Internet Engineering Task Force (IETF)
Request for Comments: 6551
Category: Standards Track
ISSN: 2070-1721

JP. Vasseur, Ed.
Cisco Systems
M. Kim, Ed.
Corporate Technology Group, KT
K. Pister
Dust Networks
N. Dejean
Elster SAS
D. Barthel
France Telecom Orange

March 2012

Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks

Abstract

Low-Power and Lossy Networks (LLNs) have unique characteristics compared with traditional wired and ad hoc networks that require the specification of new routing metrics and constraints. By contrast, with typical Interior Gateway Protocol (IGP) routing metrics using hop counts or link metrics, this document specifies a set of link and node routing metrics and constraints suitable to LLNs to be used by the Routing Protocol for Low-Power and Lossy Networks (RPL).

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6551.

Copyright Notice

Table of Contents

1. Introduction

This document makes use of the terminology defined in [ROLL-TERMS].

Low-power and Lossy Networks (LLNs) have specific routing characteristics compared with traditional wired or ad hoc networks that have been spelled out in [RFC5548], [RFC5673], [RFC5826], and [RFC5867].

Historically, IGP, such as OSPF ([RFC2328]) and IS-IS ([RFC1195]), has used quantitative static link metrics. Other mechanisms, such as Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) (see [RFC2702] and [RFC3209]), make use of other link attributes such as the available reserved bandwidth (dynamic) or link affinities (most of the time static) to compute constrained shortest paths for Traffic Engineering Label Switched Paths (TE LSPs).

This document specifies routing metrics and constraints to be used in path calculation by the Routing Protocol for Low-Power and Lossy Networks (RPL) specified in [RFC6550].

One of the prime objectives of this document is to define a flexible mechanism for the advertisement of routing metrics and constraints used by RPL. Some RPL implementations may elect to adopt an extremely simple approach based on the use of a single metric with no constraint, whereas other implementations may use a larger set of link and node routing metrics and constraints. This specification provides a high degree of flexibility and a set of routing metrics and constraints. New routing metrics and constraints could be defined in the future, as needed.

The metrics and constraints defined in this document are carried in objects that are OPTIONAL from the point of view of a RPL implementation. This means that implementations are free to include different subsets of the functions (metric, constraint) defined in this document. Specific sets of metrics/constraints and other optional RPL parameters for use in key environments will be specified as compliance profiles in applicability profile documents produced by the ROLL working group. Note that RPL can even make use of no metric, for example, using the Objective Function defined in [RFC6552].

RPL is a distance vector routing protocol variant that builds Directed Acyclic Graphs (DAGs) based on routing metrics and constraints. DAG formation rules are defined in [RFC6550]:

- o The Destination-Oriented Directed Acyclic Graph (DODAG) root, as defined in [RFC6550], may advertise a routing constraint used as a "filter" to prune links and nodes that do not satisfy specific properties. For example, it may be required for a path only to traverse nodes that are mains-powered or links that have at least a minimum reliability or a specific "color" reflecting a user-defined link characteristic (e.g., the link layer supports encryption).

- o A routing metric is a quantitative value that is used to evaluate the path cost. Link and node metrics are usually (but not always) additive.

The best path is the path that satisfies all supplied constraints (if any) and that has the lowest cost with respect to some specified metrics. It is also called the shortest constrained path (in the presence of constraints).

Routing metrics may be categorized according to the following characteristics:

- o Link versus node metrics

- o Qualitative versus quantitative

- o Dynamic versus static

Routing requirements documents (see [RFC5673], [RFC5826], [RFC5548], and [RFC5867]) observe that it must be possible to take into account a variety of node constraints/metrics during path computation.

Some link or node characteristics (e.g., link reliability, remaining energy on the node) may be used by RPL either as routing constraints or as metrics (or sometimes both). For example, the path may be computed to avoid links that do not provide a sufficient level of reliability (use as a constraint) or as the path offering most links with a specified reliability level (use as a metric). This document provides the flexibility to use link and node characteristics as constraints and/or metrics.

The use of link and node routing metrics and constraints is not exclusive (e.g., it is possible to advertise a "hop count" both as a metric to optimize the computed path and as a constraint (e.g., "Path should not exceed n hops")).

Links in LLN commonly have rapidly changing node and link characteristics; thus, routing metrics must be dynamic and techniques must be used to smooth out the dynamicity of these metrics so as to avoid routing oscillations. For instance, in addition to the dynamic nature of some links (e.g., wireless but also Power Line Communication (PLC) links), nodes' resources, such as residual energy, are changing continuously and may have to be taken into account during the path computation.

It must be noted that the use of dynamic metrics is not new and has been experimented in ARPANET 2 (see [Zinky1989]). The use of dynamic metrics is not trivial and great care must be given to the use of dynamic metrics since it may lead to potential routing instabilities. That being said, a lot of experience has been gained over the years on the use of dynamic routing metrics, which have been deployed in a number of (non-IP) networks.

Very careful attention must be given to the pace at which routing metrics and attributes values change in order to preserve routing stability. When using a dynamic routing metric, a RPL implementation should make use of a multi-threshold scheme rather than fine granular metric updates reflecting each individual change to avoid spurious and unnecessary routing changes.

