HAL
archives-ouvertes.fr

RPL Routing Protocol a case study: Precision agriculture

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

▶ To cite this version:

HAL Id: hal-00681319

https://hal.archives-ouvertes.fr/hal-00681319

Submitted on 21 Mar 2012

RPL Routing Protocol a Case Study: Precision Agriculture

Yibo CHEN[1], Jean-Pierre CHANET[1], Kun Mean HOU[2]

[1] UR TSCF
IRSTEA, Cemagref
Clermont-Ferrand, France
yibo.chen@cemagref.fr, jean-pierre.chanet@cemagref.fr

[2] LIMOS Laboratory UMR 6158 CNRS
Blaise Pascal University Clermont-Ferrand II
Clermont-Ferrand, France
kun-mean.hou@isima.fr

Abstract—The routing protocol for low power and lossy network (RPL) was designed in the ROLL working group at IETF since the year of 2008. Until the latest version of draft 19 released, this protocol algorithms and its four application scenario, such as home automation, industrial control, urban environment and building automation, have been nearly grounded. However, it is still very difficult to find effective approaches to simulate and evaluate RPL's behavior and other extensions of its application. In this paper, first we provide a brief presentation of the RPL protocol including two case studies ContikiRPL and TinyRPL, and an initial simulation experiment results obtained from the RPL capable COOJA simulator and its developed module. Second we then focus on the utilization of this protocol in the precision agriculture area and propose our dedicated instances hybrid network architecture to meet the specific requirement of this application. As a conclusion, we summarized our ongoing work and future solutions of the current technology issues.

Keywords - IoT, RPL, Network simulation, Cooja, precision agriculture

INTRODUCTION

In recent few years, Internet of Things (IoT) gradually became a very popular and hot topic in the area of Wireless Sensor Network (WSN). For enabling the convergence of WSN with the IP world and the connectivity of smart objects to the Internet, most of the core technology studies have been conducted by IETF Working Group IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) who proposed one RFC [1] to make IPv6 packets to be carried over IEEE 802.15.4 feasible. Another IETF Working Group Routing over Low power and Lossy networks (ROLL) investigated a routing protocol named IPv6 Routing Protocol for Low power and Lossy Networks (RPL). The reason why RPL was proposed is because none of the existing known protocols such as AODV, OLSR or OSPF could meet the specification of Low power and Lossy Networks (LLN) [2]. The RPL protocol targets large scale WSN and supports a variety of applications e.g., industrial, urban, home and buildings automation or smart grid [3]. ROLL's charter ensures that the designed routing protocol should operate over a variety of different link layers, including but not limited to low power WSN. This feature requires the RPL protocol to support heterogeneity in LLN, for instance with the use of WSN in the domain of Precision Agriculture (PA) technologies.

In this article, we evaluate the basic behavior of the RPL routing protocol with Cooja. This paper is organized as follows. In Section 2, related work is reviewed, such as the simulation of RPL and two existing systems. Section 3 presents the RPL protocol concepts and its multiple instance simulation. Section 4 describes our proposal use case of PA application based on the simulation of RPL on Cooja. In Section 5 and 6 we conclude the paper and discuss our ongoing work.

RELATED WORK

A. The simulation studies of RPL

Since the year of 2010, several RPL simulations and implementations have been released with the development of grounded RPL. In the IETF draft [4], the performance of RPL is evaluated through several consideration of routing metrics in both of the simulation and real world deployment scenarios, such as path quality, routing table size, loss of connectivity, etc. The simulator used in this study is OMNET++/Castalia [5] which is also adopted to analyze the stability delays of RPL in [6]. The researchers from INRIA have performed some simulations about the studies of multipoint-to-point (MP2P) performance of RPL, suggested broadcast mechanisms as well as sink mobility management on NS-2 and WSNet in [7][8][9]. In paper [10][11] SICS proposed a framework for RPL simulation, experimentation and evaluation. This framework consists of three core components: the Contiki operating system [12], the COOJA/MSPSim [11] simulator and the ContikiRPL implementation [10] (Figure 1). At Berkeley and Johns Hopkins universities, an open-source implementation of RPL in BLIP-2.0 for TinyOS 2.x [13] is under development in the OpenWSN project (Figure 2). There are also several other RPL industrial non-open source implementations. In this article, we will focus on using Cooja/Contiki platform to present our incipient proposal.

