Resource Aware Hierarchical Routing in Heterogeneous Wireless IoT Networks

Guo, J.; Orlik, P.V.; Ishibashi, K.

TR2016-082   July 2016

Abstract

Routing algorithm consumes the resources of the network nodes. Different routing algorithms require different amount of the resources. Nodes at different positions of the network topology require different amount of the resources. Routing algorithm must adapt to both available resources and required resources of the nodes. The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) considers memory and defines four modes of operation (MOPs). However, RPL allows only one MOP for all routers in a network. This paper presents a resource aware hierarchical RPL (H-RPL) to realize the mixed MOPs and the resource adaptation in heterogeneous wireless IoT networks. Taking routing preferences of the nodes into account, H-RPL also applies heterogeneous routing metrics and objective functions in hierarchical network topology construction. A new MOP is introduced to indicate the critical resource condition of the node. The required routing memory and the expected routing lifetime are proposed to determine the MOP of the node. The MOP downgrade and the MOP upgrade are introduced to address traffic congestion caused by the isolated higher resource node and to exploit the renewed resources and the resource requirement relaxation, respectively. The queue utilization based data transmission distributes data packets for load balance and network performance improvement. Simulation results show that H-RPL can improve upward data packet delivery rate by 7%, downward data packet delivery rate by 25% and extend network lifetime by 78%.

IEEE International Conference on Ubiquitous and Future Networks

Resource Aware Hierarchical Routing in Heterogeneous Wireless IoT Networks

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

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

***Abstract*—Routing algorithm consumes the resources of the network nodes. Different routing algorithms require different amount of the resources. Nodes at different positions of the network topology require different amount of the resources. Routing algorithm must adapt to both available resources and required resources of the nodes. The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) considers memory and defines four modes of operation (MOPs). However, RPL allows only one MOP for all routers in a network. This paper presents a resource aware hierarchical RPL (H-RPL) to realize the mixed MOPs and the resource adaptation in heterogeneous wireless IoT networks. Taking routing preferences of the nodes into account, H-RPL also applies heterogeneous routing metrics and objective functions in hierarchical network topology construction. A new MOP is introduced to indicate the critical resource condition of the node. The required routing memory and the expected routing lifetime are proposed to determine the MOP of the node. The MOP downgrade and the MOP upgrade are introduced to address traffic congestion caused by the isolated higher resource node and to exploit the renewed resources and the resource requirement relaxation, respectively. The queue utilization based data transmission distributes data packets for load balance and network performance improvement. Simulation results show that H-RPL can improve upward data packet delivery rate by 7%, downward data packet delivery rate by 25% and extend network lifetime by 78%.**

Keywords*—Hierarchical routing; heterogeneous nodes; resource adaptation; wireless communication; Internet of Things.

I. INTRODUCTION

The emerging Internet of Things (IOT) paradigm has been driving heterogeneous wireless networking. An IOT network may consist of tens thousands of nodes with heterogeneous resources and capabilities. The nodes are deployed to carry out multiple tasks with different requirements. The network construction must apply different routing metrics (RMs) and objective functions (OFs) at different portions of the network topology. The routing algorithm must adapt to both available resources and required resources of the nodes. Two critical issues to be addressed are: (i) How to build network topology of the nodes with heterogeneous resources, different resource requirements and multiple application tasks; (ii) How to distribute network traffic during network operation.

In a self-organized network, nodes perform router functions to form network topology and route data packets. The routing algorithms designed for homogeneous networks do not consider resource heterogeneity of the nodes. The IPv6 based routing protocol RPL [1] organizes nodes in a network as a tree-like topology called the Destination Oriented Directed Acyclic Graph (DODAG). A DODAG has only one sink node called the DODAG root. The DODAG Information Object (DIO) message is used for DODAG topology construction and upward route discovery. The DODAG Information Solicitation (DIS) message is used to solicit DIO message from RPL node. The Destination Advertisement Object (DAO) message is used for downward route configuration. The RPL routing protocol considers memory and defines four modes of operation (MOPs). The higher MOP is, the more memory is required to support more routing functions. However, RPL only allows one MOP for all routers in a network and routers must have same MOP as the root does. The node operating on a MOP different from the root MOP can only participate in network as a leaf. This homogeneous MOP requirement can partition a physically connected network. Therefore, the mixed MOP support needs to be addressed.

Authors in [2] proposed DualMOP-RPL to support mixed MOPs in a single RPL network. To solve downward routing issue caused by the mixed MOPs, authors introduced “Modified DAO Transmission” mechanism, which requires non-storing nodes to send hop-by-hop DAOs and process DAOs. However, this mechanism may not work because even the non-storing nodes process DAOs, they do not store DAO information due to the non-storing nature. As a result, the downward routes may still break at the non-storing nodes that have storing children. In addition, authors did not address MOP determination.