The requirements on reporting frequency may differ among metrics; thus, different reporting rates may be used for each metric.

The set of routing metrics and constraints used by a RPL deployment is signaled along the DAG that is built according to the Objective Function (rules governing how to build a DAG) and the routing metrics and constraints are advertised in the DODAG Information Object (DIO) message specified in [RFC6550]. RPL may be used to build DAGs with different characteristics. For example, it may be desirable to build a DAG with the goal to maximize reliability by using the link reliability metric to compute the "best" path. Another example might be to use the energy node characteristic (e.g., mains-powered versus battery-operated) as a node constraint when building the DAG so as to avoid battery-powered nodes in the DAG while optimizing the link throughput.

The specification of Objective Functions used to compute the DAG built by RPL is out of the scope of this document. This document defines routing metrics and constraints that are decoupled from the Objective Function. So a generic Objective Function could, for example, specify the rules to select the best parents in the DAG, the number of backup parents, etc., and it could be used with any routing metrics and/or constraints such as the ones specified in this document.

Some metrics are either aggregated or recorded. An aggregated metric is adjusted as the DIO message travels along the DAG. For example, if the metric is the number of hops, each node updates the path cost that reflects the number of traversed hops along the DAG. By contrast, for a recorded metric, each node adds a sub-object reflecting the local valuation of the metric. For example, it might be desirable to record the link quality level along a path. In this case, each visited node adds a sub-object recording the local link quality level. In order to limit the number of sub-objects, the use of a counter may be desirable (e.g., record the number of links with a certain link quality level), thus, compressing the information to reduce the message length. Upon receiving the DIO message from a set of parents, a node might decide, according to the OF and local policy, which node to choose as a parent based on the maximum number of links with a specific link reliability level, for example.

Note that the routing metrics and constraints specified in this document are not specific to any particular link layer. An internal API between the Medium Access Control (MAC) layer and RPL may be used to accurately reflect the metrics values of the link (wireless, wired, PLC).

Since a set of metrics and constraints will be used for links and nodes in a LLN, it is critical to ensure the use of consistent metric calculation mechanisms for all links and nodes in the network, similar to the case of inter-domain IP routing.

There are many different permutations of options that may be appropriate in different deployments. Implementations must clearly state which options they include, and they must state which are default and which are configurable as options within the implementation. Applicability statements will be developed within the ROLL working group to clarify which options are applicable to the specific deployment scenarios indicated by [RFC5673], [RFC5826], [RFC5548], and [RFC5867].

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Object Formats

2.1. DAG Metric Container Format

Routing metrics and constraints are carried within the DAG Metric Container object defined in [RFC6550]. Should multiple metrics and/or constraints be present in the DAG Metric Container, their use to determine the "best" path can be defined by an Objective Function.

The Routing Metric/Constraint objects represent a metric or a constraint of a particular type. They may appear in any order in the DAG Metric Container (specified in [RFC6550]). They have a common format consisting of one or more bytes with a common header.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Routing-MC-Type|Res Flags|P|C|O|R| A   |  Prec | Length (bytes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (object body)                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Routing Metric/Constraint Object Generic Format

The object body carries one or more sub-objects defined later in this document. Note that an object may carry a TLV, which may itself comprise other TLVs. A TLV carried within a TLV is called a TLV in this specification.

Routing-MC-Type (Routing Metric/Constraint Type - 8 bits): the Routing Metric/Constraint Type field uniquely identifies each Routing Metric/Constraint object and is managed by IANA.

Length (8 bits): this field defines the length of the object body, expressed in bytes. It ranges from 0 to 255.

Res Flags field (16 bits). The Flag field of the Routing Metric/ Constraint object is managed by IANA. Unassigned bits are considered as reserved. They MUST be set to zero on transmission and MUST be ignored on receipt.

The following bits of the Routing Metric/Constraint Flag field object are currently defined:

- 'P' flag: the P field is only used for recorded metrics. When cleared, all nodes along the path successfully recorded the corresponding metric. When set, this indicates that one or several nodes along the path could not record the metric of interest (either because of lack of knowledge or because this was prevented by policy).

- 'C' flag. When set, this indicates that the Routing Metric/Constraint object refers to a routing constraint. When cleared, the routing object refers to a routing metric.

- 'O' flag: The 'O' flag is used exclusively for routing constraints ('C' flag is set). When set, this indicates that the constraint specified in the body of the object is optional. When cleared, the constraint is mandatory. If the 'C' flag is zero, the 'O' flag MUST be set to zero on transmission and ignored on reception.

- 'R' flag: The 'R' flag is only relevant for a routing metric (C=0) and MUST be cleared for C=1. When set, this indicates that the routing metric is recorded along the path. Conversely, when cleared, the routing metric is aggregated.

A Field (3 bits): The A field is only relevant for metrics and is used to indicate whether the aggregated routing metric is additive, is multiplicative, reports a maximum, or reports a minimum.

- A=0: The routing metric is additive

- A=1: The routing metric reports a maximum

- A=2: The routing metric reports a minimum

- A=3: The routing metric is multiplicative

The A field has no meaning when the 'C' flag is set (i.e., when the Routing Metric/Constraint object refers to a routing constraint) and is only valid when the 'R' bit is cleared. Otherwise, the A field MUST be set to 0 and MUST be ignored on receipt.