Despite of the fact that a lot of studies have been conducted to evaluate the performance, feasibility and several application fields of RPL protocol, to our knowledge until recently, there has been no real open source released simulation and constructive proposal of RPL in the case of PA software platform.
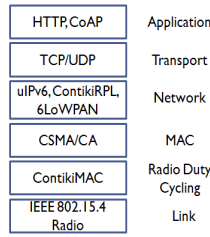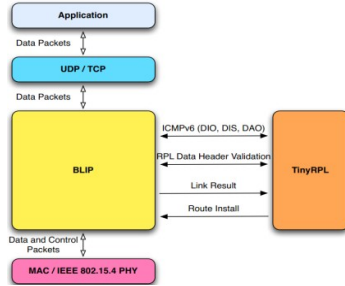
Figure 1.   ContikiRPL structure

Figure 2.   TinyRPL structure

B.   Precision agriculture

Thanks to the rapid advanced in the domain of WSN and miniaturization of the sensor boards, PA started emerging as new trends in the agricultural sector in the past few years. Generally speaking, PA system concentrates on providing the ways for observing, assessing and controlling agricultural production process, and covers a wide range from herd management to filed crop production [14]. In this article, we mainly discuss about the facet of PA focused on site-specific crop management. This surrounds several different aspects, such as monitoring soil, crop and climate in a field which is separated by some complete parcels; providing a decision support system (DSS) for possible treatments analysis, for filed-wide or specific parcel (Figure 3); and the methods for taking differential actions, for instance, adjusting in real-time an operation such as fertilizer, lime and pesticide utilization, tillage, or sowing rate [15].

Figure 3.   A croup filed seperated into parcels

STATE-OF-THE-ART

In this section, we presented the RPL protocol, a simulator survey about NS-3, OMNet++ and Cooja, as well as our initial RPL simulation on Cooja.

C.   Key concepts of the RPL protocol

RPL is a routing protocol designed for LLN with the expectation of joining to thousands of nodes network. It supports three traffic patterns: MP2P, point-to-multipoint (P2MP) and point-to-point (P2P). The basic idea of RPL is that the high degree of autonomy in the nodes level through building a Destination Oriented DAGs (DODAGs) rooted towards one sink (DAG ROOT) identified by a unique identifier DODAGID. The DODAGs can be optimized according to an Objective Function (OF) based on differential application specifications and identified by an Objective Code Point (OCP), which indicates the dynamic constraints and the metrics such as hop count, latency, expected transmission count, parents' selection, energy etc. [16]. A rank number is assigned to each node which can be used to determine its relative position and distance to the root in the DODAG.

A set of multiple DODAGs can be in a RPL INSTANCE which is a very important concept in RPL. A node can be a member of multiple RPL INSTANCEs but can belong to at most one DODAG per DAG INSTANCE. DODAG Information Object (DIO) messages are used to construct and maintain the upwards routes of the DODAG with the information, such as RPL INSTANCE, DODAGID, RANK and DODAGVersionNumber. A trickle timer [17] of RPL can regulate the transmission of DIO messages and help to eliminate redundant control messages. Each node has to monitor its neighbors' DIO messages before joining a DODAG. Then, it selects a DODAG parent set from its neighbors according to the latency they advertise, OF and computes its RANK. Destination Advertisement Object (DAO) messages are used to maintain downward routes by selecting the preferred parent with lower rank and sending a packet to the DAG ROOT through the parents set. Another common message is DODAG Information Solicitation (DIS) that can be sent by any node in RPL to solicit DIO messages from its neighborhoods for update routing information.

RPL has two mechanisms to repair the topology of the DODAG, one is to avoid the loops and allow nodes to join or rejoin a new position and another one is called global repair [18]. Global repair is an operation mode that the DODAG ROOT increments the DODAGVersionNumber to create a new DODAGVersion. Another mechanism is local repair which can allow the DODAG repaired within the DODAG Version. For example, the node can detach from the DODAG, advertise a rank of INFINITE RANK to inform its sub-DODAG, and finally re-attach to the original or a brand-new DODAG.

