

RPL based Routing for Advanced Metering Infrastructure in Smart Grid

Di Wang[†], Zhifeng Tao[★], Jinyun Zhang[★], Alhoussein A. Abouzeid[†]

[†]Department of Electrical, Computer and Systems Engineering

Rensselaer Polytechnic Institute, Troy, New York 12180-3590, USA

[★]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA

Email: wangd2@rpi.edu, tao@merl.com, jzhang@merl.com and abouzeid@ecse.rpi.edu

Abstract—In this paper, we present a routing protocol design and implementation for the Advanced Metering Infrastructure (AMI) in Smart Grid. The proposed protocol implementation is based on the framework of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL), which is proposed by IETF and currently still in its design phase. RPL is based on the idea of maintaining a directed acyclic graph (DAG) structure for the network. We provide a practical implementation of RPL with a number of proper modifications so as to fit into the AMI structure and meet stringent requirements enforced by the AMI. In particular, we propose a novel DAG rank computation method and a reverse path recording mechanism, which enables real-time automated meter reading and real-time remote utility management in the AMI. Our proposed routing protocol design for AMI networks is validated through extensive simulations.

I. INTRODUCTION

Smart grid is a fast-developing technology that aims at delivering electricity from electric suppliers to homes using digital technology to interact with home appliances in order to save energy, reduce cost and increase reliability. The concept of smart grid involves bridging information and communications technologies with energy technology to permit two-way power flow, to achieve seamless operation for electric generation, delivery, and end-use benefit, and to enable a wide adoption of renewable energy and electric vehicles.

Although smart grid technology is still in its early stage, its current deployment has already provided some benefits. For example, many electric suppliers now use the Automated Meter Reading (AMR) system [1] to collect data from electric meters. Using wireless or powerline communications technologies, AMR system can provide one-way communications from meters to a data-reading device (via a gateway). Therefore, compared to the conventional power grid, the use of AMR saves utility providers the expense of periodic trips to each physical location to read a meter.

Future development of smart grid is expected to go one step further and provide more features than AMR. A vital component of a future smart grid system is called the Advanced Metering Infrastructure (AMI) [2], which is expected to provide two-way communications that allow utilities to not only keep track of consumers' electricity usage, but also keep consumers informed of latest electricity prices and perform remote utility management, all on a real-time basis.

One possible solution to enabling these functionalities in

AMI is to deploy a static, multi-hop wireless mesh network that connects a very large number (hundreds or even thousands) of electric meters to a gateway, which in turn is connected (possibly by wireline) to a control center that coordinates all kinds of management mentioned above. Most importantly, such wireless-mesh based AMI network should provide proper routing functions ensuring highly reliable and low-latency delivery for the following two types of traffic: 1) *inward unicast traffic*, consisting of meter-reading data flowing from each meter to the gateway, and 2) *outward unicast traffic*, consisting of utility management data flowing from the gateway to each meter.

Please be aware that, although meter nodes in an AMI network are immobile, the quality of wireless links between an arbitrary pair of meters is in general unstable and varies with time due to fading effects and signal interference. Therefore, in order to satisfy the high-reliability and low-latency requirements of AMI, the routing protocol for AMI networks must have the ability to cope with frequent link state changes by providing fast and effective routing path re-computation methods, and at the same time must only produce moderate amount of protocol overhead.

In this paper, we present a practical implementation (with some modifications) of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) [3] that aims at providing reliable and low-latency routing support for large-scale AMI networks in smart grid. RPL is a routing protocol that is currently under development by the Internet Engineering Task Force (IETF) for low power and lossy networks (LLN) [4], which intends to support a variety of low-cost network applications, including industrial monitoring, building automation, connected homes, health care, environmental monitoring, urban sensor networks (e.g. smart grid), asset tracking, etc. In general, RPL proposes to use directed acyclic graphs (DAG) as an abstraction of network topology, and each node maintains its position in a DAG structure by using a rank property to determine its relations with other nodes in the DAG.