We previously proposed resource-aware adaptive mode RPL (RAM-RPL) in [3] to support the mixed MOPs in RPL network. Acting parent and acting root are introduced to realize the mixed MOPs. However, RAM-RPL may suffer from traffic congestion caused by the isolated higher resource node. It does not consider the required routing resources, which depends on routing algorithm and position of the node in network topology. Furthermore, RAM-RPL constructs flat DODAG topology that is not ideal for the networks with multiple tasks.

This paper proposes resource aware hierarchical RPL (H-RPL) to realize hierarchical routing with the mixed MOPs and multiple tasks in heterogeneous wireless IoT networks. The H-RPL considers both available routing resources and required routing resources of the node as well as resources of the neighbors. In a H-RPL network, the attribute MOP indicates the routing type of the node. Nodes operating on higher MOPs support more routing functions and nodes operating on lower MOPs support limited routing functions or do not route. The required routing memory and the expected routing lifetime are introduced to assist MOP determination. With nodes operating on heterogeneous MOPs, the H-RPL constructs hierarchical DODAG (H-DODAG) topology by applying different RMs and

OFs at different tiers of the H-DODAG topology.

The rest of this paper is organized as follows. Section II presents the related work. Section III describes the available and required resource based MOP determination. Section IV introduces the H-DODAG construction and maintenance. The queue utilization based data transmission is presented in Section V. The H-RPL performance evaluation is provided in Section VI. We conclude our work in Section VII.

II. RELATED WORK

Battery energy as one of the resources has been studied by many researchers. The Low-energy adaptive clustering hierarchy (LEACH) [4] is a commonly cited clustering protocol to balance the workload among wireless sensors. LEACH probabilistically rotates the role of cluster head (CH) to save battery energy. The authors in [5] proposed a hierarchical routing protocol by rotating CH based on residual energy level. They showed that hierarchical routing topology is better than flat routing structure in uniforming energy consumption and prolonging network lifetime. In [6], the authors proposed a multi-hop hierarchical routing protocol for wireless sensor networks (WSNs) to enhance the network lifetime and avoid the formation of energy holes. The residual energy level is used to select CH. They showed that the proposed algorithm can largely reduce the total energy consumption and significantly prolong the network lifetime compared to other routing algorithms likes LEACH. A Bayesian coalition game-based optimized clustering algorithm was proposed in [7] to improve LEACH. The results obtained show that the proposed coalition game achieved better stability and network lifetime in comparison to LEACH and other existing protocols. However, the hierarchical routing algorithms mentioned above are designed for homogeneous WSNs and only focus on energy without considering other node resources. Therefore, these algorithms do not fit heterogeneous networks well.

The RPL routing protocol has been designed with memory consideration by defining four MOPs. However, the MOP is set by the root and routers are not allowed to change it. Depending on the MOP of the network, downward routing mechanism varies. If MOP = 0, RPL does not support downward routing and therefore, nodes do not transmit DAO. If MOP = 1, RPL support downward routing via source routing. In this case, nodes send DAOs to the root and specify their parents in DAOs. Source address and destination address of the DAO are global IPv6 address. We refer this type of the DAO to as non-storing DAO (N-DAO). If MOP = 2 or 3, downward routing is realized via routing table. In this case, DAOs are sent directly to the parents and therefore, nodes do not specify parents in DAOs. Source address and destination address of the DAO are link-local IPv6 address. We call this type of the DAO as storing DAO (S-DAO). The difference between MOP = 2 and MOP = 3 is that downward multicast is not supported if MOP = 2 and is supported if MOP = 3.

Even four MOPs are defined, RPL does not support the mixed MOPs in a network. The RPL enhancements are proposed in [2] and [3] to support the mixed MOPs. However, the mechanism proposed in [2] may not work and methods provided in [3] may suffer from packet congestion and do not consider the difference of the routing resource requirements for nodes at different positions of the network topology. In this paper, we propose the H-RPL to realize resource aware hierarchical routing with the mixed MOP support and multiple tasks. The required routing memory and the expected routing lifetime are proposed for nodes to determine their MOPs. A queue utilization based data transmission method is introduced for reliable data packet delivery and load balance.

III. RESOURCE BASED MOP DETERMINATION

Based on available memory, residual battery energy percentage and sub-tree size, we proposed a method to compute MOP in [3]. However, the available memory is not same as the required routing memory, which depends on routing algorithm and positions of the nodes in the network topology. The residual battery energy percentage does not necessarily indicate the battery lifetime due to battery capacity and workload variations. The sub-tree size does not consider the routing memory requirement difference between the non-storing descendant and the storing descendant in the sub-tree. In this paper, we propose solutions to address these issues.

