ROLL Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2011

N. Dejean
Elster SAS
D. Barthel
France Telecom Orange
March 7, 2011

Selective DIS for RPL
draft-dejean-roll-selective-dis-00

Abstract

This document specifies DIS options that enrich the potential behavior of the Routing Protocol for Low Power and Lossy Networks (RPL) specified in [I-D.ietf-roll-rpl].

The goal is to enable new leaf nodes to quickly discover and attach to the routing structure, without having to wait for spontaneous DIO transmissions by neighbour routers and without causing them to reset their DIO timers.

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].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 8, 2011.

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].

[I-D.ietf-roll-rpl] has specified the minimally viable core of mechanisms for a routing protocol, called Routing Protocol for Low Power and Lossy Networks (RPL), specifically designed for LLNs.

This document specifies DIS options that enrich the behavior of RPL and that were left out of [I-D.ietf-roll-rpl] in the interest of time.

The goal is to enable new leaf nodes to quickly discover and attach to the routing structure, without having to wait for spontaneous DIO transmissions by neighbour routers and without causing them to reset their DIO timers.

Indeed, with RPL as defined in [I-D.ietf-roll-rpl], a leaf node that wants to join an already deployed LLN is confronted with the following dilemna:

- It can either wait for DIOs to be sent by neighbor routers. These transmissions may happen after a very long time, since the Trickle timers of the neighbor routers may have increased their period to a very large value, in order to save energy in a stable network. Furthermore, the transmission of a DIO packet by a neighbor router is not even guaranteed to happen during a Trickle timer period, since transmission suppression may happen (see [I-D.ietf-roll-trickle]).

- Or it elects to proactively send a DIS (DODAG Information Sollicitation). This DIS can only be sent in broadcast, since the new node does know which router to ask for. Under the specification of [I-D.ietf-roll-rpl], all routers that receive a broadcast DIS packet will reset their Trickle timer. The time to their next spontaneous DIO transmission will indeed be dramatically shortened, which is desirable, although it will not prevent potential transmission suppression. But an undesired effect is that this will induce a large energy consumption in the network for two compounding reasons: first, all neighbour routeurs will respond, irrespective of their relevance to the new node, and second, each neighbor router will send frequent DIOs until its

Trickle timer relaxes to the maximum period, even though only the first DIO is useful.

None of the choices above matches the requirements of [RFC5548].

This document defines a way to broadcast a DIS message that includes selective options and a flag in order to query responses by neighbor routers such that:

- responses are sent promptly, reducing the time the technician has to sit waiting at the customer premises to check the result of the joining process
- responses are DIOs sent using unicast, reducing the overhearing energy cost in the router neighborhood when modern MAC technologies are used
- each neighbor router only responds with a single DIO for each DIS, reducing the reception cost at the destination
- the DIO is only sent if the neighbor router matches the criteria specified in the DIS selective options, reducing the reception, collision and overhearing energy costs

Admitedly, requesting an unknown population of neighbor routers to promptly send even a single DIO may be a cause for multiple collisions. This risk is mitigated by the use of good access contention methods at the link layer and by the wise use of the DIS options. However, both conditions are beyond the control of this specification. This document therefore specifies an optional collision mitigation mechanism of its own.

2. Leaf Node bit

In the format of the DIS base object, bit 0 of the Flag field is defined as the "Leaf Node" bit.

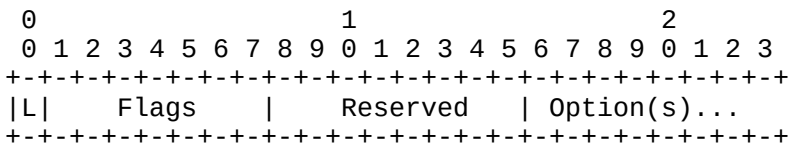


Figure 1: The DIS Base Object

A node that receives a DIS with the "Leaf Node" bit set MUST NOT reset its DIO Trickle timer, even if it matches the options carried by the DIS.

A node that receives a DIS message with the "Leaf Node" bit set and that matches the options carried in the DIS MUST reply with a unicast DIO, using the mechanism described in Section 3.2.

3. DIS Options

3.1. Metric Container

In addition to those already listed in [I-D.ietf-roll-rpl], the following option is declared valid for a DIS message:

0x02 Metric Container

A node that receives a DIS with a Metric Container option MUST ignore any Metric object in it, and MUST parse the Contraint objects in it, if any. The constraint values are compared to the values of the corresponding metrics known to the node. If both a Solicited Information option and a Metric Container option are present in a DIS message, they combine in a logical AND fashion, i.e. all predicates MUST match for the DIS to globally match.

If a Constraint objects carries a constraint for a metric the value of which is unknown to the node, it is RECOMMENDED that the node considers the constraint a match.

3.2. Response Spreading

With a wise use of the DIS options, our experience is that the population of responding routers is small enough for modern medium access techniques to efficiently resolve contention at the link layer. However, for those systems in which either above-mentioned postulate can’t be met, an optional DIO response spreading mechanism is specified here.

A new RPL control message option is defined, called "Response

Spreading", with a recommended Type value of 0x0A (to be confirmed by IANA). Its format complies with the general format of RPL options, and is described in Figure 2.

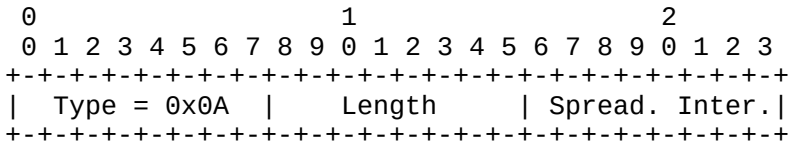


Figure 2: The Response Spreading option

A node that responds to a broadcast DIS in observance of Section 2 MUST, if that DIS includes a Response Spreading option, wait for a time uniformely drawn in the interval [O..2^SpreadingInterval], expressed in ms, before attempting to transmit its DIO. If the DIS does not include a Response Spreading option, the node is free to transmit the DIO as it otherwise would.

4. Example of use

A new leaf node that joins an established network runs an iterative algorithm by which it requests (using broadcast) network information from routers belonging to the desired network ID and which match some constraint values passed as parameters of the request. At each unsuccessful iteration, the requirements are relaxed, until one or several answers are received, or until the maximum number of iterations is reached. The answers from the routers can advantageously contain the values for other metrics than those by which the request was qualified, so that the router selection takes place based on more metrics.

The following example shows requests iterating on two constraint values (on Hop Count and Link Quality Level) and makes use of a third metric value (Node Energy) provided into the answers.

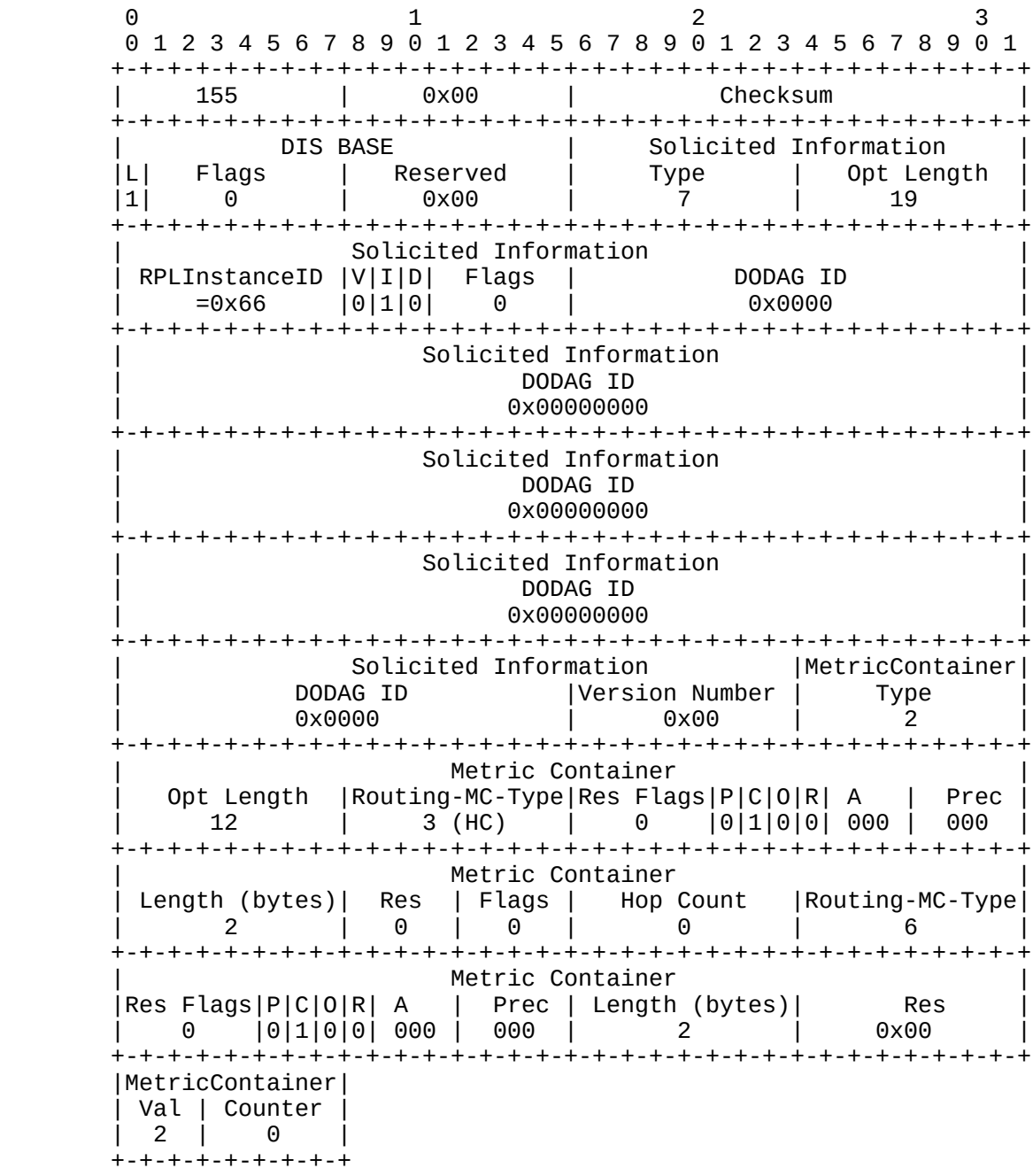
With Hop Count iterating through four different values (0-3) and Link Quality Level iterating through three possible values (2,4,6), a maximum of twelve DIS packets can be broadcast per joining node, in the following order:

- Soliciting info from routers with max Hop Count 0 and max LQL 2
- Soliciting info from routers with max Hop Count 0 and max LQL 4

- Soliciting info from routers with max Hop Count 0 and max LQL 6
- Soliciting info from routers with max Hop Count 1 and max LQL 2
- Soliciting info from routers with max Hop Count 1 and max LQL 4
- Soliciting info from routers with max Hop Count 1 and max LQL 6
- Soliciting info from routers with max Hop Count 2 and max LQL 2
- Soliciting info from routers with max Hop Count 2 and max LQL 4
- Soliciting info from routers with max Hop Count 2 and max LQL 6
- Soliciting info from routers with max Hop Count 3 and max LQL 2
- Soliciting info from routers with max Hop Count 3 and max LQL 4
- Soliciting info from routers with max Hop Count 3 and max LQL 6

Receiving any answer stops the iteration. Per our example, the new node then selects its parent router, based on the Node Energy and the Link Quality Level, according to the following algorithm:

- Reject router(s) with asymmetric connectivity (LQL seen from new node does not match the constraint value issued in the DIS request)
- Retain the router(s) that advertise the best Node Energy level
- In case of equality, select the router that boasts the best Link Quality Level.



Packet dump of DIS with Hop Count = 0, LQL <= 2

5. IANA Considerations

5.1. DIS Flag Field

IANA is requested to allocate bit O of the DIS Flag Field to become the "Leaf Node" bit, the functionality of which is described in Section 2 of this document.

Value	Meaning	Reference
0	Leaf Node	This document

5.2. RPL Control Message Options

IANA is requested to allocate a new code point in the "RPL Control Message Options" registry for the "Response Spreading" option, the behavior of which is described in Section 3.2.

Value	Meaning	Reference
0x0A	Response Spreading	This document

RPL Control Message Options

6. Security Considerations

7. Acknowledgements

8. References

8.1. Normative references

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-18 (work in progress), February 2011.

[I-D.ietf-roll-trickle]
Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", draft-ietf-roll-trickle-08 (work

in progress), January 2011.

[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.

8.2. Informative references

[I-D.ietf-roll-terminology] Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-04 (work in progress), September 2010.

[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

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-ftgroup.com

INTERNET-DRAFT
Intended Status: Experimental
Expires: June 25, 2011

A. Dvir
T. Holczer
L. Dora
L. Buttyan

January 14, 2011

<Version Number Authentication and Local Key Agreement>
<draft-dvir-roll-security-extensions-00.txt>

Abstract

Low power and Lossy Networks (LLNs) are a class of networks in which both the routers and their interconnects are constrained. LLN routers typically operate with constraints on processing power, memory, and energy (battery power). LLN router supported traffic flows include point-to-point, point-to-multipoint, and multipoint-to-point. The IPv6 Routing Protocol for LLNs (RPL) provides the mechanisms to support those traffic flows. The currently available security services in RPL will not protect against a compromised internal node that can also construct and disseminate fake messages. In this document, a service is described that prevents an internal attacker from impersonating a Destination Oriented Directed Acyclic Graph (DODAG) root. Moreover, the establishment and maintenance of any cryptographic key is out of the scope of the current RPL proposal. In this document a service that allows nodes to agree on local keys with their neighborhood is also presented.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html.

The list of Internet-Draft Shadow Directories can be accessed at

http://www.ietf.org/shadow.html.

Table of Contents

1 Terminology

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]. This document adopts and conforms to the terminology defined in [I-D.ietf-roll-terminology] and in [RFC4949].

In this document, ’compromised’ refers to taking control over a node. ’Potential DODAG roots’ are grounded DODAG roots and a small set of capable nodes that could become floating DODAG roots. ’Data authenticity’ is the assurance about the source of transmitted information (and, hereby, that information was not modified in transit).

As they form networks, LLN devices often mix the roles of ’host’ and ’router’ when compared to traditional IP networks. In this document, ’host’ refers to an LLN device that can generate but does not forward RPL [I-D.ietf-roll-rpl] traffic, ’router’ refers to an LLN device that can forward as well as generate RPL traffic, and ’node’ refers to any RPL device, either a host or a router.

2 Introduction

Low power and Lossy Networks (LLNs) consist largely of constrained nodes with limited processing power, memory, and sometimes energy, when they are battery operated. These routers are interconnected by unstable lossy links, typically supporting only low packet and data delivery rates. Another characteristic of such networks is that point-to-point is not the typical traffic pattern, but point-to-multipoint or multipoint-to-point are. Furthermore, such networks may potentially comprise up to thousands of nodes.

These characteristics offer unique challenges to a routing solution. The IETF ROLL Working Group has defined application-specific routing requirements for a Low power and Lossy Network (LLN) routing protocol, specified in [RFC5867], [RFC5826], [RFC5673], and [RFC5548]. Moreover, based on those standards, an IPv6 Routing Protocol for Low power and Lossy Networks (RPL) has been proposed [I-D.ietf-roll-rpl] and a security framework for RPL is described in [I-D-roll-security-framework].

