

## Resource Aware Routing Protocol in Heterogeneous Wireless Machine-to-Machine Networks

Guo, J.; Orlik, P.V.; Parsons, K.; Ishibashi, K.; Takita, D.

TR2015-127 December 2015

### Abstract

Routing algorithm can significantly impact network performance. Routing in a network containing heterogeneous nodes differs from routing in a network with homogeneous nodes. If the routing algorithm is designed to fit less powerful nodes, the resources of more powerful nodes are wasted and network performance can be degraded. If the routing algorithm is developed to suit more powerful nodes, less powerful nodes may not have sufficient resources to run the algorithm and network may break down. Routing algorithms developed for homogeneous networks do not work well for heterogeneous networks. The IETF designed the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) by taking into account resource heterogeneity and defined four modes of operation. However, RPL only allows one mode of operation for all routers in a network. This paper proposes a resource-aware adaptive mode RPL (RAM-RPL) to achieve adaptive mode of operation in heterogeneous wireless machine-to-machine (M2M) networks. RAM-RPL not only allows routers to have mixed modes of operation in a network but also allows routers to adaptively adjust their modes of operation during network operation. Acting parent and acting root techniques are introduced to realize adaptive mode of operation and route compression. RAM-RPL exploits resource heterogeneity and shifts routing workload from less powerful nodes to more powerful nodes. Simulation results show that RAM-RPL can improve data packet delivery rate by 26% and reduce control message overhead by 53% while maintaining similar packet latency.

*2015 IEEE Global Communications Conference (GLOBECOM)*

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



# Resource Aware Routing Protocol in Heterogeneous Wireless Machine-to-Machine Networks

Jianlin Guo, Philip Orlik, Kieran Parsons  
Electronics and Communications  
Mitsubishi Electric Research Laboratories  
Cambridge, MA 02139, USA  
{guo, porlik, parsons}@merl.com

Koichi Ishibashi, Daisuke Takita  
Wireless Modules Development Center  
Mitsubishi Electric Corporation IT R&D Center  
Ofuna, Japan  
Ishibashi.Koichi@ce, Takita.Daisuke@eb}.MitsubishiElectric.co.jp

**Abstract**—Routing algorithm can significantly impact network performance. Routing in a network containing heterogeneous nodes differs from routing in a network with homogeneous nodes. If the routing algorithm is designed to fit less powerful nodes, the resources of more powerful nodes are wasted and network performance can be degraded. If the routing algorithm is developed to suit more powerful nodes, less powerful nodes may not have sufficient resources to run the algorithm and network may break down. Routing algorithms developed for homogeneous networks do not work well for heterogeneous networks. The IETF designed the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) by taking into account resource heterogeneity and defined four modes of operation. However, RPL only allows one mode of operation for all routers in a network. This paper proposes a resource-aware adaptive mode RPL (RAM-RPL) to achieve adaptive mode of operation in heterogeneous wireless machine-to-machine (M2M) networks. RAM-RPL not only allows routers to have mixed modes of operation in a network but also allows routers to adaptively adjust their modes of operation during network operation. Acting parent and acting root techniques are introduced to realize adaptive mode of operation and route compression. RAM-RPL exploits resource heterogeneity and shifts routing workload from less powerful nodes to more powerful nodes. Simulation results show that RAM-RPL can improve data packet delivery rate by 26% and reduce control message overhead by 53% while maintaining similar packet latency.

**Keywords**—Heterogeneity; wireless M2M; mesh topology; resource aware; distributed control.

## I. INTRODUCTION

Wireless Machine-to-Machine (M2M) networks are being driven by emerging applications in Internet of Things (IOT), smart grid, industrial automation, home and building automation. The networks for such applications typically contain nodes with heterogeneous resources and capabilities. Some nodes may have more resources and other nodes may have constrained resources. Nodes with more resources can run heavier protocols and perform more functions. Nodes with constrained resources can only run lighter protocols and perform limited functions.

Unlike homogeneous networks, heterogeneous networks require resource aware routing algorithms to achieve the best performance. Conventional routing protocols such as ad-hoc on demand distance vector (AODV) do not consider resource heterogeneity of the nodes. The Internet Engineering Task Force (IETF) standardized the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) as RFC 6550 [1]. RPL considers resources of the nodes and defines four modes of operation. The higher mode of operation (MOP) is, the more

routing functions are supported. However, RPL only allows one MOP for all routers in a network and routers must have same MOP as the root does. Node operating on a MOP different from MOP of the root can only participate in network as a leaf node. This homogeneous MOP requirement of RPL may partition a physically connected network. Fig.1A shows an example of the network partition. Therefore, RPL needs further enhancements to fit heterogeneous networks well.