Although the framework for RPL has been established in [3], the details of RPL are still under development at this moment with many important issues left unresolved. Specifically, as for the smart grid application, it remains to be seen how RPL should be implemented and/or modified to order to meet stringent requirements enforced by the AMI.

The objective of this paper is to fill in this gap.

The major contributions of this paper are as follows:

1) We present a detailed implementation of RPL that integrates with CSMA-based MAC layer protocols, and provides reliable and low-latency transport for both inward and outward unicast traffic in large-scale AMI networks.

2) We adopt the Expected Transmission Time (ETX) as the link metric and propose a low-cost ETX measurement scheme based on a MAC layer feedback mechanism.

3) We propose a novel, ETX-based rank computation method that is further used by the DAG construction and maintenance mechanisms that provide high end-to-end reliability for the inward unicast traffic in AMI networks.

4) We propose a reverse path recording mechanism to establish the routes for the outward unicast traffic. This mechanism is purely based on the processing of inward unicast data traffic, and hence will not produce extra protocol overhead.

5) We provide performance evaluations for the proposed implementation of RPL through extensive simulations. The simulation results show that, in AMI networks, and in the presence of shadow fading, the proposed RPL-based routing protocol outperforms some existing routing protocols like AODV [6], and produces satisfactory performances in terms of packet delivery ratio and end-to-end delay.

The rest of this paper is organized as follows. Section II provides a brief introduction of the RPL framework drafted by the IETF. In Section III we propose a detailed implementation of RPL tailored for AMI networks. Section IV presents simulation results for our proposed implementation of RPL. In section V we conclude the paper.

II. THE RPL FRAMEWORK

RPL is a routing protocol for low power and lossy Networks (LLN) that is currently under design by an IETF working group. The key idea of RPL is to maintain network state information using one or more directed acyclic graphs (DAG).

A DAG is a directed graph wherein all edges are oriented in such a way that no cycles exist. For each DAG created in RPL, there is a root. The DAG root typically is the gateway node in AMI networks or the sink node in sensor networks. All edges in the DAG are contained in paths oriented toward and terminating at one root node. Each node in the DAG is associated with a rank value. The rank of nodes along any path to the DAG root should be monotonically decreasing in order to avoid any routing loop.

In order to construct a DAG, the gateway node will issue a control message called DAG Information Object (DIO). A DIO message conveys information about the DAG, including: 1) a DAGID used to identify the DAG as sourced from the DAG root; 2) rank information used by nodes to determine their positions in the DAG relative to each other; 3) Objective Function identified by an Objective Code Point (OCP) that specifies the metrics used within the DAG and the method for computing DAG rank.

Any other node (called client node) that receives a DIO message and is willing to join the DAG should add the DIO

sender (the previous node traveled by the DIO) to its parent list, compute its own rank (associated with the parent node) according to the OCP, and pass on the DIO message with the updated rank information. For a node having already joined the DAG, upon receiving another DIO message it may have the option to 1) discard the DIO based on several criteria recommended by RPL, or 2) process the DIO to maintain a position in an existing DAG, or 3) improve its position (by obtaining a lower rank) according to the OCP and current path cost. After the DAG is constructed, each client node will be able to forward any inward traffic (destined to the gateway) by choosing its most preferred parent as the next-hop node.

In order to support the outward traffic from the gateway to a client node, the client node should issue a control message called Destination Advertisement Object (DAO). The information conveyed in the DAO message includes 1) the rank information used by nodes to determine how far away the destination (the client node that issues the DAO message) is, and 2) reverse route information to record the node visited along the outward path. After passing this DAO message all the way from the client node to the gateway according to the inward path indicated by the DAG, and all the intermediate nodes record the reverse path information from the DAO message, a complete outward path is established from the gateway to the client node.

Please note that RPL does not specify any particular OCP or routing metric for DAG rank computation. Nor did it mention any optimizations or modifications of RPL for some particular applications like the smart grid. Therefore the current framework of RPL leaves plenty of spaces for future exploration.