The security of RPL is critical in smart object networks but implementation complexity and size is a core concern for LLNs such that it may be economically or physically impossible to include sophisticated security provisions in a RPL implementation. Thus, it is an optional extension in RPL features because we can utilize link-layer or other security mechanisms to meet our security requirements in many deployments other than the utilization of security in RPL. If we really make this feature available, RPL nodes can operate in three security modes: unsecured; pre-installed; authenticated [19]. Each RPL message has a secure variant. The level of security and the algorithms in use are indicated in the protocol messages. The secure variants provide integrity and replay protection and confidentiality and delay protection as an added option.

RPL cannot be separated with specific application which may influence the objective function of this protocol. However, after the release of OF0, this default mechanism can meet most of the required situation including our use case in the below section. The interoperability of RPL is another hot spot. ContikiRPL and TinyRPL have been proved that they can work together [20] with an acceptable performance. Except the interoperability between systems, RPL protocol under the "non-storing" operation mode and ZigBee protocol have been tested by IPSO and Zigbee/IP alliances [21].

D. A survey of available RPL simulators

This section provides a quick overview of the general features of NS-3, OMNet++ and Cooja simulator. The more complete documentations of these simulators are available on their websites. Table I compares the technical features of these three simulators.

Table 1: Key features of the RPL available simulator NS-3, OMNeT++ and Cooja

Simulator	NS-3	Castalia OMNet++	Cooja/MPSim
WSN platforms	n/a	n/a	Tmote Sky, ESB, MicaZ
GUI	MSF, NetAnim, Ns3Generator	topology definition, result analysis and visualization	Friendly GUI
Wireless channel	802.11, YANS, Free space/two-ray/shadowing /small-scale path loss model	Lognormal shadowing, experimentally measured, path loss map, packet reception rates map, temporal variation, unit disk	Multipath ray tracing with support for attenuating for obstacles, unit disk, directed graph
PHY layer	LTE, LAN, Mobility module	CC1100, CC2420	CC2420, TR1001
MAC layer	802.11, CSMA/CA(CD), QoS, mesh, bridge	TMAC, SMAC, Tunable MAC	CSMA/CA, TDMA, X-MAC, LPP, NullMAC, ContikiMAC, SicslowMAC
Network layer	AODV, dsdv, olsr, internet	Simple tree, multi path rings	RPL, AODV
Transport layer	UDP, TCP	None	UDP(not complete), TCP
Energy model	Yes	Yes	Yes

NS-3 is not the new version of the well-know NS-2 simulator that is still one of the most popular network simulator in the WSN research community. However, NS-2 has so many inconvenient technical issues and limitations, and these problems have motivated the development of NS-3 simulator which can provide better modularity and performance to replace the original simulator. The key concepts behind it are modularity, re-usability and extensibility. NS-3 is a discrete event-driven network simulator which is under the GNU GPLv2 license [22].

NS-3 is written in pure C++ unlike its predecessor, thus the developers could benefit from several useful mechanisms, such as the use of templates, smart pointer and design patterns. It is worth to talk about its tracing file function, especially it can generate both of the standard text files and simulation output in the PCAP format. They all can contain information on various events occurred during the simulation (e.g. packets sent or received), but the PCAP format can be used by the tcpdump and Wireshark tools to represent the frames exchanging over a live network which can help the developers to analyze simulation results just as live network captures.

In the internet IETF draft [4], the RPL performance is evaluated by several RPL simulations and implementations recently. The simulator used in this study is OMNET++/ Castalia [5]. The RPL's stability delays are analyzed on OMNET++ [6] also. This simulator is a C++ based discrete event simulator for modeling communication networks, multiprocessors and other distributed or parallel systems. It represents a framework approach, since its first release, its simulation models have been developed for most of the network area (e.g. wireless and ad-hoc networks, sensor networks, IP and IPv6 networks and etc.). Some of the simulation models are ported from real world protocol implementations; others were realized directly for OMNET++. In addition to university laboratory research and non-profit research organization, some companies like IBM, Intel and Cisco are also using OMNET++ successfully in commercial projects.