An IETF draft [2] proposed the mixed mode RPL. This draft tries to support mixed modes of operation by enhancing downward data packet forwarding mechanism of the storing node. The problem is that this draft does not address how to build downward routes in a network operating on mixed modes of operation. In fact, RPL constructs downward routes via the upward forwarding mechanism of the Destination Advertisement Object (DAO) message. As a result, the mixed mode RPL proposed in [2] does not work. Fig.1B shows a network, for which child-parent relationships and parent specifications are shown in the right table. The root  $R$  does not receive parent information for storing nodes  $S1$  and  $S2$ . Therefore,  $R$  can not build downward routes to nodes  $S1$  and  $S2$ . In addition, this draft requires modification to DAO message, which causes protocol compatibility issue.

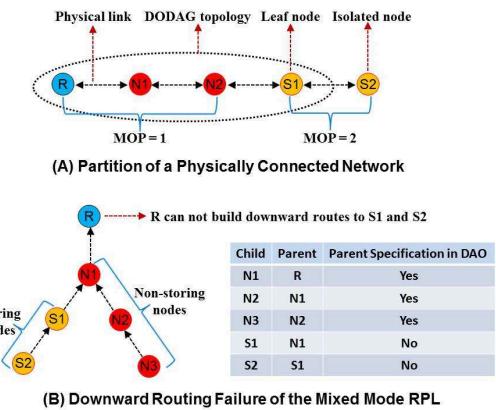


Fig. 1: Routing Issues of RPL with Mixed Modes of Operation

This paper proposes a resource-aware adaptive mode RPL (RAM-RPL) to realize adaptive mode of operation in heterogeneous wireless M2M networks containing nodes with different resources and capabilities. RAM-RPL leverages both sustainable resources such as memory and non-sustainable resources such as battery energy. Nodes with more resources perform

more routing functions and nodes with constrained resources perform limited routing functions or do not route packet. RAM-RPL provides resource aware solutions for both upward routing and downward routing. A new path level routing metric is introduced to discover resource aware upward routes. Acting parent and acting root techniques are introduced to realize resource aware downward routing and route compression, i.e., communication overhead reduction.

The rest of this paper is organized as follows. Section II presents the related work. Section III describes adaptive mode of operation. Section IV introduces a resource aware routing metric. The detailed design of the RAM-RPL is presented in Section V. Performance evaluation of the RAM-RPL is provided in Section VI. We conclude our work in Section VII.

## II. RELATED WORK

RPL organizes nodes in a network as a Destination Oriented Directed Acyclic Graph (DODAG). A DODAG has only one data sink called the DODAG root. RPL uses different types of control messages to build upward routes from nodes to the root and downward routes from the root to nodes.

DODAG Information Object (DIO) message is used for DODAG topology construction and upward route discovery. DIO transmission is initiated by the root. Routers propagate DIO to all nodes in a network. DIO contains information to build the DODAG topology. For example, the rank of a node defines the individual position of the node relative to other nodes with respect to the root of the DODAG and the MOP indicates the mode of operation of the network. MOP is set by the root and routers are not allowed to change it.

DAO message is used for downward route construction. Depending on the MOP of a network, downward routing varies. If MOP = 0, RPL does not support downward routing and therefore, does not transmit DAO message. If MOP = 1, RPL supports downward routing via source routing. In this case, nodes send DAOs to the root and specify their parents in DAOs. Source and destination addresses of the DAO are global IPv6 addresses. If MOP = 2 or 3, downward routing is realized via routing table. In this case, DAO messages are sent directly to the parents and therefore, nodes do not specify parents in DAOs. Source and destination addresses of the DAO are link-local IPv6 addresses. The difference between MOP = 2 and MOP = 3 is that downward multicast is not supported if MOP = 2 and supported if MOP = 3.

DODAG Information Solicitation (DIS) message is used to solicit a DIO message from a RPL node.

Performance of RPL has been evaluated and analyzed by many researchers. RPL shows good scalability [3] and fast network setup [4]. However, RPL may suffer from severe unreliability [3] and frequent route changes have negative impact on the performance of network [5]. Moreover, RPL does not support mixed modes of operation well in a network.

Several technologies have been proposed to improve the performance of RPL. [6] proposed a multipath opportunistic forwarding scheme to extend RPL with the possibility of forwarding packets over multiple routes. [7] introduced a load balanced technique to improve performance of RPL. [8] proposed a stability metric based mechanism to improve the reliability of RPL. However, these improvements are made for homogeneous wireless networks without considering heterogeneities of the nodes. This paper enhances RPL to realize resource aware routing via the adaptive mode of operation.

## III. ADAPTIVE MODE OF OPERATION

In this section, we introduce adaptive mode of operation (AMOP) to achieve resource aware routing and mixed modes of operation. For convenience, nodes operating on MOP = 1 are called as non-storing nodes and nodes operating on MOP = 2 or 3 are called as storing nodes. In addition, we call DAO for MOP = 1 as non-storing DAO (N-DAO) and DAO for MOP = 2 or 3 as storing DAO (S-DAO).

