Networking Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 10, 2011

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

July 09, 2010

Routing Metrics used for Path Calculation in Low Power and Lossy Networks
draft-ietf-roll-routing-metrics-08

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.

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].

Copyright Notice

Table of Contents

1. Introduction

This document makes use of the terminology defined in [I-D.ietf-roll-terminology].

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]) have 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 (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 [I-D.ietf-roll-rpl].

One of the prime objectives of this document is to propose 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.

RPL is a distance vector routing protocol that builds a Directed Acyclic Graph (DAG) based on routing metrics and constraints. DAG formation rules are defined in [I-D.ietf-roll-rpl]:

- o The DAG root 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 the path to only 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 nodes metrics are usually (but not always) additive.

The best path is the path with the lowest cost with respect to some

metrics that satisfies all constraints (if any) and is also called the shortest constrained path (in the presence of constraints).

Routing metrics can be classified according to the following set of characteristics:

o Link versus Node metrics

o Qualitative versus quantitative

o Dynamic versus static

It must be noted that the use of dynamic metrics is not new and has been experimented in ARPANET 2 [Khanna1989], with moderate success. 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.

As pointed out in various routing requirements documents (see [RFC5673], [RFC5826] [RFC5548] and [RFC5867]), it must be possible to take into account a variety of node constraints/metrics during path computation.

It is also worth mentioning that it is fairly common for links in LLNs to have fast changing node and link characteristics, which must be taken into account when specifying routing metrics. For instance, in addition to the dynamic nature of wireless connectivity, nodes' resources such as residual energy and other link's charatacteristics such as the throughput are changing continuously and may have to be taken into account during the path computation. Similarly, link attributes including throughput and reliability may drastically change over time due to multi-path interference.

Very careful attention must be given when using dynamic metrics and attributes that affect routing decisions in order to preserve routing stability. Routing metrics and constraints may either be static or dynamic. When dynamic, a RPL implementation SHOULD make use of a multi-threshold scheme rather than fine granular metric updates so as to avoid constant routing changes.

Furthermore, it is a time and energy consuming process to update dynamic metrics and recompute the routing tables on a frequent basis. Therefore, it may be desirable to use a set of discrete values to reduce computational overhead and bandwidth utilization. Of course, this comes with a cost, namely, reduced metric accuracy. In other cases, a set of flags may be defined to reflect a node state without having to define discrete values.

Some link or node characteristics (e.g. link reliability flag, energy remaining on the node) may either be used by RPL as routing constraints or metric. 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 the maximum number of links with a specified reliability level (use as a metric).

The set of routing metrics and constraints used by an RPL implementation is signalled along the Directed Acyclic Graph (DAG) that is built according to the Objective Function (rules governing how to build a DAG) and the routing metrics and constraints advertised in the Dag Information Option (DIO) message specified in [I-D.ietf-roll-rpl]. 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.

Links and nodes routing metrics and constraints are not exclusive.

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

The specification of objective functions used to compute the DAG built by RPL is out of the scope of this document and will be specified in other documents. Routing metrics and constraints 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. Such objective function can be used with any routing metrics and/or contraints such as the ones specified in this document.

Some metrics are either aggregated or recorded. In the former case, the metric is adjusted as the DIO message travels along the DAG. For example, if the metric is the link latency, each node updates the latency metric along the DAG. By contrast, metric may be recorded in which case each node adds a sub-object reflecting the local metric. For example, it might be desirable to record the link quality level along the path. In this case, each visited node adds a sub-object reporting 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). Upon receiving the DIO message from a set of parents, a node can decide which node to choose as a parent based on the maximum number of links with a specific link reliability level for example.

Notion of local versus global metric: some routing objects may have a local or a global significance. In the former case, a metric may be transmitted to a neighbor to charaterize a link or a node as opposed to a path. For example, a node may report information about its local energy without the need to propagate the energy level of all nodes along the path. In contrast, other metrics such as link latency metrics are additive and global in the sense that they characterize a path cost using the latency as a metric. In this particular example the path latency is an aggregated global and additive link metric.