The last simulator which we introduced in this section is Cooja [23]. After the comparison of existing simulators' technical features, our consideration of this aspect is to find open-source simulators for the easier and low-cost development of our experimental research platforms. Note that when the study of this paper was started, there was still no real available RPL open source implementation in a simulator.

Thus we chose to simulate and form our RPL routing behavior simulation in Cooja which provides a well-done connection between MSPsim [24] hardware emulator and Contiki OS. Cooja is able to run Contiki program directly without any modification, in fact, the ContikiRPL module can execute in this simulation environment which made it time-saving for our initial simulation work. Furthermore, it provides a effective GUI that let Cooja be a extraordinary easy to use and start tool, except this, we can build our own application level prototype in this simple platform with the help of several useful plug-ins, such as the interface vision of the simulated network and MRM (Multi-path Ray tracing Medium) obstacle

module. However, the limitation of Cooja is unavoidable, especially its adherence to Contiki OS architecture and the lack of optimization for the large scale topology.

E. Basic RPL routing behavior simulation in Cooja

In this chapter, one RPL instance building simulation in Cooja was evaluated. Different network sized scenarios, such as small, medium, large and huge network, were discussed. This study was about how the network converged and stabilized using the RPL protocol and OF0 implementation of ContikiRPL. The simulation scripts consists of RPL sender node and LLN Border Router (LBR) programs which are emulated as Tmote sky nodes and derived from Cooja and uIPv6 module including UDP, ICMPv6, IPv6, SICSLoWPAN and Rime of the Contiki kernel [10][11].

With the help of the CollectView tool [25] provided by Cooja, the following metrics could be observed: The time taken to find the first source-destination pair in the whole network (in seconds); The time taken for the network to fully converge when all nodes join the network tree (in seconds); The time taken for the network to fully stabilize after convergence, the time taken for the Estimated Transmission Count (ETX) value for each node to reduce to 1.0 (in seconds). After three sets of experiments organized by different size of network, such as 10, 20, 30 RPL nodes with one LBR, we got a serial of results in the same random seed and same size of space. The key information and observations are summarized in the table below. This simple simulation cannot provide accurate results because of the limitation of Cooja, but we still find some conclusion, one is the first two metrics didn't change in all the experiments which means these times are irrespective of the number of nodes. Another conclusion is, the third metric obviously depends on the number of nodes and the larger network will take longer time for the whole network to stabilize.

Table 2: Key information and observation on Cooja simulation

Metric observed	Time approximated
Time to find the first src-dst pair	50-60 s
Time to complete convergence	115 s
Time to achieve complete stable network	1050-1450 s

In addition, we tried to simulate a large scale network including 200 RPL nodes and configurable number of LBRs in one square kilometer. Before the simulation starts, we arrange these RPL nodes and one LBR in random position. However, one single LBR is difficult to support the network with big number of nodes due to the constraints of memory, calculating power and wireless transmission distance. After 8 LBRs were appended in the network topology (Figure 4), all of the RPL nodes can join their neighboring LBR and build the DODAGs (Figure 5).

In fact, this RPL simulation is not a well-rounded and needs much more improvement, especially the optimization of its application layer and traffic flow tracing function. Furthermore, we need another simulator to execute the same simulation and verify whether the results are completely credible. Thus, the simulation on NS-3 was put on the agenda of our next step work. However, these simple simulations on Cooja still brought us a global view of RPL, as well as proved Cooja/Contiki is an available platform for RPL protocol
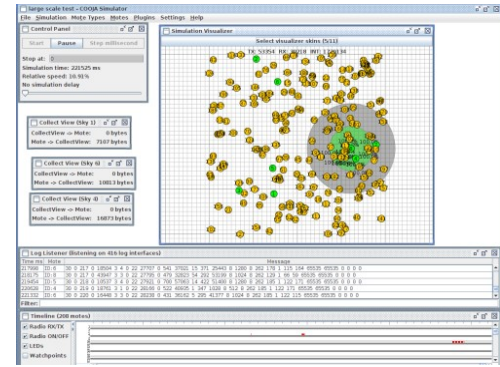
simulation.