III. RPL IMPLEMENTATION FOR AMI NETWORKS

In this section we introduce a detailed RPL implementation with some modifications that are specifically tailored for AMI networks. We consider a static multi-hop wireless AMI network that consists of n meter node and one gateway node. The routing protocol maintains one DAG structure rooted at the gateway node. First we introduce the network information defined by this implementation that must be stored and maintained by each node. Then we define the data traffic forwarding rules that must be followed by each node. After that we present a novel DAG rank computation method based on the use of Expected Transmission Time (ETX) as the link metric. Further we provide a detailed characterization for the DAG construction and maintenance procedure incorporated with the proposed DAG rank computation method. Then we present a ETX measurement scheme based on a MAC layer feedback mechanism, and the protocol operations required for DAG structure adaptation to ETX changes. After that we propose a reverse path recording mechanism based on inward traffic processing, which enables routing support for outward unicast traffic that flows from the gateway to each meter.

A. Network Information

We use node 0 to represent the gateway node in the AMI network, and number the n meters by node 1 through node n .

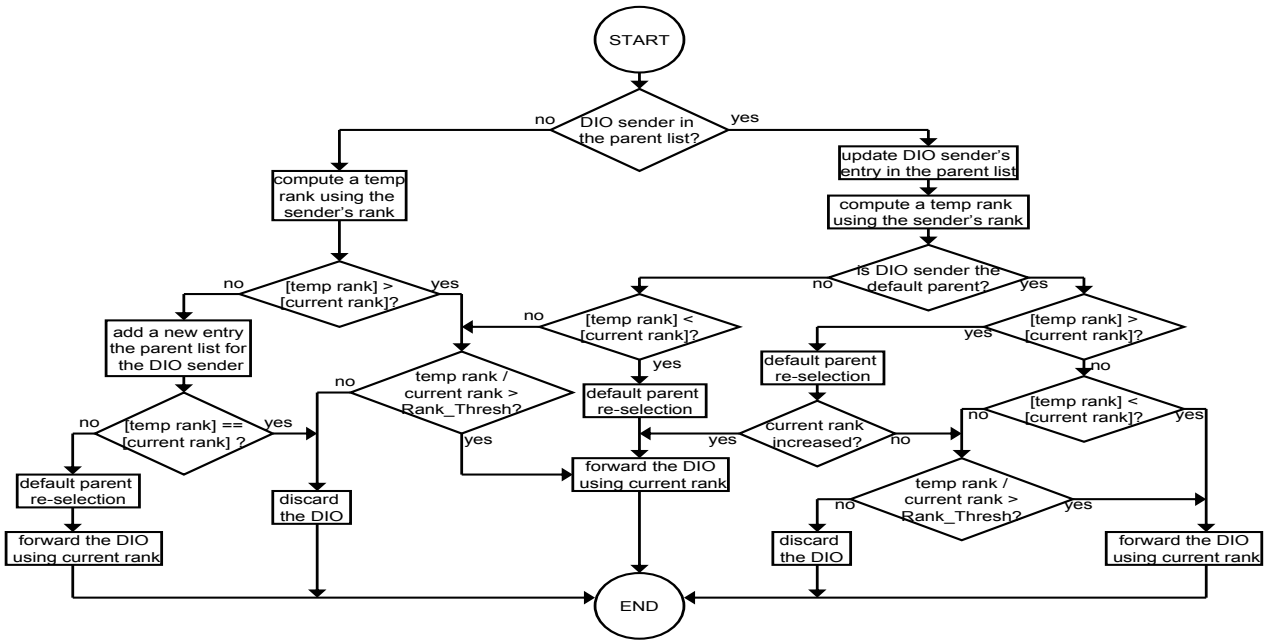


Fig. 1. the operations performed by a meter node that has already joined node 0's DAG once the node receives a DIO message

Each node in the network is uniquely identified by a node ID, e.g., an Internet Protocol (IP) address of the node.

