics of IEEE802.15.4 standard are: Low-rate, Low power, Low cost, Short distance, Multiple types of devices, Full Function Device (FFD) like coordinator and router and Reduced Function Device (RFD) which has limited functionality and sleeps most of the time and multiple transmission modes [17].

These advantages deem an optimal fit for the flood detection application, where the sensors need to be battery powered, spread out over a large area and low cost, communicating data reliably at typically low rates.

B. 6LoWPAN, RPL

6LoWPAN working group works on the research of IPv6 protocol suite based on IEEE802.15.4 standard, and constructs self-organizing 6LoWPAN network with routing protocol. 6LoWPAN technology bottom layer adopts PHY and MAC layer standards of IEEE802.15.4, and 6LoWPAN chooses IPv6 as the networking technology. By communicating natively with IP, 6LoWPAN networks are connected to other networks simply using IP routers.

The MTU size for IPv6 packets over IEEE 802.15.4 is 1280 octets. However, a full IPv6 packet does not fit in an IEEE 802.15.4 frame. 802.15.4 protocol data units have different sizes depending on how much overhead is present. Starting from a maximum physical layer packet size of 127 octets (aMaxPHYPacketSize) and a maximum frame overhead of 25 (aMaxFrameOverhead), the resultant maximum frame size at the media access control layer is 102 octets. Link-layer security imposes further overhead, which in the maximum case (21 octets of overhead in the AES-CCM-128 case, versus 9 and 13 for AES-CCM-32 and AES-CCM-64, respectively) leaves only 81 octets available. This is obviously far below the minimum IPv6 packet size of 1280 octets and a fragmentation and reassembly adaptation layer must be provided at the layer below IP [4].

The IETF ROLL working group has defined application-specific routing requirements for a Low-power and Lossy Network (LLN) routing protocol RPL. RPL provides a mechanism to disseminate information over the dynamically formed network topology. This dissemination enables minimal configuration in the nodes, allowing nodes to operate mostly autonomously. RPL involves the formation of a directed acyclic graph (DAG) having a DAG root, DODAG (destination oriented DAG) and a DODAG

root. The DODAG root may act as a border router for the DODAG; in particular, it may aggregate routes in the DODAG and may redistribute DODAG routes into other routing protocols [5].

C. CoAP

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks. The nodes often have 8-bit microcontrollers with small amounts of ROM and RAM, while constrained networks such as IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) often have high packet error rates and a typical throughput of 10s of kbit/s. CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types. CoAP is designed to easily interface with HTTP for integration with the Web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments [6].

IV. SIMULATIONS

The easiest way to enable a wireless embedded device with 6LoWPAN is by integrating an existing protocol stack, either with a network processor, a stack included with an operating system or by integrating a stack into an embedded software project. A protocol stack for 6LoWPAN typically includes, at the minimum, these basic components: radio drivers, medium access control (e.g. IEEE 802.15.4), IPv6 with 6LoWPAN (ID-6lowpan-hc), UDP, ICMPv6, Neighbor Discovery (ID-6lowpan-nd) and socket-like or other API to the stack. Currently, there are two open-source protocol stacks for embedded operating systems: uIPv6 for Contiki and BLIP for TinyOS, and three commercial protocol stacks: Sensinodes NanoStack, Jennics 6LoWPAN and the Nivis ISA100 stack.

A. Contiki OS and uIPv6

Contiki is a popular embedded open-source operating system for small microcontroller architectures such as AVR, 8051 and MSP430, led by the Swedish Institute for Computer Science (SICS) [Contiki]. Contiki includes a very small implementation of IP called uIP, along with an implementation of IPv6 with 6LoWPAN support called uIPv6. The Contiki architecture is designed for supporting IP networking over low-power radios and other network interfaces. The operating system is implemented in C and uses a make build environment for cross-compilation on most platforms. A wide variety of microcontroller and device platform ports exist, along with examples and reusable applications. The Contiki OS provides basic thread and timer support. Both built-in and user applications are run over Contiki using a lightweight thread model called protothreads [7]. Due to greater availability and flexibility, this work uses the open source solution Contiki as the default OS for simulations.

A simulator is a very useful tool for the application software development in wireless sensor networks. Before the simulator came up, the code development was very difficult and tedious for users due to the long compilation and debugs time and the transplant problem of the program. With the simulator tool, users can obtain much benefit during the software development phase and large scale test phase. The various simulators researched for this work were OPNET, ns3, COOJA and OMNET++, and COOJA was found to be the best to simulate the network we have in mind, due to its flexibility, extensibility and quick prototyping as the next section explains.