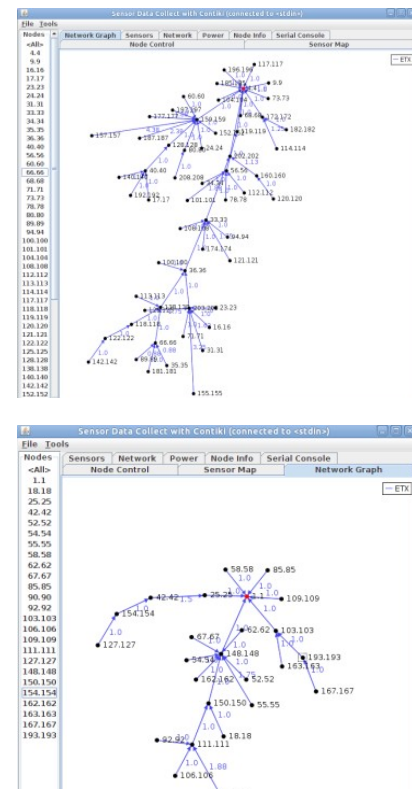Figure 4.   200 RPL nodes simulation on Cooja

Figure 5.   Network graph shows building of the DODAG

RPL AND PRECISION AGRICULTURE PLATFORM

In this chapter, we will discuss the precision agriculture application or agricultural area LLN as a use case of RPL protocol.

Precision Farming and WSN applications combine an exciting new topic of research that will greatly improve quality in agricultural production, water management and will have dramatic reduction in cost needed [26]. Using a network of strategically placed sensors, measurement data such as temperature, humidity, and soil moisture can be automatically monitored. For example, one sensor network is able to provide precise information about crops in real time, and help to reduce water, energy, and pesticide usage, and the most important is to avoid environment pollution through the utilization of

measurable fertilizer, pesticide, etc [27].

Furthermore, the ease of deployment and system maintenance open the door of the adoption of WSN and IoT technologies in precision farming. In addition the data on the field crop condition can be saved by sensor tags, which can be used in the supply chain management [27]. Using the proposed methodology of this chapter, in finding the optimal and simplified sensor architecture, we contrive to lower implementation cost and thus make our use case a more appealing solution for all kinds of fields and cultivations.

Figure 6.   Use case of automated vineyard for LLN

For a better explanation, we presented an application example called automated vineyard (Figure 6) [28]. It was supposed in a 20-acre vineyard with 8 parcels of land, 10 RPL nodes can be placed within each parcel to provide data collection (temperature and soil moisture) function. The basic idea of precision agriculture in this demo is to adopt different plan of irrigation and fertilization for these 8 parcels of farm field based on their specific soil and yield features.

In this LLN scenario, the dominant parameters [28] in this use case can be listed in the table below.

Table 3: Dominant parameters in the use case of automated vineyard

Parameter	Note
Pre-planned deployment	Considering the harsh environment, due to high exposure to water, soil, dust, in dynamic environments of moving people and machinery, with growing crop and foliage
Mobility	All static
Network size	Medium to large, low to medium density
Power source	All nodes are battery-powered except the LBRs
Security level	Light-weight security or a simple shared encryption key management can be adopted depending on this application
Multi-hop communication	RPL forwarding, and Ethernet connection
connectivity	Intermittent (existing many sleeping nodes)
Traffic pattern	MP2P/P2MP, P2P actuator triggering
Other issues	Time synchronization among the sensors; low frequent of the traffic interval (30-60 mins)

Except the common RPL end device and router, this static scenario consists of one or more fixed LBR which are mains-powered and have a high-bandwidth connection to a backbone link, which might be positioned in a control center, or connect to the data server through Internet (with predefined forward management information to a central data aggregation point). This device will be strategically located at the border of vineyard parcels, acting as data sinks (Figure 7). The LBRs should implement the 6LoWPAN adaptation mechanism [29],

IPv6 ND protocol and of course full version of Ethernet protocol stack and RPL routing protocol.
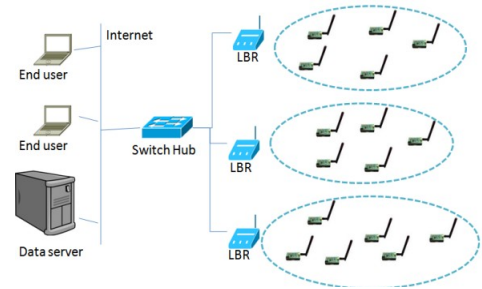

Figure 7.   PA platform architecture

The LLN nodes will be spread around these LBRs with a more energy-considering ND (including basic bootstrapping and address assignment) [30] and RPL routing. For enhanced energy efficiency, all LLN nodes are in periodic sleep state. Thus, the LBRs need to aware of sudden events from the leaf RPL nodes. Context-awareness [31], node identification and data collection on the application level of LBR system are necessary.