In a network operating on AMOP, nodes use the MOP settings to signal their resources. Nodes with more resources announce higher MOPs and nodes with constrained resources signal lower MOPs. Nodes initially configure MOPs based on their resources and adaptively adjust MOPs during network operation based on their resource usage and observed network conditions. Nodes learn neighbor's resources by monitoring the MOPs announced. Once a node changes its MOP, it must announce new MOP and its children must respond accordingly by updating their parent sets. Other neighbors may also optimize their parent sets.

IETF standard RFC 7228 [9] defines three classes of constrained devices based on random access memory (RAM) and read only memory (ROM), which can be used as a reference for MOP setting. We propose a method to determine MOP. Define three MOP thresholds  $MOP_{TH1}$ ,  $MOP_{TH2}$  and  $MOP_{TH3}$  such that  $0 < MOP_{TH1} < MOP_{TH2} < MOP_{TH3} < 1$ . A node computes a MOP index ( $MOP_I$ ) as follows.

$$MOP_I = \left( \frac{MA}{MC} \right)^{C_M} \times \left( \frac{RA}{RC} \right)^{C_R} \times \left( \frac{RE}{ME} \right)^{C_E} \times \left( \frac{1}{SS} \right)^{C_N} \quad (1)$$

where  $MA$  is available RAM,  $MC$  is RAM capacity,  $RA$  is available ROM,  $RC$  is ROM capacity,  $RE$  is residual energy level,  $ME$  is the maximum energy level,  $SS$  is sub-DODAG size.  $C_M$ ,  $C_R$ ,  $C_E$  and  $C_N$  are positive exponents to reflect the importance of RAM, ROM, energy and sub-DODAG size, respectively. For example, if RAM is a major consideration,  $C_M$  can be set to a large value and other exponents can be set to small values so that a small change of RAM can greatly impact  $MOP_I$  value. Other resources and observed networks conditions can also be utilized in  $MOP_I$  calculation. Using  $MOP_I$ , a node can then determine its MOP as follows.

$$MOP = \begin{cases} 0 & \text{if } 0 \leq MOP_I < MOP_{TH1} \\ 1 & \text{if } MOP_{TH1} \leq MOP_I < MOP_{TH2} \\ 2 & \text{if } MOP_{TH2} \leq MOP_I < MOP_{TH3} \\ 3 & \text{if } MOP_{TH3} \leq MOP_I \leq 1 \end{cases} \quad (2)$$

Equations (1) and (2) show that two nodes with same RAM, ROM and energy level may set different MOPs based on their sub-DODAG sizes since a node may store route entries for a smaller sub-DODAG but not for a larger sub-DODAG. The MOP adaptation is critical in networks with heterogeneous nodes. For example, a battery node with large memory can initially set its MOP to 2. When its battery level becomes low, the node may lower its MOP to reduce its routing functions.

Using the AMOP, a node constructs DAO message based not only on its resources but also on resources of its parents. A node operating on MOP = 0 does not transmit DAO. A child does not send DAO to a parent operating on MOP = 0. Both non-storing and storing children must construct N-DAO to be sent via a non-storing parent. A storing child constructs a S-DAO to a storing parent. A non-storing child may construct a N-DAO to be sent via a storing parent or a

S-DAO to a storing parent. If a non-storing child constructs a S-DAO to a storing parent, link-local address of parent is used as destination address and global address of the non-storing child is used as source address. A storing node sends S-DAO to the root and a non-storing node sends N-DAO to the root. Even a node operating on MOP = 0 does not route downward packet, it may receive downward packet destined to it.

#### IV. A PATH LEVEL ROUTING METRIC

The routing metrics for RPL is defined in the IETF standard RFC 6551 [10], which specifies link metrics and node metrics. We propose a path level routing metric to propagate resource information of the nodes on a route in a network operating on AMOP. The non-storing node count (NNC) of a route counts the number of non-storing nodes on a route. This metric is used to discover routes containing nodes with more resources. The NNC metric can be used with existing routing metrics. During route discovery, destination node sets NNC = 0. A node operating on MOP = 0 or 1 increases NNC by 1 and a node operating on MOP = 2 or 3 does not change NNC metric. When a node receives a route discovery message, it knows how many nodes on the route having constrained resources.

## V. RESOURCE-AWARE ADAPTIVE MODE RPL

In this section, we present the proposed resource-aware adaptive mode RPL (RAM-RPL) to leverage resources of heterogeneous nodes in route discovery and data packet transmission. We denote a non-storing node as  $Nx$ , e.g.,  $N1$ , and a storing node as  $Sy$ , e.g.,  $S1$ .

