Practice Problem

21_AIE_302

Advanced Computer Networks– SEM-V
Professor – Jaysooraj Sir

Submitted By: Vikhyat Bansal [CB.EN.U4AIE21076]

AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

RPL
(Routing Protocol for Low-Power and Lossy Networks)

INTRODUCTION

The Internet has evolved rapidly in the past few decades introducing countless applications in many fields including industry, transport, education, entertainment, etc. During these years, many devices, services and protocols were created and the Internet grew and is still exponentially. The next generation of this worldwide network is the IoT, where a large number of 'Things'. These things include sensor nodes, radio frequency identification (RFID) tags, near field communication (NFC) devices and other wired or wireless gadgets that interact with each other and with the existing network.

Low power and Lossy Networks (LLNs) consist of largely of constrained nodes (with limited processing power, memory, and sometimes energy when they are battery operated or energy scavenging). These routers are interconnected by lossy links, typically supporting only low data rates, that are usually unstable with relatively low packet delivery rates. Another characteristic of such networks is that the traffic patterns are not simply point-to-point, but in many cases point-to-multipoint or multipoint-to-point.

Wireless sensor networks (WSNs) play a key role in the creation and growth of the IoT. One of the main standards that supports low power and lossy networks (LLNs) is the IEEE 802.15.4 standard, which forms the backbone of WSNs as part of the IoT. This standard defines the physical and data-link layers of the network and provides a framework of operation at low costs.

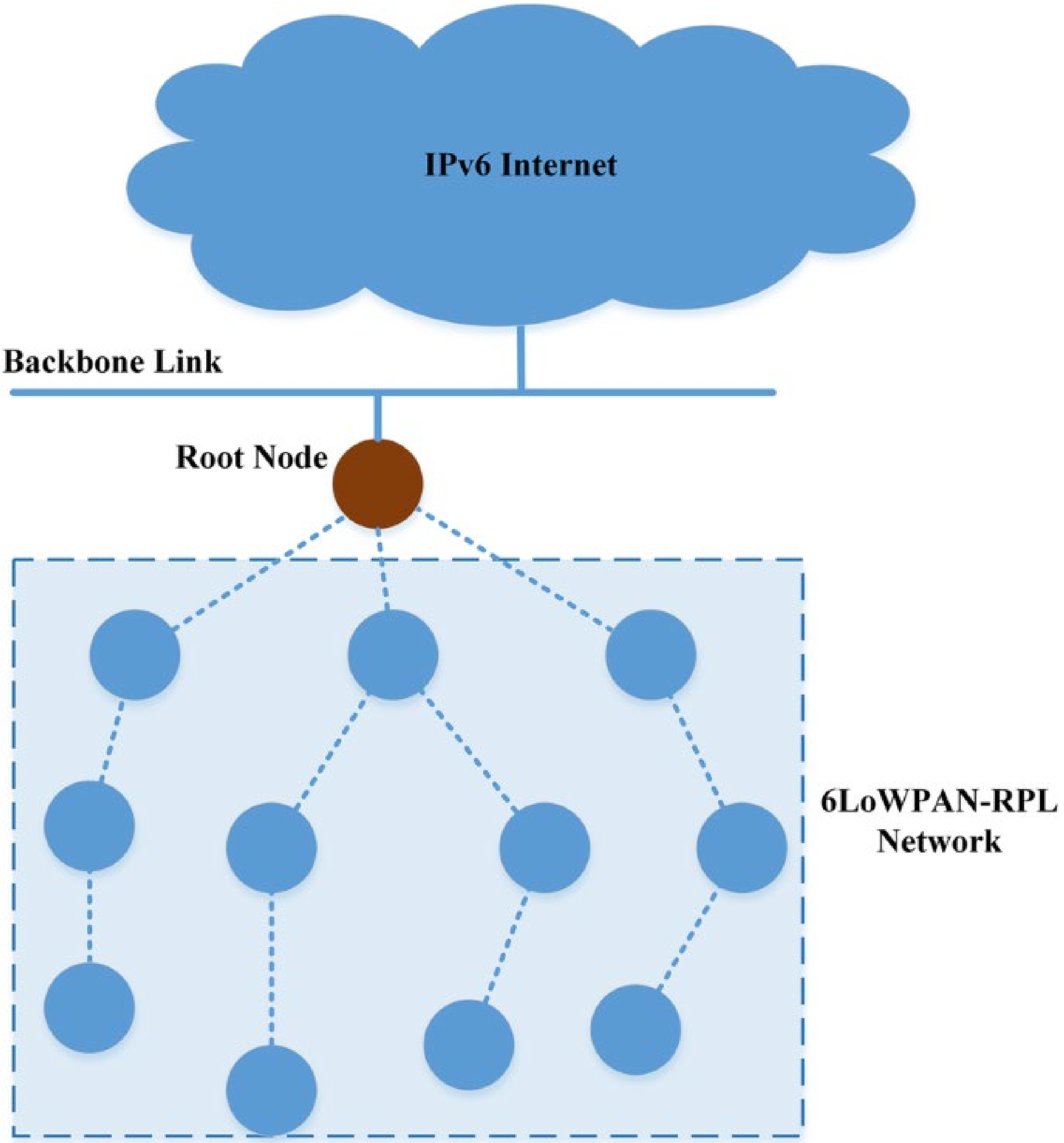
To make these low end devices a part of the Internet, the IETF developed the IPv6 low-power wirless personal area networks (6LoWPAN) which is used as an adaptation layer that allows sensor nodes to implement the Internet protocol (IP) stack and become accessible by other devices on the network. This adaptation layers allows these nodes to implement routing protocols at the network layer and provide an end-toend connectivity that enables countless applications. With the exponential growth of the Internet and the evolution of IoT, conventional routing protocols can no longer accommodate the large number of added nodes. For this reason, RPL was designed especially for LLNs and quickly gained popularity among the research community.