B. COOJA

COOJA is a wireless sensor network simulator designed for the Contiki operating system. It is a flexible java-based simulator which supports using C language to develop application software by Java Native Interface. One of the great advantages of this COOJA simulator is that it can simulate the application software simultaneously in high level algorithm development and low level hard driver development. The COOJA simulator has great extensibility. Application developer can alter parts of the simulation environment without changing any COOJA main code. It means that the system can be added new parts such as interfaces, plugins and radio mediums or reconfigured existing parts. With these advantages of COOJA, we can implement variant simulations with different conditions and system settings such as different packet generation rates, different MAC protocols and different network topology [18].

C. A word about the Californium (Cf) Framework

Californium is a powerful CoAP framework which implements CoAP in Java. It targets back end services and stronger IoT devices, providing a convenient API for RESTful Web services that support all of CoAP's features. The Californium (Cf) CoAP framework shows 33 to 64 times higher throughput than high-performance HTTP Web servers. The Californium reference implementation Cf also outperforms other CoAP systems with a three times higher throughput. It is explicitly designed for scalable IoT cloud services and is publicly available at the Eclipse Foundation [11]. All these advantages motivated the use of this framework in the simulation scenario described in the next section.

D. Simulation scenario

The simulation scenario consists of temperature and humidity sensors running CoAP servers which are periodically queried by a CoAP client. The CoAP client is a Californium (Cf) based Java program, which periodically sends GET requests to the servers in the simulation to get the current temperature and humidity values through the border router, which connects the network to the internet using the TUN-SLIP utility. The network setup in Cooja is as shown in the figure below.

The nodes are positioned in such a way that there are 1 to 5 hops from the border router to the different nodes. Node

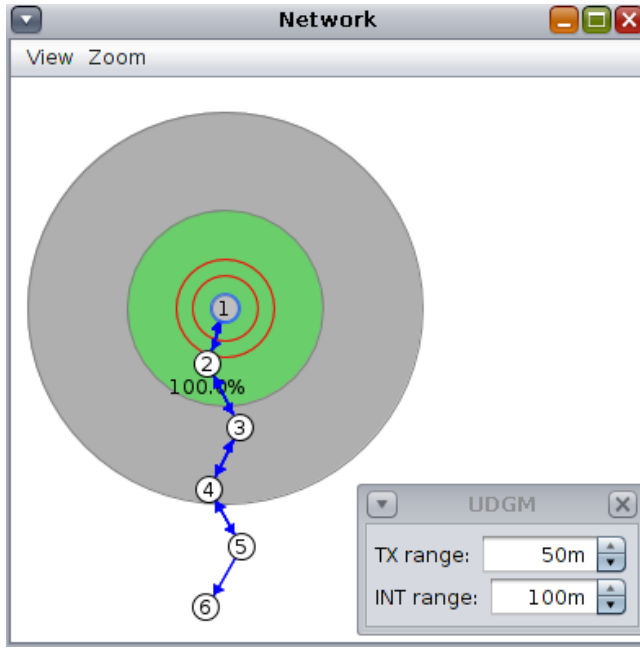


Fig. 2. Network consisting of motes and their radio environment in Cooja.

1 is the border router, which connects the sensor network to the internet, motes 2-6 are temperature and humidity sensors running CoAP servers which are queried by the external CoAP client application.

The general simulation parameters and their values are as shown below in the table,

TABLE I
GENERAL SIMULATION PARAMETERS SET IN COOJA

Parameter name	Value
Radio medium	Unit Disk Graph Medium
Mote Type/ startup delay	T-mote Sky/ 1000 ms
MAC layer	CSMA/CA
Bit rate	250 kbps
Radio duty cycling	NullRDC
Node transmission range	50 m
Node carrier sensing range	100 m
Tx / Rx ratio	100 %

The CoAP client used, Californium (Cf), can be customized and the tuned CoAP specific settings are as follows,

TABLE II
CALIFORNIVM PROPERTIES

Property name	Value
Max retransmit factor	0
Max trasmit wait	93000
Ack timeout	2000
Ack random factor	1.5

V. RESULTS AND ANALYSIS

The scenario evaluates three important parameters namely the throughput, end to end delay and the packet loss for the various motes in the network. This evaluation also tests the

bounds of the RPL routing protocol used in 6LoWPAN, as the scenario features motes with 1 to 5 hops to the router. There is a lot of data generated to maintain the DAG in the RPL protocol for which the ICMP is used as a transport. The initial network traffic is offered by the ICMP protocol trying to establish the DAG, after which it reduces gradually. Also, the resilience of the network was tested for dynamic changes in the topology and the RPL successfully adjusted the routes to ensure packet transmissions. We also demonstrated the reliability of the Californium (Cf) framework, which ensured that negligible packet loss occurred even when the packet generation rate was as high as 1000 packets/sec for a distant mote 5 hops away.