A. A New MOP and K-Hop Neighbor Routing Preference

In a H-RPL network, nodes use MOPs to indicate their routing types. We define a new MOP = -1 to signal that one or combination of the node's resources is in critical condition and therefore, the node can not route any more. The MOP = -1 is different from the infinite rank, which is a transient topology perspective. We refer -1 to as leaf MOP, 0 to as upward MOP, 1 to as non-storing MOP, 2 to as storing MOP and 3 to as multicast storing MOP. The node operating on leaf MOP must act as leaf and is referred to as leaf node. The node operating on upward MOP does not route downward packet and is referred to as upward node. The node operating on non-storing MOP does not store downward route information and is referred to as non-storing node. The node operating on storing MOP stores the downward route entries and K-Hop neighbor routing preference, and is referred to as storing node. The node operating on multicast storing MOP stores the downward route entries, K-Hop neighbor routing preference and downward multicast information, and is referred to as multicast storing node. The leaf nodes and upward nodes may not send DAO message. Non-storing nodes send N-DAO message. Storing nodes and multicast storing nodes sends S-DAO message.

The K-Hop neighbor routing preference is expressed as RMs and OFs preferred by a K-Hop neighbor. Before H-DODAG construction, the root and storing nodes broadcast a K-Hop neighbor request message to inquire the routing preferences from their K-Hop neighbors, which send back response messages. The root and storing nodes determine their K values based on application requirements.

B. Required Routing Memory

The required routing memory (RRM) is defined as the memory required by a node to run a routing algorithm. The routing memory can be divided into leaf memory and router memory. The leaf memory is the memory required by a node to join a network as a leaf and the router memory is the additional memory required by a node to act as a router. Different routing algorithm requires different routing memory. The RRM is different from the allocated (available) routing memory (ARM) that is the memory allocated by a node to run the routing algorithm. It is critical for a node to compute RRM and determine appropriate MOP. In this section, we introduce the RRM estimation techniques for H-RPL routing protocol.

To join a H-DODAG topology, the required leaf memory (M_L) can be estimated as

$$M_L = N_P \times (|P_{ID}| + |P_{MOP}| + |DL| + |LU|) + \\ |HR_{ID}| + |HD_{ID}| + |HD_{VN}| + |N_{MOP}| + OL \tag{1}$$

where N_P is the number of parents (1 byte), P_{ID} is the parent ID (16 bytes), P_{MOP} is the parent MOP (4 bits), DL is the default lifetime (1 byte), LU is the lifetime unit (2 bytes), HR_{ID} is the H-RPL Instance ID (1 byte), HD_{ID} is the H-DODAG ID (16 bytes), HD_{VN} is the H-DODAG Version Number (1 byte), N_{MOP} is the MOP of the node (4 bits) and OL is other required memory by the leaf, e.g., if a node joins multiple sub H-DODAGs (sub-trees) it may store sub H-DODAG IDs and the corresponding parents, routing metrics and objective functions. Therefore, the minimum leaf memory (M_L^{min}) is at least 38 bytes, which is for only 1 parent.

For a node in H-DODAG topology to be an upward router, the node needs to transmit regular DIO message. Therefore, the node must maintain three Trickle Timer configuration parameters: DIO interval minimum (D_{IM}) (1 byte), DIO interval doubling (D_{ID}) (1 byte) and DIO redundancy constant (D_{RC}) (1 byte). The node also needs to maintain three Trickle Timer variables: the current interval size I (4 bytes), a time T (4 bytes) within current interval and a counter C (1 byte). In addition, an upward router needs to maintain RMs and OFs as well as one to many mappings between each OF and the corresponding RMs ($O2M_{MAP}$). A RM container (RC) object is at least 1 byte and an OF is identified by an objective code point (OCP) (1 byte). An upward router also needs to maintain parent rank (R_P) (2 bytes), its own rank (R_N) (2 bytes), MaxRankIncrease (MRI) (2 bytes) and MinHopRankIncrease ($MHRI$) (2 bytes). As a result, the RRM for an upward router (M_{UR}) can be estimated as

$$M_{UR} = M_L + |D_{IM}| + |D_{ID}| + |D_{RC}| + |I| + |T| + \\ |C| + N_{OCP} \times (|OCP| + |O2M_{MAP}|) + N_{RC} \times |RC| + \\ N_P \times |R_P| + |R_N| + |MRI| + |MHRI| + |Q_U| + OU \tag{2}$$

where N_{OCP} is the number of OCPs, N_{RC} is the number of RM containers, Q_U is the queue to buffer upward relay packets and OU is the other required memory by an upward router. Therefore, the minimum routing memory for an upward router (M_{UR}^{min}) is at least 60 bytes, which is for 1 parent, 1 RM, 1 OF and 0 queue size.