The network information stored at each meter node i ($i = 1, 2, \dots, n$) includes 1) rank of node i , 2) a parent list, 3) node ID of a default parent, and 4) a destination list.

Each entry of the parent list includes 1) the ID of the parent node, 2) rank of the parent node, and 3) the ETX of the link from node i to this parent node.

Each entry of the destination list includes 1) the ID of a destination node, and 2) the ID of a next-hop node.

The default parent of node i is in the parent list of the node and has a lowest rank among all the nodes in the parent list.

The network information stored at the gateway node 0 includes 1) rank of the gateway node (which is set to a constant value n), and 2) a destination list.

B. Data Traffic Forwarding Rules

1) *Inward Unicast Forwarding*: A meter node that generates or receives an inward data packet destined to the gateway should forward this packet to its default parent. The packet should be dropped if the node does not have a default parent.

2) *Outward Unicast Forwarding*: A node that generates or receives an outward data packet destined to a meter node should search for the entry of the destination node in the destination list, and forward the packet to the next-hop node indicated by that entry. The packet should be dropped if the entry cannot be found.

C. DAG Rank Computation

It is already stated that the rank value of the gateway node 0 (root of the DAG) is the constant n . The rank value of meter nodes in the DAG will be determined as follows.

Let $R(i)$ denote the rank of node i , $i = 1, 2, \dots, n$. Let $p(i)$ denote the default parent of node i . Let $X(i, j)$ denote the expected transmission time (ETX) of link (i, j) . Then the rank of node i is determined by

$$R(i) = R(p(i)) \cdot X(i, p(i)) + 1.0 \quad (1)$$

The initial value of ETX of any link is set to be 1.0. Once a link starts to carry data traffic, its ETX value will be measured and updated continuously. Therefore the ETX value of each link may keep changing over time. The method of measuring a link's ETX and adapting the DAG structure according to ETX changes will be introduced later in Section III-E.

Please note that, the proposed rank computation method imposes strong penalty on using links with high ETX. For example, suppose the ETX of a link from node i to its default parent $p(i)$ is increased by Δ . Then, if node i still keeps $p(i)$ as its default parent after the ETX change, then node i 's rank $R(i)$ will be increased by $n\Delta$.

Therefore with this DAG rank computation method, the routing protocol will be inclined to use links with very low ETX and hence provide good end-to-end reliability.

D. DAG Construction and Maintenance

The DAG construction and maintenance procedure can be briefly summarized as follows.

The gateway triggers the DAG construction procedure by broadcasting a DAG Information Object (DIO) message using the User Data Protocol (UDP). The DIO includes information such as DAGID (gateway's node ID) and the rank value (n).

For any meter node (say node i , $i = 1, 2, \dots, n$) that receives a DIO message but have not yet joined node 0's DAG (i.e. with an empty parent list), the node should add the DIO sender to its parent list by recording the node ID and the rank in a

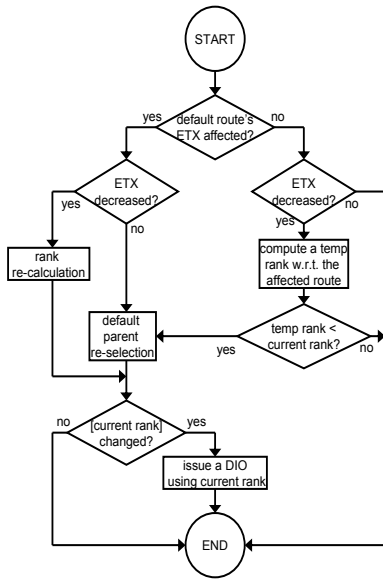


Fig. 2. the operations performed by a meter node due to ETX change

new entry in the parent list, and makes the sender node the default parent node. Node i then determines its rank according Equation (1) and forwards the DIO message with the rank information updated with its rank.

