A Comprehensive Review of Routing Protocols for Internet of Things

[1]Divya Sharma, [2]Sanjay Jain, [3]Reema Sharma

[1] *Senior Assistant Professor, Department of ECE, New Horizon College of Engineering*
[2] *Principal, CMR Institute of Technology*
[3] *Associate Professor Department of ECE, New Horizon College of Engineering*
[1]*divyas@newhorizonindia.edu,*[2]*dr_sanjay.jain@yahoo.com,* [3]*sharma80reema@gmail.com*

Abstract *— With the dawn of Internet of Things (IoT), intelligent device implanted in things can be connected at anytime, anywhere, with anything and anyone by possibly utilizing any network and any service. IoT networks are self-organizing and decentralized in nature which results in dynamic changes in node's position. Hence routing in IoT becomes crucial for successful delivery of the data. Further limited energy and processing capabilities of the connected things makes routing more challenging in IoT networks. This research work focuses on the study of some of the major routing algorithms designed for IoT network and their extensions. This review work also includes a comparison of these protocols on the basis of various performance metrics.*

Index Terms *— Internet of Things, Low power and Lossy networks (LLNs), RPL, LOADng, CTP LOADng-CTP*

I. INTRODUCTION

IoT is a future-facing development of the internet wherein objects and systems are embedded with sensors and computing power, with the intention of being able to communicate with each other. IoT finds its applications in diversified fields such as e-health, industrial manufacturing, energy, smart cities, agriculture, transportation sector, etc [1]. The need for advancement in IoT network protocols is mandated due to rise in number of physical things/objects connected to the Internet. The data collected from these devices should be made accessible to concerned parties, which can be smart phone, web services, cloud resource, etc. In IoT, the interaction between devices is done by using sensors and actuators. A sensor is used to collect, store and process the data. An actuator is used to maintain the change in the environment of a device. The processed data is stored at the remote server. Sometimes the storage and processing will be restricted to some available resources due to the limitations of size, energy consumption and computational capability of an IoT objects. The process of collecting, sharing and transmitting information will involve communication between nodes which act as both host and router with or without human intervention. Hence, due to weak processing and low power devices in IoT, new network algorithms or adjustments in the

existing ones are required to be compatible with the new type of network. Routing protocols for IoT network has been a rising research topic since last few years and the research area has witnessed many achievements. In most of the routing protocols, flooding of route request packets, results in increase in overhead. Based upon algorithms in routing it decide upon the finest route between the source and the destination node. Recently a lot of research work has been carried out focusing on design and implementation of routing algorithms to improve the efficiency and longevity of the network.

In [2], the authors have provided the detailed classification of routing protocols for IoT network. In their work authors have done an extensive survey of key routing protocols for IoT network and studied and compared their unique steering methods concurring with numerous measurements and parameters such as topology, power usage, mobility, query based, multipath, etc. Also various routing protocols for IoT network have been studied by the authors in [3] wherein they have compared the routing algorithms based on server technologies, security, data and storage management. In [4], the authors have focused on Routing protocol for low power and lossy networks (RPL) and have provided an exclusive classification of the same.

This survey paper focuses on the design and principle of operation of Collection tree routing protocol (CTP), RPL, Lightweight on-demand ad hoc distance-vector routing(LOADng), extend collection tree protocol (XCTP) and CTP variant of LOADng (LOADng-CTP). Further the performance of these routing algorithms on the basis of message delivery ratio, delay and control overhead has also been investigated. CTP routing protocol stands as a predecessor to RPL and was considered the de-facto routing standard for Tiny OS. RPL is a standard routing

protocol for IoT network by IETF in 2012. It is basically a distance vector routing protocol. Another standard distance vector based routing approach is the LOADng protocol. It is a lightweight variation of AODV for LLNs.

The contents of this paper are arranged as: Section II presents a detailed insight into the recently emerged routing protocols and their successors for IoT network. The results of the survey are discussed in section III. At the end, the conclusion is given in section IV of the paper.

II. ROUTING PROTOCOLS FOR IOT NETWORK

Several routing algorithms have been designed to perform efficiently in IoT network to overcome the various challenges posed by the constrained environment. This section discusses the standard and non-standard routing mechanisms, designed specifically for IoT scenarios. The key features, working principle and performance has been studied for CTP, RPL, LOADng, LOADng-CTP , CARP.