For a node in H-DODAG to be a non-storing router, besides routing memory required for an upward router, additional routing memory is needed to maintain the destination advertisement trigger sequence number (DTSN) (1 byte) and to buffer downward relay packets (Q_D). Therefore, the RRM for a non-storing router (M_{NR}) can be estimated as

$$M_{NR} = M_{UR} + N_P \times |P_{DTSN}| + |Q_D| + ON \tag{3}$$

where ON is other required memory by a non-storing router. The minimum routing memory for a non-storing router (M_{NR}^{min}) is at least 61 bytes, which is for 1 parent, 1 RM, 1 OF and 0 queue size.

For a node in H-DODAG topology to be a storing router, besides routing memory required by a non-storing router, additional routing memory is required to store downward route entries, child-parent entries and K-Hop neighbor routing preferences. The size of routing table is computed by counting the number of the destinations contained in both N-DAOs and S-DAOs sent or forwarded by storing children. The size of child-parent table is computed by counting the number of destinations contained in N-DAOs sent or forwarded by non-storing children. The memory size to store K-Hop neighbor routing preferences is computed by counting the number of the K-Hop neighbor response messages. Assume a storing node has N_R entires in its routing table, N_C entries in its child-parent table and N_K K-Hop neighbors. The RRM for a storing router (M_{SR}) can be estimated as

$$M_{SR} = M_{NR} + N_R \times |R_E| + N_C \times |CP_E| \\ + N_K \times |KHN_E| + OS \tag{4}$$

where a Route Entry (R_E) contains at least target ID (16 bytes), next hop ID (16 bytes), next hop MOP (4 bits) and path lifetime (1 byte), and therefore, $|R_E| \geq 33.5$ bytes. A Child-Parent Entry (CP_E) contains at least child ID (16 bytes), parent ID (16 bytes) and path lifetime (1 byte), and therefore, $|CP_E| \geq 33$ bytes. A K-Hop Neighbor Entry (KHN_E) contains at least neighbor ID (16 bytes), routing preference (2 bytes), hop count (1 byte), and timestamp (4 bytes), and therefore, $|KHN_E| \geq 23$ bytes. OS is other required memory by a storing router. To be a storing router, a node needs to have at least 1 parent, store at least one R_E or CP_E and at least 2 KHN_E. Therefore, the minimum routing memory for a storing router (M_{SR}^{min}) is at least 140 bytes.

For a multicast storing router to support downward multicast, besides the RRM required by a storing router, the node needs additional multicast memory (M_M) to store downward multicast information such as multicast address and members of the multicast groups. Therefore, the RRM for a multicast storing router (M_{MSR}) can be estimated as

$$M_{MSR} = M_{SR} + |M_M| + OMS \tag{5}$$

where OMS is other required memory by a multicast storing router. The minimum routing memory for a multicast storing router (M_{MSR}^{min}) is at least 140 bytes plus the minimum multicast memory.

Using equations (1) to (5), a node can define five RRM thresholds $RRM_L^{min}(\geq M_L^{min})$, $RRM_{UR}^{min}(\geq M_{UR}^{min})$, $RRM_{NR}^{min}(\geq M_{NR}^{min})$, $RRM_{SR}^{min}(\geq M_{SR}^{min})$ and $RRM_{MSR}^{min}(\geq M_{MSR}^{min})$. Based on the RRM thresholds and its ARM, a node can determine the memory based MOP MOP_M as

$$MOP_M = \begin{cases} -1 & \text{if } RRM_L^{min} \leq ARM < RRM_{UR}^{min} \\ 0 & \text{if } RRM_{UR}^{min} \leq ARM < RRM_{NR}^{min} \\ 1 & \text{if } RRM_{NR}^{min} \leq ARM < RRM_{SR}^{min} \\ 2 & \text{if } RRM_{SR}^{min} \leq ARM < RRM_{MSR}^{min} \\ 3 & \text{if } RRM_{MSR}^{min} \leq ARM \end{cases} \tag{6}$$

The MOP_M can be used if a node is only constrained on memory or the other resource information is not available.

C. Expected Routing Lifetime

The expected routing lifetime (ERL) is defined as the time a node can act as a router. When routing lifetime is up, the node acts as a leaf. In a heterogeneous network, nodes may have different power sources and different battery capacities. A mains powered node is considered to have infinite routing lifetime. However, a battery powered node has finite routing lifetime. Due to battery capacity and workload variation, the residual energy percentage does not reflect the battery lifetime.

For a mains powered node, the ERL depends on the size of the parent set (PS) only. If $|PS| > 0$, the ERL is defined as ∞. Otherwise, the ERL is define as 0.