Many LLN systems are deployed in unattended or remote locations, such as urban environments [RFC5548]. Hence, security mechanisms that provide confidentiality and authentication are critical for the operation of many applications. The currently available security services in RPL proposed in [I-D.ietf-roll-rpl] will not protect

against a compromised internal node that can also construct and disseminate fake messages. Moreover, the establishment and maintenance of any cryptographic key is out of the scope of the current RPL proposal [I-D.ietf-roll-rpl]. Therefore, this document presents two new security services for RPL:

- o DIO Message Broadcast Authentication - secures the network from misbehaving nodes to become a DODAG root and to increase the Version Number.

- o Local Key Agreement - allows each node to agree on local keys with its neighborhood.

The implementation of the security services described in this document are OPTIONAL. A given implementation MAY support a subset (including the empty set) of the described security services; for example, the implementation could support Local Key Agreement, but not DIO Message Authentication. An implementer SHOULD clearly specify which security services are supported, and it is RECOMMENDED that implementers carefully consider security requirements and the availability of security mechanisms in their network.

3 Security Services

This section describes two protocols; the first enables nodes to authenticate DIO Messages. The second protocol enables nodes to a) agree on a pairwise key, with each of its neighbors; and b) generate and disseminate a cluster key, a shared key between a node and all of its neighbors. The hash functions, MAC functions, and the digital signatures used in the protocols are based on sections 10.1 and 10.9.2 of [I-D.ietf-roll-rpl], e. g., SHA-256 hash function specified in Section 6.2 of [FIPS180], message encoding rules of Section 8.1 of [RFC3447]. The elliptic curve cryptography (ECC) used in section 3.1 is based on section 2.7 of [SECG2]. The Counter with CBC-MAC (CCM) used in section 3.2, is described in [RFC3610]. Note that although [RFC3610] disallows the CCM mode with M=0, RPL explicitly allows the CCM mode with M=0 when used in conjunction with a signature, because the signature provides sufficient data authentication. Here, the CCM mode with M=0 is specified as in [RFC3610], but where the M field in Section 2.2 of [RFC3610] MUST be set to 0. The Hashed Message Authentication Mode (HMAC) in the protocols is described in [RFC4868].

3.1 DIO Message Authentication

A grounded DODAG offers connectivity to hosts that are required to satisfy the application-defined goal. An attacker may try to become a DODAG root by sending a well-constructed DIO message where the

grounded flag is set. The scope of the current RPL security services is the link; therefore, the authenticity of the messages sent by the DODAG root relies on the trustworthiness of all intermediate nodes and the fact that none of the keys are compromised. Any key that is compromised allows an attacker to send an authentic DIO message that will be accepted by all the nodes. Therefore, a node that received the DIO message from the attacker will multicast to its neighbors the DIO message using uncompromised keys. The content of the message from the attacker will affect other nodes participating in the DODAG.

RPL [I-D.ietf-roll-rpl] allows the Version Number to be increased regularly or occasionally. Moreover, the whole network can be reconstructed by sending a DIO message with an increased Version Number. Therefore, preventing any misbehaving node from impersonating the actual DODAG root by increasing the Version Number is essential. In particular, only those parts of the DIO message that do not need to be updated when the nodes forward the DIO message can be protected. The static fields are the following:

- DIO Base Object:
  - RPLInstanceID
  - G|A|T|MOP|Prf
  - DODAGID
  - Version
- Routing Information (option)
- DODAG Configuration (option)

By authenticating the DIO message, each node can securely forward the DIO message in order to bootstrap or update the DODAG.

The Authentication procedure starts/updates from a DODAG root toward the nodes as follows:

1. The DODAG root first generates a random number r.
2. The DODAG root calculates h(h...(h(h(r)))), also denoted by h^n(r), where h() is a hash function and n is the length of the chain. This value is called the hash chain root [L1981].
3. The DODAG root authenticates the h^n(r) value as well as the static fields using any supported integrity protection

algorithm (e. g., digital signature or a MAC function).

4. The DODAG root sends a DIO message with the authenticated value.

5. Each node receiving a DIO message verifies the authenticity of the static fields of the message.

6. If the message is authentic, the node saves the Version Number value (init or update value), the hash chain value (root or current chain value), and the integrity protection data (MAC value or signature) for future use, and multicast to all neighbors the DIO message after updating the fields as described in section 6.3.1. of [I-D.ietf-roll-rpl].

7. If the message is not authentic, the receiver MUST ignore the message.

In case the implementer decides to authenticate the hash chain root with an integrity protection mechanism, steps 1-7 MUST be implemented. If not, only steps 1-2 MUST be implemented. When digital signature is used, each node has to know the public signature verification key. When symmetric keys are used, all nodes must have a preshared key K. In order to minimize the computation time and memory usage of the hash chain, the implementer can use the technique in [OptHash] on the DODAG root side.

When the DODAG root increases the Version Number (by k from the initial Version Number value), the DODAG root reveals the value of h^(n-k)(r) and inserts this value in the DIO message with Broadcast Authentication Option. When node v receives the DIO message it can easily verify the message because, if the Version Number is increased by the DODAG root, h^k(h^(n-k)(r)) must be equal to h^n(r). For an attacker, computing the previous element h^(i-1)(r) knowing h^i(r) is hard when r is not known and h() is a cryptographic one-way function.

In order to authenticate the static fields of a DIO message and the Version Number, a DIO MUST carry one or more "Broadcast Authentication" options. A Broadcast Authentication option consists of the following fields:

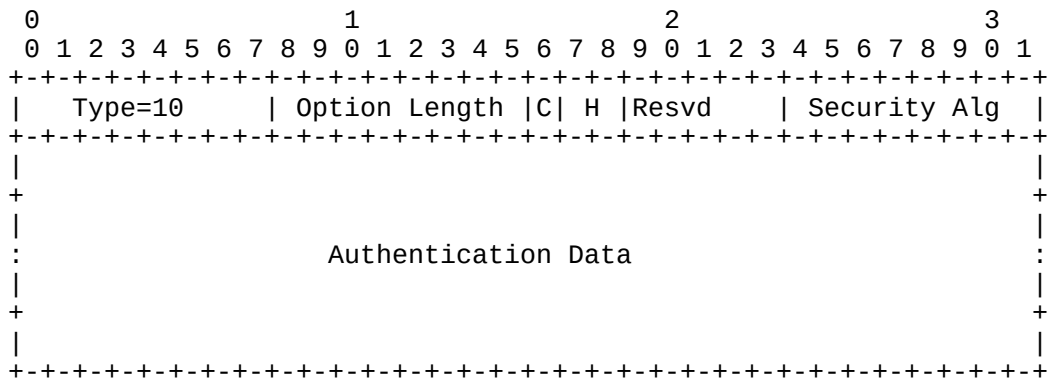


Figure 1: Format of the Broadcast Authentication Option

Option Type: 0x0A (to be confirmed by IANA)

Option Length: 8-bit unsigned integer, variable length of the option in octets excluding the Type and Length fields.

C: Continues bit, the C bit is set whenever the signature/Hash/MAC output has length greater than maximum option data length; the receiver needs to merge it with the other Broadcast Authentication Options with the same H type until the C bit is unset.

H: 2-bit field, indicating which part of the hash chain is in the Authentication data field.

Bit Number	Hash Value Type
0	No Hash Value
1	Hash Root Chain Value
2	Current Hash Chain Value
3	Unassigned

Figure 2: Hash Value Type

Resvd: 5-bit unused field. The field MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Security Algorithm: The Security Algorithm field specifies the encryption, MAC, and signature scheme used by the network. The high order bit (0x80) of the code denotes whether Integrity Protection has been enabled. The second high order bit (0xC0) of the code denotes whether the Integrity Protection is using symmetric or asymmetric key algorithms. Supported values of this field are as follows:

Bit Number	Security Algorithm
0x00	No Security Algorithm
0x01	SHA-256
0x02	SHA-512
0x80	HMAC-SHA-256
0x81	HMAC-SHA-512
0xC0	RSA with SHA-256
0xC1	ECC-SECP256K1 with SHA-256
else	Unassigned

Figure 3: Security Algorithm

Authentication Data: Contains the authentication data compatible with the Hash and Protection Type fields.

Unassigned bits of the Broadcast Authentication option are reserved. They MUST be set to zero on transmission and MUST be ignored on reception.

3.1.1 Sequence Diagram

The sequence diagram of the DIO Message Authentication has three parts: authentication procedure, Version Number update, and, admission of a new node in the DODAG.

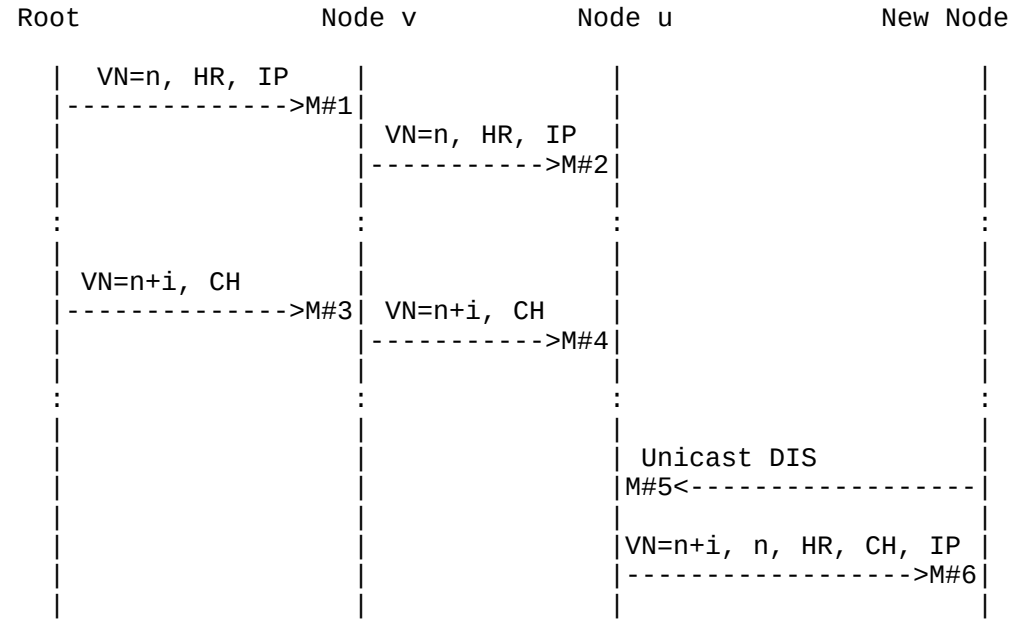
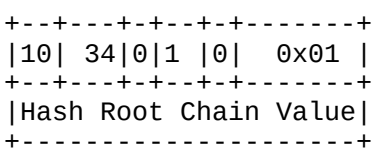


Figure 4: Sequence Diagram of DIO Message Authentication

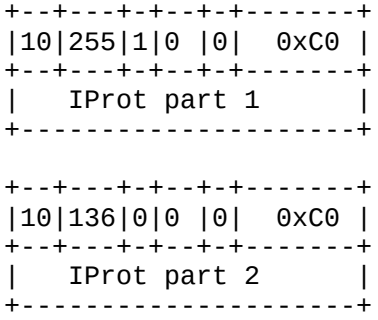
M - Message
VN - Version Number
n - Initial value of the Version Number
HR - Hash root chain value
CH - Node chain value
IP - Integrity protection

Messages #1 and #2 refer to the authentication procedure. The DIO messages (messages #1 and #2) consist of the following Broadcast Authentication Options (the format of the option is described in Figure 1):

- The value of the chain root, HR value:



- The integrity protection, IP value:

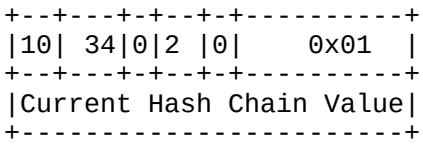


The length of the integrity protection value (3096 bits in this example) can be larger than the maximum length of the Authentication data.

Each DODAG node saves the IP value, Root value, and the initial Version Number (taken from the DIO message). Each DODAG node sends the DIO message to its neighbors.

In the case when a root wants to update the Version Number, the DIO messages (messages #3 and #4) consist of the following Broadcast Authentication Option(the format of the option is described in Figure 1):

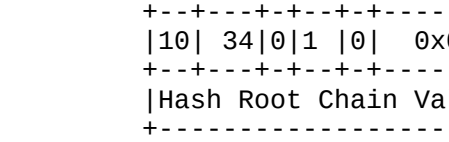
- One of the node’s value of the hash chain, CH value:



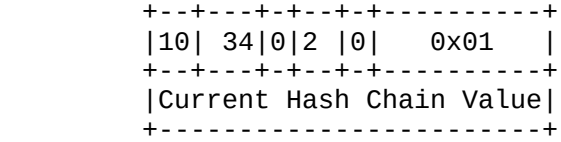
Each DODAG node verifies the values as explained above and saves the current hash value and the current Version Number (taken from the DIO message).

In the case when a new node (newcomer) wants to join the DODAG, a node receiving a unicast DIS message (message #5) from the new node (newcomer) must reply with a DIO message (message #6), consisting of the following Broadcast Authentication Options(the format of the option is described in Figure 1):

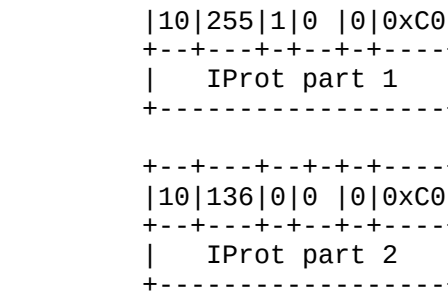
- The root chain value (HR value, as sent in message #1 and #2):



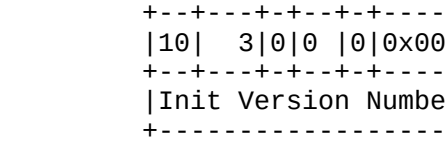
- The current hash value (CH value, as sent in messages #3, #4):



- The integrity protection, IP value, as sent in message #1 and #2:



- The initial Version Number, VN value as sent in message #1 and #2:



The new node saves the IP value, Root value, current Version Number (taken from the DIO message), and the initial Version

Number.

3.2 Local Key Agreement

Providing security is particularly challenging to LLN networks due to the resource limitations. If a group key is used for peer-to-peer communication, protection is provided only against outsider devices and not against potential malicious devices in the key-sharing group. However, local key agreements can be used despite the node limitation in order to authenticate MAC layer one-hop unicast and multicast for all neighbors' messages. The establishment and maintenance of any cryptographic key for security services is out of the scope of the current RPL proposal. This section describes two protocols, establishment of a pairwise key and establishment of a cluster key. Both protocols assume the following:

- T is defined as the lower bound on the time for an adversary to compromise a node. T is measured from the boot/restart time of the node.
- T is greater than the accumulated time required to construct a DODAG and the time to create local key agreements.
- Each node has preshared key K at boot/restart.