The RPL protocol is a distance vector proactive routing protocol that creates a tree-like routing topology called the destination-oriented directed acyclic graph (DODAG), rooted towards one or more nodes called the root node or sink node. The directed acyclic graphs (DAGs) are created based on a user-specified specific objective function (OF).

It operates on the IEEE 802.15.4 standard with the support of 6LoWPAN adaptation layer. The routing over LLNs (RoLL) working group introduced the routing requirements for LLNs in general taking into account the resources limitations in terms of energy, processing and memory in a vision to allow large number of nodes to communicate in a peer-to-peer topology or an extended star topology . This protocol creates a multi-hop hierarchical topology for nodes, where each node can send data to its parent node which in turn forwards it upward until it reaches the sink or gateway node.

In the same way, the sink node can send a unicast message to target a specific node in its network. RPL successfully and efficiently manages data routing for nodes that have restricted resources, it provides an operation framework that ensures bidirectional connectivity, robustness, reliability, flexibility and scalability. The key features of RPL come from its efficient hierarchy, the use of timers to minimise control messages and the flexibility of the objective function.

In compliance with the layered architecture of IP, RPL does not rely on any particular features of a specific link layer technology. RPL is designed to be able to operate over a variety of different link layers, including ones that are constrained, potentially lossy, or typically utilized in conjunction with highly constrained host or router devices, such as but not limited to, low power wireless or PLC (Power Line Communication) technologies.


IPv6 Internet
Backbone Link
Root Node
6LoWPAN-RPL Network

Control Messages used in RPL

A RPL Control Message is identified by a code, and composed of a base that depends on the code, and a series of options.

Most RPL Control Message have the scope of a link. The only exception is for the DAO / DAO-ACK messages in non-storing mode, which are exchanged using a unicast address over multiple hops and thus uses global or unique-local addresses for both the source and destination addresses. For all other RPL Control messages, the source address is a link-local address, and the destination address is either the all-RPL-nodes multicast address or a link-local unicast address of the destination. The all-RPL-nodes multicast address is a new address with a requested value of FF02::1A (to be confirmed by IANA).

The RPL Control Message consists of an ICMPv6 header followed by a message body. The message body is comprised of a message base.