#### A. Resource Aware Upward Route Discovery

We provide an algorithm to build DODAG topology using NNC, hop count (HC) and MOP. The root initiates DODAG construction. Nodes select upward routes containing nodes with more resources, i.e., higher MOPs. Fig.2 shows resource aware upward route discovery algorithm. If the first route discovered contains zero non-storing nodes, a node  $N$  sets DIO sender as its default parent. Otherwise, node  $N$  adds DIO sender into its parent set and starts a timer to wait for a more preferred DIO. Node  $N$  optimizes its routes if the better routes are subsequently discovered. A route with smaller NNC is considered better than a route with larger NNC if the HC is same. A route with smaller HC is considered better than a route with larger HC if the NNC is same. MOP of the parent is used to break tie. When DIO waiting timer expires, node  $N$  selects one parent from parent set as its default parent. Once default parent is selected, node  $N$  extends network by transmitting a DIO carrying its MOP and the updated NNC and HC. As DIO propagation continues, node  $N$  receives more information about the network and therefore, it may adjust its MOP value. Node  $N$  stores MOP for each parent and uses its MOP and MOP of the parent to determine type of DAO. Node  $N$  computes its rank using metrics NNC and HC received from default parent and its MOP as follows.

$$Rank = R_D + C_N \times NNC + C_M \times HC + C_M(MOP+1) \quad (3)$$

where  $R_D$  is the rank of default parent,  $C_N$ ,  $C_H$  and  $C_M$  are non-negative coefficients to reflect importance of metrics NNC, HC and MOP, respectively. These coefficients are selected such that the rank computed must satisfy requirements specified in RPL.  $C_N$  and  $C_H$  should be proportional to NNC and

HC, respectively, and  $C_M$  should be inversely proportional to MOP+1. The computed rank should be smaller if a node discovers a better default route.

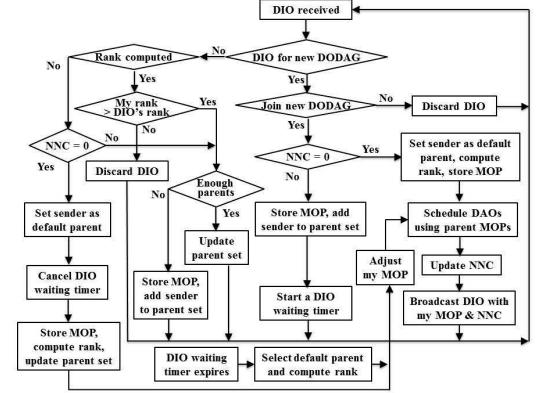


Fig. 2: RAM-RPL Upward Route Discovery Flow Chart

### *B. Acting Parent, Acting Root and Enhanced DAO Processing*

Downward routing in a network operating on mixed MOPs is different from that in a network operating on same MOP, where either routing table or source routing is used. With mixed MOPs, both routing table and source routing are needed. The source routing presents the challenge. For example, in Fig.1B, the root  $R$  must build source routing headers for all nodes since next hop  $N1$  is a non-storing node. However,  $R$  does not receive parent information to build downward routes to  $S1$  and  $S2$ . To overcome this challenge, DAO processing mechanism must be enhanced. We introduce acting parent (AP) and acting root (AR) techniques to achieve mixed MOPs. Acting parent and acting root techniques not only realize downward routing but also reduce communication overhead.

An acting parent of a node  $N$  (storing or non-storing) is a storing node  $S$  on an upward route from the node  $N$  to the root  $R$  and satisfies following two conditions: (1) there is no non-storing node between  $N$  and  $S$  on the route and (2) next hop of  $S$  towards the root is a non-storing node. In fact, an acting parent on an upward route is a border node from a storing section to a non-storing section. The role of an acting parent is to specify itself as the parent in the N-DAO for those target nodes that have no parent specified in the received S-DAOs. To generate a N-DAO to be sent via a non-storing parent, a storing node determines if it needs to perform acting parent role. In Fig.3A, there are three acting parents  $S_2$ ,  $S_3$  and  $S_4$ . For example, on route  $S_8 \rightarrow S_7 \rightarrow S_4 \rightarrow S_2 \rightarrow N_1 \rightarrow R$ ,  $S_2$  is acting parent of  $S_4$ ,  $S_7$  and  $S_8$ . On route  $S_6 \rightarrow S_5 \rightarrow S_3 \rightarrow N_1 \rightarrow R$ ,  $S_3$  is acting parent of  $S_5$  and  $S_6$ . If  $N_3$  sends a S-DAO to  $S_3$ ,  $S_3$  is also acting parent of  $N_3$ .