3.2.1 Pairwise Shared Key

This section describes a pairwise shared key agreement protocol based on the Localized Encryption and Authentication Protocol (LEAP) [LEAP]. This section does not provide results on LEAP's performance or behavior, nor does it explain the algorithm's design in detail. Interested readers should refer to [LEAP].

The pairwise key agreement consists of the following steps:

- Each node sets the safe period timer; the pairwise key agreement protocol assumes that the nodes are not compromised before this timer expires.
- Each u node derives its own key Ku=MAC(K,u), K is a preshared master key, and u is the IPv6 address of the node.
- Each Node u multicasts its identifier to all neighbors.
- Each node v receiving the identifier from u, responds with message (v, MAC(Kv, u|v)).

- The pairwise key Kuv is generated as: Kuv=MAC(Kv,u).
- After the safe period timer expires, each node deletes the preshared key K (from its memory).
- Each node has a set of pairwise keys, one for each neighbor.
- In case of conflict, a node chooses the pairwise key generated by the node with the lower id.

Figure 5 presents the messages exchanged between two neighbors in the pairwise key agreement:

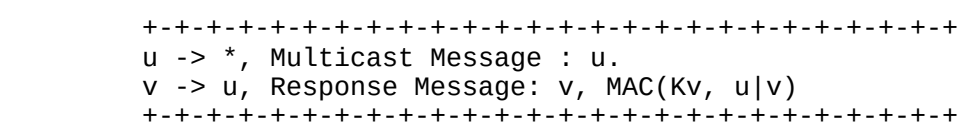


Figure 5: Messages Flow of Pairwise Key Agreement

In order to realize the pairwise key agreement, the LEAP option is presented. A LEAP option consists of the following fields:

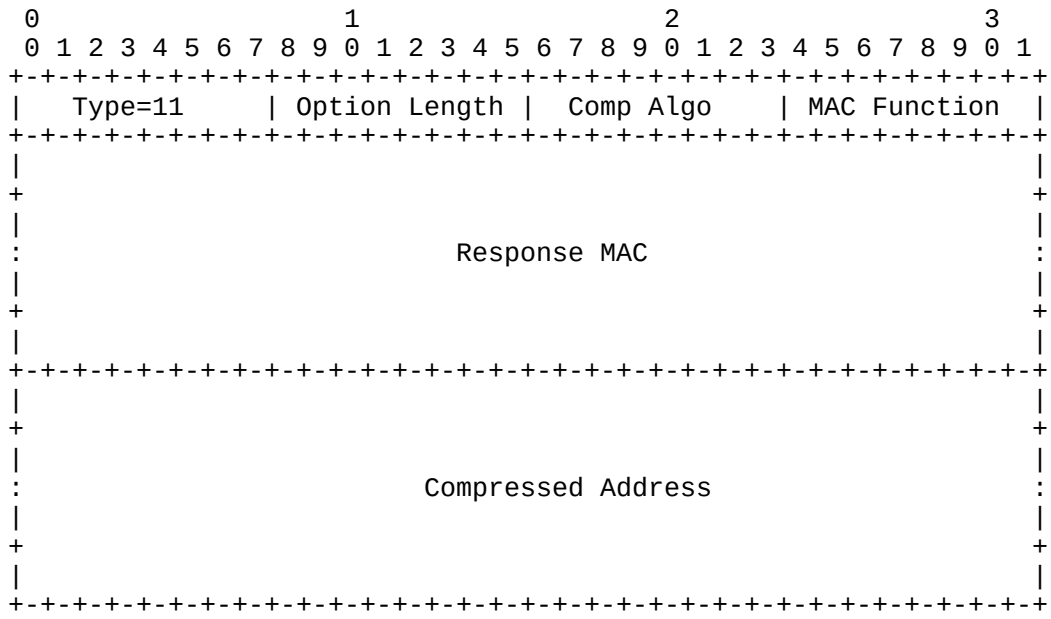


Figure 6: Format of the LEAP Response Option

Option Type: 0x0B (to be confirmed by IANA)

Option Length: Variable, length of the option in octets excluding the Type and Length fields.

Comp Algo: 8-bit field. In order to store a short version of the id (IPv6) a collision resistant hash function or the method used in Prefix Information Option(as described in section 6.7.1. of [I-D.ietf-roll-rpl]) can be used. The Compression Algorithm field indicates which (if any) compression algorithm is being used. The Compression Algorithm is encoded as in the table below:

Bit Number	Comp Algo
0x00	No Address
0x01	No Compression
0x02	SHA-1
0x03	SHA-256
0x04	Prefix Information
else	Unassigned

Figure 7: Compression Algorithm

MAC Function: 8-bit field, indicating which MAC function is being used (interested readers should refer to [PseuFun]). The length of the MAC Function is set by the algorithm. The MAC Function is encoded as in the table below:

Bit Number	MAC Function
0	HMAC-SHA-256
1	HMAC-SHA-512
else	Unassigned

Figure 8: MAC Function

Response MAC: The message authentication code is computed on the address of the sender, address of the recipient, and the key

of the sender.

Compressed Address: Indicates the compressed IPv6 destination address. The sender truncates the compressed address from the Comp Algo result. The receiver can calculate the Compressed Address length by excluding the comp, MAC, and Response MAC fields from the Option length

The pairwise key establishment can be based on RPL messages, by piggy-backing the key agreement message on RPL messages. Implementers may choose to use the LEAP option on any of the one-hop bi-directional message exchanges done in RPL based on the design considerations of their implementation. Below are lists of design considerations, possible message exchange schemes, and a matrix summarizing which design considerations are covered by each message exchange scheme.

3.2.1.1 Message Design Considerations List

The design considerations are as follows:

- RPLM: The scheme should not introduce a new RPL message type.
- RPLF: The scheme should not change RPL functionality.
- EFFI: The scheme should be efficient (low communication and computation overhead).
- STP: The local key agreement must be completed before the safe time period expires.
- BN: The scheme must work when the network boots and when a new node joins the DODAG.
- NEI: The scheme must find all of a node’s neighbors.
- MAND: The scheme should prefer mandatory RPL message types (i. e., DIO, DIS).
- RELY: The scheme should not rely on DODAG or DODAGID.

3.2.1.2 Message Exchange Schemes

The possible message exchange schemes that can be used to implement the key agreement protocol are as follows:

- S1: u -> * DAO Multicast
  v -> u DAO Unicast Ack

- S2: u -> * DAO Multicast
  v -> u DAO Multicast Ack

- S3: u -> * DAO Multicast
  v -> u DAO Multicast

- S4: u -> * DIO Multicast
  v -> u DIO Unicast

- S5: u -> * DIS Multicast
  v -> u DIO Unicast

- S6: u -> * DIS Multicast or DIO Multicast
  v -> u DIO Multicast

- S7: u -> * New RPL Base Message
  v -> u New RPL Base Message

In case the response message is a Multicast, the sender may add a number of IPv6 addresses. In order to save overhead, any algorithm to compress the addresses can be used, e. g., a collision resistance hash function, the method used in Prefix Information Option. Selecting at least one is mandatory in order to use the LEAP option.

3.2.1.3 Design Consideration vs. Message Exchange Scheme

The following matrix analyzes the design considerations vs. the message exchange schemes. The implementer needs to choose which scheme is most appropriate for its application requirements:

S	MES	RPLM	RPLF	EFFI	STP	BN	NEI	MAND	RELY
S1	DAO-M DAO-MA	+	- #0	+ #1	+	+	+	-	+
S2	DAO-M DAO-MA	- #2	+	+ #0	+	+	+	-	+
S3	DAO-M DAO-M	+	- #3	+ #4	+	+	+	-	+
S4	DIO-M DIO-U	+	- #5	+ #6	+	+	+	+	- #8
S5	DIS-M DIO-U	+	- #7	+ #6	+	+	+	+	-
S6	DIS-M DIO-M	+	+	+ #6	+	+	+	+	- #8
S7	NEW NEW	-	+	+ #9	+	+	+	-	-

Figure 9: Design Consideration vs. Message Exchange

#0 - Acknowledgement of DAO Multicast required, while the RPL [I-D.ietf-roll-rpl] states that Ack is sent to unicast messages.

#1 - The number of extra Ack messages is proportional to the number of neighbors. Those messages may potentially cause congestion and collisions.

#2 - DAO-Multi-Ack is a new type.

#3 - According to the RPL specification [I-D.ietf-roll-rpl], DAO Multicast is not sent automatically as a response to DAO Multicast.

#4 - The number of extra DAO Multicast messages is proportional to the number of neighbors. This number can be reduced with longer aggregated messages.

#5 - According to the RPL [I-D.ietf-roll-rpl], DIO Unicast is not sent automatically to DIO Multicast.

#6 - The number of extra DIO messages has an order of magnitude of the number of neighbors. Compared to other base messages, the length of a DIO message is longer.

#7 - According to the RPL [I-D.ietf-roll-rpl], the response message to DIS Multicast is DIO Multicast and not DIO Unicast.

#8 - Part of the DODAG construction.

#9 - The number of the extra new RPL messages is proportional to the number of neighbors.

For example, if S6 (using DIS Multicast and DIO Multicast) is selected for implementation, the following apply:

1. Each node periodically sends a DIS message before joining the DODAG (as described in section 17.2.1.1. of [I-D.ietf-roll-rpl]).

2. A non-DODAG node, a node that is not part of the DODAG, when receiving a DIS message, MUST ignore the message.

3. A non-DODAG node, when receiving a DIO message, follows the RPL.

4. A DODAG node, when receiving a DIS or DIO message during the Trickle interval, checks whether a pairwise key exists with the sender.

   4a. If not, the node adds a new LEAP option with the compressed address to its next DIO message, and copies the pairwise key it generates. The node also initializes a retransmission value, a maximum number each node will try to retransmit to a neighbor (can be different for different neighbors).

   4b. If a pairwise key exists, the node checks the retransmission value.

      I. If the retransmission value is greater than zero, the node adds a LEAP option to its next DIO message.

      II. Otherwise, it does not add a LEAP option.

      III. It always decreases the retransmission value.

3.2.2 Cluster Key

This section describes a cluster key agreement procedure based on the LEAP algorithm [LEAP]. This section does not provide results on LEAP's performance or behavior, nor does it explain the algorithm's design in detail. Interested readers should refer to [LEAP].

The cluster key establishment phase follows the pairwise key establishment phase. The cluster key agreement has the following steps:

- Node u first generates a random key.
- For each neighbor, node u encrypts this random key with the neighbor's pairwise key.
- For each neighbor, node u sends the encrypted random key.

The cluster key agreement can be realized with RPL messages; any RPL Unicast message is OPTIONAL. For example, a node sends a DAO unicast message with a Cluster Key Option that can carry the cluster key encrypted to each neighbor.

In order to generate a cluster key, an RPL message MUST carry a "Cluster Key" option. A Cluster Key option consists of the following fields:

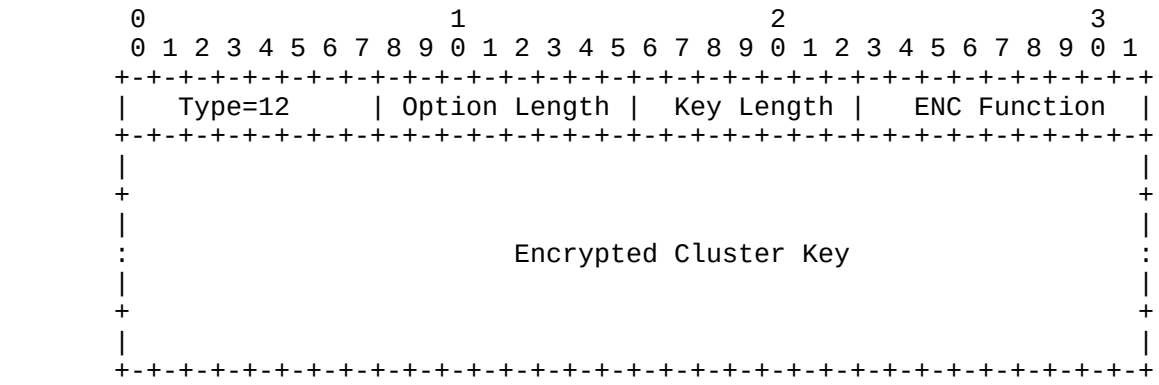


Figure 10: Format of the LEAP Cluster Key Option

Option Type: 0x0C (to be confirmed by IANA)

Option Length: Variable, length of the option in octets excluding the Type and Length fields.

Key Length: Variable, length of the Encrypted Cluster Key in octets.

ENC Function: 8-bit field, indicating which encrypted function is being used. The ENC Function is encoded as in the table below:

Bit Number	ENC Function
0	CCM with AES-128, M=0
else	Unassigned

Figure 11: Encryption Function

Encrypted Cluster Key: The encrypted value of the cluster key computed on the random key and the neighbor pairwise key.

4 Security Considerations

The security mechanisms in this standard extend the RPL security mechanisms, sections 6.1 and 10 of [I-D.ietf-roll-rpl]. Therefore, the security consideration described in section 18 of [I-D.ietf-roll-rpl] exists in this document. The scope of the current RPL security services is the link; the authenticity of the messages sent by the DODAG root relies on the trustworthiness of all intermediate nodes and the fact that none of the keys are compromised. The herein proposed DIO Message Authentication extends the data integrity and data origin authentication [RFC3552] into network level, by authenticating the static fields of the DIO message for all nodes in the DODAG.

The security mechanisms in RPL [I-D.ietf-roll-rpl] are based on symmetric-key and public-key cryptography, and use keys that are to be provided by higher/lower layer processes. However, the establishment and maintenance of these keys are out of the scope of the current RPL. The proposed local key agreement gives new procedures in order to establish and maintain pairwise and cluster keys for peer entity authentication [RFC3552]. The cryptographic protection using pairwise and cluster keys allows some flexibility and application specific tradeoffs between key storage and key maintenance costs versus the cryptographic protection provided.

The security services in this document are based on symmetric-key and public-key cryptography and assume a safe time interval after bootstrapping, during which an attacker cannot compromise a node.

The current RPL security services [I-D.ietf-roll-rpl] assume that a node wishing to join a secured network has been preconfigured with a shared key; for example, each node MAY use a secure message with KIM=0. Moreover, to join a secure RPL network, a node either listens for secure DIO messages or triggers secure DIOs by sending a secure DIS.

5 IANA Considerations

5.1 RPL Control Message Option

IANA is requested to create a registry for the RPL Control Message Options.

New values may be allocated only by an IETF Review. Each value should be tracked with the following qualities:

- Value
- Capability description
- Defining RFC

The following bits are currently defined:

Value	Description	Reference
0x0A	Broadcast Authentication	This document
0x0B	LEAP Response	This document
0x0C	Cluster Key	This Document

RPL Control Message Options

5.2 New Registry for the Hash Value Type

IANA is requested to create a registry for the Hash Value Type Field, which is contained in the Broadcast Authentication option.