2. Object formats

Routing metrics and constraints are carried within the DAG Metric Container object defined in [I-D.ietf-roll-rpl].

```
  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 2 3 4
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
 |     Type=2    |  Option Len   |  Routing Metric/Constraint objects
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

Figure 1: DAG Metric Container format

The Routing Metric/Contraints objects have a common format consisting of one or more 8-bit words 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|R|G| A |O|C|   Object Length (bytes)       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   //                        (object body)                        //
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Routing Metric/Constraint object generic format

The object body carries one or more sub-objects.

Note that the Routing Metric/Constraint objects defined in this document can appear in any order in the DAG Metric Container. However, for some of them, the order is significant (as described in Section 8 and Section 3.2, for example).

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

Res flags (2 bits). Reserved field. This field MUST be set to zero on transmission and MUST be ignored on receipt.

- o 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 O Flag: The O flag is used exclusively for routing constraints (C flag is set) and has no meaning for routing metrics. When set, this indicates that the constraint is optional. When cleared, the constraint is mandatory.

- o A Field: The A field is used to indicate whether a routing metric is additive, multiplicative, reports a maximum or a minimum.

  * A=0x00: The routing metric is additive

  * A=0x01: The routing metric reports a maximum

  * A=0x02: The routing metric reports a minimum

  * A=0x03: 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).

- o G Flag: When set, the Routing Metric/Constraint object is global. When cleared it is local (see details below).

- o R Flag: The R Flag is only relevant for global routing metric (C=0 and G=1) and MUST be cleared for all other values of C and G. When set, this indicates that the routing metric is recorded along the path. Conversely, when cleared the routing metric is aggregated.

Example 1: A DAG formed by RPL where all nodes must be mains powered and the link metric is the link quality characterized by the ETX. In this case the DAG Metric container carries two Routing Metric/ Constraint objects: the link metric is the link ETX (C=0, O=0, A=00, G=1, R=0) and the node constraint is power (C=1, O=0, A=00, G=0, R=0). Note that in this example, the link quality is a global additive aggregated link metric. Note that a RPL implementation may use the metric to report a maximum (A=0x01) or a minimum (A=0x02). If the best path is characterized by the path avoiding low quality links for example, then the path metric reports a maximum (A=0x02):

when the link quality metric (ETX) is processed each node updates it if the link quality (ETX) is higher than the current value reported by the link quality metric.

Example 2: A DAG formed by RPL where the link metric is the link quality level 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 (C=0, O=0, A=00, G=1, R=1) containing multiple sub-objects.

A Routing Metric/Constraint object may also include one or more 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 Length field defines the length of the value field in bytes.

Unrecognized TLVs MUST be ignored.

IANA management of the Routing Metric/Constraint objects identifier codespace is described in Section 9.

3. Node Metric/Constraint objects

It is fairly common for LLNs to be made of nodes with heterogeneous attributes and capabilities (e.g. nodes being battery operated or not, amount of memory, etc). More capable and stable nodes may assist the most constrained ones for routing packets, which results in extension of network lifetime and efficient network operations. This is a typical use of constraint-based routing where the computed path may not be the shortest path according to some specified metrics.

3.1. Node State and Attributes object

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

The NSA object MAY be present in the DAG Metric Container. There MUST be no more than one NSA object as a constraint per DAG Metric Container, and no 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 is to be assigned by IANA (recommended value=1).

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 3: NSA object format

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, similarly to the "overload" bit used in routing protocols such as IS-IS. Algorithms used to set the overload bit and to compute path to potentially avoid node with their overload bit set are outside the scope of this document.

Data Aggregation Attribute: data fusion involves more complicated processing to improve accuracy of the output data while data aggregation mostly aims at reducing the amount of data. This 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 life time in battery operated environments. Applications where high directional data flow is expected on a regular basis may take advantage of data aggregation supported routing.