For any meter node i already in the DAG, upon receiving a DIO message from a node j , the node should either discard the DIO, or modify and forward the DIO according to the following rules illustrated in fig. 1 and explained as follows.

Node i first checks if the DIO sender j is in the parent list. If node j is not in the node i 's parent list, then node i computes a temporary rank T according to Equation (1) using the sender j 's rank, and then compares the temporary rank to the current rank C of the node.

If $[T] \leq [C]$, wherein $[x]$ denote the nearest integer value to x , then node i adds a new entry to the parent list, in which the parent node is node j , and the rank value is node j 's rank indicated by the DIO. If $[T] = [C]$, then node i discards the DIO message in order to avoid creating any routing loop. Otherwise, node i re-selects its default parent, re-determines the rank according to Equation (1), and forwards the DIO by broadcasting using its current rank value.

If $[T] > [C]$, node i checks if the ratio T/C has exceeded a predefined threshold value R_T . If $T/C > R_T$, node i forwards the DIO using its current rank for the purpose of improving the rank of node j ; otherwise node i discards the DIO.

If node j is in the parent list of node i , then node i updates node j 's entry using the information provided by the DIO. Then node i computes a temporary rank T according to Equation (1) using node j 's rank, and compares the temporary rank T with node i 's current rank C .

If node j is not the default parent of node i and $[T] \geq [C]$, then node i computes the ratio T/C and compares with R_T . Depending on the output of the comparison node i will decide either to discard the DIO, or modify and forward it for the

purpose of improving node j 's rank.

If node j is not the default parent of node i and $[T] < [C]$, then node i will perform default parent re-selection, re-compute the rank value according to Equation (1), and forward the DIO using its current rank value.

If node j is the default parent of node i and $[T] > [C]$, then node i re-selects the default parent and re-computes the rank according to Equation (1). If the value of $[C]$ has increased after the default parent re-selection, node i will forward the DIO using its current rank. If $[C]$ is not increased, then node i will compute the ratio T/C and compare with R_T and check to see if it needs to forward the DIO or not.

If node j is the default parent of node i and if $[T] < [C]$, quite similarly to what mentioned above, node i will compute the ratio T/C and compare with R_T and check to see if it needs to forward the DIO or not.

In this way, node i will always keep a position in the DAG by maintaining a parent list and adapting its rank value according to DIO messages from other nodes. Node i will also notify other nodes with its own rank via DIO messages when needed and hence nodes will maintain a consistent and efficient DAG structure that provides routing support for any inward unicast traffic generated from meters to the gateway.

E. ETX Measurement and ETX-Change triggered Operations

The ETX measurement of a link (i, j) is based on a number s of successful network-layer transmissions of m data packets transmitted from node i to node j in the past τ seconds using

$$X(i, j) = m/s \quad (2)$$

The information of successful/failed network-layer transmissions can be obtained via a MAC layer feedback mechanism. For example, in IEEE 802.11, after successfully receiving a unicast packet at the MAC layer, the receiver will reply with an ACK packet to the sender and hence the sender will know the transmission was successful. If the receiver does not receive the ACK and the transmission reaches the maximum retry limit specified by the IEEE 802.11 MAC layer protocol, a MAC layer failure will be reported to the network layer.

Therefore this ETX measurement method will allow each meter node i to monitor the ETX of links to any of its parent nodes. If node i detects a ETX change in any link to one of its parents, node i will perform a number of operations illustrated in Fig. 2 and explained as follows.

Assume node i detects a ETX change in a link to one of its parents, node j . If node j is node i 's default parent and the ETX has decreased, then node i re-computes its rank C according to (1). If the value of $[C]$ has changed after the re-computation, node i issues a DIO using the updated rank.

If node j is the default parent of node i and the ETX has increased, then node i re-selects the default parent and re-calculates the rank C according to Equation (1). If the value of $[C]$ has changed after the re-calculation, then node i issues a DIO message using the updated rank.