A. Throughput

The packet generation rate was varied from 1 packet per second up to 100 packets per second in a constant delay by the Java client. These GET requests were sent to the temperature and humidity sensor motes running CoAP servers and the corresponding throughput was noted in each of the cases which is depicted in the table below,

TABLE III
THROUGHPUT VARIATION (IN KBPS) AS A FUNCTION OF THE PACKET GENERATION RATE PGR (IN PACKETS/SECOND)

PGR	Throughput in Kbps				
	1-hop	2-hops	3-hops	4-hops	5-hops
1	0.8977	0.8635	0.7583	0.6904	0.6508
2	1.4425	1.292	1.02	0.8529	0.7129
5	2.1026	1.6152	1.203	0.8504	0.7424
10	2.45	1.6062	1.3027	0.8758	0.7832
20	2.098	1.3228	1.099	0.8065	0.6837
50	1.5733	0.9643	0.7021	0.5926	0.5464
100	1.2866	0.8082	0.5506	0.5126	0.4806

The graph below depicts the variation of the throughput for the various motes (in Kbps) at hops 1 to 5 against the packet generation rate in packets/sec,

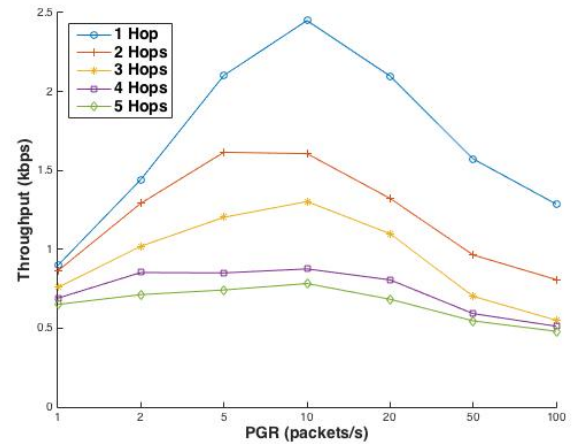


Fig. 3. Plot of average throughput for various motes at 1 to 5 hops against the rate of packet generation.

The throughput is calculated as (in Kbps) = (No of successful CoAP request/response pairs * (length of request + length of response in bits)) / total time of simulation. We can see from the graph that the throughput peaks at around 10 packets/sec and then declines. This can be attributed to the greater packet loss occurring due to increased incoming traffic.

B. End to End delay

The table below depicts the various values for the end to end delay in a similar simulation scenario varying the packet generation rate from 1 to 100 packets/sec,

TABLE IV

END TO END DELAY VARIATION (IN MS) AS A FUNCTION OF THE PACKET GENERATION RATE PGR (IN PACKETS/SECOND)

PGR	End to End delay in ms				
	1-hop	2-hops	3-hops	4-hops	5-hops
1	121.01	136.95	152.95	163.95	173.2
2	125.91	142.59	153.77	167.76	173.06
5	181.2	193.8	209.77	227.82	255.914
10	254.2	266.86	322.85	339.41	382.92
20	233.89	295.05	343.97	390.26	430.48
50	387.66	425.2	539.96	575.16	652.34
100	406.9	586.88	747.95	775.38	828.03

The end to end delay is an important parameter of evaluation as it is a direct indicator of the network congestion as well as the processing delays. It is computed as a function of the round trip time by the californium client Cf as the time difference between the time of the successful response reception for a particular request and the time instant at which the request is sent. It is the sum of 2 * max latency and the processing delay.

The plot below illustrates a sort of almost linear relationship for the different hop scenarios,

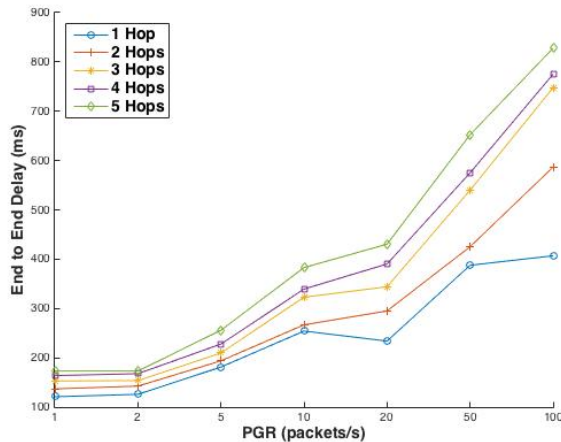


Fig. 4. Plot of end to end delay for various motes at 1 to 5 hops against the rate of packet generation.