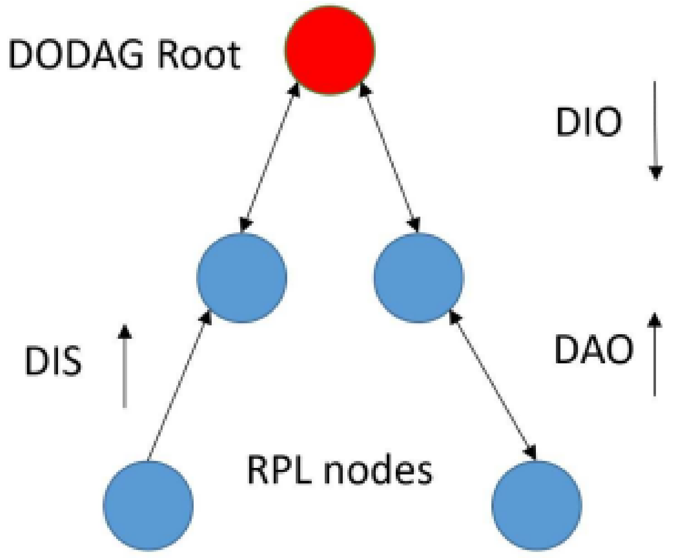


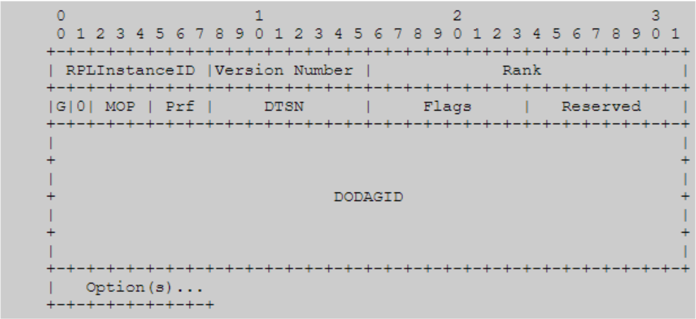
Fig. 1. Control messages in RPL

DODAG Information Object (DIO)

The DODAG Information Object carries information that allows a node to discover a RPL Instance, learn its configuration parameters, select a DODAG parent set, and maintain the DODAG.

The DODAG information object (DIO) is sent from the root node with information about the rank of the sending node, the instance ID, the version number and the DODAG-ID. This allows nodes to decide whether or not to act upon receiving this message, in addition to keeping valuable information about the network that can contribute to making an informed decision.

6.3.1. Format of the DIO Base Object



```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | RPLInstanceID |Version Number |             Rank              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |G|0| MOP | Prf |     DTSN      |     Flags     |   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +                                                               +
   |                                                               |
   +                            DODAGID                            +
   |                                                               |
   +                                                               +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Option(s)...
   +-+-+-+-+-+-+-+-+
```

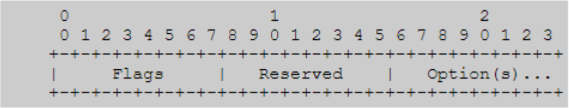
Figure 14: The DIO Base Object

DODAG Information Solicitation (DIS)

The DODAG Information Solicitation (DIS) message may be used to solicit a DODAG Information Object from a RPL node. Its use is analogous to that of a Router Solicitation as specified in IPv6 Neighbor Discovery; a node may use DIS to probe its neighborhood for nearby DODAGs.

The DODAG information solicitation is another form of upward control messages that is used to request a DIO from the parent node, this is one of the most relevant and important features that RPL uses to maintain connectivity.