A. Collection Tree Protocol

CTP, a distance-vector routing protocol, discovers routes to one or few number of chosen sinks in a wireless sensor network [5]. A collection based routing protocol involves construction and maintenance of least cost trees to nodes which announce themselves as the root. The trees are built and maintained in such a way that the root node is positioned at the sink of the network. An approximation of the path cost to a collection point is maintained by each node. It is an address-free algorithm. The control messages for routing are broadcasted using adaptive beaconing mechanism.

Most distance-vector protocols, however, are impacted by issues such as routing loops that degrade their performance. In [5] the authors have focused on two principles to ensure that the routing protocol responds instantly as soon as the topology alters and at the same time continue to be robust and efficient. At the outset, routing topology is validated by verifying a data path by detecting any loops present in it. For achieving this, the local cost estimation of the transmitting node is included in all data packets. A probable routing loop can be detected upon reception of a packet to be forwarded from a node with equal or lesser distance to the destination. Instead of discarding this packet, the topology is repaired and the packet is forwarded normally. Data packets can thus identify routing inconsistencies accurately, even when the rate of control packets is low. Secondly, extension of Trickle algorithm, by the use of adaptive beaconing, allows dynamic adjustment to control traffic variations. This allows rapid detection of new nodes and failure recovery, as well as supporting lengthy beacon intervals in a stable network. Moreover, in CTP topology is formed with the help of a specific link-layer technology. CTP exhibit enhanced Packet Reception Ratio (PRR) and efficient energy consumption.

In CTP, the routes exist only towards the root node and hence it isn't capable of requesting for the data lost or using acknowledgements. In [10], it has been observed that only 96.5 % of a file with 512KB data is transmitted successfully using CTP. The accountability of malfunctioning of application lies with the fact that it is not possible to request the remaining fragments.

B. RPL Protocol:

In contrast to CTP, RPL [6] provides routes that allow exchange of feedback messages among the base station and the sensor nodes.

RPL, a distance-vector routing protocol, is based on source routing which operates on top of numerous link layer mechanisms. These layers can be characterized as potentially constrained and lossy in nature, and classically used along with highly constrained host or devices, such as PLC (Power Line Communication) or low-power wireless technologies. RPL is proactive in nature, well-suited for LLNs, which forms a directed acyclic graph (DAG) of the nodes present in the network [5]. Several DODAGs (Destination Oriented Acyclic graph) can be formed from a DAG. Each DODAG comprises of one sink and numerous sensor nodes. RPL involves periodic transmission of different control messages using trickle timer [7]. The process of forming DODAG topology is initiated by the root node of DODAG with the flow of DIO (DODAG Information Object) messages. Topology formation and maintenance is achieved by the use of DIO messages. The sensor node advances the DAO (DODAG Advertisement Object) message towards the sink node and they are used by the sink to update its network view. Any node that wants to join a DODAG topology uses DIS message (DODAG Information Solicitation). The flow of these control packets (DIO, DAO) messages are shown in Fig. 1. Upon reception of a DIO packet, a node checks whether packet is sent from a probable neighbor or not. After that it checks if or not the packet belongs to same DODAG of which the current node is an existing member. Next it verifies the rank of the current link, post which the packet is considered to be processed. Further, downward traffic can be forwarded in one of the two possible modes: Storing, in which before being moved down the packets must traverse up to a DODAG root, or Non-Storing, wherein the packet may possibly be forwarded by a mutual predecessor of the source, down towards the destination. Three basic traffic flows are supported by RPL namely Point-to-Point Traffic (P2P), Point-to-Multipoint Traffic (P2M), Multipoint-to-Point Traffic (M2P). It is well optimized for M2P as the routing

tables store upwards roots. It also gives rational support for P2MP in non-storing mode, apart from providing elementary features for P2P [8].

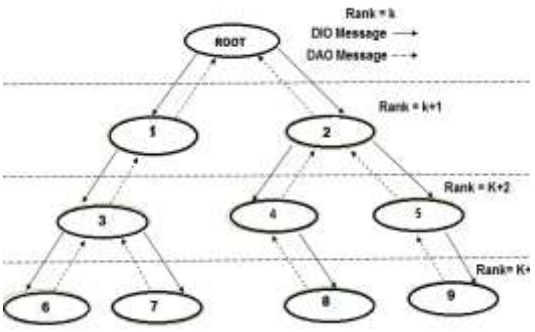


Fig. 1 Control Messages Flow in RPL

In RPL detection and repairing of a loop is done as soon as it is used but it does not assures a loop free route selection or hard delay convergence time [6]. The data packets should be buffered upon loop detection and it should trigger route repair process. However, it is unlikely to buffer all incoming packets throughout the route repair, thereby resulting in packet drop.