The following two bits of the NSA object are defined:

- O Flag: When set, this indicates that the node is overloaded and may not be able to process traffic.

- A Flag: When set, this indicates that the node can act as a traffic aggregator. An implementation MAY decide to add optional

TLVs (not currently defined) to further describe the node traffic aggregator functionality.

The Flag field of the NSA 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.

3.2. Node Energy object

Whenever possible, a node with low residual energy should not be selected 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.

The routing engine may prefer a "longer" path that traverses mains-powered nodes or nodes equipped with energy scavenging, rather than a "shorter" path through battery operated nodes.

Power and energy are clearly critical resources in LLNs. As yet there is no simple abstraction which adequately covers the broad range of power sources and energy storage devices used in existing LLN nodes. These include line-power, 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 4-20mA 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 which 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 like 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 LLN 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 three levels of fidelity in the solution.

The simplest solution relies on a 2-bit field encoding three types of power sources: "powered", "battery", "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, H=P_in/P_out, in units of percent. Nodes which are scavenging more power than they are consuming will register above 100. The time period for averaging power in this calculation is out of the scope of this document but something related to the discharge time of the energy storage device on the node is probably appropriate. For battery powered devices, H is the current expected lifetime divided by the desired minimum lifetime. 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 H can be calculated as the ratio of desired max power (initial energy E_0 divided by desired lifetime T) to actual power H=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 H can be calculated as the total stored energy remaining versus the target energy remaining: H= E_bat / [E_0 (T-t)/T].

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

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 be no more than one NE object as a constraint per DAG Metric Container, and no more than one NE object as a metric per DAG Metric Container.

The NE object Type is to be assigned by IANA (recommended value=2).

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 4: NE 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 5: NE sub-object format

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

The following flags are currently defined:

- o T (node Type): 2-bit field indicating the node type. When E=0x00, the node is mains powered. When E=0x01 is battery powered. When E=0x02 the node is powered by a scavenger.

- o 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 node. Conversely, battery operated node must be excluded. The I bit is used to stipulate inclusion versus exclusion. When set, this indicates that nodes of 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.

- o 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 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 is that I bit is an inclusion.

No TLV is currently defined.

The most complex solution involves a half dozen TLV parameters representing energy storage, consumption, and generation capabilities of the node, as well as desired lifetime, and will appear in a future version of this document.

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 be no more than one HP object as a constraint per DAG Metric Container, and no 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 is to be assigned by IANA (recommended value=3)

The HP routing metric object is a global routing object that characterizes a path.

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

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

Figure 6: Hop Count object format

No Flag is currently defined.

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.

3.4. Node Fanout Ratio object

The Node Fanout Ratio (NFR) object is used to provide information on the nodes current forwarding load.

The NFR object MAY be present in the DAG Metric Container. There MUST be no more than one NFR object as a constraint per DAG Metric Container, and no more than one NFR object as a metric per DAG Metric Container.

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

The NFR object Type is to be assigned by IANA (recommended value=9).

The format of the NFR 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                 |  F R  |  Optional TLVs
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ...
```