6.2.1. Format of the DIS Base Object



```
    0                   1                   2
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Flags     |   Reserved    |   Option(s)...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: The DIS Base Object

Flags:
    8-bit unused field reserved for flags. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Reserved:
    8-bit unused field. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

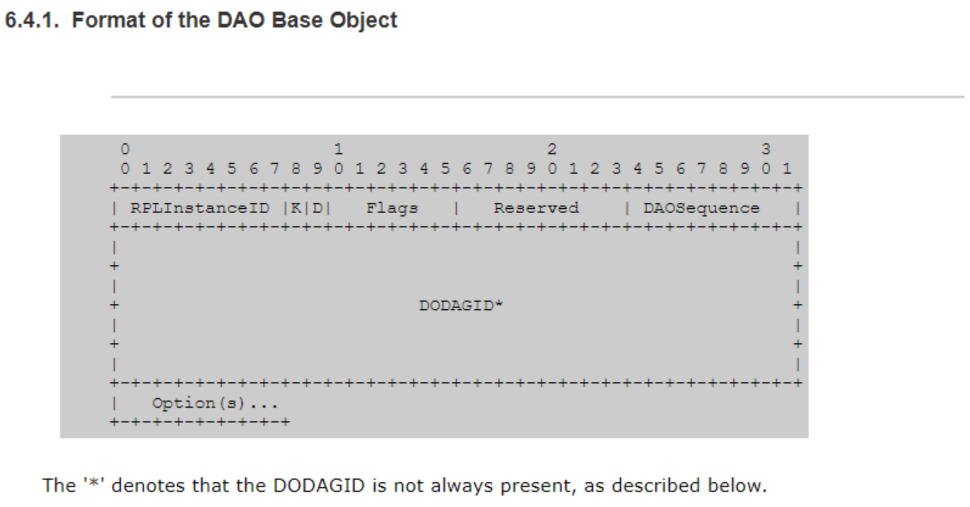
Unassigned bits of the DIS Base are reserved. They MUST be set to zero on transmission and MUST be ignored on reception.

Destination Advertisement Object (DAO)

The Destination Advertisement Object (DAO) is used to propagate destination information upwards along the DODAG. In storing mode the DAO message is unicast by the child to the selected parent(s). In non-storing mode the DAO message is unicast to the DODAG root. The DAO message may optionally, upon explicit request or error, be acknowledged by its destination with a Destination Advertisement Acknowledgement (DAO-ACK) message back to the sender of the DAO.

The destination advertisement object (DAO) is sent from the child node to its parent (the DAG root or the DODAG root) and it contains destination

information which practically informs the root that this node is still available.



6.4.1. Format of the DAO Base Object

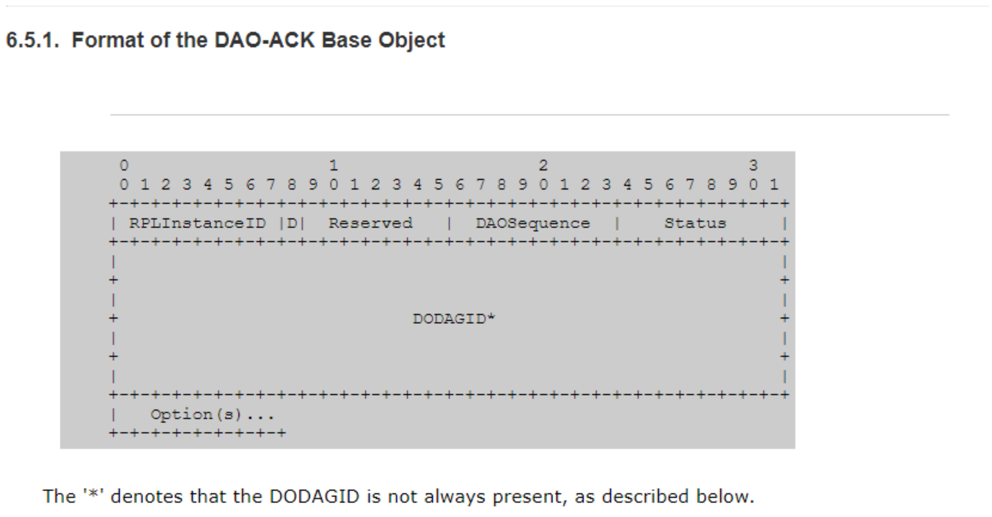
```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | RPLInstanceID |K|D|   Flags   |   Reserved    | DAOSequence   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    +                                                               +
    |                                                               |
    +                            DODAGID*                           +
    |                                                               |
    +                                                               +
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Option(s)...
    +-+-+-+-+-+-+-+-+
```

The '*' denotes that the DODAGID is not always present, as described below.

Figure 16: The DAO Base Object

Destination Advertisement Object Acknowledgement (DAO-ACK)

The DAO-ACK message is sent as a unicast packet by a DAO recipient (a DAO parent or DODAG root) in response to a unicast DAO message. The root node may optionally send a DAOack acknowledgement if required for DAO.



6.5.1. Format of the DAO-ACK Base Object

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | RPLInstanceID |D|  Reserved   |  DAOSequence  |    Status     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    +                                                               +
    |                                                               |
    +                            DODAGID*                           +
    |                                                               |
    +                                                               +
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   Option(s)...
    +-+-+-+-+-+-+-+-+
```

The '*' denotes that the DODAGID is not always present, as described below.

Figure 17: The DAO ACK Base Object

Working of RPL

The DODAG is a directed graph with no loops oriented towards a root node. The nodes that provide Internet access (gateways) are called root nodes, and the other nodes in the network are linked with it either directly or indirectly through a sequence of parent nodes. Furthermore, each node is responsible for selecting the desired parent, who is then used for forwarding the application packets. The parent node selection depends on the rank value that a device (node) can achieve. Moreover, the rank value refers to the position of a node in the DODAG. Hence, the rank value is affected by the node’s distance from the root and the Objective Function (OF). The OF determines numerous metrics, such as rank of nodes, selection of the parent node, and route optimization. The versatility of RPL to interact with many limited devices is the primary reason for its adoption in LLNs.

The directed acyclic graphs (DAGs) are created based on a user-specified specific objective function (OF). The OF defines the method to find the best-optimized route among the number of sensor devices.

The IETF ROLL working group standardized the objective function zero (OF0) and the minimum rank with hysteresis objective function (MRHOF) as default routing metrics.