If node j is not the default parent of node i and the ETX has decreased, then node i computes a temporary rank T using

the rank of node j . If $T < C$, then node i selects node j as the default parent and update the rank C . If the value of $[C]$ has changed after the re-selection, then node i issues a DIO message using the updated rank.

Therefore, those DIO messages issued by nodes due to ETX change will propagate over the network and trigger the aforementioned DAG maintenance procedure, allowing automated DAG structure adaption to the ETX.

F. Reverse Path Recording Mechanism

The reverse path recording mechanism enables outward unicast routing support by letting each node in the AMI network record the source and the last-hop node of the inward data packets traveling through that node. This mechanism is illustrated by Fig. 3 and explained as follows.

Assume a node i ($i = 1, 2, \dots, n$) receives an inward unicast packet sourced from another node j . Then node i checks if there is an entry for node j in the destination list of node i . If not, then a new entry is created, with the destination node ID being the node j 's ID, and the next-hop node ID being the packet's last-hop node ID. If the entry for node j exists, the entry should be updated accordingly.

In this way, each node is able to record all of its descendants in its destination list, with the next-hop node ID indicating the direction towards which the descendant node can be reached, therefore enabling the routing support for outward unicast traffic from the gateway to each meter node.

As compared to the IETF's RPL framework, our proposed reverse path recording mechanism establishes outward paths only by processing the inward data traffic and hence will not introduce any additional protocol overhead like the DAO messages used in [3]. The philosophy of this design lies on the fact that AMI networks performs frequent, periodical meter reading operations which periodically generate inward unicast data traffic with sufficient amount.

IV. SIMULATION RESULTS

We validate our routing protocol design for AMI networks by simulations using ns-2 [5]. A number of $n = 1000$ meter nodes are randomly distributed in a $300\text{m} \times 300\text{m}$ square, with the gateway node placed at the center. The wireless channel between any two nodes are under the effect of shadow fading with path loss exponent β and shadowing deviation σ (in dB). The transmit power of nodes is set to the minimum value that ensures successful transmissions between any two nodes within a distance $r = 17\text{m}$ if there is no interference or shadow fading effects (i.e. $\sigma = 0$). The inward traffic (due to meter-data reading) is simulated by letting each meter node generate a constant-bit-rate (CBR) session destined to the gateway using UDP at the rate $\lambda_1 = 1$ pkt/min, with packet size $b_1 = 200$ bytes. The outward traffic (due to remote utility management) is simulated by letting the gateway node produce a Poisson traffic destined to each meter using UDP at the rate $\lambda_2 = 0.1$ pkt/min, with packet size $b_2 = 150$ bytes. The total simulation time is 6000 seconds.