New values may be allocated only by an IETF Review. Each value should be tracked with the following qualities:

- Value

o Capability description

o Defining RFC

The following bits are currently defined:

Value	Hash Value Type	Reference
0	No hash Value	This document
1	Hash Root Chain Value	This document
2	Current Hash Chain Value	This document
3	Unassigned	This document

Hash Field in Broadcast Authentication Option

5.3 New Registry for the Security Algorithm Type

IANA is requested to create a registry for the Security Algorithm Field, which is contained in the Broadcast Authentication option.

New values may be allocated only by an IETF Review. Each value should be tracked with the following qualities:

o Value

o Capability description

o Defining RFC

The following bits are currently defined:

Value	Security Algorithm	Reference
0x00	No Security Algorithm	This document
0x01	SHA-256	This document
0x02	SHA-512	This document
0x80	HMAC-SHA-256	This document
0x81	HMAC-SHA-512	This document
0xC0	RSA with SHA-256	This document
0xC1	ECC-SECP256K1 with SHA-256	This document
else	Unassigned	This document

Security Algorithm Field in Broadcast Authentication Option

5.4 New Registry for the Comp Algo Type

IANA is requested to create a registry for the Comp Algo Field, which is contained in the LEAP Response Option.

New values may be allocated only by an IETF Review. Each value should be tracked with the following qualities:

- Value
- Capability description
- Defining RFC

The following bits are currently defined:

Value	Comp Algo	Reference
0x00	No Address	This document
0x01	No Compression	This document
0x02	SHA-1	This document
0x03	SHA-256	This document
0x04	Prefix Information	This document
else	Unassigned	This document

Comp Algo Field in LEAP Response Option

5.5 New Registry for the MAC Function Type

IANA is requested to create a registry for the MAC Function Field, which is contained in the LEAP Response Option.

New values may be allocated only by an IETF Review. Each value should be tracked with the following qualities:

- Value
- Capability description
- Defining RFC

The following bits are currently defined:

Value	MAC Function	Reference
0	HMAC-SHA-256	This document
1	HMAC-SHA-512	This document
else	Unassigned	This document

MAC Function Field in LEAP Response Option

5.6 New Registry for the ENC Function

IANA is requested to create a registry for the ENC Function Field, which is contained in the LEAP Cluster Key Option.

New values may be allocated only by an IETF Review. Each value should be tracked with the following qualities:

- Value
- Capability description
- Defining RFC

The following bits are currently defined:

Value	ENC Function	Reference
0	CCM with AES-128, M=0	This document
else	Unassigned	This document

ENC Function Field in LEAP Cluster Key Option

6 Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 225186. The authors would also like to acknowledge the review and comments from Yoav Ben Yehezkel.

7 References

7.1 Normative References

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-17 (work in progress), December 2010.

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

[RFC3552] Rescorla E., and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, March 2003.

[RFC3447] Jonsson, J., and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

7.2 Informative References

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-04 (work in progress), September 2010.

[I-D-roll-security-framework]
Tsao T., Alexander R., Dohler M., Daza V., and A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", |%draft-ietf-roll-security-framework-03, (work in progress) December 2010.

[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.

[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.

[RFC4949] R. Shirey, "Internet Security Glossary", RFC 4949, FYI 36, August 2007.

[RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.

[RFC4868] Kelly, S., and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-

384, and HMAC-SHA-512 with IPsec", RFC 4868, May 2007.

[PseuFun] Goldreich, O., Goldwasser, S., and S. Micali, "How to Construct Random Functions", Journal of the ACM, Volume 33, Number. 4, 1986, pp 210-217.

[L1981] Lamport L., "Password Authentication with Insecure Communication", ACM Journal of Communications Volume 24 Issue 11, pp 770-772, Nov. 1981.

[LEAP] Zhu, S., Setia, S., and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks ", ACM conference on Computer and communications security, pp. 62-72, 2003.

[SECG2] D. R. L. Brown, "Standards for Efficient Cryptography Group (SECG), "SEC 2: Recommended Elliptic Curve Domain Parameters version 2.0", Version 2.0, January 2010.

[OptHash] Don, C., and M. Jakobsson, "Almost Optimal Hash Sequence Traversal", Fourth Conference on Financial Cryptography, 2002.

[FIPS180] National Institute of Standards and Technology, "FIPS Pub 180-3, Secure Hash Standard (SHS)", US Department of Commerce , February 2008, <http://www.nist.gov/itl/upload/fips180-3_final.pdf>.

Authors' Addresses

Amit Dvir
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Technology and Economics
BME-HIT, PO Box 91, 1521 Budapest
Hungary

EMail: azdvir@gmail.com

Tamas Holczer
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Technology and Economics
BME-HIT, PO Box 91, 1521 Budapest
Hungary

EMail: tamas.holczer@crysys.hu

Laszlo Dora
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Technology and Economics
BME-HIT, PO Box 91, 1521 Budapest
Hungary

EMail: laszlo.dora@crysys.hu

Levente Buttyan
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Technology and Economics
BME-HIT, PO Box 91, 1521 Budapest
Hungary

EMail: buttyan@crysys.hu

Networking Working Group
Internet-Draft
Intended status: BCP
Expires: September 14, 2011

O. Gnawali
P. Levis
Stanford University

March 13, 2011

Recommendations for Efficient Implementation of RPL
draft-gnawali-roll-rpl-recommendations-01

Abstract

RPL is a flexible routing protocol applicable to a wide range of Low Power and Lossy Networks. To enable this wide applicability, RPL provides many configuration options and gives implementers choices on how to implement various components of RPL. Drawing on our experiences, we distill the design choices and configuration parameters that lead to efficient RPL implementations and operations.

Table of Contents

1. Introduction

RPL [I-D.ietf-roll-rpl] is a routing protocol that is applicable in a wide range of settings in networks characterized by low power and lossy links (LLN). Because RPL is designed to work in a wide range of settings, it offers many configuration parameters and choices in how different mechanisms are implemented. This flexibility is essential to ensure the wide applicability of this protocol.

One can take advantage of this flexibility to implement and configure RPL in the most efficient way for a given network. However, it is easy to inadvertently configure RPL to work inefficiently in the network. These design choices must be made carefully drawing on implementation and operational experiences.

In this document, we describe aspects of configuration and mechanisms that impact the performance of RPL. We hope these descriptions serve as guidelines and best practices for RPL implementers and enables them to understand why certain design and configuration choices are favored over others.

2. Terminology

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

This terminology used in this document is consistent with the terminologies described in [I-D.ietf-roll-terminology], [I-D.ietf-roll-rpl], and [I-D.ietf-roll-routing-metrics].

This document does not introduce new terms.

3. Set the Minimum Trickle Interval with Care

The minimum Trickle interval determines the fastest rate at which RPL will send DIOs. It is not useful to have multiple DIOs in the transmit queue at a given node. The information in the older DIOs is likely already stale when the new DIO is generated. In systems that cannot cancel the packets that are already in the queue, it is advisable to set the minimum interval to be much larger than the minimum link layer packet time.

4. Use Large Maximum Trickle Interval

The maximum Trickle interval determines the slowest rate at which RPL will send DIOs. It is recommended that the maximum interval is set to several hours. A large interval does not necessarily make RPL less agile or the routing information stale. Trickle will operate at a rate between the minimum and maximum interval depending on the dynamics in the network.

5. Use Small Trickle Redundancy Constant

If a node receives more DIOs than the redundancy constant, it does not transmit, i.e., suppresses, its DIO. The rationale for this suppression is that the additional DIOs do not help discover new or better paths if certain number of DIOs have already been transmitted in the neighborhood of a node. In general, the smaller this number the more efficient the route discovery. Setting this value too small can lead to network partitioning as many nodes will suppress their DIOs and will not be discovered. A constant of 3-5 has been found adequate in deployments.

6. Poison Route Sparingly

It is often not necessary for a node to poison a route explicitly by advertising a rank of INFINITY. With datapath validation, it is easy to detect a loop and coupled with adaptive beaconing, the routes can be repaired quickly without additional explicit mechanism for route poisoning. Poisoning the route does not prevent loops because the control packet can get dropped on the lossy link.

7. Preserve Neighbor Information

The neighborhood information is useful even when a node detects that it has lost a route. It is recommended that the nodes not flush the entire or subset of the neighbor table even when a node loses its route or detects a loop. It is sufficient to mark the nodes in the table with the updated information that resulted in route loss or loops, e.g., marking the particular parent with a rank of INFINITY.

8. Slow-Down Datapath Traffic During Path Inconsistency

When a node detects that a path is inconsistent through datapath validation, it tasks the control plane to repair the topology and make it consistent. During this time, although the route is

available, it is advisable that the data packets are sent at lower rates to reduce contention with the control packets. This slow-down can increase data packet latency or lead to queue overflow.

9. Choose Better Path Cost Over Route Stability

With bursty links, a link metric designed to reflect link quality accurately can change rapidly. Other link metrics may also change rapidly. As a result, the path cost computed using these agile metrics can change rapidly. Selecting the best path then implies frequent parent changes. Route flapping is not detrimental to the performance of many network protocols such as sensor data collection over UDP. Hence, oftentimes, it is better to optimize for path cost than for path stability.

10. Acknowledgements

Thanks to Ulrich Herberg and Mukul Goyal for valuable comments.

11. IANA Considerations

None.

12. Security Considerations

Security considerations to be developed in accordance to the output of the WG.

13. References

13.1. Normative References

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

13.2. Informative References

[I-D.ietf-roll-routing-metrics]
Vasseur, J. and D. Networks, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-01 (work in progress), October 2009.

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-18 (work in progress), February 2011.

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-01 (work in progress), May 2009.

Authors' Addresses

Omprakash Gnawali
Stanford University
S255 Clark Center, 318 Campus Drive
Stanford, CA 94305
USA

Phone: +1 650 725 6086
Email: gnawali@cs.stanford.edu

Philip Levis
Stanford University
358 Gates Hall, Stanford University
Stanford, CA 94305
USA

Email: pal@cs.stanford.edu

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: August 27, 2011

M. Goyal, Ed.
University of Wisconsin Milwaukee
E. Baccelli
INRIA
J. Martocci
Johnson Controls
February 23, 2011

Identifying Defunct DAGs in RPL
draft-goyal-roll-defunct-dags-00

Abstract

This document specifies a mechanism for an RPL node to identify defunct directed acyclic graphs.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 27, 2011.

Copyright Notice

Table of Contents

1. Introduction

RPL [I-D.ietf-roll-rpl], an IPv6 routing protocol for low power and lossy networks (LLNs), allows the formation of directed acyclic graphs (DAGs) in an LLN. These DAGs are used for routing data traffic within the LLN as well as to reach destinations outside the LLN via the DAG root(s). A DAG can be categorized as "grounded" or "floating" based on whether joining the DAG allows a node to meet an application specific goal or not. A DAG can be categorized as "global" or "local" depending on whether the RPL Instance (identified by the RPLInstanceID), to which the DAG belongs, is globally unique or not. A DAG may be permanent in nature or exist temporarily [I-D.ietf-roll-rpl] [I-D.ietf-roll-p2p-rpl]. A DAG is uniquely identified by the combination of its RPLInstanceID, DODAGID and DODAGVersionNumber. A node, running RPL, can join at most one DAG within an RPL Instance.

As described in Section 17.4.2 in [I-D.ietf-roll-rpl], an RPL node needs to maintain a certain state about each DAG it belongs to. This state includes the tuple (RPLInstanceID, DODAGID, DODAGVersionNumber) to identify the DAG, the node’s current Rank as well as the minimum Rank (L) the node has had in this DAG, the set of parents the node has in the DAG and the Trickle timers that govern the sending of DODAG Information Object (DIO) messages by the node for the DAG [I-D.ietf-roll-trickle]. This state, except the Trickle timers, needs to be maintained for a certain time duration even when the node has no parent left in the DAG. This is done to ensure that the node does not join an earlier version of the DAG and it does not rejoin the DAG version represented by the DODAGVersionNumber value at a rank higher than L + DAGMaxRankIncrease, where DAGMaxRankIncrease is a configurable RPL parameter [I-D.ietf-roll-rpl].

Given the strict memory constraints faced by nodes in an LLN [RFC5548] [RFC5673] [RFC5826] [RFC5867], it is imperative that RPL protocol has a mechanism that allows a node to identify defunct DAGs and delete the state it maintains for such DAGs. This document specifies such a mechanism.

1.1. Defunct DAGs

An RPL node removes a neighbor from its parent set for a DAG:

- If the neighbor is no longer reachable, as determined using a mechanism such as Neighbor Unreachanility Detection (NUD) [RFC4861], Bidirectional Forwarding Detection (BFD) [RFC5881] or L2 triggers [RFC5184]; or

- o If the neighbor advertises in its DIO an infinite rank in the DAG; or
- o If keeping the neighbor as a parent would required the node to increase its rank beyond L + DAGMaxRankIncrease; or
- o If the neighbor advertises in its DIO membership in a different DAG within the same RPL Instance, where a different DAG is recognised by a different DODAGID or a different DODAGVersionNumber.

Even if the conditions listed above exist, an RPL node may fail to remove a neighbor from its parent set because:

- o The node fails to receive the neighbor's DIOs advertising an increased rank or the neighbor's membership in a different DAG;
- o The node may not check, and hence may not detect, the neighbor's unreachability for a long time. For example, the node may not have any data to send to this neighbor and hence may not encounter any event (such as failure to send data to this neighbor) that would trigger a check for the neighbor's reachability.

In such cases, a node would continue to consider itself attached to a DAG even if all its parents in the DAG are unreachable or have moved to different DAGs. Such a DAG can be characterized as being defunct from the node's perspective. If the node maintains state about a large number of defunct DAGs, such state may prevent a considerable portion of the total memory in the node from being available for more useful purposes.

To alleviate the problem described above, this document specifies a mechanism for an RPL node to identify the defunct DAGs and delete the state it maintains for such DAGs. Note that, given the proactive nature of RPL protocol, the lack of data traffic using a DAG can not be considered a reliable indication of the DAG's defunction. Further, the Trickle timer based control of DIO transmissions means the possibility of an indefinite delay in the receipt of a new DIO from a functional DAG parent. Hence, the mechanism specified in this document is based on the use of a multicast DODAG Information Solicitation (DIS) message to solicit DIOs about a DAG suspected of defunction.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and

"OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Additionally, this document uses terminology from [I-D.ietf-roll-terminology] and [I-D.ietf-roll-rpl]. Specifically, the term RPL node refers to an RPL router or an RPL host as defined in [I-D.ietf-roll-rpl].

3. The No Inconsistency Flag in the DIS Base Object

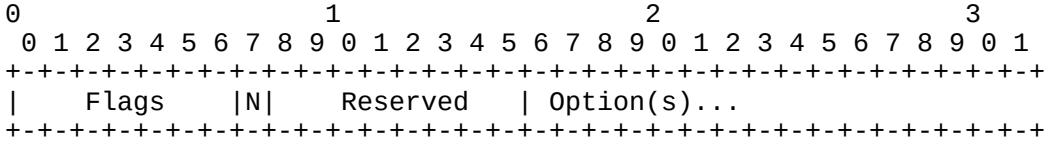


Figure 1: The No Inconsistency Flag in DIS Base Object

An RPL node can use a DODAG Information Solicitation (DIS) message to solicit DODAG Information Object (DIO) messages from its neighbors. A DIS may carry a Solicited Information option that specifies the predicates of the DAG(s) the node is interested in. In the absence of a Solicited Information option, it is assumed that the node generating the DIS is interested in receiving DIOs for all the DAGs. In the following discussion, we use the term "DIS predicates" to refer to both cases. If the DIS does not contain a Solicited Information option, all DAGs will match the DIS predicates; otherwise only those DAGs match the DIS predicates that satisfy the predicates specified in the Solicited Information option contained in the DIS.