Acting parent has route scope only. In Fig.3A, on route  $S6 \rightarrow S5 \rightarrow S3 \rightarrow N1 \rightarrow R$ ,  $S3$  is acting parent of  $S6$ . However, on route  $S6 \rightarrow N3 \rightarrow S3 \rightarrow N1 \rightarrow R$ ,  $S3$  is not acting parent of  $S6$  since there is a non-storing node  $N3$  in between  $S6$  and  $S3$ . An acting parent can be a real parent, e.g.,  $S3$  is  $S5$ 's real parent and acting parent. A node can have multiple acting parents, e.g., both  $S2$  and  $S3$  are acting parents of  $S6$ .

An acting root is a storing node that is a border node from a storing section to a non-storing section on a downward route.

The role of an acting root is to store parents or acting parents for the target nodes in N-DAOs received from non-storing children, and manage source routing headers for downward data packets. A storing node performs acting root role if it is DAO parent of any non-storing child. In Fig.3A,  $S_1$  performs acting root role for its sub-DODAG via  $N_2$  and  $S_3$  performs acting root role for its sub-DODAG via  $N_3$ .

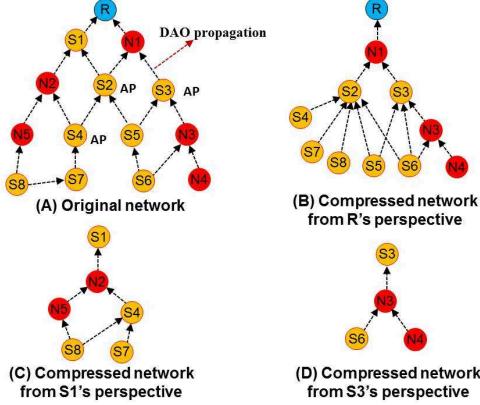


Fig. 3: Illustration of Acting Parent and Route Compression in RAM-RPL

In a network operating on AMOP, a non-storing node only receives N-DAO and handles the received N-DAO same as in RPL. A storing node or the root may receive both N-DAO and S-DAO. We enhance DAO processing mechanism for a storing node  $S$  or the root  $R$ . Firstly,  $S$  or  $R$  may need to identify and store the MOP of a child. Secondly,  $S$  may perform acting parent role. Thirdly,  $S$  may perform acting root role. Fourthly,  $S$  may need to process and relay the received N-DAO.

If  $S$  or  $R$  receives a S-DAO, source node is a child. If destination and source addresses are same type, the child is a storing node. Otherwise, the child is a non-storing node. If  $S$  or  $R$  receives a N-DAO, source node is a non-storing node. If source node specifies  $S$  or  $R$  as parent, source node is a child. Otherwise, the source node is not a child.

Using AMOP, a routing table contains entries for all target nodes in both S-DAOs and N-DAOs received from storing children. A child-parent table contains {child, parent} or {child, acting parent} pairs for target nodes in N-DAOs received from non-storing children. The root maintains a routing table if it is DAO parent of any storing child and a child-parent table if it is DAO parent of any non-storing child. Acting root maintains a child-parent table and a routing table if it is DAO parent of any storing child, e.g., in Fig.3A,  $S_1$  maintains a routing table and a child-parent table. Other storing node maintains a routing table if it is DAO parent of any storing child.

The AMOP and acting parent technique solve the problem in [2]. In Fig.1B,  $S_1$  sends a N-DAO via  $N_1$  and specifies  $N_1$  as its parent.  $S_1$  performs acting parent role for  $S_2$  by specifying itself as parent of  $S_2$ . The root  $R$  learns parent (acting parent) of  $S_1$  and  $S_2$  and therefore, can build downward routes to  $S_1$  and  $S_2$ .

### C. Compressed Network and Downward Route Compression

Acting parent and acting root techniques provide a mechanism to build downward routes by utilizing {child, parent} and {child, acting parent} pairs collected in the N-DAOs. A

{child, parent} pair reflects actual network topology. However, a {child, acting parent} pair may present a compressed network topology different from the original network. Fig.3B shows the compressed topology of the network shown in Fig.3A from the root  $R$ 's perspective. A compressed network may have smaller size than the original network and only shows nodes for which source routing is needed, e.g., size of network in Fig.3A is 14 and size of network in Fig.3B is 11. A compressed network may have smaller network diameter than the original network, e.g., diameter of network in Fig.3A is 5 and diameter of network in Fig.3B is 4. The topology of a compressed network depends on the view point. Fig.3C shows the compressed network from  $S_1$ 's perspective and Fig.3D shows the compressed network from  $S_3$ 's perspective. Only the root or an acting root may have a compressed network.

A downward route built from a compressed network may be a compressed route that is shorter than hop-by-hop route, e.g., based on network topology in Fig.3B,  $R$  can build a compressed downward route to  $S_8$  as  $N_1 \rightarrow S_2 \rightarrow S_8$ , which is shorter than actual hop-by-hop route  $N_1 \rightarrow S_2 \rightarrow S_4 \rightarrow S_7 \rightarrow S_8$ . Compressed routes reduce communication overhead by shortening the source routing headers. On a compressed route, a node may not directly communicate with the next hop.