Each RPL node, has its predefined objective function (OF), this function carries the metrics upon which nodes select the ”better” parent among competing nodes. There are currently two objective functions presented by the IETF, the first one is Objective Function zero (OF0) which is a simple and basic objective function that has only one metric, it uses the rank of the node to determine its distance from the root and selects the node with the lower (better) rank. The OF0 is designed as a general objective function used as a guide and base for other implementations. The second one and the arguably most popular one is the minimum rank with hysteresis objective function (MRHOF) which is based on routing metric containers. It allows the user to configure the metrics inside the metric container which is transmitted as part of DIO messages.

This function uses the expected transmission count (ETX) as the default metric and provides support for using path-specific expected energy consumption as a routing metric.

The OF0 finds the shortest path to the sink node by selecting the candidate parent node with minimum rank in terms of the distance from the sink (i.e., its position in the routing tree). The MRHOF finds the routes through the sensor nodes that minimize the link cost associated with the routes. It selects the new routing path if the cost associated with it is less than the current path cost by a given threshold value.

This is known as ‘hysteresis.’ As prescribed in the standard, MRHOF utilizes the expected transmission count (ETX) metric which calculates the link quality. Furthermore, ETX considers link-layer congestion but does not reflect node level congestion. Therefore, selecting a routing path based on the smallest hop count and link quality in a heterogeneous traffic environment does not lead to an efficient load-balancing solution. The nodes closer to the sink node suffer from packet loss due to a high relay burden in a dense networking environment. The chosen parent node in RPL can have multiple child nodes; consequently, the overloaded preferred parent becomes prone to failure because its energy drains much faster than other nodes. The inefficient OFs lead to building a routing topology that experiences an excessively unbalanced load and energy distribution, particularly for those nodes that are closest to the sink node.

RPL uses a specific OF that defines the use of specific metrics for rank calculation to make an intelligent routing decision. It constructs a tree-like routing topology called DODAG. The primary objectives of the RPL protocol are to create an optimal DODAG and to adapt the topology with respect to various network situations. To manage various routing challenges, the routing metrics must consider different constraints and requirements of the LLN network. The RPL protocol does not specify any particular OF or routing metric to be used. Rather, it provides a great degree of flexibly for defining any OF based on network applications and design.

Traditionally, ETX in MRHOF and hop count in OF0 are provided by the RPL standard implementation. The OF0 implementation does not use any specific routing metric for rank calculation other than hop count. MRHOF accesses the link quality between the nodes to select the preferred parent node. This approach is more efficient and is favored in RPL models.

RANK OF A NODE

The rank of a node is a scalar representation of the location of that node within a DODAG Version. The rank is used to avoid and detect loops, and as such must demonstrate certain properties. The exact calculation of the rank is left to the Objective Function. Even though the specific computation of the rank is left to the Objective Function, the rank must implement generic properties regardless of the Objective Function.

In particular, the rank of the nodes must monotonically decrease as the DODAG version is followed towards the DODAG destination. In that regard, the rank can be regarded as a scalar representation of the location or radius of a node within a DODAG Version.

The details of how the Objective Function computes rank are out of scope for this specification, although that computation may depend, for example, on parents, link metrics, node metrics, and the node configuration and policies. See Section 14 for more information.

The rank is not a path cost, although its value can be derived from and influenced by path metrics. The rank has properties of its own that are not necessarily those of all metrics:

Type:

The rank is an abstract numeric value.

Function:

The rank is the expression of a relative position within a DODAG Version with regard to neighbors and is not necessarily a good indication or a proper expression of a distance or a path cost to the root.

Stability:

The stability of the rank determines the stability of the routing topology. Some dampening or filtering is RECOMMENDED to keep the topology stable, and thus the rank does not necessarily change as fast as some link or node metrics would. A new DODAG Version would be a good opportunity to reconcile the discrepancies that might form over time between metrics and ranks within a DODAG Version.

Properties:

The rank is incremented in a strictly monotonic fashion, and can be used to validate a progression from or towards the root. A metric, like bandwidth or jitter, does not necessarily exhibit this property.

Abstract:

The rank does not have a physical unit, but rather a range of increment per hop, where the assignment of each increment is to be determined by the Objective Function.

The rank value feeds into DODAG parent selection, according to the RPL loop-avoidance strategy. Once a parent has been added, and a rank value for the node within the DODAG has been advertised, the node's further options with regard to DODAG parent selection and movement within the DODAG are restricted in favor of loop avoidance.