Prec field (4 bits): The Prec field indicates the precedence of this Routing Metric/Constraint object relative to other objects in the container. This is useful when a DAG Metric Container contains several Routing Metric objects. Its value ranges from 0 to 15. The value 0 means the highest precedence.

Example 1: A DAG formed by RPL where all nodes must be mains-powered and the best path is the one with lower aggregated expected transmission count (ETX). In this case, the DAG Metric Container carries two Routing Metric/Constraint objects: one is an ETX metric object with header (C=0, O=0, A=00, R=0) and the second one is a Node Energy constraint object with header (C=1, O=0, A=00, R=0). Note that a RPL Instance may use the metric object to report a maximum (A=1) or a minimum (A=2). If, for example, the best path is characterized by the path avoiding low quality links, then the path metric reports a maximum (A=1) (the higher the ETX, the lower the link quality): when the DIO message reporting the link quality metric (ETX) is processed by a node, each node selecting the advertising node as a parent updates the value carried in the metric object by replacing it with its local link ETX value if and only if the latter is higher. As far as the constraint is concerned, the object body will carry a Node Energy constraint object defined in Section 3.1 indicating that nodes must be mains-powered: if the constraint signaled in the DIO message is not satisfied, the advertising node is just not selected as a parent by the node that processes the DIO message.

Example 2: A DAG formed by RPL where the link metric is the link quality level (defined in Section 4) and link quality levels must be recorded along the path. In this case, the DAG Metric Container carries a Routing Metric/Constraint object: link quality level metric (C=0, O=0, A=00, R=1) containing multiple sub-objects.

A Routing Metric/Constraint object may also include one or more additional type-length-value (TLV) encoded data sets. Each Routing Metric/Constraint TLV has the same structure:

Type: 1 byte
Length: 1 byte
Value: variable

A Routing Metric/Constraint TLV is comprised of 1 byte for the type, 1 byte specifying the TLV length, and a value field. The TLV length field defines the length of the value field in bytes (from 0 to 255).

Unrecognized TLVs MUST be silently ignored while still being propagated in DIOs generated by the receiving node.

IANA manages the codepoints for all TLVs carried in routing constraint/metric objects.

IANA management of the Routing Metric/Constraint objects identifier codespace is described in Section 6.

2.2. Use of Multiple DAG Metric Containers

Since the length of RPL options is encoded using 1 octet, they cannot exceed 255 bytes, which also applies to the DAG Metric Container. In the vast majority of cases, the advertised routing metrics and constraints will not require that much space. However, there might be circumstances where larger space is required, should, for example, a set of routing metrics be recorded along a long path. In this case, in order to avoid overflow, as specified in [RFC6550], routing metrics will be carried using multiple DAG Metric Container objects.

In the rest of this document, this use of multiple DAG Metric Container objects will be considered as if they were actually just one long DAG Metric Container object.

2.3. Metric Usage

When the DAG Metric Container contains a single aggregated metric (scalar value), the order relation to select the best path is implicitly derived from the metric type. For example, lower is better for Hop Count, Link Latency, and ETX. Conversely, for Node Energy or Throughput, higher is better.

An example of using such a single aggregated metric is optimizing routing for node energy. The Node Energy metric (E_E field) defined in Section 3.2 is aggregated along paths with an explicit min function (A field), and the best path is selected through an implied Max function because the metric is Energy.

When the DAG Metric Container contains several aggregated metrics, they are to be used as tiebreakers according to their precedence defined by their Prec field values.

An example of such use of multiple aggregated metrics is the following: Hop Count as the primary criterion, Link Quality Level (LQL) as the secondary criterion, and Node Energy as the ultimate tiebreaker. In such a case, the Hop Count, LQL, and Node Energy metric objects' Prec fields should bear strictly increasing values such as 0, 1, and 2, respectively.

If several aggregated metrics happen to bear the same Prec value, the behavior is implementation dependent.

3. Node Metric/Constraint Objects

Sections 3 and 4 specify several link and node metric/constraint objects. In some cases, it is stated that there must not be more than one object of a specific type. In that case, if a RPL implementation receives more than one object of that type, the second object MUST silently be ignored.

In the presence of a constraint, a node MUST include a metric of the same type. That metric is used to check whether or not the constraint is met. In all cases, a node MUST not change the content of the constraint.

3.1. Node State and Attribute Object

The Node State and Attribute (NSA) object is used to provide information on node characteristics.

The NSA object MAY be present in the DAG Metric Container. There MUST NOT be more than one NSA object as a constraint per DAG Metric Container, and there MUST NOT be more than one NSA object as a metric per DAG Metric Container.

The NSA object may also contain a set of TLVs used to convey various node characteristics. No TLV is currently defined.

The NSA Routing Metric/Constraint Type has been assigned value 1 by IANA.