For a battery powered node to compute the ERL, the node first decides a leaf lifetime (LLT), within which the node acts as leaf only. The ERL is defined as the battery lifetime minus the leaf lifetime. Assume E_i is residual battery energy at time t_i ($i = 0, 1, 2, ...$) with E_0 being initial energy. The battery lifetime at time t_i (BLT_i) can be estimated as

$$BLT_i = \frac{E_i \times (t_i - t_{i-1})}{E_{i-1} - E_i} \tag{7}$$

and therefore, the ERL for a battery powered node at time t_i is defined as

$$ERL = \begin{cases} BLT_i - LLT & \text{if } |PS| > 0 \\ 0 & \text{if } |PS| = 0 \end{cases} \tag{8}$$

A node can define four ERL thresholds $ERL_{UR}^{min} = 0$, ERL_{NR}^{min}, ERL_{SR}^{min} and ERL_{MSR}^{min}. Using the ERL thresholds and the ERL, the node can determine the lifetime based MOP (MOP_{LT}) as

$$MOP_{LT} = \begin{cases} -1 & \text{if } ERL \leq ERL_{UR}^{min} \\ 0 & \text{if } ERL_{UR}^{min} < ERL \leq ERL_{NR}^{min} \\ 1 & \text{if } ERL_{NR}^{min} < ERL \leq ERL_{SR}^{min} \\ 2 & \text{if } ERL_{SR}^{min} < ERL \leq ERL_{MSR}^{min} \\ 3 & \text{if } ERL_{MSR}^{min} < ERL \end{cases} \tag{9}$$

The MOP_L can be used if a node is only constrained on battery lifetime or other resource information is unavailable.

D. Node MOP Determination

Based on the random access memory and the flash memory of the node, the RFC 7228 [8] defines three classes of constrained nodes. The definitions can be used for initial MOP configuration since initially, nodes can not estimate the RRM and the ERL due to lack of topology information such as their positions in H-DODAG. Once network operation starts, nodes can determine their MOPs using the RRM and the ERL as

$$MOP = \begin{cases} -1 & \text{if } MOP_M = -1 \, or \, MOP_{LT} = -1 \\ 0 & \text{if } MOP_M = 0 \, and \, MOP_{LT} \geq 0 \\ 0 & \text{if } MOP_M \geq 0 \, and \, MOP_{LT} = 0 \\ 1 & \text{if } MOP_M = 1 \, and \, MOP_{LT} \geq 1 \\ 1 & \text{if } MOP_M \geq 1 \, and \, MOP_{LT} = 1 \\ 2 & \text{if } MOP_M = 2 \, and \, MOP_{LT} \geq 2 \\ 2 & \text{if } MOP_M \geq 2 \, and \, MOP_{LT} = 2 \\ 3 & \text{if } MOP_M \geq 3 \, and \, MOP_{LT} \geq 3 \end{cases} \tag{10}$$

A node in H-DODAG topology can change its MOP. The MOP downgrade is to avoid traffic congestion caused by the isolated higher MOP node and the MOP upgrade is to fully capitalize the renewed resources and/or the resource requirement relaxation.

IV. H-DODAG CONSTRUCTION AND MAINTENANCE

This section presents the H-DODAG construction and maintenance based on the mixed MOPs, heterogeneous RMs and OFs. We mainly describe new mechanisms proposed in the H-RPL. Due to the limited space, this section restricts MOP to -1, 0, 1, and 2. We use OF1, OF2 and a set of RMs to illustrate H-DODAG construction and maintenance. The root specifies OF1, OF2, RMs and one-to-many maps MP1 and MP2, which map each OF to the corresponding RMs. The first set {OF1, MP1} is used to construct the first tier H-DODAG topology and the second set {OF2, MP2} is used to construct second tiers of H-DODAG topology. The root selects OFs and RMs based on overall network consideration and the collected K-Hop neighbor routing preferences, respectively. The second set {OF2, MP2} can be modified by the storing nodes later.

The root starts H-DODAG construction by broadcasting a DIO message with a new sub H-DODAG ID field set to its ID. If its MOP = 2, besides the routing table, the root also maintains a child parent table. After transmitting the DIO message, the root receives and processes DAO messages. For N-DAOs or S-DAOs sent or forwarded by storing children, the root adds downward route entries into routing table and for N-DAOs sent or forward by non-storing children, the root adds child-parent entries into child parent table.