We provide a method that can be used by the root or an acting root to build downward routes. Algorithm 1 shows downward route construction procedure for the root  $R$ . An acting root ( $AR$ ) constructs downward routes similarly as the root  $R$  does. Only difference is the initialization, in which  $AR$  sets  $SS = \{\text{All nodes in } AR\text{'s sub-DODAG learned from the N-DAOs received from non-storing children}\}$ ,  $FS = \{AR\}$ .

---

#### Algorithm 1 A Downward Route Construction Method

---

```

1: // Initialization
2: Start Set (SS) = {All nodes the root  $R$  learned from the
   N-DAOs received from the non-storing children}
3: Finish Set (FS) = { $R$ }
4: // Downward Route Construction
5: If set  $SS$  is empty, process completes. Otherwise,
6:  $R$  checks if set  $SS$  contains any node that has the root  $R$ 
   as only DAO parent
7: if No then
8:   An error occurs and process stops
9: else if a node  $N$  in  $SS$  has  $R$  as only DAO parent then
10:    $R$  build a route for  $N$  as  $R \rightarrow N$ 
11:    $R$  stores route and process continues
12: After check,  $R$  moves all such nodes from  $SS$  to  $FS$ 
13: If set  $SS$  is empty, process completes. Otherwise,
14:  $R$  checks if set  $SS$  contains any node that has all its
   DAO parents in set  $FS$ 
15: if No then
16:   An error occurs and process stops
17: else if a node  $M$  in  $SS$  has nodes  $P_1, \dots, P_k$  in  $FS$  as
   its DAO parents then
18:    $R$  build routes for  $M$  as  $R \rightarrow \dots \rightarrow P_1 \rightarrow M, \dots,$ 
       $R \rightarrow \dots \rightarrow P_k \rightarrow M$ 
19:    $R$  stores routes and process continues
20: After check,  $R$  moves all such nodes from  $SS$  to  $FS$ 
21: If set  $SS$  is empty, process completes. Otherwise,
   process goes back to line 14
22: end if
23: end if

```

---

#### D. Handling Change of the Mode of Operation

When a node  $N$  changes its MOP, it announces new MOP by transmitting a DIO and schedules appropriate DAOs. In the DIO, node  $N$  may trigger its sub-DODAG to send new DAOs. Children and storing DAO parents update the MOP of node  $N$ . If the MOP increases, a child may add  $N$  into its DAO parent set. Furthermore, if the MOP increases from 1 to 2 or 3, a non-storing child having  $N$  as DAO parent may schedule a S-DAO, a storing child having  $N$  as DAO parent schedules a S-DAO in which the storing child may remove its acting parent role, and a storing DAO parent removes its acting root role. Other neighbors may also add  $N$  into their parent sets. If the MOP decreases from 2 or 3 to 1, a child may remove  $N$  from its parent set and/or DAO parent set. Furthermore, a non-storing child having  $N$  as DAO parent schedules a N-DAO, a storing child having  $N$  as DAO parent schedules a N-DAO in which the storing child performs acting parent role, and a storing DAO parent performs acting root role. If node  $N$  decreases its MOP from non-zero to 0, children remove  $N$  from their DAO parent sets and  $N$  may send a N-DAO without parent specification prior to announcing MOP = 0. Such N-DAO is forwarded to the root, which then makes necessary update on its routing table, child-parent table and downward routes that have been affected. If the MOP switches between 2 and 3, no required action is needed.

#### E. Low Overhead Downward Packet Delivery

In RAM-RPL, a downward packet can be delivered in three ways: source routing, routing table or combination of source routing and routing table. The root uses routing table if next hop is a storing node or source routing if next hop is a non-storing node. A source routing header may contain a hop-by-hop route or a compressed route. The compressed route achieves low overhead downward packet delivery.

During downward packet forwarding, a storing node may remove source routing header or replace a longer source routing header with a shorter one. When a storing node receives a downward packet for forwarding and the packet contains a source routing header, it removes source routing header from the packet and forwards the packet using routing table if next hop is a storing node. For example, in Fig.3A,  $R$  may send a packet to  $S_6$  via  $N_1$  by attaching source routing header  $N_1 \rightarrow S_3 \rightarrow S_6$ . When  $S_3$  receives packet, it removes source routing header. If next hop is a non-storing node, it can replace a longer source routing header with a shorter one. When a storing node receives a downward packet for forwarding and the packet does not contain a source routing header, it forwards packet to next hop via routing table if the destination is in its routing table and next hop is a storing node. If the destination is in its routing table and next hop is a non-storing node, it performs acting root role by attaching a source header. If the destination is not in its routing table, it checks if the destination is reachable. If yes, it uses direct P2P forwarding the packet. If no, it forwards packet to its default parent.