The format of the NSA object body is as follows:

```
 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 4
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
|     Res       |  Flags    |A|O|  Optional TLVs
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

Figure 2: NSA Object Body Format

Res flags (8 bits): Reserved field. This field MUST be set to zero on transmission and MUST be ignored on receipt.

Flags field (8 bits). The following two bits of the NSA object are currently defined:

o 'A' flag: data Aggregation Attribute. Data aggregation is listed as a requirement in Section 6.2 of [RFC5548]. Some applications may make use of the aggregation node attribute in their routing decision so as to minimize the amount of traffic on the network, thus, potentially increasing its lifetime in battery operated environments. Applications where highly directional data flow is expected on a regular basis may take advantage of data aggregation supported routing. When set, this indicates that the node can act as a traffic aggregator. Further documents MAY define optional TLVs to describe the node traffic aggregator functionality.

- 'O' flag: node workload may be hard to determine and express in some scalar form. However, node workload could be a useful metric to consider during path calculation, in particular when queuing delays must be minimized for highly sensitive traffic considering Medium Access Control (MAC) layer delay. Node workload MAY be set upon CPU overload, lack of memory, or any other node related conditions. Using a simple 1-bit flag to characterize the node workload provides a sufficient level of granularity, similar to the "overload" bit used in routing protocols such as IS-IS. Algorithms used to set the overload bit and to compute paths to potentially avoid nodes with their overload bit set are outside the scope of this document, but it is RECOMMENDED to avoid frequent changes of this bit to avoid routing oscillations. When set, this indicates that the node is overloaded and may not be able to process traffic.

The unspecified flag fields MUST be set to zero on transmission and MUST be ignored on receipt.

The Flags field of the NSA Routing Metric/Constraint object is managed by IANA. Unassigned bits are considered as reserved.

3.2. Node Energy Object

It may sometimes be desirable to avoid selecting a node with low residual energy as a router; thus, the support for constraint-based routing is needed. In such cases, the routing protocol engine may compute a longer path (constraint based) for some traffic in order to increase the network life duration.

Power and energy are clearly critical resources in most LLNs. As yet, there is no simple abstraction that adequately covers the broad range of power sources and energy storage devices used in existing LLN nodes. These include mains-powered, primary batteries, energy scavengers, and a variety of secondary storage mechanisms. Scavengers may provide a reliable low level of power, such as might be available from a 4-20 mA loop; a reliable but periodic stream of power, such as provided by a well-positioned solar cell; or unpredictable power, such as might be provided by a vibrational energy scavenger on an intermittently powered pump. Routes that are viable when the sun is shining may disappear at night. A pump turning on may connect two previously disconnected sections of a network.

Storage systems, such as rechargeable batteries, often suffer substantial degradation if regularly used to full discharge, leading to different residual energy numbers for regular versus emergency operation. A route for emergency traffic may have a different optimum than one for regular reporting.

Batteries used in LLNs often degrade substantially if their average current consumption exceeds a small fraction of the peak current that they can deliver. It is not uncommon for self-supporting nodes to have a combination of primary storage, energy scavenging, and secondary storage, leading to three different values for acceptable average current depending on the time frame being considered, e.g., milliseconds, seconds, and hours/years.

Raw power and energy values are meaningless without knowledge of the energy cost of sending and receiving packets, and lifetime estimates have no value without some higher-level constraint on the lifetime required of a device. In some cases, the path that exhausts the battery of a node on the bed table in a month may be preferable to a route that reduces the lifetime of a node in the wall to a decade.

Given the complexity of trying to address such a broad collection of constraints, this document defines two levels of fidelity in the solution.

The simplest solution relies on a 2-bit field encoding three types of power sources: "powered", "battery", and "scavenger". This simple approach may be sufficient for many applications.

The mid-complexity solution is a single parameter that can be used to encode the energetic happiness of both battery-powered and scavenging nodes. For scavenging nodes, the 8-bit quantity is the power provided by the scavenger divided by the power consumed by the application, E_E=P_in/P_out, in units of percent. Nodes that are scavenging more power than they are consuming will register above 100. A good time period for averaging power in this calculation may be related to the discharge time of the energy storage device on the node, but specifying this is out of the scope of this document. For battery-powered devices, E_E is the current expected lifetime divided by the desired minimum lifetime, in units of percent. The estimation of remaining battery energy and actual power consumption can be difficult, and the specifics of this calculation are out of scope of this document, but two examples are presented. If the node can measure its average power consumption, then E_E can be calculated as the ratio of desired max power (initial energy E_0 divided by desired lifetime T) to actual power, E_E=P_max/P_now. Alternatively, if the energy in the battery E_bat can be estimated, and the total elapsed lifetime, t, is available, then E_E can be calculated as the total stored energy remaining versus the target energy remaining: E_E= E_bat / [E_0 (T-t)/T].

An example of an optimized route is max(min(E_E)) for all battery-operated nodes along the route, subject to the constraint that E_E>=100 for all scavengers along the route.

Note that the estimated percentage of remaining energy indicated in the E_E field may not be useful in the presence of nodes powered by battery or energy scavengers when the amount of energy accumulated by the device significantly differ. Indeed, X% of remaining energy on a node that can store a large amount of energy cannot be easily compared to the same percentage of remaining energy on a node powered by a tiny source of energy. That being said, in networks where nodes have similar energy storage, such a percentage of remaining energy is useful.

The Node Energy (NE) object is used to provide information related to node energy and may be used as a metric or as constraint.

The NE object MAY be present in the DAG Metric Container. There MUST NOT be more than one NE object as a constraint per DAG Metric Container, and there MUST NOT be more than one NE object as a metric per DAG Metric Container.

The NE object Type has been assigned value 2 by IANA.

The format of the NE object body is as follows:

```
 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 4
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
|     NE Sub-objects
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

Figure 3: NE Sub-Object Format

The format of the NE sub-object body is as follows:

```
  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 4
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
 | Flags |I| T |E|      E_E      |   Optional TLVs
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

Figure 4: NE Sub-Object Format

The NE sub-object may also contain a set of TLVs used to convey various nodes' characteristics.

Flags field (8 bits). The following flags are currently defined:

- I (Included): the 'I' bit is only relevant when the node type is used as a constraint. For example, the path must only traverse mains-powered nodes. Conversely, battery-operated nodes must be excluded. The 'I' bit is used to stipulate inclusion versus exclusion. When set, this indicates that nodes of the type specified in the node type field MUST be included. Conversely, when cleared, this indicates that nodes of type specified in the node type field MUST be excluded.

- T (node Type): 2-bit field indicating the node type. T=0 designates a mains-powered node, T=1 a battery-powered node, and T=2 a node powered by an energy scavenger.

- E (Estimation): when the 'E' bit is set for a metric, the estimated percentage of remaining energy on the node is indicated in the E_E 8-bit field. When cleared, the estimated percentage of remaining energy is not provided. When the 'E' bit is set for a constraint, the E_E field defines a threshold for the inclusion/exclusion: if an inclusion, nodes with values higher than the threshold are to be included; if an exclusion, nodes with values lower than the threshold are to be excluded.

E_E (Estimated-Energy): 8-bit unsigned integer field indicating an estimated percentage of remaining energy. The E_E field is only relevant when the 'E' flag is set, and it MUST be set to 0 when the 'E' flag is cleared.

If the NE object comprises several sub-objects when used as a constraint, each sub-object adds or subtracts node subsets as the sub-objects are parsed in order. The initial set (full or empty) is defined by the 'I' bit of the first sub-object: full if that 'I' bit is an exclusion, empty if that 'I' bit is an inclusion.

No TLV is currently defined.

Future documents may define more complex solutions involving TLV parameters representing energy storage, consumption, and generation capabilities of the node, as well as desired lifetime.

3.3. Hop Count Object

The Hop Count (HP) object is used to report the number of traversed nodes along the path.

The HP object MAY be present in the DAG Metric Container. There MUST NOT be more than one HP object as a constraint per DAG Metric Container, and there MUST NOT be more than one HP object as a metric per DAG Metric Container.

The HP object may also contain a set of TLVs used to convey various node characteristics. No TLV is currently defined.

The HP routing metric object Type has been assigned value 3 by IANA.

The format of the Hop Count object body is as follows:

```
 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 4
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
|  Res  | Flags |   Hop Count   |  Optional TLVs
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

Figure 5: Hop Count Object Body Format

Res flags (4 bits): Reserved field. This field MUST be set to zero on transmission and MUST be ignored on receipt.

No Flag is currently defined. Unassigned bits are considered reserved. They MUST be set to zero on transmission and MUST be ignored on receipt.

The HP object may be used as a constraint or a metric. When used as a constraint, the DAG root indicates the maximum number of hops that a path may traverse. When that number is reached, no other node can join that path. When used as a metric, each visited node simply increments the Hop Count field.

Note that the first node along a path inserting a Hop Count metric object MUST set the Hop Count field value to 1.

4. Link Metric/Constraint Objects

4.1. Throughput

Many LLNs support a wide range of throughputs. For some links, this may be due to variable coding. For the deeply duty-cycled links found in many LLNs, the variability comes as a result of trading power consumption for bit rate. There are several MAC layer protocols that allow for the effective bit rate of a link to vary over more than three orders of magnitude with a corresponding change in power consumption. For efficient operation, it may be desirable for nodes to report the range of throughput that their links can handle in addition to the currently available throughput.

The Throughput object MAY be present in the DAG Metric Container. There MUST NOT be more than one Throughput object as a constraint per DAG Metric Container, and there MUST NOT be more than one Throughput object as a metric per DAG Metric Container.

The Throughput object is made of throughput sub-objects and MUST at least comprise one Throughput sub-object. The first Throughput sub-object MUST be the most recently estimated actual throughput. The actual estimation of the throughput is outside the scope of this document.

Each Throughput sub-object has a fixed length of 4 bytes.

The Throughput object does not contain any additional TLVs.

The Throughput object Type has been assigned value 4 by IANA.

The format of the Throughput object body is as follows:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  (sub-object) .....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Throughput Object Body Format

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Throughput                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Throughput Sub-Object Format

Throughput: 32 bits. The Throughput is encoded in 32 bits in unsigned integer format, expressed in bytes per second.

4.2. Latency

Similar to throughput, the latency of many LLN MAC sub-layers can vary over many orders of magnitude, again with a corresponding change in power consumption. Some LLN MAC link layers will allow the latency to be adjusted globally on the subnet, on a link-by-link basis, or not at all. Some will insist that it be fixed for a given link, but allow it to be variable from link to link.