The new concepts of Cooja are very attractive and endow this simulator with feature of cross-level sensor network simulation. Going on the major premise of adaptation of Contiki OS, we are able to easily organize our designed application layer on this simplified platform. In this article, we may imagine that the above application is developed initially with the RPL, IPv6, and 6LoWPAN modules of Cooja, to implement specific programs for RPL end device, router and border router. Even under the consideration of wireless communication encumbrance of windbreak or plant shelves, the MRM plug-in [32] in Cooja can make contribute to provide a solution of this issue (Figure 8).
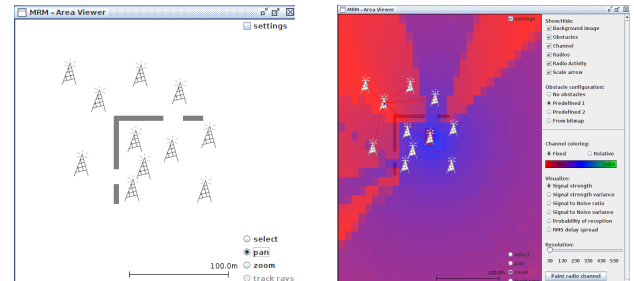

Figure 8.   MRM obstacle module in Cooja

This method can be seen as a solution to avoid the development difficulties brought by the embedded system software programming and debugging, and its low cost and time saving specialties should not be ignored when the developers plan to build any application systems with the latest RPL-based network stack.

SUMMARY AND OUTLOOK

Although the IP support in WSNs is nowadays a reality, the large scale applications in real world and simulation scenarios are still scarce. This paper discusses an IoT system application as a case study in PA platform using RPL routing protocol. We have introduced the basic RPL concepts, current existent RPL

implementations and simulation. We also tried to learn and gain more experience from the utilization of Cooja/Contiki platform, and proposed our idea of protocol stack simulation and a low-cost simplified PA platform.

Finally, in spite of some drawbacks, the IoT technology is nowadays quite grounded and we believe that in the next coming few years, most of daily living objects may be connected to internet. The RPL routing protocol is motivated by the need to support the upcoming automated metering infrastructure, agricultural domain and home area networking applications. Energized by a decade of research in WSN, the protocols provided by ROLL and 6LoWPAN working group will change the way of metering, monitoring, control and diagnosing. In general, although these efforts are only the beginning, it will be important to recognize that the presence of IEFT protocol stack for LLN provides plentiful opportunities for profound innovations in the related fields, which should be a fertile ground for continued research in versatile application systems.

References

[1] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15.4 networks," IETF, RFC 4944, 2007..

[2] P. Levis, A. Tavakoli, and S. Dawson-Haggerty, "Overview of existing routing protocols for low power and lossy networks," Draft, 2009.

[3] T. Winter and P. Thubert, "RPL: IPv6 routing protocol for low power and lossy networks," IETF, Draft, 2010.

[4] J. Tripathi, J. de Oliveira, and J. Vasseur, "Performance Evaluation of Routing Protocol for Low Power and Lossy Networks (RPL)," IETF, Draft, January 2011.

[5] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in Int Conf on Simulation Tools (ICST). ICST, 2008.

[6] M. Nuvolone, "Stability analysis of the delays of the routing protocol over low power and lossy networks," Master's thesis, Sweden Royal Institute of Technology, 2010.

[7] T. H. Clausen and U. Herberg, "Comparative Study of RPL-Enabled Optimized Broadcast in Wireless Sensor Networks," INRIA, Tech. Rep. 7296, 2010.