C. LOADng Protocol:

RPL, being a pro-active protocol, keeps a routing table to all potential destinations. Thus, it transmits control packets irrespective of whether data packets are present or not in the network which is liable for increased control overhead. This can be overcome by opting for a reactive routing approach wherein the route discovery to a particular destination commences only when a data packet arrives for transmission.

In [9] authors have proposed a standard routing protocol, LOADng for LLNs, which is a reactive routing protocol derived from Ad hoc On-Demand Distance-vector routing (AODV). The key features of this protocol are its simplicity and low memory storage requirements. There are four types of control messages [10] involved in the path discovery process:

i) Route Request (RREQ): On arrival of a data packet with a destination for which there is no valid route available in its cache, a router prepares a RREQ containing the destination address.

ii) Route Reply (RREP): Once RREQ is received and processed by a router then it verifies, whether or not, that destination is available in its cache If it is not present in its routing set then it prepares a RREP.

iii) Route Reply Acknowledgement (RREP-ACK): When a RREP is successfully received at a router then it notifies its neighboring RREP sender node with a RREP_ACK.

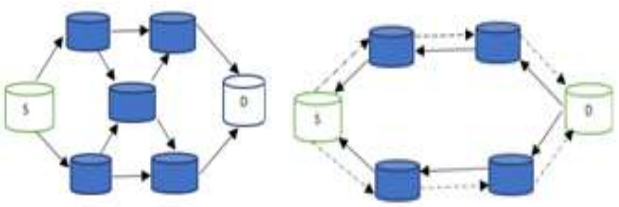
iv) Route Error (RERR): When a router detects a route failure to the destination then it sends a RERR message.

LOADng is an inheritance of AODV [11] which adopts its basic operations of generating and forwarding control packets (RREQ, RREP, RREP-ACK), as illustrated in Fig. 2, to find route to a particular destination. Node A initiates a RRREQ for finding a suitable path to destination node B (Fig. 2a). Intermediate nodes forward the RREQ after appending it's id in RREQ. Once the RREQ reaches the destination then node B prepares a RREP and send it to node A via the path by which it received the RREQ (Fig. 2b). Finally, the data packets are sent along that route towards the destination as shown in Fig. 2c.

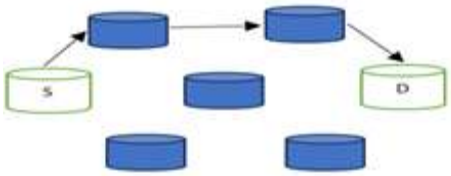
1. Route Discovery: It involves flooding of RREQ messages throughout the network. In contrast to AODV, LOADng protocol only the destination node, which is present in the RREQ, is allowed to generate RREPs. All other intermediate nodes intermediate LOADng routers are refrained from generating any RREPs, despite having active routes to the required destination. This eradicates intermediate RREPs, multiple RREPs, Destination Sequence Number and Source Sequence Number in RREQ messages thus resulting in decreased per-packed overhead.

2) Route Maintenance: LOADng employs end to end signaling upon router failure detection. The node prepares a RERR message and sends it along the route to the source of data packet using unicast transmission. The source node initiates a new route discovery process whenever it receives a RERR message.

A single distinctive monotonically increasing sequence number is included in RREQ/RREP messages that are generated by a given LOADng router [9]. Due to elimination of additional RREPs, the message size and the complexity of protocol operation is reduced which is certainly appropriate to low-power and memory constrained networks. The authors [12] have shown that LOADng exhibits lower control overhead and higher end-end delay as compared to RPL. However, end to end delay in LOADng increases since the data packets need to wait for finding route before being transmitted.