C. Packet loss

The GET requests sent by the CoAP client in the simulation scenario, which are not acknowledged determine the

packet loss. Packet losses occur in environments where there is atmospheric noise present, and this can be simulated in Cooja by varying the Tx/Rx ratio (in %). The CoAP retransmission factor is set to 0 and the statistics are as shown in the table below,

TABLE V

PACKET LOSS (IN %) AS A FUNCTION OF THE PACKET GENERATION RATE PGR (IN PACKETS/SECOND)

PGR	Packet loss in %				
	1-hop	2-hops	3-hops	4-hops	5-hops
1	1.8	2.7	6.7	9.8	11.7
2	3.7	5.8	11.6	16.4	21.4
5	4.4	9	14.9	23.3	26.7
10	3.1	7.2	18.7	22.9	25.3
20	8.1	15.4	20.5	25.6	29.7
50	9.6	20.7	27.5	32.7	33.9
100	14.2	22.3	31.8	33.9	35.3

This parameter is obtained directly from the CoAP client as the no of unsuccessful packet transmissions. The figure below illustrates this variation,

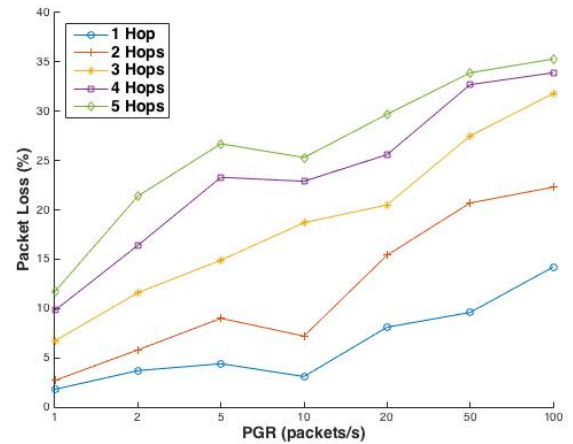


Fig. 5. Plot of packet loss for various motes at 1 to 5 hops against the rate of packet generation.

The acceptable limits for these parameters is not definite and is a trade-off between energy consumption and performance. This compensation depends upon the application in mind and its performance requirements.

VI. CONCLUSIONS AND FUTURE WORK

6LoWPAN potentially solves the interoperability problem. It is based on IPv6 and operates in a fully asynchronous way. In the case with ZigBee, bridging between ZigBee and non ZigBee needs more complex gateways unlike 6LoWPAN. The interoperability prowess of 6LoWPAN emerges as the single biggest advantage over Zigbee as the simulation demonstrates the communication between the motes in the network and applications outside the network using CoAP, employing nothing else than a normal router. The network operation is also demonstrated for heterogeneous networks in

different address spaces, like in this case the CoAP client Cf is in IPv4 whereas the CoAP servers reside in the simulation in an IPv6 address space.

Unlike HTTP, CoAP is built on top of UDP and has a compact packet overhead. This has a considerable impact on the energy consumption and response time of the motes. This is quantified by the simulation scenario where a CoAP server run on a temperature and humidity sensor is sent GET requests which are 13 bytes whereas a CoAP response is just 17 bytes. In addition, we demonstrated that CoAP is not particularly sensitive to the increase of client request generation time. This is an important advantage because the application of flood detection requires frequent access to WSN data. CoAP also allows the tuning of parameters like the CoAP re-transmission factor, which can be customized to the use case and application scenario. CoAP with its congestion control mechanism based on the request time out (RTO) and re-transmission factor also ensures reliability, we observed that for a default CoAP re-transmission factor of 4, even with a packet generation rate of 1000 packets/sec and the scenario of 5 hops, negligible packet losses occurred. This is an important advantage of CoAP, which implements congestion control mechanisms over the unreliable transport layer protocol UDP.

The stack simulated in this work is inter-operable as it can be connected to different mote platforms and doesn't require proprietary gateways to connect to the internet. Cooja enables emulation with real hardware motes which makes the network simple and re-usable, and security is ensured by the inherent IPsec mechanism derived from IPv6. Based on these simulations, one can design the physical deployment effectively in terms of how many hops are supportable and positioning of base station/router. In addition throughput and latency requirements for the real-time applications can be analyzed. All these metrics make the suggested stack an optimal fit for real time applications like flood detection.