To construct H-DODAG topology, nodes with MOP $\geq$ 0 propagate DIO to extend network coverage. When a node receives a DIO, it processes DIO to get OF1, OF2, RMs, one-to-many maps MP1, MP2, and the MOP of the DIO transmitter. It is possible that the second set {OF2, MP2} is different in DIOs received from different transmitters. A node processes DIOs based on its MOP. (i) If MOP = -1, the node first selects parents with higher MOPs based on its routing preference. The node then selects a default parent with the highest MOP. (ii) If MOP = 0, the node processes DIOs same as the node with MOP = -1 does. In addition, the node computes a rank by using {OF2, MP2} from default parent and transmits DIOs containing its MOP with the set {OF2, MP2} and sub H-DODAG ID same as those of corresponding parents. (iii) If MOP = 1, the node processes DIO same as the node with MOP = 0 does. In addition, the node degrades its MOP to 0 if the MOP of the default parent is 0. In this case, the node is called acting upward node. Finally, if its MOP is not degraded, the node constructs a N-DAO including its MOP and forwards the N-DAO to its default parent. (iv) If MOP = 2, the node selects storing parents using the set {OF1, MP1} and non-storing or upward parents using the set {OF2, MP2}. The node selects a default parent with the highest MOP. It computes rank and adjusts its MOP based on the MOP of the default parent. (a) If the MOP of the default parent is 0, the node degrades its MOP to 0 to be an acting upward node and computes rank using {OF2, MP2} from default parent. In this case, the node does not transmit DAO. (b) If the MOP of the default parent is 1, it degrade its MOP to 1 to be an acting non-storing node and computes rank using {OF2, RMs} from default parent. In this case, the node send a N-DAO with its MOP to default parent. (c) If the MOP of the default parent is 2, it does not degrade its MOP and computes rank using {OF1, MP1} from default parent. Most importantly, the node replaces sub H-DODAG ID with its own ID in DIOs and if necessary, the node also modifies OF2 and the corresponding RMs in DIOs based on its K-HOP neighbor routing preferences. The node sends S-DAOs to the preferred storing parents. The node may also send a N-DAO to a preferred non-storing parent.

The H-RPL shits workload from lower MOP nodes to higher MOP nodes. In the H-DODAG construction process, the nodes with MOP = -1 or 0 should not receive DAO. The non-storing nodes and acting non-storing nodes receive and forward N-DAOs. In fact, they should only receive N-DAOs. The storing nodes receive and process N-DAOs and S-DAOs. In addition, storing nodes also maintain a routing table and a child parent table. A storing node adds a downward route entry into routing table for each destination contained in N-DAOs or S-DAOs sent or forwarded by storing children and adds a child-parent entry into child parent table for each destination contained in N-DAOs sent or forwarded by non-storing children.

The H-DODAG is maintained based on resource availability and resource requirement. Nodes may change their MOPs. The MOP decrease can be caused by the required resource increase, neighbor MOP decrease, neighbor unreachability or node resource decrease. The MOP increase can be caused by the required resource decrease, neighbor MOP increase, new neighbor availability or node resource increase. Once a node changes its MOP, it must announce new MOP and its neighbors, especially children, may need to adjust their MOPs, select new parents and/or default parents accordingly. If the default parent becomes unreachable or changes MOP to -1, children must remove this parent from their parent sets, a child with no backup parent must degrade its MOP to -1, non-storing child, acting non-storing child and storing child having no backup parent with MOP $\geq$ 1 must degrade MOP to 0, storing child with only non-storing backup parent must degrade MOP to 1. If the default parent changes MOP to 0, non-storing child, acting non-storing child and storing child having no backup parent with MOP $\geq$ 1 must degrade MOP to 0, storing child with only non-storing backup parent must degrade MOP to 1. If the default parent changes MOP to 1, storing child with only non-storing backup parent must degrade MOP to 1, acting upward child may upgrade its MOP to 1. If the default parent changes MOP to 2, acting upward child may upgrade MOP to 1 or 2, acting non-storing child may upgrade MOP to 2. If a backup parent changes MOP to -1, children must remove this parent from their parent sets. If a backup parent upgrades its MOP, children may update their parent sets and upgrade their MOPs accordingly. If a non-parent neighbor upgrades its MOP, a node may add this neighbor into parent set and upgrade its MOP. If the routing resource requirement relaxes, a node may upgrade its MOP. If the routing resource requirement increases, a node may downgrade its MOP.

V. DATA TRANSMISSION IN H-RPL NETWORK

In multipath routing, packet transmission has flexibility. A RPL node sends all upward packets to default parent without considering queue overflow. For downward data, RPL uses either source routing or routing table depending the MOP of the network. A H-RPL node sends upward packets to parents based on task classification. For same task, a H-RPL node distributes upward packets to the corresponding parents based on the queue utilization. H-RPL uses combination of the source routing and the routing table for downward data transmission.