A DIS can be multicast to all the in-range neighbors or it can be unicast to a specific neighbor. Unless restricted by a DIS flag, an RPL node must consider the receipt of a multicast DIS as an inconsistency and hence reset its Trickle timers [I-D.ietf-roll-trickle] for the DAGs that match the DIS predicates. The receipt of a unicast DIS causes an RPL node to generate the DIOs for all the DAGs matching the DIS predicates without resetting the Trickle timers.

This document defines a "No Inconsistency" (N) flag inside the DIS base object. The modified DIS base object format is shown in Figure 1. An RPL node, generating a DIS, MUST set this flag if it solicits DIOs for the purpose of identifying the defunct DAGs as specified in this document. On receiving a unicast/multicast DIS with N flag set, an RPL node MUST NOT reset the trickle timers for the DAGs that match the DIS predicates. For each DAG matching the

predicates of a multicast DIS received with N flag set, an RPL node MUST schedule a DIO transmission after a time duration between Imin/2 and Imin, where Imin is the minimum Trickle interval size [I-D.ietf-roll-trickle] associated with the DAG. For each DAG matching the predicates of a unicast DIS received with N flag set, an RPL node MUST immediately generate a DIO.

4. Identifying A Defunct DAG

When an RPL node has not received a DIO from any of its parents in a DAG for more than MaxSilence*Imax seconds, where MaxSilence is a configurable parameter greater than 1 and Imax is the maximum Trickle interval size [I-D.ietf-roll-trickle] associated with the DAG:

- The node MUST generate a multicast DIS message that carries a Solicited Information option and has N flag set. The Solicited Information option MUST have the I and D flags set and the RPLInstanceID/DODAGID fields MUST be set to values identifying the DAG. The V flag inside the Solicited Information option SHOULD NOT be set so as to allow neighbors to send DIOs advertising the latest version of the DAG.

- After sending the DIS, the node MUST wait for Imin duration, where Imin is the minimum Trickle interval size associated with the DAG, to receive the DIOs generates by its neighbors.

- At the conclusion of the wait period:

  * If the node has received one or more DIOs advertising newer version(s) of the DAG, it MUST join the latest version of the DAG, select a new parent set among the neighbors advertising the latest DAG version and mark the DAG status as functional.

  * Otherwise, if the node has not received a DIO advertising the current version of the DAG from a neighbor in the parent set, it MUST remove that neighbor from the parent set. As a result, if the node has no parent left in the DAG, it MUST mark the DAG as defunct and schedule the deletion of the state it has maintained for the DAG after DAGHoldTime duration, a configurable parameter.

An RPL node SHOULD check the functional status of a DAG it belongs to in the manner described above at least once during a CheckDAGStatusTime interval, which is a configurable parameter.

5. Security Considerations

TBA

6. IANA Considerations

TBA

7. Acknowledgements

We gratefully acknowledge Thomas Clausen for motivating this draft.

8. References

8.1. Normative References

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-18 (work in progress), February 2011.

[I-D.ietf-roll-trickle]
Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", draft-ietf-roll-trickle-08 (work in progress), January 2011.

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

8.2. Informative References

[I-D.ietf-roll-p2p-rpl]
Goyal, M., Baccelli, E., Brandt, A., Cragie, R., Martocci, J., and C. Perkins, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-02 (work in progress), February 2011.

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-04 (work in progress), September 2010.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5184] Teraoka, F., Gogo, K., Mitsuya, K., Shibui, R., and K. Mitani, "Unified Layer 2 (L2) Abstractions for Layer 3 (L3)-Driven Fast Handover", RFC 5184, May 2008.

[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.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.

Authors' Addresses

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53201
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: http://www.emmanuelbaccelli.org/

Jerald Martocci
Johnson Controls
507 E Michigan St
Milwaukee, WI 53202
USA

Phone: +1 414-524-4010
Email: jerald.p.martocci@jci.com

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: August 27, 2011

M. Goyal, Ed.
University of Wisconsin Milwaukee
E. Baccelli
INRIA
J. Martocci
Johnson Controls
February 23, 2011

The Direction Field in Routing Metric/Constraint Objects Used in RPL
draft-goyal-roll-metrics-direction-00

Abstract

This document specifies a Direction field in the Routing Metric/ Constraint objects used in RPL operation in low power and lossy networks.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 27, 2011.

Copyright Notice

Table of Contents

1. Introduction

Asymmetric links are a common observation in low power and lossy networks (LLNs) [sang_2010]. Many link-level routing metrics have a directional aspect. Although such routing metrics can be defined in a bidirectional manner so as to account for the link properties in both directions, this is not always desirable. In the context of RPL [I-D.ietf-roll-rpl], the IPv6 routing protocol for LLNs, it may be necessary to measure a link-level routing metric in a particular direction. For example, if the intent is to build a directional acyclic graph (DAG) specifically for the purpose of low latency communication to the DAG root, the routing metric must measure the link latency in Up direction, i.e., towards the DAG root, as defined in [I-D.ietf-roll-rpl]. Similarly, if a temporary DAG is being constructed to discover a point-to-point route towards a destination [I-D.ietf-roll-p2p-rpl], the routing metric must calculate the relevant link characteristic in Down direction, i.e., away from the DAG root, as defined in [I-D.ietf-roll-rpl]. Thus, there is a need to specify the directional aspect of a link-level routing metric.

Accordingly, this document defines a Direction field inside the Routing Metric/Constraint object header, defined in [I-D.ietf-roll-routing-metrics]. The Direction field is defined in two previously reserved bits inside the Routing Metric/Constraint object header. The modified Routing Metric/Constraint object header is backward compatible with its definition in [I-D.ietf-roll-routing-metrics].

2. Terminology

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

Additionally, this document uses terminology from [I-D.ietf-roll-terminology] and [I-D.ietf-roll-rpl]. Specifically, the term RPL node refers to an RPL router or an RPL host as defined in [I-D.ietf-roll-rpl].

3. The Direction Field

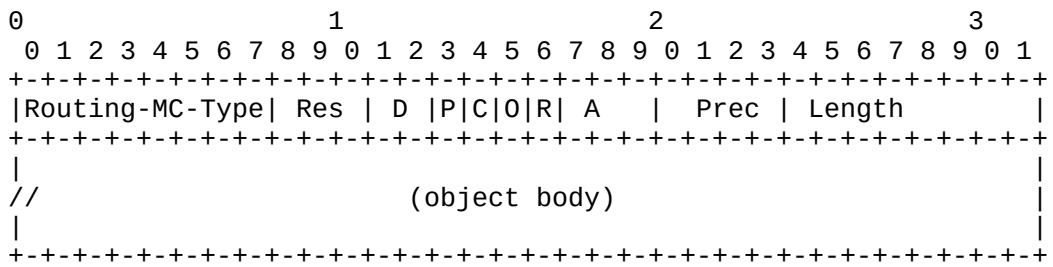


Figure 1: Routing Metric/Constraint object generic format

The modified Routing Metric/Constraint object header is illustrated in Figure 1. The Direction (or D) field is a 2-bit field that indicates the direction associated with the routing metric/ constraint:

- D = 0x00: undefined;
- D = 0x01: Up;
- D = 0x02: Down;
- D = 0x03: Bidirectional.

If the D field has value 0x00, the direction associated with the routing metric/constraint is undefined as in [I-D.ietf-roll-routing-metrics]. A value 0x00 for the D field may be suitable for node-level routing metrics/constraints defined in [I-D.ietf-roll-routing-metrics]. The D field value in link-level routing metrics/constraints SHOULD NOT be set to 0x00.

This document does not specify how to measure/evaluate a routing metric/constraint object in the direction specified by the D field. The measurement/evaluation methodology for specific routing metrics/ constraints, taking in account the D field, may be specified in a separate document.

A routing metric/constraint object MUST be measured/evaluated in accordance with its D field value if defined. In case, an RPL node can not measure/evaluate the routing metric/constraint object in the specified direction, the following rules MUST be applied:

- If the object is a recorded metric, i.e., has C=0 and R=1 fields, the RPL node MUST set the P flag inside the object, thereby

indicating the partial nature of the recorded metric.

- If the object is an aggregated metric, i.e., has C=0 and R=0 fields, the RPL node MUST drop the DIO containing the object.
- If the object is a mandatory constraint, i.e., has C=1 and O=0 fields, the RPL node MUST drop the DIO containing the object.
- If the object is an optional constraint, i.e., has C=1 and O=1 fields, the RPL node MAY drop the DIO containing the object or it MAY continue processing rest of the DIO ignoring this object.

4. Security Considerations

TBA

5. IANA Considerations

This document does not have any IANA considerations.

6. References

6.1. Normative References

[I-D.ietf-roll-routing-metrics]
Vasseur, J., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-17 (work in progress), January 2011.

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-18 (work in progress), February 2011.

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

6.2. Informative References

[I-D.ietf-roll-p2p-rpl]
Goyal, M., Baccelli, E., Brandt, A., Cragie, R., Martocci,

J., and C. Perkins, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-02 (work in progress), February 2011.

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-04 (work in progress), September 2010.

[sang_2010]
Sang, L., Arora, A., and H. Zhang, "On Link Asymmetry and One-way Estimation in Wireless Sensor Networks", ACM Transactions on Sensor Networks Volume 6, Number 2, February 2010.

Authors' Addresses

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53201
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: http://www.emmanuelbaccelli.org/

Jerald Martocci
Johnson Controls
507 E Michigan St
Milwaukee, WI 53202
USA

Phone: +1 414-524-4010
Email: jerald.p.martocci@jci.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 29, 2011

M. Goyal, Ed.
University of Wisconsin Milwaukee
E. Baccelli, Ed.
INRIA
P2P. Team
October 26, 2010

A Mechanism to Measure the Quality of a Point-to-point Route in a Low Power and Lossy Network
draft-goyal-roll-p2p-measurement-01

Abstract

This document specifies a mechanism that enables an RPL node to measure the quality of an existing route to/from another RPL node in a low power and lossy network, thereby allowing the node to decide if it wants to initiate the discovery of a more optimal route.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 29, 2011.

Table of Contents

1. Introduction

Point to point (P2P) communication between arbitrary nodes in a Low power and Lossy Network (LLN) is a key requirement for many applications [RFC5826][RFC5867]. RPL [I-D.ietf-roll-rpl], the IPv6 Routing Protocol for LLNs, constrains the LLN topology to a Directed Acyclic Graph (DAG) built to optimize routing costs to reach the DAG's root and requires the P2P routes to use the DAG links only. Such P2P routes may potentially be suboptimal and may lead to traffic congestion near the DAG root. Additionally, RPL is a proactive routing protocol and hence all P2P routes must be established ahead of the time they are used.

To ameliorate situations, where RPL's P2P routing functionality does not meet the requirements, [I-D.ietf-roll-p2p-rpl] describes a reactive mechanism to discover P2P routes that meet the specified performance characteristics. This mechanism, henceforth referred to as the reactive P2P route discovery, requires the specification of "good enough criteria", in terms of constraints on aggregated values of the relevant routing metrics [I-D.ietf-roll-routing-metrics], that the discovered routes must satisfy. In some cases, the application requirements or the LLN's topological features allow a node to infer the good enough criteria intrinsically. For example, the application may require the end-to-end loss rate and/or latency on the route to be below certain thresholds or the LLN topology may be such that a router can safely assume its destination to be less than a certain number of hops away from itself.

When the existing P2P routes are deemed unsatisfactory by the application layer but the node does not intrinsically know the good enough criteria, it may be necessary for the node to determine the aggregated values of relevant routing metrics along the existing routes. This knowledge will allow the node to frame a reasonable good enough criteria and initiate a reactive P2P route discovery to determine better routes. For example, if the router determines the aggregate ETX [I-D.ietf-roll-routing-metrics] along an existing route to be "x", it can use "ETX < x*y", where y is a certain fraction, as a constraint in the good enough criteria. Note that it is important that the good enough criteria is not overly strict; otherwise the route discovery may fail even though routes, much better than the ones being currently used, exist.

This document specifies a mechanism that enables an RPL node to measure the aggregated values of the routing metrics along an existing route to/from another RPL node in an LLN, thereby allowing the node to decide if it wants to initiate the reactive discovery of a more optimal route and determine the good enough criteria to be used for this purpose.

1.1. Terminology

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

Additionally, this document uses terminology from [I-D.ietf-roll-terminology], [I-D.ietf-roll-rpl] and [I-D.ietf-roll-p2p-rpl]. Specifically, the term node refers to an RPL router or an RPL host as defined in [I-D.ietf-roll-rpl]. The following terms, originally defined in [I-D.ietf-roll-p2p-rpl], are redefined in the following manner.

Origin Node: The origin node refers to the node that initiates the measurement process defined in this document and is one end point of the P2P route being measured.

Target Node: The target node refers to the other end of the P2P route being measured.

Intermediate Router: A router, other than the origin and the target node, on the P2P route being measured.

2. Functional Overview

The mechanism described in this document can be used by an origin node to measure the aggregated values of the routing metrics along a P2P route to/from a target node in the LLN. Such a route could be a source route or a hop-by-hop route established using RPL [I-D.ietf-roll-rpl] or the reactive P2P route discovery [I-D.ietf-roll-p2p-rpl].

When an origin node desires to measure the aggregated values of the routing metrics along a P2P route from itself to a target node, it sends a Measurement Request message along that route. The Measurement Request message accumulates the values of the relevant routing metrics as it travels towards the target node. Upon receiving the Measurement Request message, the target node unicasts a Measurement Reply message, carrying the accumulated values of the routing metrics, back to the origin node.

When an origin node desires to measure the aggregated values of the routing metrics along a P2P route from a target node to itself, it unicasts a Measurement Request message, specifying the routing metrics to be measured, to the target node. On receiving the Measurement Request message, the target node sends a Measurement

Reply message to the origin node along the P2P route to be measured. The Measurement Reply message accumulates the values of the relevant routing metrics as it travels towards the origin node.