a. Flow of RREQ b.Flow of RREP and RREP-ACK



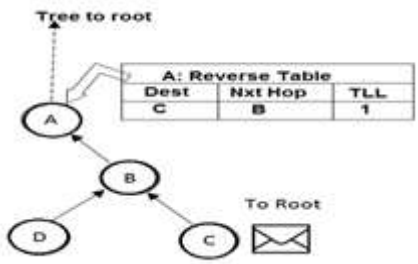
c. Data Flow

Fig.2 Control Packets' Flow in LOADng Routing Protocol

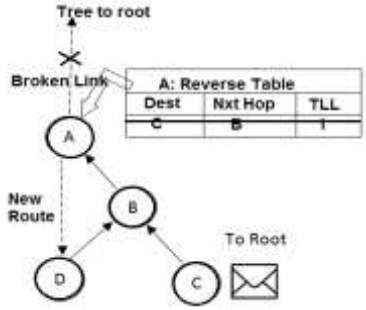
D. Extend Collection Tree Protocol

As discussed earlier, a routing tree is formed for data transmission from any sensor node to the root (sink) node in CTP routing. But it is limited to forward path finding and does not uncover path in reverse direction i.e. from the root to the sensor nodes. This route is required for sending acknowledgment packets to confirm reliable delivery of data packets. In [13], the authors have proposed eXtend Collection Tree Protocol (XCTP) which provides bi-directional paths between sensor nodes and the root of CTP using additional storage to hold reverse routes. In their work, the authors altered the CTP architecture by modifying protocols at data plane and control plane. In order to permit the transmission of data packet in reverse path, a 16 bits field for destination address was included in the data packet. An acknowledgement packet has also been introduced in XCTP for feedback. In XCTP, the control plane is accountable for manipulating the reverse table. Four basic operations Create, Read, Update, and Delete are implemented over the reverse table. Initially there are no entries in the table. A reverse route is identified whenever a sensor node transmits a packet to the root. The entries for 1-hop neighbors are not recorded in the reverse table because the information about 1-hop node neighbors is stored by the link estimator. Fig. 4a illustrates reverse flow table in XCTP. A reverse path is recorded when a data packet, sent from the source node C to the root, is intercepted by an intermediate node (excluding C's 1-hop neighbor). On receiving a packet, first a node determines whether it is a data or acknowledgement packet. Then it checks the destination address. If the node itself is the destination, then the packet is considered to have reached the destination. Otherwise the packet is progressed further by an intermediate node if the destination is one of its descendants. If the recipient does not exist in any of the routing tables then it is either forwarded to the root or dropped. In the former case, the root, being aware of the complete topology, can forward the packet.

Also, when a link failure or loop is detected, the data plane is modified in order to maintain the consistencies in the routes for reverse paths (Fig. 4c). Upon link failure, the nodes that were descendants earlier tend to become parents in the routing tree thereby forming loops. The data plane directs the control plane for deleting the corresponding reverse flow table entries by marking them in the reverse table.



a. Reverse flow table for XCTP routing



b. Outdated Reverse flow table owing to routing loop

Fig4. XCTP routing and route maintenance.

Thus, unlike CTP, which varies the beacon count with change in stability of the network, XCTP does not need additional beacons. Further in order to maintain the consistencies in the routes, upon detection of any link failure or loops, XCTP control plane modifies the data plane for reverse paths.

The results in [13] show the agility of XCTP even when simultaneous flows and faults are present in the network. Also XCTP is scalable and consumes less memory as compared to RPL. Control overhead is

significantly less in XCTP as compared to RPL as it does not transmit extra signals to build reverse routes.

E. Collection Tree Extension of LOADng Protocol (LOADng-CTP)

A large number of MAC layer collisions can be witnessed in LOADng routing because for every router it initiates route discovery. Thus it has a reduced packet/data delivery ratio, specifically when the topology is larger in size. The collection tree generated in CTP routing contains bi-directional paths between the data concentrator and other sensor nodes. On the other hand, LOADng routing protocol offers P2P paths amongst the devices present in the network. If both are deployed unanimously, the resultant provides both collection tree and point-to-point routes. LOADng-CTP [14] is an extension to LOADng, for building a "collection tree" in environments, restricted in terms of energy, memory, and processing power. This extension uses the mechanisms from LOADng, levies minimal overhead and complexity, and allows a deployment to competently maintain "sensor-to-root" traffic, evading issues associated with uni-directional links in the collection tree.