Rank Comparison (DAGRank())

Rank may be thought of as a fixed point number, where the position of the radix point between the integer part and the fractional part is determined by MinHopRankIncrease. MinHopRankIncrease is the minimum increase in rank between a node and any of its DODAG parents. A DODAG Root provisions MinHopRankIncrease. MinHopRankIncrease creates a tradeoff between hop cost precision and the maximum number of hops a network can support. A very large MinHopRankIncrease, for example, allows precise characterization of a given hop's affect on Rank but cannot support many hops.

When an objective function computes rank, the objective function operates on the entire (i.e. 16-bit) rank quantity. When rank is compared, e.g. for determination of parent relationships or loop detection, the integer portion of the rank is to be used. The integer portion of the Rank is computed by the DAGRank() macro as follows, where floor(x) is the function that evaluates to the greatest integer less than or equal to x:

DAGRank(rank) = floor(rank/MinHopRankIncrease)

For example, if a 16-bit rank quantity is decimal 27, and the MinHopRankIncrease is decimal 16, then DAGRank(27) = floor(1.6875) = 1. The integer part of the rank is 1 and the fractional part is 11/16.

By convention in this document, using the macro DAGRank(node) may be interpreted as DAGRank(node.rank), where node.rank is the rank value as maintained by the node.

A node A has a rank less than the rank of a node B if DAGRank(A) is less than DAGRank(B).

A node A has a rank equal to the rank of a node B if DAGRank(A) is equal to DAGRank(B).

A node A has a rank greater than the rank of a node B if DAGRank(A) is greater than DAGRank(B).

Rank Relationships

Rank computations maintain the following properties for any nodes M and N that are neighbors in the LLN:

DAGRank(M) is less than DAGRank(N):

In this case, the position of M is closer to the DODAG root than the position of N. Node M may safely be a DODAG parent for Node N without risk of creating a loop. Further, for a node N, all parents in the DODAG parent set must be of rank less than DAGRank(N). In other words, the rank presented by a node N MUST be greater than that presented by any of its parents.

DAGRank(M) equals DAGRank(N):

In this case the positions of M and N within the DODAG and with respect to the DODAG root are similar (identical). Routing through a node with equal Rank may cause a routing loop (i.e., if that node chooses to route through a node with equal Rank as well).

DAGRank(M) is greater than DAGRank(N):

In this case, the position of M is farther from the DODAG root than the position of N. Further, Node M may in fact be in the sub-DODAG of Node N. If node N selects node M as DODAG parent there is a risk to create a loop.

As an example, the rank could be computed in such a way so as to closely track ETX (Expected Transmission Count, a fairly common routing metric used in LLN and defined in [I-D.ietf-roll-routing-metrics]) when the metric that an objective function minimizes is ETX, or latency, or in a more complicated way as appropriate to the objective function being used within the DODAG.

RPL INSTANCES

Within a given LLN, there may be multiple, logically independent RPL instances. A RPL node may belong to multiple RPL instances, and may act as a router in some and as a leaf in others. This document describes how a single instance behaves.

There are two types of RPL Instances: local and global. RPL divides the RPLInstanceID space between Global and Local instances to allow for both coordinated and unilateral allocation of RPLInstanceIDs. Global RPL Instances are coordinated, have one or more DODAGs, and are typically long-lived. Local RPL Instances are always a single DODAG whose singular root owns the corresponding DODAGID and allocates the Local RPLInstanceID in a unilateral manner. Local RPL Instances can be used, for example, for constructing DODAGs in support of a future on-demand routing solution. The mode of operation of Local RPL Instances is out of scope for this specification and may be described in other companion specifications.

The definition and provisioning of RPL instances are out of scope for this specification. Guidelines may be application and implementation specific, and are expected to be elaborated in future companion specifications. Those operations are expected to be such that data packets coming from the outside of the RPL network can unambiguously be associated to at least one RPL instance, and be safely routed over any instance that would match the packet.

Control and data packets within RPL network are tagged to unambiguously identify what RPL Instance they are part of.

Every RPL control message has a RPLInstanceID field. Some RPL control messages, when referring to a local RPLInstanceID as defined below, may also include a DODAGID.

For data packets coming from outside the RPL network, the RPLInstanceID is determined by the RPL network ingress router and placed in the IPv6 Hop-by-Hop RPL Option that is added to the packet.