3. The Measurement Object (MO)

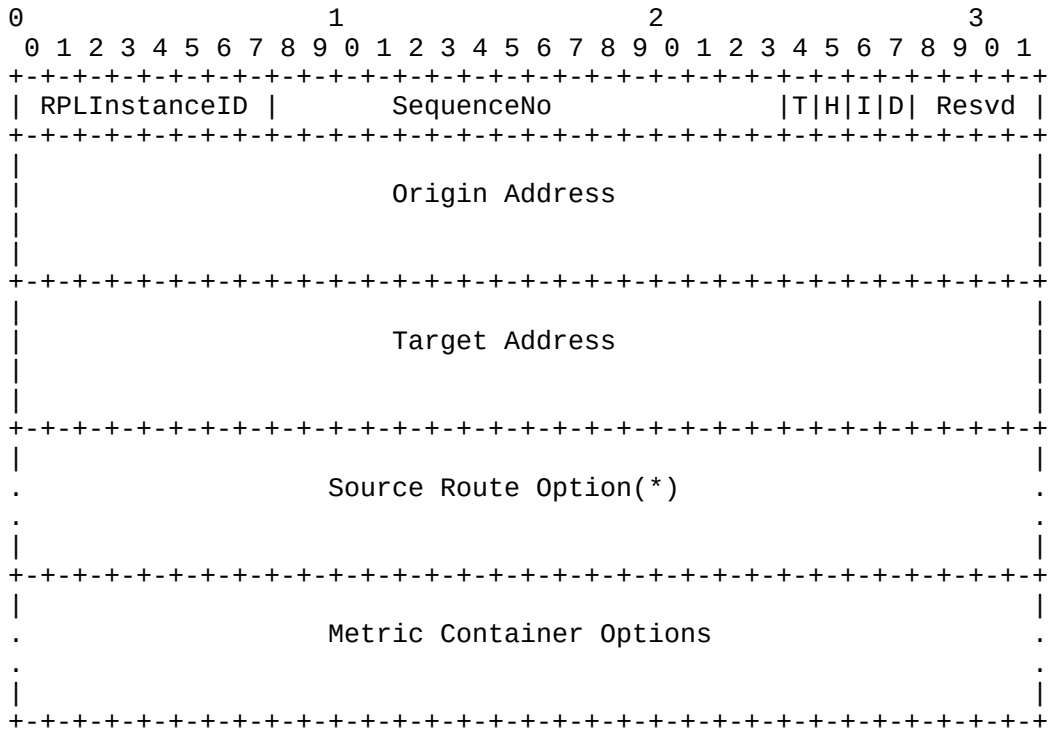


Figure 1: Format of the Measurement Object (MO)

This document defines a new RPL Control Message type, the Measurement Object (MO) with code 0x05 (to be confirmed by IANA) that serves as both Measurement Request and Measurement Reply. The format of an MO is shown in Figure 1. An MO consists of the following fields:

- RPLInstanceID: Relevant only if the MO travels along a hop-by-hop route. This field identifies the RPLInstanceID of the hop-by-hop route.

- SequenceNo: A 16-bit sequence number that uniquely identifies a Measurement Request and the corresponding Measurement Reply to the origin node.

- T: The type flag. This flag is set if the MO represents a Measurement Request. The flag is cleared if the MO is a Measurement Reply.

- H: This flag is set if the MO travels along a hop-by-hop route. In that case, the hop-by-hop route is identified by the RPLInstanceID and, if required, the Origin/Target Address serving as the DODAGID. This flag is cleared if the MO travels along a source route. In that case, the MO MUST contain a Source Route option [I-D.ietf-roll-p2p-rpl]. Note that, in case of a P2P route along a non-storing DAG, it is possible that an MO message travels along a hop-by-hop route till the DAG's root, which then sends it along a source route to its destination. In that case, the DAG root will reset the H flag and also insert a Source Route option in the MO.

- I: A flag that indicates which of the two - the Origin Address and the Target Address - indicates the DODAGID for the hop-by-hop route. This flag is relevant only if the MO travels along a hop-by-hop route (i.e., H flag is set) and a local RPLInstanceID has been specified to identify the hop-by-hop route. This flag is set if the Origin Address indicates the DODAGID; the flag is cleared if the Target Address indicates the DODAGID.

- D: A flag that indicates the direction of the P2P route. This flag is set when the route to be measured is from the origin node to the target node. Otherwise, the flag is cleared.

- Reserved: These bits are reserved for future use. These bits MUST be set to zero on transmission and MUST be ignored on reception.

- Origin Address: The IPv6 address of the origin node.

- Target Address: The IPv6 address of the target node.

- Source Route Option: An MO MUST contain one Source Route option if it travels along a source route.

- Metric Container Options: An MO MUST contain one or more Metric Container options to carry the routing metric objects [I-D.ietf-roll-routing-metrics].

4. Originating an MO To Measure a P2P Route

4.1. From the Origin Node to the Target Node

If the origin node intends to measure the routing metric values along a P2P route towards a target node, it generates an MO message and sets its fields as follows:

- RPLInstanceID: If the P2P route is a hop-by-hop route, the origin node specifies the RPLInstanceID to identify the route in this field. This field is not relevant if the P2P route is a source route specified in the Source Route option. This document RECOMMENDS a value 10000000 for this field if the P2P route is a source route.

- SequenceNo: The origin node assigns a sequence number to the MO to uniquely identify the corresponding Measurement Reply.

- T: The T flag is set to indicate that MO represents a Measurement Request.

- H: The H flag is set if the MO travels along a hop-by-hop route.

- I: This field in relevant only if the H flag is set and the RPLInstanceID is a local value. The origin node sets this flag if the Origin Address indicates the DODAGID. The origin node clears this flag if the Target Address indicates the DODAGID.

- D: This flag is set.

- Origin Address, Target Address: These fields are set to the IPv6 addresses of the origin and target nodes respectively. If the H flag is set and the RPLInstanceID is a local value, the Origin Address or the Target Address MUST also indicate the DODAGID value required to identify the hop-by-hop route.

- Source Route Option: If the P2P route is a source route (i.e., the H flag is cleared), the Source Route option MUST be present and MUST include a complete source route to the target node in forward direction (excluding the addresses of the origin and target nodes).

- Metric Container Options: The origin node MUST also include one or more Metric Container options containing relevant routing metric objects to accumulate the costs for these metrics along the P2P route. The origin node also initiates the routing metric objects by including the local values of the routing metrics for the first hop on the P2P route.

After setting the MO fields as described above, the origin node MUST

unicast the MO message to the next hop on the P2P route. The origin node MAY include a Record Route IPv6 Extension Header, proposed in [I-D.thubert-6man-reverse-routing-header], in the MO message to accumulate a reverse route that the target node can use to send the Measurement Reply back to the origin node.

4.2. From the Target Node to the Origin Node

If the origin node intends to measure the routing metric values along a P2P route from a target node to itself, it generates an MO message and sets its fields as follows:

- SequenceNo: The origin node assigns a sequence number to the MO to uniquely identify the corresponding Measurement Reply.
- T: The T flag is set to indicate that MO represents a Measurement Request.
- D: This flag is cleared.
- Origin Address, Target Address: These fields are set to the IPv6 addresses of the origin and target nodes respectively.
- Source Route Option: In this case, the MO SHOULD NOT include any Source Route option.
- Metric Container Options: The origin node MUST include one or more Metric Container options containing relevant routing metric objects to accumulate the costs for these metrics along the P2P route. These routing metric objects MUST be empty.

The other fields in the MO are not relevant in this case and SHOULD be set to zero. After setting the MO fields as described above, the origin node MUST unicast the MO message to the target node.

5. Processing a Received MO at an Intermediate Router

When a node receives an MO, it examines if one of its IPv6 addresses is listed as the Origin Address or the Target Address. If not, the node checks if H bit is clear (i.e., the MO is traveling along a source route). If yes, the node checks the Address[0] field inside the Source Route Option contained in the MO. The node MUST drop the MO with no further processing and send an ICMPv6 Destination Unreachable error message to the source of the message (the Origin Address if the MO is a Measurement Request; otherwise the Target Address) if the received MO has a clear H bit but does not contain a Source Route Option or if the Address[0] inside the Source Route

option does not match one of the node's IPv6 address.

The node then determines the next hop on the P2P route being measured. In case the received MO has a clear H flag, the Address[1] field (the second element in the Address vector) inside the Source Route Option is taken as the next hop. If the Source Route Option does not contain Address[1] element, the node checks the T flag inside the MO. If T flag is set, i.e., MO is a Measurement Request, the Target Address is taken as the next hop; otherwise the Origin Address is the next hop. If the received MO has H flag set, the node uses the RPLInstanceID, the ultimate destination of the MO (Target Address if T flag is set; otherwise the Origin Address) and, if RPLInstanceID is a local value, the DODAGID (the Origin Address if I flag is set; otherwise the Target Address) to determine the next hop for the MO. If the H flag in the MO is set and the node is the root of a non-storing DAG, indicated by the RPLInstanceID, the node MAY reset the H flag and insert a Source Route option in the MO to indicate a source route along which the MO should travel on rest of its way to its destination. The node MUST drop the MO with no further processing and send an ICMPv6 Destination Unreachable error message to the source of the message if it can not determine the next hop for the message.

After determining the next hop, the node updates the routing metric objects, contained in the Metric Container options inside the MO, either by updating the aggregated value for the routing metric or by attaching the local values for the metric inside the object. The node MUST drop the MO with no further processing and send a suitable ICMPv6 error message to the source of the message if the node does not know the relevant routing metric values for the next hop.

After updating the routing metrics, the node MUST unicast the MO to the next hop. If the MO to be forwarded has a clear H flag, the node MUST ensure that the Address vector in the Source Route option contains the next hop address as the first element.

6. Processing a Received MO at the Target Node

When a node receives an MO, it examines if one of its IPv6 addresses is listed as the Target Address. If yes, the node checks the T flag. The node MUST drop the MO with no further processing and optionally log an error if the T flag is clear (i.e. the received MO is a Measurement Reply).

The target node then checks the D flag to determine the direction of the P2P route to be measured. If the D flag is set (i.e., the P2P route to be measured is from the origin node to the target node), the

target node updates the routing metrics objects in the Metric Container options if required, removes the Source Route Option if present and clears the T bit thereby converting the MO into a Measurement Reply. The target node then unicasts the updated MO back to the origin node. For this purpose, the target node MAY use the reverse route accumulated in the Record Route IPv6 Extension Header [I-D.thubert-6man-reverse-routing-header] if present in the received MO message.

If the D flag in the received MO message is clear (i.e., the P2P route to be measured is from the target node to the origin node), the target node selects the P2P route to be measured and modifies the following MO fields:

- RPLInstanceID: If the P2P route is a hop-by-hop route, the target node specifies in this field the RPLInstanceID associated with the route. This field is not relevant if the P2P route is a source route. This document RECOMMENDS a value 10000000 for this field if the P2P route is a source route.

- T: The T flag is cleared to indicate that MO represents a Measurement Reply.

- H: The H flag is set if the P2P route is a hop-by-hop one.

- I: If the H flag is set and the RPLInstanceID is a local value, the target node sets this flag if the Origin Address indicates the DODAGID. The target node clears this flag if the Target Address indicates the DODAGID.

- D: This flag is cleared.

- Source Route Option: If the P2P route is a source route, the Source Route option MUST be present and MUST include a complete source route from the target node to the origin node (excluding the addresses of the target and origin nodes).

- Metric Container Options: The target node MUST initiate the routing metric objects inside the Metric Container options by including the local values of the routing metrics for the first hop on the P2P route.

The target node need not modify the other fields in the received MO. After these modifications, the target node MUST unicast the MO message to the next hop on the P2P route.

7. Processing a Received MO at the Origin Node

When a node receives an MO, it examines if one of its IPv6 addresses is listed as the Origin Address. If yes, the node checks the T flag. The node MUST drop the MO with no further processing and optionally log an error if the T flag is set (i.e. the received MO is a Measurement Request) or if the node has no recollection of sending a Measurement Request with the sequence number listed in the received MO.

If the D flag in the received MO is clear (i.e., the P2P route to be measured is from the target node to the origin node), the origin node MUST update the routing metrics objects in the Metric Container options if required.

The origin node can now examine the routing metric objects inside the Metric Container options to evaluate the quality of the measured P2P route. If a routing metric object contains local metric values recorded by enroute nodes, the origin node MAY aggregate these local values into an end-to-end value as per the aggregation rules for the metric.

8. Security Considerations

TBA

9. IANA Considerations

TBA

10. Authors and Contributors

In addition to the editors, the authors of this document include the following individuals (listed in alphabetical order).

Anders Brandt, Sigma Designs, Emdrupvej 26A, 1., Copenhagen, Dk-2100, Denmark. Phone: +45 29609501; Email: abr@sdesigns.dk

Robert Cragie, Gridmerge Ltd, 89 Greenfield Crescent, Wakefieldm WF4 4WA, UK. Phone: +44 1924910888; Email: robert.cragie@gridmerge.com

Jerald Martocci, Johnson Controls, Milwaukee, WI 53202, USA. Phone: +1 414 524 4010; Email:jerald.p.martocci@jci.com

Charles Perkins, Tellabs Inc., USA. Email:charliep@computer.org

Authors gratefully acknowledge the contributions of Richard Kelsey and Zach Shelby in the development of this document.

11. References

11.1. Normative References

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

11.2. Informative References

[I-D.ietf-roll-p2p-rpl]
Goyal, M. and E. Baccelli, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-00 (work in progress), August 2010.

[I-D.ietf-roll-routing-metrics]
Vasseur, J., Kim, M., Networks, D., Dejean, N., and D. Barthel, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-11 (work in progress), October 2010.

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Networks, D., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-13 (work in progress), October 2010.

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-04 (work in progress), September 2010.

[I-D.thubert-6man-reverse-routing-header]
Thubert, P., "Reverse Routing Header", draft-thubert-6man-reverse-routing-header-00 (work in progress), June 2010.

[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

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53211
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli (editor)
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: http://www.emmanuelbaccelli.org/

P2P Team

Networking Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 29, 2011

O. Gnawali
P. Levis
Stanford University
April 27, 2011

The Minimum Rank Objective Function with Hysteresis
draft-ietf-roll-minrank-hysteresis-of-02

Abstract

The Routing Protocol for Low Power and Lossy Networks (RPL) uses objective functions to construct routes that optimize or constrain the routes it selects and uses. This specification describes the Minimum Rank Objective Function with Hysteresis (MRHOF), an objective function that selects routes that minimize a metric, while using hysteresis to reduce churn in response to small metric changes. MRHOF works with metrics that are additive along a route, and the metric it uses is determined by the metrics RPL Destination Information Object (DIO) messages advertise.

Table of Contents

1. Introduction

An objective function specifies how RPL [I-D.ietf-roll-rpl] selects paths. Objective functions can choose paths based on routing metrics or constraints. For example, if an RPL instance uses an objective function that minimizes hop-count, RPL will select paths with minimum hop count.

The nodes running RPL might use a number of metrics to describe a link or a node [I-D.ietf-roll-routing-metrics] and make it available for route selection. These metrics are advertised in RPL Destination Information Object (DIO) messages using a Metric Container suboption. An objective function can use these metrics to choose routes.

To decouple the details of an individual metric or objective function from forwarding and routing, RPL describes routes through a value called Rank. Rank, roughly speaking, corresponds to the distance associated with a route. An objective function is responsible for computing a node’s advertised Rank value based on the Rank of its potential parents, metrics, and other network properties.

This specification describes MRHOF, an objective function for RPL. MRHOF uses hysteresis while selecting the path with the smallest metric value. The metric that MRHOF uses is determined by the metrics in the DIO Metric Container. For example, the use of MRHOF with the latency metric allows RPL to find stable minimum-latency paths from the nodes to a root in the DAG instance. The use of MRHOF with the ETX metric allows RPL to find the stable minimum-ETX paths from the nodes to a root in the DAG instance.