The authors [14] have introduced two flags namely RREQ Collection_Tree _Trigger and RREQ Collection Tree Build in a RREQ. Additionally, a HELLO message is included for verification of bidirectional links before admitting them to the collection tree. The collection tree formation involves i) Triggering of collection tree, ii) Discovery of bidirectional neighbor iii) Building collection tree iv) Formation of path from root to sensor. Initially the root generates and forwards the RREQ_Trigger. When a router receives RREQ_Trigger then it includes the address of the forwarding router in its neighbor set of the sender router and marks it as HEARD. If RREQ_TRIGGER is retransmitted provided it is not a duplicate one, and a HELLO message is generated. The router, upon receiving the HELLO packet, checks for its own address in it. If it is present then it updates its neighbor set with the address of the sender and sets the status as bidirectional else discards it silently. Thus, each router learns about what type of link, one-way (HEARD) or a two-way (SYM), does it share with its neighbor nodes.. Next the root generates a RREQ_BUILD, wherein on receiving it, the router whether it has been received over a bidirectional link or not. If it is received over a bidirectional link then RREQ_BUILD is said to indicate the shortest path to the root with next hop to the root as the previous hop in it. Thus, all the sensors present in the network can find a path to the root with the use of only bi-directional links via exchange of two control messages namely RREQ TRIGGER and RREQ BUILD. If the application requires transmission of data from root to sensor node then it should set RREP_REQUIRED flag to TRUE. In such cases, a RREP is sent as unicast message to the root with by the sensor node.

Route Maintenance in LOADng-CTP is done on per path basis in case of any path failure. Further in their work, the authors have proposed a smart route request scheme for route maintenance. In this scheme, if an intermediate router receives a RREQ for a destination, for which currently the route is unavailable, then RREQ is forwarded in a normal way. On the other hand, if the path to the root is available at the intermediate router, then it the RREQ is sent as unicast to the destination according to the routing table.

The routing overhead in LOADng-CTP is O(N) as compared to $O(N^2)$ for LOADng protocol [9]. LOADng-CTP and RPL exhibit similar delays as they have routes readily available. Control overhead is comparatively lesser in LOADng-CTP as compared to RPL and LOADng.

III. RESULTS

In this section, the key features of the routing protocols for IoT network have been summarized. Table 1 illustrates the significant characteristics of these protocols.

Further the performance of these protocols have been evaluated and summarized in Table 2. The performance parameters considered for evaluation are packet delivery ratio, end to end delay, control overhead and memory consumption.

CTP has high packet delivery ratio when the network is small [3] and it reduces with increase in the number of nodes. RPL, XCTP and LOADng-CTP have higher packet delivery ratio (almost 100%). Packet delivery ratio in LOADng drops as the number of nodes increases due to increased number of MAC layer collisions [15]. It is observed that CTP, RPL and XCTP routing protocols have higher control overhead and lower end-end delay due to their proactive nature. RPL makes use of control packets to establish reverse paths whereas XCTP utilizes data packets to accomplish the same. Also XCTP relies on TTL-based strategy for paths which are not utilized potentially and records on-demand reverse paths. This accounts for reduced the control overhead in XCTP as compared to RPL. The number of beacons per node is 0.9beacons/min for RPL whereas it's only 0.2beacons/min in case of XCTP [13]. On the other hand, LOADng and LOADng-CTP are reactive protocols and hence have reduced control overhead and higher end-end delay.

TABLE 1 SURVEY OF ROUTING PROTOCOLS FOR IoT

S. No.	Routing Protocol	Key Features
1	CTP	• Collection tree based distance-vector protocol. • Forms only upward routes towards the root. • Packet Delivery ratio is better for smaller networks. • Efficient energy consumption.
2	RPL	• Proactive distance-vector source based routing. • Both upward and downward routes are available. • Lower end to end delay and Increased control overhead due to proactive nature. • Path length is comparatively more as compared to LOADng routing.
3	LOADng	• Reactive protocol based on AODV routing approach. • Packet delivery ratio drops if there is growth in network size. • Memory consumption is less and delay is more due to proactive nature. • Reduced packet size results in high control overhead.
4	Xtend-CTP	• It is extension of CTP routing to provide both forward and reverse paths. • Additional memory is required for storing reverse paths. • It is fault-tolerant and scalable. • Initially, higher control overhead is observed but it gradually decreases and stabilizes over the period of time.
5	LOADng-CTP	• Collection based extension of LOADng. • Path finding between root and other nodes is done using bidirectional links. • Maintains route on per-path basis and hence exempts the need of re-building the entire collection tree. • Packet delivery ratio is high and almost independent of number of nodes. • Control overhead is O(N) and is comparatively lesser than RPL and LOADng.