The Latency object MAY be present in the DAG Metric Container. There MUST NOT be more than one Latency object as a constraint per DAG Metric Container, and there MUST NOT be more than one Latency object as a metric per DAG Metric Container.

The Latency object is made of Latency sub-objects and MUST at least comprise one Latency sub-object. Each Latency sub-object has a fixed length of 4 bytes.

The Latency object does not contain any additional TLVs.

The Latency object Type has been assigned value 5 by IANA.

The Latency object is a metric or constraint.

The format of the Latency object body is as follows:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  (sub-object) .....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Latency Object Body Format

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Latency                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Latency Sub-Object Format

Latency: 32 bits. The Latency is encoded in 32 bits in unsigned integer format, expressed in microseconds.

The Latency object may be used as a constraint or a path metric. For example, one may want the latency not to exceed some value. In this case, the Latency object common header indicates that the provided value relates to a constraint. In another example, the Latency object may be used as an aggregated additive metric where the value is updated along the path to reflect the path latency.

4.3. Link Reliability

In LLNs, link reliability could be degraded for a number of reasons: signal attenuation, interferences of various forms, etc. Time scales vary from milliseconds to days, and are often periodic and linked to human activity. Packet error rates can generally be measured directly, and other metrics (e.g., bit error rate, mean time between failures) are typically derived from that. Note that such variability is not specific to wireless link but also applies to PLC links.

A change in link quality can affect network connectivity; thus, link quality may be taken into account as a critical routing metric.

A number of link reliability metrics could be defined reflecting several reliability aspects. Two link reliability metrics are defined in this document: the Link Quality Level (LQL) and the ETX Metric.

Note that a RPL deployment MAY use the LQL, the ETX, or both.

4.3.1. The Link Quality Level Reliability Metric

The Link Quality Level (LQL) object is used to quantify the link reliability using a discrete value, from 0 to 7, where 0 indicates that the link quality level is unknown and 1 reports the highest link quality level. The mechanisms and algorithms used to compute the LQL are implementation specific and outside of the scope of this document.

The LQL can be used either as a metric or a constraint. When used as a metric, the LQL metric can only be recorded. For example, the DAG Metric object may request all traversed nodes to record the LQL of their incoming link into the LQL object. Each node can then use the LQL record to select its parent based on some user defined rules (e.g., something like "select the path with most links reporting a LQL value of 3 or less").

Counters are used to compress the information: for each encountered LQL value, only the number of matching links is reported.

The LQL object MAY be present in the DAG Metric Container. There MUST NOT be more than one LQL object as a constraint per DAG Metric Container, and there MUST NOT be more than one LQL object as a metric per DAG Metric Container.

The LQL object MUST contain one or more sub-object used to report the number of links along with their LQL.

The LQL object Type has been assigned value 6 by IANA.

The format of the LQL object body is as follows:

```
 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 4
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
|       Res     | LQL sub-object
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

Figure 10: LQL Object Body Format

Res flags (8 bits): Reserved field. This field MUST be set to zero on transmission and MUST be ignored on receipt.

When the LQL metric is recorded, the LQL object body comprises one or more LQL Type 1 sub-object.

The format of the LQL Type 1 sub-object is as follows

```
 0
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
| Val | Counter |
+-+-+-+-+-+-+-+-+
```

Figure 11: LQL Type 1 Sub-Object Format

Val: LQL value from 0 to 7 where 0 means undetermined and 1 indicates the highest link quality.

Counter: number of links with that value.

4.3.2. The ETX Reliability Object

The ETX metric is the number of transmissions a node expects to make to a destination in order to successfully deliver a packet. In contrast with the LQL routing metric, the ETX provides a discrete value (which may not be an integer) computed according to a specific formula: for example, an implementation may use the following formula: ETX= 1 / (Df * Dr) where Df is the measured probability that a packet is received by the neighbor and Dr is the measured probability that the acknowledgment packet is successfully received. This document does not mandate the use of a specific formula to compute the ETX value.

The ETX object MAY be present in the DAG Metric Container. There MUST NOT be more than one ETX object as a constraint per DAG Metric Container, and there MUST NOT be more than one ETX object as a metric per DAG Metric Container.

The ETX object is made of ETX sub-objects and MUST at least comprise one ETX sub-object. Each ETX sub-object has a fixed length of 16 bits.

The ETX object does not contain any additional TLVs.

The ETX object Type has been assigned value 7 by IANA.

The format of the ETX object body is as follows:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  (sub-object) .....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: ETX Object Body Format

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ETX              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: ETX Sub-Object Format

ETX: 16 bits. The ETX * 128 is encoded using 16 bits in unsigned integer format, rounded off to the nearest whole number. For example, if ETX = 3.569, the object value will be 457. If ETX > 511.9921875, the object value will be the maximum, which is 65535.

The ETX object may be used as a constraint or a path metric. For example, it may be required that the ETX must not exceed some specified value. In this case, the ETX object common header indicates that the value relates to a constraint. In another example, the ETX object may be used as an aggregated additive metric where the value is updated along the path to reflect the path quality: when a node receives the aggregated additive ETX value of the path (cumulative path ETX calculated as the sum of the link ETX of all of the traversed links from the advertising node to the DAG root), if it selects that node as its preferred parent, the node updates the path ETX by adding the ETX of the local link between itself and the preferred parent to the received path cost (path ETX) before potentially advertising itself the new path ETX.

4.4. Link Color Object

4.4.1. Link Color Object Description

The Link Color (LC) object is an administrative 10-bit link constraint (which may be either static or dynamically adjusted) used to avoid or attract specific links for specific traffic types.

The LC object can be used either as a metric or as a constraint. When used as a metric, the LC metric can only be recorded. For example, the DAG may require recording the link colors for all traversed links. A color is defined as a specific set of bit values: in other words, that 10-bit field is a flag field, and not a scalar value. Each node can then use the LC to select the parent based on user defined rules (e.g., "select the path with the maximum number of links having their first bit set 1 (e.g., encrypted links)"). The LC object may also be used as a constraint.

When used as a recorded metric, a counter is used to compress the information where the number of links for each Link Color is reported.

The Link Color (LC) object MAY be present in the DAG Metric Container. There MUST NOT be more than one LC object as a constraint per DAG Metric Container, and there MUST NOT be more than one LC object as a metric per DAG Metric Container.

There MUST be a at least one LC sub-object per LC object.

The LC object does not contain any additional TLVs.

The LC object Type has been assigned value 8 by IANA.

The format of the LC object body is as follows:

```
  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 4
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
 |      Res      | LC sub-objects
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

Figure 14: LC Object Format

Res flags (8 bits): Reserved field. This field MUST be set to zero on transmission and MUST be ignored on receipt.

When the LC object is used as a recorded metric, the LC object body comprises one or more LC Type 1 sub-objects.

The format of the LC Type 1 sub-object body is as follows:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Link Color     |  Counter  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: LC Type 1 Sub-Object Format

When the LC object is used as a constraint, the LC object body comprises one or more LC Type 2 sub-objects.

The format of the LC Type 2 sub-object body is as follows:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Link Color    |Reserved |I|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: LC Type 2 Sub-Object Format

Reserved (5 bits): Reserved field. This field MUST be set to zero on transmission and MUST be ignored on receipt.

'I' Bit: The 'I' bit is only relevant when the Link Color is used as a constraint. When set, this indicates that links with the specified color must be included. When cleared, this indicates that links with the specified color must be excluded.

It is left to the implementer to define the meaning of each bit of the 10-bit Link Color Flag field.

4.4.2. Mode of Operation

The link color may be used as a constraint or a metric.

- o When used as constraint, the LC object may be inserted in the DAG Metric Container to indicate that links with a specific color should be included or excluded from the computed path.

- o When used as recorded metric, each node along the path may insert an LC object in the DAG Metric Container to report the color of the local link. If there is already an LC object reporting a similar color, the node MUST NOT add another identical LC sub-object and MUST increment the counter field.

5. Computation of Dynamic Metrics and Attributes

As already pointed out, dynamically calculated metrics are of the utmost importance in many circumstances in LLNs. This is mainly because a variety of metrics change on a frequent basis, thus, implying the need to adapt the routing decisions. That being said, care must be given to the pace at which changes are reported in the network. The attributes will change according to their own time scales. RPL controls the reporting rate.

To minimize metric updates, multi-threshold algorithms MAY be used to determine when updates should be sent. When practical, low-pass filtering and/or hysteresis should be used to avoid rapid fluctuations of these values. Finally, although the specification of path computation algorithms using dynamic metrics is out of the scope of this document, it is RECOMMENDED to carefully design the route optimization algorithm to avoid too frequent computation of new routes upon metric values changes.

Controlled adaptation of the routing metrics and rate at which paths are computed are critical to avoid undesirable routing instabilities resulting in increased latencies and packet loss because of temporary micro-loops. Furthermore, excessive route changes will adversely impact the traffic and power consumption in the network, thus, potentially impacting its scalability.

6. IANA Considerations

IANA has established a new top-level registry, called "RPL Routing Metric/Constraint", to contain all Routing Metric/Constraint objects codepoints and sub-registries.

The allocation policy for each new registry is by IETF review: new values are assigned through the IETF review process (see [RFC5226]). Specifically, new assignments are made via RFCs approved by the IESG. Typically, the IESG will seek input on prospective assignments from appropriate persons (e.g., a relevant working group if one exists).

New bit numbers may be allocated only by an IETF Review action. Each bit should be tracked with the following qualities:

- o Bit number
- o Capability Description
- o Defining RFC

6.1. Routing Metric/Constraint Type

IANA has created a sub-registry, called "Routing Metric/Constraint Type", for Routing Metric/Constraint object types, which range from 0 to 255. Value 0 is unassigned.

Value	Meaning	Reference
1	Node State and Attribute	This document
2	Node Energy	This document
3	Hop Count	This document
4	Link Throughput	This document
5	Link Latency	This document
6	Link Quality Level	This document
7	Link ETX	This document
8	Link Color	This document

6.2. Routing Metric/Constraint TLVs

IANA has created a sub-registry, called "Routing Metric/Constraint TLVs", used for all TLVs carried within Routing Metric/Constraint objects. The Type field is an 8-bit field whose value is comprised between 0 and 255. Value 0 is unassigned. The Length field is an 8-bit field whose value ranges from 0 to 255. The Value field has value ranges depending on the Type; therefore, they are not defined here, since no Type is registered at this time.