In terms of future work, 6LoWPAN adopts a mesh topology and uses a routing algorithm which does not take care of the sleeping mode thus requiring approaches such as low-power listening for energy saving purpose. Such energy saving modes will be an engaging area of research. In addition to this, computing the re-transmission time out (RTO) in CoAP is an interesting research question, which helps determine better congestion control mechanisms for such stacks. Finally, the hardware deployment of the stack simulated for a pilot test study of flood detection and a comparison of simulated and observed values of parameters would be a crucial validation of the proposition made.

ACKNOWLEDGEMENT

We would like to extend our gratitude to the Swedish Research Council (under grant 2014-4251) for the funding and the provision of resources to conduct this research. Special thanks to the Erasmus Mundus program PERCCOM (pervasive computing and communications for sustainable development) as an enabler for this research work.

REFERENCES

- [1] Mulligan, G., "The 6LoWPAN architecture", Proceedings of the 4th workshop on Embedded networked sensors (EmNets07), pp. 78-82, ACM, 2007.
- [2] Astorga, J., Aguado, M., Jacob, E., and Matias, J., "Evaluating the Viability of IPv6-Based Communications Over IEEE 802.15.4 Networks Using 6LoWPAN", 2014.
- [3] Akyildiz, I. and Vuran, M., *Wireless Sensor Networks*, Wiley, 2010.
- [4] Montenegro, G., Kushalnagar, N., Hui, J., and Culler, D., "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", IETF RFC4944, 2007.
- [5] Winter, T.E., Thubert, P.E., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J.P., and Alexander, R., "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", IETF RFC6550, 2012.
- [6] Shelby, Z., Hartke, K., and Bormann, C., "The Constrained Application Protocol (CoAP)", IETF RFC7252, 2014.
- [7] Shelby, Z. and Bormann, C., "The wireless embedded internet", Wiley, 2009.
- [8] Contiki: The open source operating system for the internet of things. Available - <http://www.contiki-os.org/index.html>. Retrieved on December 9, 2015.
- [9] White paper, Zigbee Vs 6LoWPAN for sensor networks, LSR. Available - <https://www.lsr.com/white-papers/zigbee-vs-6lowpan-for-sensor-networks>. Retrieved on December 17, 2015.
- [10] White Paper, Thread stack fundamentals, v2.0, The THREAD group. Available - <http://www.threadgroup.org/>. Retrieved on January 3, 2016.
- [11] Kovatsch, M., Lanter, M. and Shelby, Z., "Californium: Scalable Cloud Services for the Internet of Things with CoAP", 2014 International Conference on the Internet of Things, IEEE, 2014. Available - <http://www.eclipse.org/californium/>. Retrieved on January 4, 2016.
- [12] Mehmood, W., Hassan, A., Tabish, R., Touati, F., Mnaouer, A.B. and Gaabab, B., "Performance Evaluation of 6LoWPAN Based Networks for Ubiquitous Health Monitoring System", Proceedings of the 13th International Conference on Wireless Networks (ICWN'14), pp. 286291, 2014.
- [13] Robles, T., Alcarria, R., Martn, D., Navarro, M., Calero, R., Iglesias, S. and Lopez, M., "An IoT based reference architecture for smart water management processes", Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), pp. 4-23, 2014.
- [14] Pokri, B., Kro, S., Draji, D., Pokri, M., Rajs, V., Mihajlovi, ., Kneevi, P. and Jovanovic, D., "Augmented Reality Enabled IoT Services for Environmental Monitoring Utilising Serious Gaming Concept", Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), pp. 37-55, 2015.
- [15] Andersson, K. and Hossain, M.S., "Smart Risk Assessment Systems using Belief-rule-based DSS and WSN Technologies", Proceedings of 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace and Electronic Systems (VITAE), pp. 1-5, IEEE, 2014.
- [16] Andersson, K. and Hossain, M.S., "Heterogeneous Wireless Sensor Networks for Flood Prediction Decision Support Systems, Proceedings of 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 133-137, IEEE, 2015.
- [17] Ma, X. and Luo, W., "The analysis of 6LowPAN technology", Proceedings of Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIA 08), pp. 963-966, IEEE, 2008.
- [18] Saad, L.B., Chauvenet, C. and Tourancheau, B., "Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies", Proceedings of International Conference on Sensor Technologies and Applications (SENSORCOMM 2011), 2011.