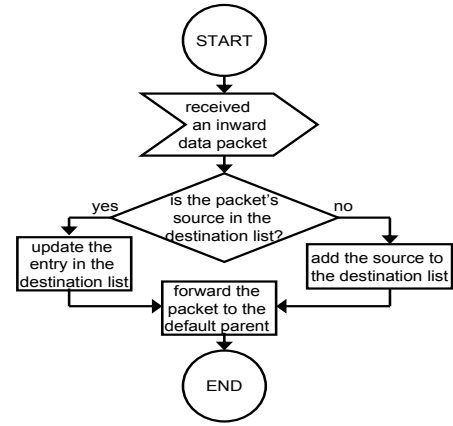


Fig. 3. the illustration of reverse path recording mechanism

we evaluated the performance of RPL under the network scenario described above and make comparisons with the performance of the Ad hoc On-Demand Distance Vector (AODV) routing Protocol [6]. Fig. 4(a) shows the plot of the inward data packet delivery ratio for each meter against the distance from the node to the gateway. It is indicated in the figure that the overall inward packet delivery ratio (of all meters) is around 99.9% for RPL and only 37.3% for AODV. For RPL, The per-node packet delivery ratio is not sensitive to the distance. However, for AODV there is a clear trend that the per-node packet delivery ratio decreases linearly with respect to the distance. Fig. 4(b) shows the plot of the average end-to-end delay of inward traffic against nodes' distance to the gateway. This figure indicates that the average end-to-end delay (averaged over all nodes) is around 160ms for RPL and 1870ms for AODV. For RPL the per-node end-to-end delay is also not very sensitive to the distance, however, for AODV the per-node delay increases with respect to the distance. Further, Fig. 4(c) and Fig. 5(b) plot a 95% confidence interval for the end-to-end delay of inward data traffic for a subset of nodes in the network, which suggest that a much larger delay jitter is incurred by AODV than that by RPL.

The reason for the fact that AODV fails to provide satisfactory performance in a large scale AIM network is that, in AODV each node has to initiate a route request (RREQ) to establish a path to the gateway. Due to frequent link breakage (resulted from the shadow fading), a large amount of RREQ packets will be generated and propagated over the network. For nodes that are far away from the gateway, their paths (to the gateway) are more likely to fail than the nodes close to the gateway, and the delay for a far-away node to establish a path in AODV is much higher than those near the gateway.

We also evaluated the performance of RPL for AMI networks under different levels of the shadowing effect and results are shown in Fig. 5. Here Fig. 5(a) shows the plot of the per-node packet delivery ratio for the inward data traffic against the distance from nodes to the gateway, when the shadowing deviation is 1dB and 2dB, respectively. It is indicated in the figure that, for 1dB fading, the overall delivery ratio is 99.9%