A. *Queue Utilization Based Upward Data Transmission*

The queue utilization is critical to avoid packet drop due to queue overflow. Authors in [9] use the queue utilization as a routing metric in parent selection and rank computation to balance load in RPL network. However, a RPL node may not perform parent selection and rank computation in a long time period. As a result, the queue utilization used may be stale. To utilize fresh queue utilization information in data transmission, we introduce the queue occupancy index (QOI), which is defined as the average number of packets queued divided by the queue size. Each parent computes its QOI and transmits the QOI piggybacked in other packets or in separate packets. The child monitors QOIs of the parents and ranks parents according to QOIs. The parent with higher QOI gets a lower ranking and the parent with lower QOI gets a higher ranking. The child distributes upward packets to parents proportional to the ranking of the parents.

Let NoP_i be the number of packets pushed into queue in time period $[t_{i-1}, t_i]$ and QT_i be the average amount of time a packet queued in $[t_{i-1}, t_i]$, then using Little's theorem, the average number of packets queued can be estimated as $\frac{QT_i \times NoP_i}{t_i - t_{i-1}}$. Let Q_S be the queue size, the QOI can be estimated as

$$QOI = \frac{QT_i \times NoP_i}{Q_S \times (t_i - t_{i-1})} \tag{11}$$

B. *Mixed Mode Downward Data Transmission*

In a H-DODAG topology, the root can send downward packets to nodes with MOP = 1, 2, or 3. A downward packet can be delivered via routing table, source routing or the mixed routing table and source routing. Since the first tier of H-DODAG topology is formed by the root and storing nodes, the root sends downward packets to storing nodes using routing table. For non-storing nodes and acting non-storing nodes, the root uses routing table if the next hop is a storing node or source routing if the next hop is a non-storing node. To relay downward packets, non-storing nodes uses source routes carried in the packets and storing nodes use routing table or source routing similarly as the root does. The root and storing nodes use the child parent table to construct source routes.

VI. H-RPL PERFORMANCE EVALUATION

This section presents performance evaluation of the H-RPL protocol using the NS2 simulator with IEEE 802.15.4 MAC and PHY. 500 nodes are deployed in a 690m×660m rectangle area with the root at the center. The nodes are configured with different memory sizes and power sources. Non-storing nodes can buffer 5 packets and are battery powered. Storing nodes can buffer 10 packets and are mains powered. We show the results with 15% of storing nodes, which are randomly deployed. The standard RPL is used as benchmark for comparison. The performance metrics are upward data packet delivery rate (PDR), downward data PDR and battery energy comsuption. We simulated a bi-directional traffic scenario. For upward traffic, each node sends 1 packet per second to the root. For downward traffic, the root sends 20 packets per second and packets are uniformly distributed to 500 nodes. The payload is 50 bytes. The simulation runs 10000 seconds. The ETX and hop count metrics are used with the minimum ETX and the shortest path as objective functions, respectively.

A. *Data Packet Delivery Rate*

The data PDR is key performance metric for a routing protocol. Fig.1 illustrates variation of the upward PDR with respect to simulation time. H-RPL operates on the mixed MOP = 1 and 2, and achieves 96.34% of PDR by fully utilizing the extra memory provided by storing nodes. Presence of the small memory nodes causes RPL to operate on MOP = 1 to avoid network partition. Therefore, RPL ignores extra memory provided by storing nodes. As a result, RPL obtains 89.17% of PDR due to queue overflow of small memory nodes.

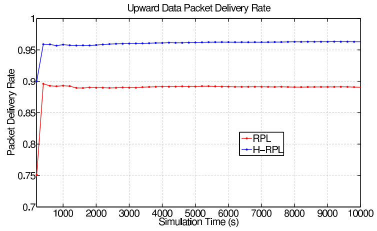

Fig. 1: Upward Data Packet Delivery Rate

Fig.2 and Fig.3 show 3D surfaces of the downward data PDR by RPL and H-RPL, respectively. The RPL obtains 72% of average delivery rate and 1% of the minimum delivery rate. Operating on MOP = 1, RPL uses full length source route that causes packet fragmentation and traffic congestion. On the other hand, the H-RPL uses routing table as much as possible. Packet size and chance of packet fragmentation are much smaller. As a result, H-RPL achieves 97% of the average delivery rate and 74% of the minimum delivery rate.