MRHOF can only be used with an additive metric that must be minimized on the paths selected for routing.

2. Terminology

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

This terminology used in this document is consistent with the terminologies described in [I-D.ietf-roll-terminology], [I-D.ietf-roll-rpl], and [I-D.ietf-roll-routing-metrics].

This document introduces two terms:

Selected metric: The metric chosen by the network operator to use for path selection. This metric can be any additive metric listed in [I-D.ietf-roll-routing-metrics].

Path cost: Path cost quantifies a property of an end-to-end path. Path cost is obtained by summing up the selected metric of the links or nodes along the path. Path cost can be used by RPL to compare different paths.

Worst parent: The node in the parent set with the largest path cost.

3. The Minimum Rank Objective Function with Hysteresis

The Minimum Rank Objective Function with Hysteresis, MRHOF, is designed to find the paths with the smallest path cost while preventing excessive churn in the network. It does so by finding the minimum cost path and switching to that path only if it is shorter (in terms of path cost) than the current path by at least a given threshold. MRHOF may be used with any additive metric listed in [I-D.ietf-roll-routing-metrics] as long the routing objective is to minimize the given routing metric.

3.1. Computing the Path cost

Nodes compute the path cost for each candidate neighbor reachable on an interface. The Path cost represents the cost of the path, in terms of the selected metric, from a node to the root of the DODAG through the neighbor.

Root nodes (Grounded or Floating) set the variable cur_min_path_cost to MIN_PATH_COST.

A non-root node computes the path cost for a path to the root through each candidate neighbor by adding these two components:

1. If the selected metric is a link metric, the selected metric for the link to a candidate neighbor. If the selected metric is a node metric, the selected metric for the node.

2. The value of the selected metric in the metric container in the DIO sent by that neighbor.

A node SHOULD compute the path cost for the path through each candidate neighbor reachable through an interface. If a node cannot compute the path cost for the path through a candidate neighbor, the node MUST NOT select the candidate neighbor as its preferred parent, with one exception. If the node does not have metrics to compute the

path cost through any of the candidate neighbors, it MUST join one of the candidate neighbors as a leaf node.

If the selected metric is a link metric and the metric of the link to a neighbor is not available, the path cost for the path through that neighbor SHOULD be set to MAX_PATH_COST. This cost value will prevent this path from being considered for path selection.

If the selected metric is a node metric, and the metric is not available, the path cost through all the neighbors SHOULD be set to MAX_PATH_COST.

The path cost corresponding to a neighbor SHOULD be re-computed each time:

1. The selected metric of the link to the candidate neighbor is updated.

2. If the selected metric is a node metric and the metric is updated.

3. A node receives a new metric advertisement from the candidate neighbor.

This computation MAY also be performed periodically. Too much delay in updating the path cost after the metric is updated or a new metric advertisement is received can lead to stale Rank or parent set.

3.2. Parent Selection

After computing the path cost for all the candidate neighbors reachable through an interface for the current DODAG iteration, a node selects the preferred parent. This process is called parent selection. Parent Selection SHOULD be performed each time:

1. The path cost for an existing candidate neighbor, including the preferred parent, changes. This condition can be checked immediately after the path cost is computed.

2. A new candidate neighbor is inserted into the neighbor table.

The parent selection MAY be deferred until a later time. Deferring the parent selection can delay the use of better paths available in the network.

A node MUST select a candidate neighbor as its preferred parent if the path cost corresponding to that neighbor is smaller than the path cost corresponding to the rest of the neighbors, except as indicated

below:

1. If the smallest path cost for paths through the candidate neighbors is smaller than cur_min_path_cost by less than PARENT_SWITCH_THRESHOLD, the node MAY continue to use the current preferred parent.

2. If there are multiple paths with the smallest path cost and the smallest path cost is smaller than cur_min_path_cost by at least PARENT_SWITCH_THRESHOLD, a node MAY use a different objective function to select the preferred parent among the candidate neighbors on the path with the minimum cost.

3. A node MAY declare itself as a Floating root, and hence no preferred parent, depending on the configuration.

4. If the selected metric for a link is greater than MAX_LINK_METRIC, the node SHOULD exclude that link from consideration for parent selection.

5. If cur_min_path_cost is greater than MAX_PATH_COST, the node MAY declare itself as a Floating root.

6. If the configuration disallows a node to be a Floating root and no neighbors are discovered, the node does not have a preferred parent, and MUST set cur_min_path_cost to MAX_PATH_COST.

Except in the cases above, the candidate neighbor on the path with the smallest path cost is the preferred parent. A node MAY include a total of PARENT_SET_SIZE candidate neighbors in the parent set. The cost of path through the nodes in the parent set is smaller than or equal to the cost of the paths through any of the nodes that are not in the parent set. If the cost of the path through the preferred parent and the worst parent is too large, a node MAY keep a smaller parent set.

3.3. Computing Rank

The DAG roots set their rank to MIN_PATH_COST for the selected metric.

Once a non-root node selects its parent set, it can use the following table to covert the the path cost of the worst parent (written as Cost in the table) to its rank:

Node/link Metric	Rank
Node Energy	255 - Cost
Hop-Count	Cost
Latency	Cost/65536
Link Quality Level	Cost
ETX	Cost

Table 1: Conversion of metric to rank.

Nodes MUST support at least one of the above metrics. Nodes SHOULD support the ETX metric.

Node rank is undefined for these node/link metrics: Node state and attributes, throughput, and link color. If the rank is undefined, the node MUST join one of the neighbors as a leaf node.

3.4. Advertising the Path Cost

Once the preferred parent is selected, the node sets its cur_min_path_cost variable to the path cost corresponding to the preferred parent. Thus, cur_min_path_cost is the cost of the minimum cost path from the node to the root. The value of the cur_min_path_cost is carried in the metric container corresponding to the selected metric when DIO messages are sent.

3.5. Working Without Metric Containers

In the absence of metric container, MRHOF uses ETX as its metric. It locally computes the ETX of links to its neighbors and adds this value to their advertised Rank to compute the associated Rank of routes. Once parent selection and rank computation is performed using the ETX metric, the node advertises a Rank equal to the ETX cost and SHOULD NOT include a metric container in its DIO messages.

4. Using MRHOF for Metric Maximization

MRHOF cannot be directly used for parent selection using metrics which require finding paths with maximum value of the selected metric, such as path reliability. It is possible to convert such a metric maximization problem to a metric minimization problem and use MRHOF provided:

There is a fixed and well-known maximum metric value corresponding to the best path. This is the path cost for the DAG root. Example, the best link reliability has a value of 1.

Metrics are all positive. Example, link reliability is always positive.

For metrics meeting the above conditions, the problem of maximizing the metric value is equivalent to minimizing the negative of the metric value. MRHOF is not required to work with these metrics.

5. Settings of RPL parameters

The MinHopRankIncrease parameter MUST be set to 1.

6. MRHOF Variables and Parameters

MRHOF uses the following variable:

cur_min_path_cost: The cost of the path from a node through its preferred parent to the root computed at the last parent selection.

MRHOF uses the following parameters:

MAX_LINK_METRIC: Maximum allowed value for the selected link metric for each link on the path.

MAX_PATH_COST: Maximum allowed value for the path metric of a selected path.

MIN_PATH_COST: The minimum allowed value for the path metric of the selected path.

PARENT_SWITCH_THRESHOLD: The difference between metric of the path through the preferred parent and the minimum-metric path in order to trigger the selection of a new preferred parent.

PARENT_SET_SIZE: The number of candidate parents, including the preferred parent, in the parent set.

The parameter values are assigned depending on the selected metric. The best values for these parameters should be experimentally determined. The working group has long experience routing with the ETX metric. Based on those experiences, these ETX parameters are known to work in many settings:

MAX_LINK_METRIC: 10. Disallow links with greater than 10 expected transmission count on the selected path.

MAX_PATH_COST: 100. Disallow paths with greater than 100 expected transmission count.

MIN_PATH_COST: 0. At root, the expected transmission count is 0.

PARENT_SWITCH_THRESHOLD: 1.5. Switch to a new path only if it is expected to require at least 1.5 fewer transmission than the current path.

PARENT_SET_SIZE: 3. If the preferred parent is not available, two candidate parents are still available without triggering a new round of route discovery.

7. Acknowledgements

Thanks to Antonio Grilo, Nicolas Tsiftes, Matteo Paris, JP Vasseur, and Phoebus Chen for their comments.

8. IANA Considerations

This specification requires an allocated OCP. A value of 1 is requested.

9. Security Considerations

Security considerations to be developed in accordance to the output of the WG.

10. References

10.1. Normative References

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

10.2. Informative References

[I-D.ietf-roll-routing-metrics]
Vasseur, J. and D. Networks, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-01 (work in progress),

October 2009.

[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-05 (work in progress), December 2009.

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-01 (work in progress), May 2009.

Authors' Addresses

Omprakash Gnawali
Stanford University
S255 Clark Center, 318 Campus Drive
Stanford, CA 94305
USA

Phone: +1 650 725 6086
Email: gnawali@cs.stanford.edu

Philip Levis
Stanford University
358 Gates Hall, Stanford University
Stanford, CA 94305
USA

Email: pal@cs.stanford.edu

ROLL
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

P. Thubert, Ed.
Cisco Systems

March 14, 2011

RPL Objective Function 0
draft-ietf-roll-of0-07

Abstract

The Routing Protocol for Low Power and Lossy Networks (RPL) defines a generic Distance Vector protocol for Low Power and Lossy Networks (LLNs). RPL is instantiated to honor a particular routing objective/ constraint by the adding a specific Objective Function (OF) that is designed to solve that problem. This specification defines a basic OF, OF0, that uses only the abstract properties exposed in RPL messages with no metric container.

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].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

The IETF ROLL Working Group has defined application-specific routing requirements for a Low Power and Lossy Network (LLN) routing protocol, specified in [I-D.ietf-roll-building-routing-reqs], [I-D.ietf-roll-home-routing-reqs], [RFC5673], and [RFC5548].

Considering the wide variety of use cases, link types and metrics, the Routing Protocol for Low Power and Lossy Networks [I-D.ietf-roll-rpl] was designed as a generic core that is agnostic to metrics and instantiated using Objective Functions.

RPL forms Destination Oriented Directed Acyclic Graphs (DODAGs) within instances of the protocol, each instance being set up to honor a particular routing objective/constraint of a given deployment. This instantiation is achieved by plugging into the RPL core a specific Objective Function (OF) that is designed to solve that problem to be addressed by that instance.

An Objective Function selects the DODAG version that a device joins, and a number of neighbor routers within that version as parents and siblings. The OF is also responsible for computing the Rank of the device, that abstracts a relative position within the DODAG and is used by the RPL core to enable a degree of loop avoidance and verify forward progression towards a destination, as specified in [I-D.ietf-roll-rpl].

Since there is no default OF or metric container in the RPL main specification, it might happen that, unless given two implementations follow a same guidance for a specific problem or environment, those implementations will not support a common OF with which they could interoperate. This specification fills the need for an Objective Function that can be used as a common denominator between all generic implementations. This is why OF0 is very abstract as to how the link properties are transformed into a Rank, giving only normalized values for what a normal link and what the acceptable range is for a step of Rank are, as opposed to formulating the details of the step of Rank computation.

Indeed, it is the general design in RPL that the metrics are passed from parent to children in a specific container and that the OF will derive the Rank from the natural metric. The separation of Rank and metrics avoids a loss of information as the various metrics are propagated down the DAG. This specification can be used when the link properties that are considered are such that they can be turned in a scalar step of Rank in a reversible fashion and the resulting step of rank is additive over multiple hops.

The Objective Function 0 (OF0) corresponds to the Objective Code Point 0 (OCP0). OF0 does not leverage metric containers such as described in the metrics draft [I-D.ietf-roll-routing-metrics]. OF0 does not require information in the RPL messages but the abstract information from the DIO base container, such as Rank and an administrative preference, that is transported in DIOs as DODAGPreference in [I-D.ietf-roll-rpl]. The Rank of a node is obtained by adding a step of Rank multiplied by a Rank Factor to the Rank of a selected preferred parent. OF0 uses a MinHopRankIncrease of 0x100 so that Rank value can be stored in one octet. This allows up to at least 28 hops even when each hop has the worst step of Rank of 9 and a Rank Factor of 1. How the link properties are transformed into a step of Rank for a given hop depends on the link type and on the implementation. It can be as simple as an administrative cost, but might also derive from a statistical metric with some hysteresis.

2. Terminology

The terminology used in this document is consistent with and incorporates that described in 'Terminology in Low power And Lossy Networks' [I-D.ietf-roll-terminology] and [I-D.ietf-roll-rpl].

3. Goal

The Goal of the OF0 is to join a DODAG version that offers connectivity to a specific set of nodes or to a larger routing infrastructure. For the purpose of OF0, Grounded thus means that the root provides such connectivity. How that connectivity is asserted and maintained is out of scope.

Objective Function 0 is designed to find the nearest Grounded root. In the absence of a Grounded root, LLN inner connectivity is still desirable and floating DAGs will form, rooted at the nodes with the highest administrative preference.

The metric used in OF0 can be an administratively defined scalar cost that is trivially added up along a path to compute the RPL Rank, as defined in [I-D.ietf-roll-rpl]. Depending on how the step of Rank is computed by an implementation, the Rank of a node might be analogous to a weighted hop count of the path to the root. Using a metric that in essence is similar to hop count implies that the quality of the connectivity should be asserted so that only neighbors with a good enough connectivity are presented to the OF. How that connectivity is asserted and maintained is not covered by this specification.

In wireless networks, Hop Count will tend to favor paths with long

distance links and non optimal connectivity properties. In some situations, this might end up partitioning the network. As a result, the link selection must be very conservative, and the available link set is thus constrained. For those reasons, though it can be used on wired links and wired link emulations such as WIFI infrastructure mode, a metric derived from hop count is generally not recommended for wireless networks. Instead, careful thinking should be applied to determine how the step of Rank is computed from the link properties. For instance, the Minimum Rank Objective Function with Hysteresis [I-D.ietf-roll-minrank-hysteresis-of] provides guidance on how hysteresis can be used to maintain a certain stability of the resulting Rank.

The default step of Rank is DEFAULT_RANK_INCREMENT for each hop. An implementation MAY allow a step between MINIMUM_RANK_INCREMENT and MAXIMUM_RANK_INCREMENT to reflect a large variation of link quality by units of MINIMUM_RANK_INCREMENT. In other words, the least significant octet in the Rank is not used.

A node MAY stretch its step of Rank by up to MAXIMUM_RANK_STRETCH in order to enable the selection of a sibling when only one parent is available. For instance, say that a node computes a step of Rank of 4 units of MINIMUM_RANK_INCREMENT from a preferred parent with a Rank of 6 units resulting in a Rank of 10 units for this node. Say that with that Rank of 10 units, this node would end up with only one parent and no sibling, though there is a neighbor with a Rank of 12 units. In that case, the node is entitled to stretch its step of Rank by a value of 2 units, thus using a step of Rank of 6 units so as to reach a Rank of 12 units and find a sibling. But the node is not entitled to use a step of Rank larger than 6 units since that would be a greedy behavior that would deprive the neighbor of this node of a successor. Also, if the neighbor had exposed a Rank of 16 units, the stretch of Rank from 10 to 16 units would have exceeded MAXIMUM_RANK_STRETCH of 5 units and thus the neighbor would not have been selectable even as a sibling.