Figure 7: NFR object format

When the data traffic of the application supported by the network is known a priori, energy depletion in the network can be equalized simply by controlling the fanout ratio of router nodes.

Algorithms describing how to compute the FR value and how to use it are outside the scope of this document.

The following field of the NFR object is defined:

- FR Field: a 4-bit unsigned integer that indicates a relative fanout of the node. A value of 15 indicates a node that is very close to, or at its maximum supported fanout capability. A value of 0 indicates a very small fanout.

Unassigned bits are considered as reserved. They MUST be set to zero on transmission and MUST be ignored on receipt.

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 sub-layer protocols which allow the effective bit rate and power consumption 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 be no more than one Throughput object as a constraint per DAG Metric Container, and no 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. Each Throughput sub-object has a fixed length of 4 bytes.

The Throughput object does not contain any additional TLV.

The Throughput object Type is to be assigned by IANA (recommended value=4)

The Throughput object is a global metric.

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 8: 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 9: 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

Similarly to throughput, the latency of many LLN MAC sub-layers can be varied over many orders of magnitude, again with a corresponding change in current consumption. Some LLN MAC link layers will allow the latency to be adjusted globally on the subnet, or 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 be no more than one Latency object as a constraint per DAG Metric Container, and no 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 TLV.

The Latency object Type is to be assigned by IANA (recommended value=5)

The Latency object is a global 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 10: 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 11: 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 is degraded by external interference and multi-path interference. Multipath typically affects both directions on the link equally, whereas external interference is sometimes uni-directional. 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.

A change in link quality can affect network connectivity, thus, link quality may be taken into account as a critical routing metric. Link quality metric should be applied to each directional link unless bi-directionality is one of routing metrics.

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 Expected Transmission count Metric (ETX).

Note that an RPL implementation MAY either 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 is implementation specific and outside the scope of this document.

The LQL is global and can either be used as a metric or a constraint. When used as a metric, the LQL metric can be recorded or aggregated. For example, the DAG may require to record the LQL for all traversed links. Each node can then use the LQL to select the parent based on user defined rules (e.g. "select the path with the maximum number of links reporting a LQL value of 3"). By contrast the LQL link metric may be aggregated, in which case the sum of all LQL may be reported (additive metric) or the minimum value may be reported along the path.

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

The LQL object MAY be present in the DAG Metric Container. There MUST be no more than one LQL object as a constraint per DAG Metric Container, and no 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 is to be assigned by IANA (recommended value=6)

The LQL object is a global object that characterizes the path reliability.

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 12: LQL object format

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

Figure 13: 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.

When the LQL metric is aggregated, the LQL object body comprises one LQL Type 2 sub-object:

The format of the LQL Type 2 sub-object is as follows

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Aggregated LQL Value     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: LQL Type 2 sub-object format

Aggregated LQL Value: when used an an additive metric (A=0x00), the aggregated LQL value reports the sum of all the LQL values for all links along the path. When used to report a minimum (A=0x02), the field reports the minimum LQL value of all links along the path ignoring undetermined LQLs (Aggregated LQL Value = 0). When used to report a maximum (A=0x01), the field reports the maximum LQL value of all links along the path. When used to report a multiplication (A=0x03), and the LQL field of one of the links along the path is undetermined (LQL=0), the undetermined LQL will be ignored and not be aggregated (i.e. no reset to Aggregated LQL Value field).

4.3.2. The Expected Transmission Count (ETX) reliability object

The Expected Transmission Count (ETX) metric is the number of transmissions a node expects to make to a destination in order to successfully deliver a packet.

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 be no more than one ETX object as a constraint per DAG Metric Container, and no 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 8 bits.

The ETX object does not contain any additional TLV.

The ETX object Type is to be assigned by IANA (recommended value=7)

The ETX object is a global metric or constraint.

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 15: ETX object body format

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

Figure 16: ETX sub-object format

ETX: 16 bits. The ETX * 128 is encoded in 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 to path quality.

4.4. Link Color object

4.4.1. Link Color object description

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

The LC object can either be used 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. 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 be no more than one LC object as a constraint per DAG Metric Container, and no 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 TLV.

The LC object Type is to be assigned by IANA (recommended value=8)

The LC object may either be local or global.

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 17: LC object format

When the LC object is used as a global 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 18: 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        |I|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19: LC Type 2 sub-object format

I Bit: When cleared, this indicates that links with the specified color must be included. When set, this indicates that links with the specified color must be excluded.

The use of the LC object is outside the scope of this document.

4.4.2. Mode of operation

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

- o When used as global 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 global recorded metric, each node along the path may insert a LC object in the DAG Metric Container to report the color of the local link. If there is already a LC object reported 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 are out the scope of this document, the route optimization algorithm should be designed carefully 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.