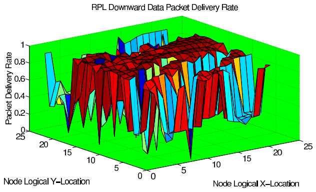

Fig. 2: RPL Downward PDR

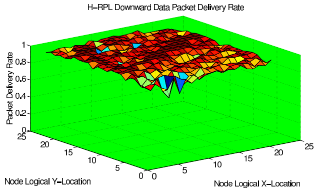

Fig. 3: H-RPL Downward PDR

B. *Battery Energy Consumption*

Fig.4 and Fig.5 show the detailed energy consumption of the battery nodes. For both RPL and H-RPL, battery node consume 3.88% of energy on idle since battery nodes do not sleep. Battery nodes in the central area consume more energy on packet transmission (TX) and receiving (RX) since the root is placed at the center. The RX energy consumption is higher than the TX energy consumption due to overhearing. However, the maximum RX energy consumption is 10.3% for RPL and is 5.2% for H-RPL and the maximum TX energy consumption is 3.6% for RPL and is 0.45% for H-RPL. These results show that H-RPL shifts more routing workload to mains powered nodes. As a result, total energy consumption of battery nodes in RPL is higher compared with H-RPL as shown in Fig.6. The maximum battery energy consumption is 15.2% for RPL and is 8.5% for H-RPL. Fig.7 illustrates the minimum battery energy level of battery nodes with respect to time, which indicates the network lifetime. With H-RPL, the network can run 32.68 hours. With RPL, network can only run 18.27 hours. These results indicate that H-RPL extends network lifetime by 78%.

VII. CONCLUSION

We propose a resource aware hierarchical routing protocol called H-RPL for heterogeneous wireless IoT networks containing nodes with different resources. The H-RPL supports the mixed MOPs in a network. The nodes use their MOPs to signal their routing types. A new MOP is introduced to indicate the critical resource condition of the node. The MOP is dynamically determined based on the available resources, the required resources and the MOPs of the neighboring nodes. The required routing memory and the expected routing lifetime are proposed to compute the MOP. Heterogeneous routing metrics and objective functions are used at different tiers and different portions of the H-DODAG topology construction. The H-RPL shifts routing workload from the nodes with lower MOPs to the nodes with higher MOPs. In addition, a queue utilization based upward data packet transmission method is introduced to distribute upward traffic and balance workload. The mixed MOPs reduce downward packet overhead by shortening source routes. An application scenario with upward data collection and downward service providing is simulated using NS2 simulator. Simulation results show that H-RPL outperforms standard RPL in terms of upward data packet delivery rate, downward data packet delivery rate and battery energy consumption. With 15% of large memory nodes and mains powered nodes, H-RPL can improve upward data packet delivery rate by 7% and downward service packet delivery rate by 25% and extend network lifetime by 78%.

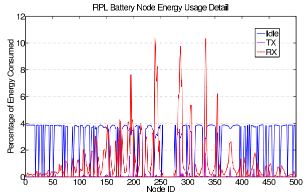

Fig. 4: RPL Energy Usage Detail

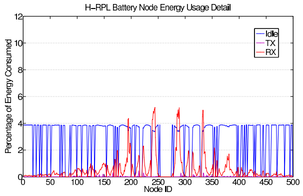

Fig. 5: H-RPL Energy Usage Detail

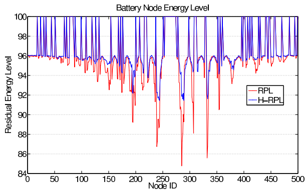

Fig. 6: Residual Energy Level

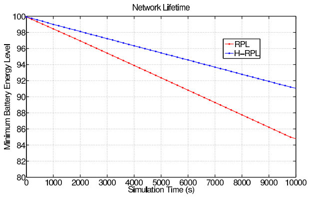

Fig. 7: Network Lifetime

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, 2012.

[2] J. Ko, J. Jeong, and et al, "DualMOP-RPL: Supporting Multiple Modes of Downward Routing in a Single RPL Network," in *ACM Transactions on Sensor Network*, 2015.

[3] J. Guo, P. Orlik, and et al, "Resource Aware Routing Protocol in Heterogeneous Wireless Machine-to-Machine Networks," in *IEEE Global Communications*, 2015.

[4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrisshnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *Proceedings of the 33th Hawaii International Conference on System Sciences 2000*, 2000.

[5] R. S. Bisht, M. C. Lohani, and S. K. Budhani, "Energy Efficient Hierarchical Routing Protocol for Wireless Sensor Network," in *International Journal of Computer Applications*, 2014.

[6] J. Wang, X. Yang, and et al, "An Energy-Efficient Multi-hop Hierarchical Routing Protocol for Wireless Sensor Networks," in *International Journal of Future Generation Communication and Networking*, 2012.

[7] S. Tyagi, S. Tanwar, and et al, "Bayesian Coalition Game-Based Optimized Clustering in Wireless Sensor Networks," in *IEEE International Conference on Communications*, 2015.

[8] C. Bormann, M. Ersue, and A. Keranen, "Terminology for Constrained-Node Networks," http://tools.ietf.org/html/rfc7228, 2014.

[9] H.-S. Kim, J. Paek, and S. Bahk, "QU-RPL: Queue Utilization Based RPL for Load Balancing in Large Scale Industrial Applications," in *IEEE International Conference on Sensing, Communication, and Networking*, 2015.