6.3. Routing Metric/Constraint Common Header Flag Field

IANA has created a sub-registry, called "Routing Metric/Constraint Common Header Flag field", to manage the 9-bit Flag field of the Routing Metric/Constraint common header.

Several bits are defined for the Routing Metric/Constraint common header Flag field in this document. The following values have been assigned:

Codespace of the Flag field (Routing Metric/Constraint common header)

Bit	Description	Reference
8	Recorded/Aggregated	This document
7	Optional Constraint	This document
6	Constraint/Metric	This document
5	P (Partial)	This document

Bits 0-4 are currently reserved.

6.4. Routing Metric/Constraint Common Header A Field

IANA has created a sub-registry, called "Routing Metric/Constraint Common Header A field", to manage the codespace of the A field of the Routing Metric/Constraint common header.

The A field is 3 bits in length, and it has values ranging from 0 to 7.

Codespace of the A field (Routing Metric/Constraint common header)

Value	Meaning	Reference
0	Routing metric is additive	This document
1	Routing metric reports a maximum	This document
2	Routing metric reports a minimum	This document
3	Routing metric is multiplicative	This document

6.5. NSA Object Flags Field

IANA has created a sub-registry, called "NSA Object Flag field", to manage the codespace of the 8-bit Flag field of the NSA object.

Several bits are defined for the NSA Object Flag field in this document. The following values have been assigned:

Codespace of the Flag field (NSA object)

Bit	Description	Reference
6	Aggregator	This document
7	Overloaded	This document

Bits 0-5 are reserved.

6.6. Hop-Count Object Flags Field

IANA has created a sub-registry, called "Hop-Count Object Flag field", to manage the codespace of the 4-bit Flag field of the Hop Count object.

No Flag is currently defined.

6.7. Node Type Field

IANA has created a sub-registry, called "Node Type Field", to manage the codespace of the field of the Routing Metric/Constraint common header.

The T field is 2 bits in length, and it has values ranging from 0 to 3.

Codespace of the T field (Routing Metric/Constraint common header)

Value	Description	Reference
0	a mains-powered node	This document
1	a battery-powered node	This document
2	a node powered by an energy scavenger	This document

7. Security Considerations

Routing metrics should be handled in a secure and trustful manner. For instance, RPL should not allow a malicious node to falsely advertise that it has good metrics for routing so as to be selected as preferred next-hop router for other nodes' traffic and intercept packets. Another attack may consist of making intermittent attacks on a link in an attempt to constantly modify the link quality and consequently the associated routing metric, thus, leading to potential fluctuation in the DODAG. Thus, it is RECOMMENDED for a RPL implementation to put in place mechanisms so as to stop advertising routing metrics for highly unstable links that may be subject to attacks.

Some routing metrics may also be used to identify some areas of weaknesses in the network (a highly unreliable link, a node running low in terms of energy, etc.). Such information may be used by a potential attacker. Thus, it is RECOMMENDED to carefully consider which metrics should be used by RPL and the level of visibility that they provide about the network state or to use appropriate the security measures as specified in [RFC6550] to protect that information.

Since the routing metrics/constraints are carried within RPL message, the security routing mechanisms defined in [RFC6550] apply here.

8. Acknowledgements

The authors would like to acknowledge the contributions of Young Jae Kim, Hakjin Chong, David Meyer, Mischa Dohler, Anders Brandt, Philip Levis, Pascal Thubert, Richard Kelsey, Jonathan Hui, Alexandru Petrescu, Richard Kelsey, Mathilde Durvy, Phoebus Chen, Tim Winter, Yoav Ben-Yehezkel, Matteo Paris, Omprakash Gnawali, Mads Westergreen, Mukul Goyal, Joseph Saloway, David Culler, and Jari Arkko for their review and valuable comments. Special thanks to Adrian Farrel for his thorough review.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

9.2. Informative References

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.

[RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.

[RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, April 2010.

[RFC5867] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.

[RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.

[ROLL-TERMS] Vasseur, JP., "Terminology in Low power And Lossy Networks", Work in Progress, September 2011.

[Zinky1989] Zinky, J., Vichniac, G., and A. Khanna, "Performance of the Revised Routing Metric for ARPANET and MILNET", Military Communications Conference, MILCOM '89, March 1989.

Authors' Addresses

JP. Vasseur (editor)
Cisco Systems
11, Rue Camille Desmoulins
Issy Les Moulineaux 92782
France

EMail: jpv@cisco.com

Mijeom Kim (editor)
Corporate Technology Group, KT
17 Woomyeon-dong, Seocho-gu
Seoul 137-792
Korea

EMail: mjkim@kt.com

Kris Pister
Dust Networks
30695 Huntwood Ave.
Hayward, CA 95544
USA

EMail: kpister@dustnetworks.com

Nicolas Dejean
Elster SAS
Espace Concorde, 120 impasse JB Say
Perols 34470
France

EMail: nicolas.dejean@coronis.com

Dominique Barthel
France Telecom Orange
28 chemin du Vieux Chene, BP 98
Meylan 38243
France

EMail: dominique.barthel@orange.com