TABLE 2.PERFORMANCE COMPARISON of IOT ROUTING PROTOCOLS

Routing Protocol	Packet Delivery Ratio	End-End Delay	Control Overhead
CTP	High (only for smaller networks)	Low	High (as compared to XCTP)
RPL	High	Low	High
LOADng	High (reduces with increase in number of nodes)	High (as compared to RPL)	Low (as compared to RPL)
XCTP	High	Low	High initially (during network

			start up). Reduces and becomes stable over the period of time.
LOADNG-CTP	High	Low	Low (as compared to RPL and LOADng)

IV. CONCLUSION

This paper has surveyed the routing protocols for IoT networks. The distinct characteristics of five such protocols have been highlighted. Further the performance of these protocols has also been compared. RPL is one of the standard protocols for LLN's having high PRR and low latency. However, due to high control overhead, RPL might not be suitable for network scenarios which are highly constrained In such networks LOADng protocol would win over RPL, provided the network is smaller in size and is suitable for non-real-time applications. XCTP depicts a balanced result for all the metrics. Amongst all, LOADng exhibits the most satisfactory performance and could be a preferred choice over the others.

Yet, there are several parameters, such as security, energy consumption, that must be considered for routing in IoT networks. Also none of these algorithms have emphasized upon the message urgency while route finding process. Thus, priority based routing approaches, for medical and other emergency applications, of IoT network need to be addressed in future.

REFERENCES

[1] Z. Kamal, A. Mohammed, E. Sayed, and A. Ahmed, "*Internet of Things Applications, Challenges and Related Future Technologies,*" WSN World Sci. News, vol. 67, no. 672, pp. 126–148, 2017.

[2] Ravi Kumar Poluru and Shaik Naseera "*A Literature Review on Routing Strategy in the Internet of Things,*" Journal of Engineering Science and Technology Review, 2017.

[3] Hanumat Prasad Alahari, Suresh Babu Yalavarthi,"*A Survey on Routing Protocols in Internet of Things*", International Journal of Computer Applications, Volume 160 – No 2, Feb 2017.

[4] A. Bhat, V Geetha, "*Survey of Routing Protocols in Internet of Things*", 7th International Symposium on Embedded Computing and System Design (ISED), 2017.

[5] Omprakash Gnawali, Rodrigo Fonseca, Kylie Jamieson David Moss, Philip Levis "*Collection Tree Protocol*", Sensys'09, Proceedings of the ACM.

[6] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur, "*RPL: IPv6 Routing Protocol for Low power and Lossy Networks*," March 2012, IETD RFC 6550.

[7] P. Levis, T. Clausen, J. Hui, O. Gnawali, J. Ko, "*RFC 6206:The Trickle Algorithm draft-ietf-roll-trickle-08,"* Internet Engineering Task Force (IETF) Request For Comments, Jan 2011.

[8] Martocci, Ed., P. De Mil, N. Riou,W. Vermeylen, "*Building Automation Routing Requirements in Low-Power and Lossy Networks*", IETF RFC5867, June 2010.

[9] T. Clausen, A Colin, J Yi, A. Niktash, Y Igarashi, H Satoh, and U Herberg, "*The LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng)*", July, 2014.

[10] J.V. Sobral, J.J. Rodrigues, K. Saleem, J. AI-Muhtadi *"Performance evaluation of loadng routing protocol in IoT p2p and mp2p applications*", Computer and Energy Science SpliTech) International Multidisciplinary Conference on, pp. 1-6, 2016.

[11] José V. V. Sobral, Joel José P. C. Rodrigues, Neeraj Kumar, Chunsheng Zhu, Raja Wasim Ahmad, "*Perrformance Evaluation of Routing Metrics in the LOADng Routing Protocol*", Journal Of Communications Software And Systems, Vol. 14, No. 2, June 2017.

[12] J. Yi, T. Clausen, Y. Igarashi, "*Evaluation of routing protocol for low power and lossy networks: LOADng and RPL*", Wireless Sensor (ICWISE) 2014 IEEE Conference,pp. 19-24,2014.

[13] P. Bruno . A. Santos and Marcos,. M. Vieira, and F. Luiz , "*eXtend Collection Tree Protocol*", IEEE Wireless Communications and NetworkingConference-Track4,2015.

[14] J. Yi, T. Clausen, "*Collection Tree Extension of Reactive Routing Protocol for Low-Power and Lossy Networks*", International Journal o Distributed Sensor Networks, vol 2014, Article ID 452421, 12 pages, 2014.

[15] S Elyengui, R. Bouhouchi, T Ezzedine, "*A comparative performance study of the routing protocols RPL, LOADng and LOADng-CTP with bidirectional traffic for AMI scenario*", arXiv, Jan. 2016.