6. Use of multiple DAG Metric Container

Since RPL options length are coded using 1 octet, their length cannot exceed 256 bytes, which also applies to the DAG Metric Container. Although in the vast majority of cases, the advertised routing metrics and constraints will not require that much space, there might be circumstances where larger space will be required, should for example a set of routing metrics be recorded along a long path. In this case, as specified in [I-D.ietf-roll-rpl], routing metrics will be carried using multiple DAG Metric Containers.

In the rest of this document, this use of multiple DAG Metric Containers will be considered as if they were actually just one long DAG Metric Container. For this to hold, nodes propagating multiple DAG Metric Containers MUST keep their order unchanged.

7. Metric consistency

Since a set of metrics and constraints will be used for links and nodes in LLN, it is particularly critical to ensure the use of consistent metric calculation mechanisms for all links and nodes in the network.

8. Metric usage

This section describes how metrics carried in the DAG Metric Container shall be used.

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

An exemple of using such a single aggregated metric is optimizing routing for node energy. The Node Energy metric (E-E field) is aggregated along pathes 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 tie-brakers in the order that they appear in the DAG Metric Container. A node propagating a DAG Metric Container MUST keep the order of metric objects unchanged.

An example of such use of multiple aggregated metrics is the following: Hop-Count as the primary criterion, LQL as the secondary criterion and Fanout Ratio as the ultimate tie-braker. In such a case, the Hop-Count, LQL and Fanout Ratio metric objects need to appear in that order in the DAG Metric Container.

The use of compound metrics, such as a polynomial function of individual metric values, will be described in a future revision of this document.

The use of recorded metrics will be described in a future revision of this document.

9. IANA Considerations

IANA is requested to establish a new top-level registry to contain all Routing Metric/Constraint objects codepoints and sub-registries.

The allocation policy for each new registry is by IETF Consensus: new values are assigned through the IETF consensus 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).

9.1. Routing Metric/Constraint type

IANA is requested to create a registry for Routing Metric/Constraint objects. Each Routing Metric/Constraint object has a type value.

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
9	Node Fanout Ratio	This document

9.2. Routing Metric/Constraint common header

IANA is requested to create a registry to manage the codespace of A field of the Routing Metric/Constraint common header.

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

IANA is requested to create a registry to manage the Flag field of the Routing Metric/Constraint common header.

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

- Bit number
- Capability Description
- Defining RFC

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

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

Bit	Description	Reference
8	Constraint/metric	This document
7	Optional Constraint	This document
5-6	Additive/Max/Min/Multi	This document
4	Global/Local	This document
3	Recorded/Aggregated	This document

9.3. NSA object

IANA is requested to create a registry to manage the codespace of the Flag field of the NSA object.

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

o Bit number

o Capability Description

o Defining RFC

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
14	Aggregator	This document
15	Overloaded	This document

9.4. Hop-Count object

IANA is requested to create a registry to manage the codespace of the Flag field of the Hop-count object.

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

o Bit number

o Capability Description

o Defining RFC

No Flag is currently defined.

10. Security considerations

Routing metrics should be handled in a secure and trustful manner. For instance, a malicious node can not advertise falsely that it has good metrics for routing and belong to the established path to have a chance to intercept packets.

11. 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, Ricahrd Kelsey, Mathilde Durvy, Phoebus Chen, Tim Winter, Mukul Goyal, Yoav Ben-Yehezkel, Matteo Paris, Omprakash Gnawali, Mads Westergreen and Mukul Goyal for their review and comments.

12. References

12.1. 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.

12.2. Informative references

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., and R. Team, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-10 (work in progress), June 2010.

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-03 (work in progress), March 2010.

[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.

[RFC3471] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.

[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.

Authors' Addresses

JP Vasseur (editor)
Cisco Systems, Inc
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
Coronis 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-ftgroup.com