## VI. EVALUATION

This section provides performance evaluation of RAM-RPL using the NS2 simulator with IEEE 802.15.4 MAC and PHY. We simulated a heterogeneous wireless network with 500 nodes deployed in a  $690m \times 660m$  rectangle area, which is divided into a  $23 \times 22$  grid. Each grid unit represents a  $10 \times 10$

square meter field and contains a randomly placed node. A sink node (the root) is deployed at the center of rectangle area. 500 nodes are configured with different memory sizes. Small memory nodes can buffer 15 packets and operate on MOP = 1. Large memory nodes can buffer 50 packets and operate on MOP = 2. The percentage of large memory nodes varies from 0% to 100%. The standard RPL is used as benchmark for comparison. The performance metrics are data packet delivery rate (PDR), data packet latency and control message overhead. Performance depends on data packet generation rate. The lower data packet generation rate is, the better performance is.

We demonstrate a bi-directional traffic scenario with upward data collection in which each node transmits 1 data packet to the sink node every 90 seconds and downward service providing in which the root sends 1 service packet to nodes every 0.05 second. Service packets are uniformly distributed to 500 nodes. Packet payload is 50 bytes. The simulation runs 10000 seconds. RPL uses HC metric and RAM-RPL utilizes HC metric and the introduced NNC metric.

#### A. Data Packet Delivery Rate

The PDR is key performance metric for a routing protocol. Fig.4 illustrates variation of the PDR with respect to the percentage of the storing nodes (POSN). As the POSN increases, RAM-RPL fully takes the advantage of the extra memory provided by storing nodes. More improvement is obtained for both upward and downward PDR. Upward PDR improvement is achieved from route optimization by utilizing large memory nodes to reduce packet drop due to buffer overflow. Downward PDR improvement is realized by communication overhead reduction. Presence of the small memory nodes causes RPL to operate on MOP = 1 to avoid network partition, which results in larger packet size due to full length source routing header and packet fragmentation. However, RAM-RPL uses routing table or compressed source routing header. Packet size and chance of packet fragmentation are much smaller. For upward data packets, PDR increases from 75.2% to 99.8% as POSN increases from 0% to 100%. However, RAM-RPL achieves a higher PDR for  $10\% \leq \text{POSN} \leq 90\%$ . RPL drops more data packets due to buffer overflow. As POSN increases from 10% to 20%, RAM-RPL improves PDR by 10% since RAM-RPL fully utilizes extra memory provided by these 10% to 20% of large memory nodes. For downward service packets, RAM-RPL improves PDR by 26%. PDR of RAM-RPL continuously increases from 72% to 98% as POSN increases from 0% to 50%. Once POSN reaches 50%, PDR gradually decreases to 95.7%. This is because more upward data packets are buffered by more large memory nodes. Extra upward data packets collide with downward service packets. PDR of RPL stays at 72% if  $\text{POSN} < 100\%$  since RPL routes do not change as POSN increases. Longer source routing headers cause the sink node to fragment downward service packets, which results in buffer overflow at sink node. Until POSN reaches 100%, RPL operates on MOP = 2 and PDR increases to 95.7%.

#### B. Data Packet Latency

Fig.5 shows that RPL has a slightly shorter data packet latency than RAM-RPL does. RPL uses HC metric and discovers shorter routes, which reduces packet queuing time. On the other hand, RAM-RPL utilizes NNC and HC metrics. Combination of NNC and HC discovers routes containing

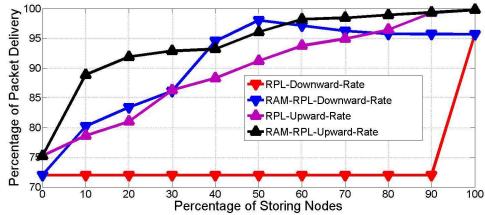


Fig. 4: Upward and Downward Data Packet Delivery Rate

more large memory nodes. However, routes are longer. As the number of hops increases, the queuing time increases correspondingly. For both RPL and RAM-RPL, 90% of packets are delivered under 200 milliseconds and 80% of packets are delivered under 100 milliseconds.