RPL Instance ID

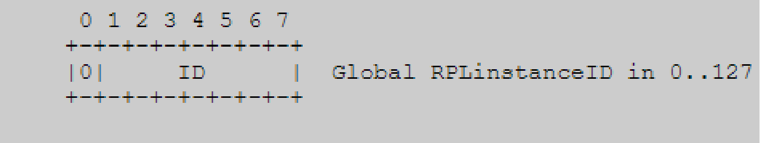
There are two type of RPL instance ID:

Global InstanceID

Local InstanceID

A global RPLInstanceID MUST be unique to the whole LLN. Mechanisms for allocating and provisioning global RPLInstanceID are out of scope for this specification. There can be up to 128 global instance in the whole network. Local instances are always used in conjunction with a DODAGID (which is either given explicitly or implicitly in some cases), and up 64 local instances per DODAGID can be supported. Local instances are allocated and managed by the node that owns the DODAGID, without any explicit coordination with other nodes, as further detailed below.

A global RPLinstanceID is encoded in a RPLinstanceID field as follows:

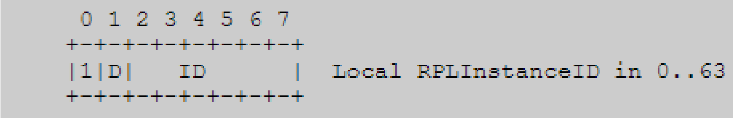


```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|0|     ID      |  Global RPLinstanceID in 0..127
+-+-+-+-+-+-+-+-+
```

Figure 4: RPL Instance ID field format for global instances

A local RPLInstanceID is autoconfigured by the node that owns the DODAGID and it MUST be unique for that DODAGID. The DODAGID used to configure the local RPLInstanceID MUST be a reachable IPv6 address of the node, and MUST be used as an endpoint of all communications within that local instance.

A local RPLinstanceID is encoded in a RPLinstanceID field as follows:



```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|1|D|   ID      |  Local RPLInstanceID in 0..63
+-+-+-+-+-+-+-+-+
```

Figure 5: RPL Instance ID field format for local instances

The D flag in a Local RPLInstanceID is always set to 0 in RPL control messages. It is used in data packets to indicate whether the DODAGID is the source or the destination of the packet. If the D flag is set to 1 then the destination address of the IPv6 packet MUST be the DODAGID. If the D flag is cleared then the source address of the IPv6 packet MUST be the DODAGID.

For example, consider a node A that is the DODAG Root of a local RPL Instance, and has allocated a local RPLInstanceID. By definition, all traffic traversing that local RPL Instance will either originate or terminate at node A. The DODAGID in this case will be the reachable IPv6 address of node A, and all traffic will contain the address of node A, thus the DODAGID, in either the source or destination address. Thus the Local RPLInstanceID may indicate that the DODAGID is equivalent to either the source address or the destination address by setting the D flag appropriately.

CHALLENGES/LIMITATIONS IN RPL

The main drivers for improving RPL are energy efficiency, mobility, Reliability,congestion and security.

Energy Consumption

Limited energy is one of the most significant problems that LLNs confront; the IEEE 802.15.4 and RPL designs both include energy consumption and offer ways to reduce it. The trickling timer is designed to alleviate the issue of energy usage in RPL by minimizing how many pointless control messages there are. Nevertheless, it has been demonstrated that the trickle timer has drawbacks of its own when it comes to dynamic situations. These drawbacks include inefficient data transfer and significant energy loss as a result of unsuccessful packet delivery.

When making recommendations for improving RPL, many studies consider energy usage; one popular method is to include energy as a routing metric in the goal function. According to a study, RPL in its original form is also energy-efficient, with nodes having a long lifespan. Based on simulations where nodes produced 40 packets per minute, these results were drawn. A different study that also used energy consumption as a criterion verified the findings that were already known, noting that larger networks and higher node densities are associated with higher energy consumption.

This is predicted given that nodes in these situations see an increase in noise and a higher volume of transmissions.

Mobility

Many studies have been conducted on routing for mobile WSNs, and within IoT applications, the majority of the latest research is based on RPL because it has emerged as the industry standard for IoT routing protocols. Using RPL as a standard makes it easy to create an interoperable solution for any application, allowing it to be used in the Internet of Things. RPL is a versatile and scalable routing protocol. Because of RPL's adaptable and scalable architecture, numerous attempts are being made to enhance and improve it. The fact that RPL does not support mobility is a clear drawback, hence many academics concentrate on developing ways to support mobile nodes.

Using a multipath method with redundant routes and a DODAG maintenance and repair technique, the DAG-based Multipath Routing for Mobile Sensor Networks (DMR) was created based on RPL with rank information and link quality identifier (LQI) as routing metrics. These techniques are already covered by RPL, albeit DMR performs better than the protocols for ad hoc on-demand multipath distance vector (AOMDV) and ad hoc on-demand distance vector (AODV), which were not intended for LLNs and were not contrasted with native RPL.