[8] T. H. Clausen and U. Herberg, "Multipoint-to-Point and Broadcast in RPL," INRIA, Tech. Rep. 7244, 2010.

[9] Leila Ben Saad, Cedric Chauvenet, "Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies", INRIA, SENSORCOMM 2011.

[10] N. Tsiftes, J. Eriksson, and A. Dunkels, "Low-power wireless IPv6 routing with ContikiRPL," in Int Conf on Information Processing in Sensor Networks. ACM/IEEE, 2010.

[11] N. Tsiftes, J. Eriksson, N. Finne, F.¨Osterlind, J. H oglund, and A. Dunkels, "A framework for low-power ipv6 routing simu-lation, experimentation, and evaluation," SIGCOMM Comput. Commun. Rev., vol. 40, pp. 479–480, August 2010.

[12] A. Dunkels et al., "Contiki-a lightweight and flexible operat-ing system for tiny networked sensors," 2004.

[13] M. Afanasyev, D. O'Rourke, B. Kusy, and W. Hu, "Het-erogeneous Traffic Performance Comparison for 6LoWPAN Enabled Low-Power Transceivers," 2010.

[14] Xiangyu Hu, Songrong Qian, "IoT application system with crop growth models in facility agriculture", School of Information Sci. & Eng. Fudan University.

[15] Aline Baggio, "Wireless sensor networks in precision agriculture", Delft University of Technology – The Netherlands.

[16] J. Vasseur, M. Kim, and K. Pister, "Routing metrics used for path calculation in low power and lossy networks."

[17] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," IETF, Draft, 2010.

[18] Tieu Tuan hao, "An overview of draft-ietf-roll-rpl-12", DCN lab.

[19] JP Vasseur, "IoT Workshop RPL Tutorial", CISCO, April 2011.

[20] JeongGil Ko, Joakim Eriksson, Nicolas Tsiftes, etc, "ContikiRPL and TinyRPL: Happy Together", IPSN' 11, Aprial 12-14 2011,SICS.

[21] JP Vasseur, Navneet Agarwal, Jonathan Hui, etc, "RPL: the IP routing protocol designed for low power and lossy networks – internet protocol for smart objects(IPSO) Alliance", April 2011.

[22] http://www.nsnam.org/docs/release/3.13/manual/singlehtml/index.html

[23] Fredrik ¨Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, Thiemo Voigt, Demo Abstract: Cross-level Simulation in COOJA, Swedish Institute of Computer Science Email: {fros,adam,joakime,nfi,thiemo}@sics.se

[24] J. Eriksson, A. Dunkels, N. Finne, F. Osterlind, and T. Voigt, "MSPsim-an extensible simulator for MSP430-equipped sen-sor boards." 2007.

[25] http://www.sics.se/contiki/wiki/index.php/Using_CollectView_in_Cooja

[26] Alberto Camilli, Carlos E. Cugnasca, etc, "From wireless sensors to field mapping: Anatomy of an application for precision agriculture", Computer and Digital Systems Engineering Department, São Paulo, Brazil, 15 October 2006.

[27] Ji-chun Zhao, Jun-fen Whang, etc, "The study and application of the IoT technology in agriculture", 978-1-4244-5540-9, 2010 IEEE.

[28] E.Kim, D.Kaspar, "Design and Application spaces for 6LoWPANs", draft-ietf-6lowpan-usecases-10, 6LoWPAN Working Group.

[29] N. Kushalnagar, G. Montenegro and C. Schumacher. IPv6 over Low-power wireless personal area networks (6LoWPAN): overview, assumptions, problem statement, and goals. RFC 4919.

[30] Simon Duquennoy, Niklas Wirström, Adam Dunkels, etc, "Leveraging IP for Sensor Network Deployment", Swedish Institute of Computer Science.

[31] Keun-Wang Lee and Si-Ho Cha, "Ontology-based context-aware management for wireless sensor networks", Communications in Computer and Information Science, 2011, Volume 214, 353-358, DOI: 10.1007/978-3-642-23321-0_55

[32] Carlo Alberto Boano and Kay Römer, "Demo Abstract: Realistic Simulation of Radio Interference in COOJA", Universität zu Lübeck, Germany, Swedish Institute of Computer Science, 2011.