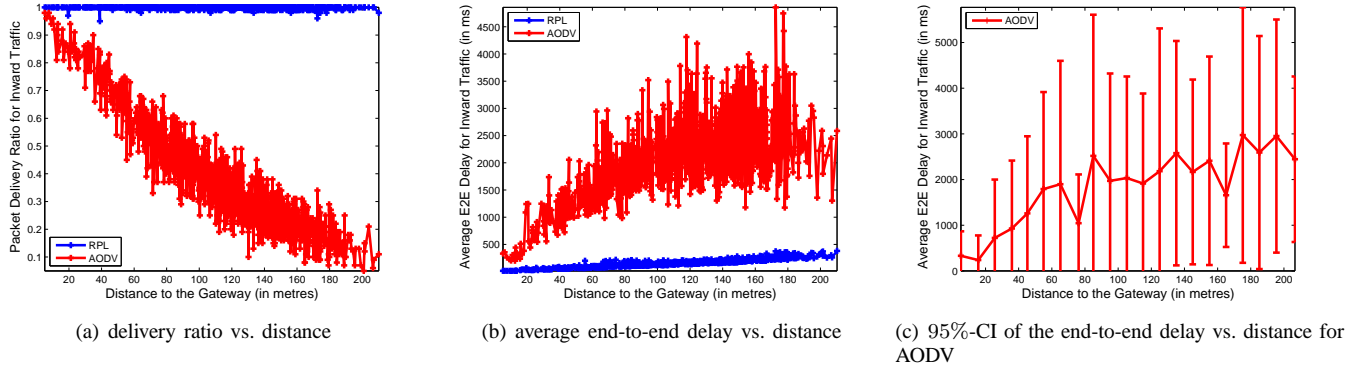


Fig. 4. performance evaluation of RPL and AODV under 1dB shadow fading

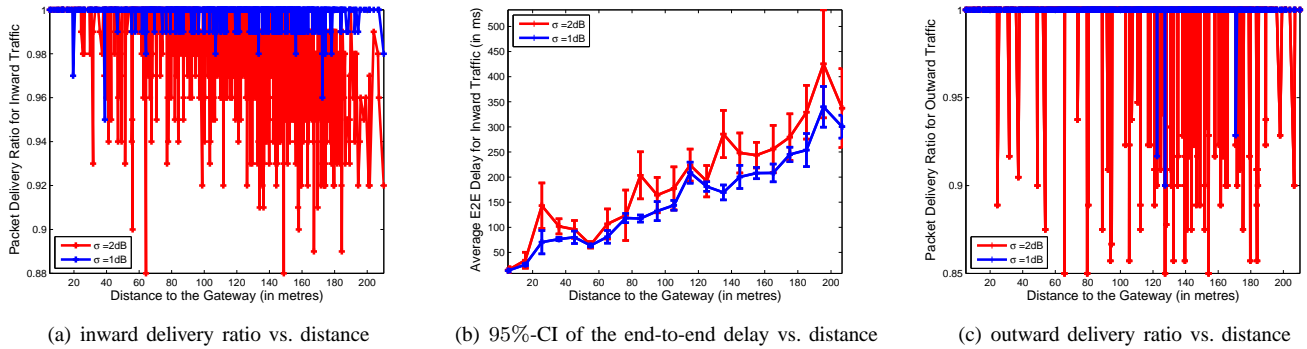


Fig. 5. performance evaluation of RPL under 1dB and 2dB shadow fading

and the worst-case per-node delivery ratio is 95%; for 2dB fading, the overall delivery ratio is 97.9% and the worst-case per-node delivery ratio is 88%. Fig. 5(b) plots a 95% confidence interval for the end-to-end delay of inward data traffic for a subset of nodes in the network. This figure tells that, for 1dB fading, the overall average end-to-end delay is 161ms, with a 95%-CI worst-case end-to-end delay around 350ms; for 2dB fading, the overall average end-to-end delay is 208ms, with a 95%-CI worst-case end-to-end delay around 550ms. Fig. 5(c) shows the plot of the per-node packet delivery ratio for the outward data traffic against the distance from nodes to the gateway. this figure indicates that, for 1dB fading, the overall delivery ratio is 99.98% and the worst-case per-node delivery ratio is 90%; for 2dB fading, the overall delivery ratio is 99.2% and the worst-case per-node delivery ratio is 85%. In all, under the described network scenario, RPL produces a satisfactory performance for both inward and outward unicast data traffic in AMI networks.

V. CONCLUSION

In this paper, we proposed a RPL based routing protocol design for AMI networks in smart grid. RPL is a routing framework proposed by an IETF working group that is currently still under development. We presented a detailed implementation of RPL with a number of modifications specifically tailored for the AMI. In particular, We adopted the Expected Transmission Time (ETX) as the link metric and proposed a low-cost ETX

measurement scheme. We proposed a novel, ETX-based rank computation method serving as the foundation of the DAG construction and maintenance mechanisms, which provide high end-to-end reliability for the inward unicast traffic in AMI networks. Further, We proposed a reverse path recording mechanism that establishes the paths for the outward unicast traffic in AMI networks. This mechanism is purely based on handling the inward unicast data traffic, and hence will not produce extra protocol overhead. We provided performance evaluations for the proposed RPL implementation via extensive simulations. The simulation results shows that, in the presence of shadow fading, the proposed RPL-based routing protocol produces satisfactory performances in terms of packet delivery ratio and end-to-end delay.

REFERENCES

- [1] R. Fischer, N. Schulz, and G. H. Anderson, Information Management For an Automated Meter Reading System, Proceedings of the 62nd American Power Conference, April 2000.
- [2] S. Karnouskos, O. Terzidis and P. Karnouskos, An Advanced Metering Infrastructure for Future Energy Networks, IFIP/IEEE 1st International Conference on New Technologies, Mobility and Security, May 2007.
- [3] T. Winter and P. Thubert, RPL: IPv6 Routing Protocol for Low Power and Lossy Networks, draft-ietf-roll-rpl-04.txt, October 2009.
- [4] M. Dohler, T. Watteyne, T. Winter and D. Barthel, RFC 5548: Routing Requirements for Urban Low-Power and Lossy Networks, May 2009.
- [5] Network Simulator 2 (ns-2), <http://www.isi.edu/nsnam/ns/>.
- [6] C. Perkins and S. Das, RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing, July 2003.