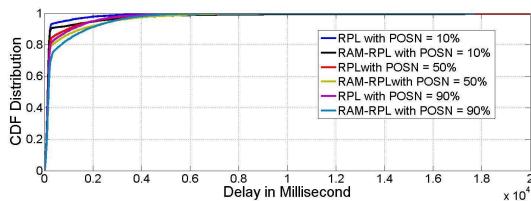


Fig. 5: CDF Distribution of Data Packet Latency

### C. Control Message Overhead

We collected the number of control messages sent or forwarded by each node and present the results for POSN = 30% and 70% as 3D mesh graphs shown in Fig.6. The control messages include DIO, DAO and DIS. As the data sink is placed at the center, the nodes in the central area transmit much more control messages than the nodes on the side do. However, RAM-RPL reduces control message overhead significantly. With RPL, packet congestion results in collision and link break, which causes nodes to discover new routes and send more control messages. Fig.6a and Fig.6c show the number of control messages each node transmitted using RPL. Fig.6b and Fig.6d show the number of control messages each node transmitted using RAM-RPL. With RPL, the peak numbers of control messages transmitted are 1195 and 2042, respectively, and average numbers of control messages transmitted are 66.6 and 133.4, respectively. With RAM-RPL, the peak numbers of control messages transmitted are 761 and 1672, respectively, and average numbers of control messages transmitted are 31.5 and 85.8, respectively. RAM-RPL can reduce peak control message overhead by 36% and average control message overhead by 53%. In addition, as the POSN increases, control message transmission also increases due to more data packet transmission.

## VII. CONCLUSION

We propose a resource aware routing protocol called RAM-RPL for heterogeneous wireless M2M networks containing nodes with different resources and capabilities. The AMOP is introduced to achieve resource aware routing. Operating on AMOP, a node initially configures its MOP based on its resources and adaptively changes its MOP based on its resource usage and observed network conditions. Nodes signal

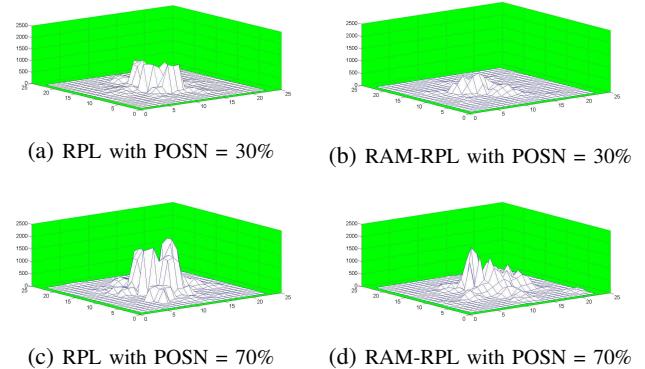


Fig. 6: Distribution of Control Message Transmissions

their resources and capabilities by announcing their MOP settings. Acting parent and acting root techniques are proposed to realize mixed modes of operation and downward route compression. RAM-RPL utilizes extra resources provided by more powerful nodes and shifts routing workload from less powerful nodes to more powerful nodes. In addition, a path level routing metric called non-storing node count (NNC) is introduced to discover routes containing nodes with more resources. An application scenario with upward data collection and downward service providing is simulated using NS2 simulator. Simulation results show that RAM-RPL outperforms standard RPL in terms of data packet delivery rate and control message overhead reduction while maintaining similar packet latency. RAM-RPL can improve upward data packet delivery rate by 10% and downward service packet delivery rate by 26%, reduce peak control message overhead by 36% and average control message overhead by 53%.

## REFERENCES

- [1] T. Winter, P. Thuber, and et al, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," <http://tools.ietf.org/html/rfc6550>, IETF, 2012.
- [2] J. Ko, J. Jeong, and et al, "RPL Routing Pathology in a Network with a Mix of Nodes Operating in Storing and Non-Storing Modes," <https://tools.ietf.org/html/draft-ko-roll-mix-network-pathology-04>, IETF, 2014.
- [3] E. Ancillotti, R. Bruno, and M. Conti, "The Role of the RPL Routing Protocol for Smart Grid Communications," *Communications Magazine, IEEE*, vol. 51, no. 1, pp. 75–83, 2013.
- [4] N. Accettura, L. Grieco, and P. Camarda, "Performance Analysis of the RPL Routing Protocol," in *2011 IEEE International Conference on Mechatronics*. IEEE, 2011.
- [5] M. R. Khan, "Performance and Route Stability Analysis of RPL Protocol," KTH Royal Institute of Technology, 2012.
- [6] B. Pavkovic, F. Theoleyre, and A. Duda, "Multipath Opportunistic RPL Routing over IEEE 802.15.4," in *MSWiM'11*. ACM, 2011.
- [7] X. Liu, J. Guo, G. Bhatti, P. Orlík, and K. Parsons, "Load Balanced Routing for Low Power and Lossy Networks," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013.
- [8] X. Yang, J. Guo, P. Orlík, K. Parsons, and K. Ishibashi, "Stability Metric Based Routing Protocol for Low-Power and Lossy Networks," in *Proceedings of International Conference on Communications (ICC), 2014 IEEE*. IEEE, 2014.
- [9] C. Bormann, M. Ersue, and A. Keranen, "Terminology for Constrained-Node Networks," <http://tools.ietf.org/html/rfc7228>, IETF, 2014.
- [10] J. Vasseur, M. Kim, and et al, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks," <http://tools.ietf.org/html/rfc6551>, IETF, 2012.