QoS

Most Internet of Things applications need reliable data transmission, which is accomplished by reducing packet loss, increasing throughput, and avoiding lengthy delays. Improved routing choices, maximized transmission rates, and effective topology repair are necessary to achieve high QoS. The authors describe a reactive method in which link quality updates are sent based on the quantity of received data packets rather than relying on control messages. By maintaining a list of various link quality measurements for nearby nodes, this technique increases the reliability of transmitted data by forcing nodes to switch parents in order to measure link quality.

To enhance link quality estimation in RPL, researchers suggested a cross-layer design. This algorithm also employs an adaptive approach to ensure dependable data transmission, low energy consumption, and a reduction in end-to-end delay in comparison to the native RPL. They also presented a way to use unicast DIS messages to update link quality data according to priority.

Fuzzy logic was used to introduce a novel objective function in that year. It divides the network into circular coronas around the DODAG root using a corona mechanism, making it easy for nodes to find a different parent without changing the DODAG. The routing metrics used by the fuzzy logic objective function (FL-OF) are end-to-end delay, hop count, link quality, and residual energy. This protocol achieves higher PDR, improved responsiveness and decreased energy consumption, it also has the ability to manage mobility at low speeds due to the corona mechanism.

Congestion

Congestion is one of the most difficult parts of multi-hop routing; as the number of hops rises, more data accumulates and causes congestion, particularly at the node level. Both the wireless channel and the nodes buffer become crowded when numerous nodes transmit at high rates, increasing the likelihood of congestion. Significant declines in energy consumption, dependability, and latency are caused by congestion. While there are several ways to address congestion, resource management, traffic management, and hybrid schemes are the most widely used ones.

In order to manage traffic in 6LoWPAN networks, researchers have proposed a duty cycle aware congestion control (DCCC6). This system uses RPL to manage routing and modifies traffic in response to RDC and buffer occupancy. Similarly, three congestion control schemes—Gripping, Deaf, and Fuse—were introduced by researchers. These schemes make use of queue length, buffer length, and hybrid forms of each.

Simulation results show that the last scheme (Fuse) performs better at managing congestion than the other two because it combines queue and buffer length.

Researchers present a congestion control algorithm that uses buffer occupancy to identify the least congested paths in resource control strategies. This proposal was compared to the CON and NON transactions in CoAP and was created for CoAP/RPL networks. When applied to noncongested networks, this strategy becomes counterproductive, but it enhances network performance in the presence of congestion. It's also important to note that this algorithm uses "eavesdropping," which involves listening in passively to received packets and using a lot of energy.

Security

Depending on the kind of application, the deployment location, and the sensitive nature of the data being transmitted, most Internet of Things applications demand a certain level of security. Generally speaking, integrity, confidentiality, availability, privacy, authentication, and trust are required of IoT applications. Because sensor nodes' hardware is relatively simple, there are many attacks that can easily target them in an attempt to gain an advantage by using their data for malicious purposes or by simply blocking their services. From a routing standpoint, denial of service (DoS), man-in-the-middle, spoofing, black hole, sink hole, worm hole, and Sybil attacks are the most frequent attacks that target sensor nodes.

A DOS attack that forces the trickle timer to reset by causing inconsistencies in the DODAG, this results in a loop of DODAG reformation and global repair. Attacks of this kind impede nodes from processing data packets and drain their energy reserves for unnecessary repairs. A proposed IETF standard looks into setting a limit on the quantity of trickle resets that are permitted each hour. While this approach does not address the issue of dropped packets, it does reduce the amount of energy required for DODAG reformation once the threshold is crossed. This concept was refined in another study [102], which suggested an adaptive threshold based on the attack type and network conditions. In terms of energy consumption, the strategy performs significantly better.

A study in proposed an intrusion detection system (IDS) to detect the problems of black hole and grey hole attacks where malicious nodes silently drop all or some of the data packets. The algorithm detects malicious nodes by monitoring the number of DIO messages, packet loss and delays. According to their results, this approach successfully prevents malicious nodes from participating in the DODAG formation process.

In case of a sink hole attack, where a node advertises itself with a high rank to attract data from neighbouring nodes, the authors in [104] propose an algorithm to use signed DIO messages to detect fake rank advertisements. The algorithm was also studied and improved by to cover

spoofing and replay attacks.

It is believed that RPL can significantly benefit from a new standard design that takes into account its current state and opens the door for new optimisation studies.

THANK YOU!!!