The gap between MINIMUM_RANK_INCREMENT and MAXIMUM_RANK_STRETCH may not be sufficient in every case to strongly distinguish links of different types or categories in order to favor, say, powered over battery-operated or wired over wireless, within a same DAG. An implementation SHOULD allow a configurable factor called Rank Factor and to apply the factor on all links and peers. An implementation MAY recognizes sub-categories of peers and links, such as different MAC types, in which case it SHOULD be able to configure a more specific Rank Factor to those categories. The Rank Factor SHOULD be set between MINIMUM_RANK_FACTOR and MAXIMUM_RANK_FACTOR. Once a step of Rank is computed along the rules specified in this document, the result of the computation is multipled by the Rank Factor and the

result is what gets added to the Rank of preferred parent in order to obtain the Rank of this node.

Optionally, the administrative preference of a root MAY be configured to supercede the goal to reach Grounded root. In that case, nodes will associate to the root with the highest preference available, regardless of whether that root is Grounded or not. Compared to a deployment with a multitude of Grounded roots that would result in a same multitude of DODAGs, such a configuration may result in possibly less but larger DODAGs, as many as roots configured with the highest priority in the reachable vincinity.

OF0 selects a preferred parent and a backup next_hop if one is available. The backup next_hop might be but does not have to be a parent or a sibling. All the upward traffic is normally routed via the preferred parent. When the link conditions do not let an upward packet through the preferred parent, the packet is passed to the backup next_hop.

4. Selection of the Preferred Parent

As it scans all the candidate neighbors, OF0 keeps the parent that is the best for the following criteria (in order):

1. [I-D.ietf-roll-rpl] spells out the generic rules for a node to reparent and in particular the boundaries to augment its Rank within a DODAG version. A candidate that would not satisfy those rules MUST NOT be considered.

2. An implementation should validate a router prior to selecting it as preferred. This validation process is implementation and link type dependent, and is out of scope. A router that has been validated is preferrable.

3. When multiple interfaces are available, a policy might be locally configured to prioritize them and that policy applies first; that is a router on a higher order interface is preferable.

4. In the absence of a Grounded DODAG version, the router with a higher administrative preference SHOULD be preferred. Optionally, this selection applies regardless of whether the DODAG is Grounded or not.

5. A router that offers connectivity to a grounded DODAG version SHOULD be preferred over one that does not.

6. When comparing 2 routers that belong to the same DODAG, a router that offers connectivity to the freshest sequence SHOULD be preferred.

7. When computing a resulting Rank for this node from a parent Rank and a Step of Rank from that parent, the parent that causes the lesser resulting Rank SHOULD be preferred.

8. A DODAG version for which there is an alternate parent SHOULD be preferred. This check is optional. It is performed by computing the backup next_hop while assuming that the router that is currently examined is finally selected as preferred parent.

9. The DODAG version that was in use already SHOULD be preferred.

10. The preferred parent that was in use already SHOULD be preferred.

11. A router that has announced a DIO message more recently SHOULD be preferred.

5. Selection of the Backup next_hop

- When multiple interfaces are available, a router on a higher order interface is preferable.

- The backup next_hop MUST NOT be the preferred parent.

- The backup next_hop MUST be either in the same DODAG version as the preferred parent or in an subsequent version. Note that if the backup next_hop is not from the current version then it can not be used as parent.

- A Router with a Rank that is higher than the Rank computed for this node out of the preferred parent SHOULD NOT be selected as parent, to avoid greedy behaviors. It MAY still be selected as sibling if no better Back-up next hop is found.

- A router with a lesser Rank SHOULD be preferred.

- A router that has been validated as usable by an implementation dependant validation process SHOULD be preferred.

- The backup next_hop that was in use already SHOULD be preferred.

6. Abstract Interface with RPL core

Objective Function 0 interacts with the core RPL in the following ways:

Processing DIO: This core RPL triggers the OF when a new DIO was received. OF0 analyses the information in the DIO and may select the source as a parent or sibling.

Providing DAG information The OF0 support can be required to provide the DAG information for a given instance to the RPL core. This includes the material that is contained in a DIO base header.

Providing a Parent List The OF0 support can be required to provide the list of the parents for a given instance to the RPL core. This includes the material that is contained in the transit option for that parent.

Trigger The OF0 support may trigger the RPL core to inform it that a change occurred. This can be used to indicate whether the change requires a new DIO to be fired or whether trickle timers need to be reset.

7. OF0 Constants and Variables

OF0 uses the following constants:

MinHopRankIncrease: 256

DEFAULT_RANK_INCREMENT: 3 * MinHopRankIncrease

MINIMUM_RANK_INCREMENT: 1 * MinHopRankIncrease

MAXIMUM_RANK_INCREMENT: 9 * MinHopRankIncrease

MAXIMUM_RANK_STRETCH: 5 * MinHopRankIncrease

DEFAULT_RANK_FACTOR: 1

MINIMUM_RANK_FACTOR: 1

MAXIMUM_RANK_FACTOR: 4

8. IANA Considerations

IThis specification requires the assignment of an OCP for OF0. The value of 0 is suggested.

9. Security Considerations

Security Considerations for OCP/OF are to be developed in accordance with recommendations laid out in, for example, [I-D.tsao-roll-security-framework].

10. Acknowledgements

Most specific thanks to Philip Levis for his help in finalizing this document, in particular WRT wireless links, to Tim Winter, JP Vasseur, Julien Abeille, Mathilde Durvy, Teco Boot, Navneet Agarwal and Henning Rogge for in-depth review and first hand implementer’s feedback.

11. References

11.1. Normative References

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

11.2. Informative References

[I-D.ietf-roll-building-routing-reqs]
Martocci, J., Riou, N., Mil, P., and W. Vermeylen, "Building Automation Routing Requirements in Low Power and Lossy Networks", draft-ietf-roll-building-routing-reqs-07 (work in progress), September 2009.

[I-D.ietf-roll-home-routing-reqs]
Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low Power and Lossy Networks", draft-ietf-roll-home-routing-reqs-08 (work in progress), September 2009.

[I-D.ietf-roll-minrank-hysteresis-of]
Gnawali, O. and P. Levis, "The Minimum Rank Objective Function with Hysteresis", draft-ietf-roll-minrank-hysteresis-of-01 (work in progress), February 2011.

[I-D.ietf-roll-routing-metrics]
Vasseur, J., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-19 (work in progress), March 2011.

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-18 (work in progress), February 2011.

[I-D.ietf-roll-terminology]
Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-04 (work in progress), September 2010.

[I-D.tsao-roll-security-framework]
Tsao, T., Alexander, R., Daza, V., and A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", draft-tsao-roll-security-framework-02 (work in progress), March 2010.

[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.

Author's Address

Pascal Thubert (editor)
Cisco Systems
Village d'Entreprises Green Side
400, Avenue de Roumanille
Batiment T3
Biot - Sophia Antipolis 06410
FRANCE

Phone: +33 497 23 26 34
Email: pthubert@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: August 11, 2011

M. Goyal, Ed.
University of Wisconsin Milwaukee
E. Baccelli
INRIA
A. Brandt
Sigma Designs
R. Cragie
Gridmerge Ltd
J. Martocci
Johnson Controls
C. Perkins
Tellabs Inc

February 7, 2011

Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks
draft-ietf-roll-p2p-rpl-02

Abstract

Point to point (P2P) communication between arbitrary IPv6 routers and hosts in a Low power and Lossy Network (LLN) is a key requirement for many applications. RPL, the IPv6 Routing Protocol for LLNs, constrains the LLN topology to a Directed Acyclic Graph (DAG) and requires the P2P routing to take place along the DAG links. Such P2P routes may be suboptimal and may lead to traffic congestion near the DAG root. This document specifies a P2P route discovery mechanism, complementary to the RPL base functionality. This mechanism allows an RPL-aware IPv6 router or host to discover and establish, on demand, one or more routes to another RPL-aware IPv6 router or host in the LLN such that the discovered routes meet a specified cost criteria.

Status of this Memo

Table of Contents

1. Introduction

RPL [I-D.ietf-roll-rpl] provides multipoint-to-point (MP2P) routes from nodes in a Low power and Lossy Network (LLN) to a sink node by organizing the nodes along a Directed Acyclic Graph (DAG) rooted at the sink. The nodes determine their position in the DAG so as to optimize their routing cost on the path towards the DAG root. A node advertises its position (the "rank") in the DAG by originating a DODAG Information Object (DIO) message. The DIO message is sent via link-local multicast and also includes information such as the DAG root's identity, routing metrics/constraints [I-D.ietf-roll-routing-metrics] and the objective function (OF) in use. When a node joins the DAG, it determines its own rank in the DAG based on that advertised by its neighbors and originates its own DIO message.

RPL enables point-to-multipoint (P2MP) routing from a node to its descendants in the DAG by allowing a node to send a Destination Advertisement Object (DAO) upwards along the DAG. The DAO carries potentially aggregated information regarding the descendants (and other local prefixes) reachable through the node originating this DAO.

RPL also provides mechanisms for point-to-point (P2P) routing between any two nodes in the DAG. If the destination is within the source's radio range, the source may directly send packets to the destination. Otherwise, a packet's path from the source to the destination depends on the storing/non-storing operation mode of the DAG. In non-storing mode operation, only the DAG root maintains downward routing information and hence a packet travels all the way to the DAG root, which then sends it towards its destination using a source route. In storing mode operation, if the destination is a DAG descendant and the source maintains "downwards" hop-by-hop routing state about this descendant, it can forward the packet to a descendant router closer to the destination. Otherwise, the source sends the packet to a DAG parent, which then applies the same set of rules to forward the packet further. Thus, a packet travels up the DAG until it reaches a node that knows of the downwards route to the destination and then it travels down the DAG towards its destination. A node may or may not maintain routing state about a descendant depending on whether its immediate children send it such information in their DAOs. Thus, in the best case with storing mode operation, the "upwards" segment of the P2P route between a source and a destination ends at the first common ancestor of the source and the destination. In the worst case, the "upwards" segment would extend all the way to the DAG root. In both storing and non-storing mode operations, if the destination did not originate a DAO, the packet will travel all the way to the DAG's root, where it will be dropped.

The P2P routing functionality available in RPL may be inadequate for applications in the home and commercial building domains for the following reasons [I-D.brandt-roll-rpl-applicability-home-building] [RFC5826][RFC5867]:

- o The need to maintain routes "proactively", i.e., every possible destination in the DAG must originate a DAO.

- o Depending on the network topology and OF/metrics in use, the constraint to route only along a DAG may cause significantly suboptimal P2P routes and severe traffic congestion near the DAG root.

Thus, there is a need for a mechanism that provides source-initiated discovery of P2P routes that are not along an existing DAG. This document describes such a mechanism, complementary to the basic RPL functionality.

The specified mechanism is based on a reactive on-demand approach, which enables a node to discover one or more routes in either direction between itself and another node in the LLN without any restrictions regarding the existing DAG-membership of the links that such routes may use. The discovered routes may be source routes or hop-by-hop routes. The discovered routes may not be the best available but are guaranteed to satisfy the desired constraints in terms of the routing metrics and are thus considered "good enough" from the application’s perspective.

A complementary functionality, necessary to help decide whether to initiate a route discovery, is a mechanism to measure the end-to-end cost of an existing route. Section 4 provides further details on how such functionality, described in [I-D.goyal-roll-p2p-measurement], can be used to determine the metric constraints for use in the route discovery mechanism described in this document.

2. The Use Cases

The mechanisms described in this document are intended to be employed as complementary to RPL in specific scenarios that need point-to-point (P2P) routes between arbitrary routers.

One use case, common in a home environment, involves a remote control (or a motion sensor) that suddenly needs to communicate with a lamp module, whose network address is a-priori known. In this case, the source of data (the remote control or the motion sensor) must be able to discover a route to the destination (the lamp module) "on demand".

Another use case, common in a large commercial building environment, involves a large LLN deployment where P2P communication along a particular DAG among hundreds (or thousands) of routers creates severe traffic congestion near that DAG's root, and thus routes across this DAG are desirable.

The use cases also include scenarios where energy or latency constraints are not satisfied by the P2P routes along a DAG because they involve traversing many more intermediate routers than necessary to reach the destination.

3. Terminology

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

Additionally, this document uses terminology from [I-D.ietf-roll-terminology] and [I-D.ietf-roll-rpl]. Specifically, the term node refers to an RPL router or an RPL host as defined in [I-D.ietf-roll-rpl]. This document introduces the following terms:

Origin : The RPL node initiating the route discovery. The origin acts as one end point of the routes to be discovered.

Target : The RPL node at the other end point of the routes to be discovered.

Intermediate Router: An RPL router that is neither the origin nor the target.

Forward Route: A route from the origin to the target.

Backward Route: A route from the target to the origin.

Bidirectional Route: A route that can carry traffic in both directions.

Source Route: A complete and ordered list of routers that can be used by a packet to travel from a source node to a destination node. Such source routes can be carried by a packet in a Type 4 Routing Header [I-D.ietf-6man-rpl-routing-header].

Hop-by-hop Route: The route characterized by each router on the route using its routing table to determine the next hop on the route.

Propagation Constraints: The constraints on the routing metrics [I-D.ietf-roll-routing-metrics] that MUST be satisfied before an intermediate router or the target will process the Route Discovery Option (defined in this document) contained inside a DODAG Information Object (DIO).

Route Constraints: Additional constraints on the routing metrics [I-D.ietf-roll-routing-metrics] that the target MUST enforce on the received DIOs.

4. Applicability

The route discovery mechanism, described in this document, may be invoked by an origin when no route exists between itself and the target or when the existing routes do not satisfy the desired performance requirements. The mechanism is designed to discover one or more "good enough" routes in either direction between an origin and a target. In some application contexts, the metric constraints that the discovered routes must satisfy are intrinsically known or can be specified by the application. For example, an origin that expects a target to be less than 5 hops away may use "hop-count < 5" as the propagation or route constraint. In other application contexts, the origin may need to measure the cost of an existing route to the target to determine the propagation/route constraints. For example, an origin that measures the total ETX of its along-DAG route to the target to be 20 may use "ETX < x*20", where x is a fraction that the origin decides, as the propagation/route constraint. The functionality required to measure the cost of an existing route between the origin and the target is described in [I-D.goyal-roll-p2p-measurement]. In case, there is no existing route between the origin and target or the cost measurement for the existing route fails, the origin will have to guess the propagation/ route constraints used in the initial route discovery. Once, the initial route discovery succeeds or fails, the origin will have a better estimate for the constraints to be used in the subsequent route discovery.

This document describes an on-demand discovery mechanism for P2P routes that is complementary to the proactive routes offered by RPL base functionality. The mechanism described in this document may result in discovery of better P2P routes than the ones available along a DAG designed to optimize routing cost to the DAG’s root. The improvement in route quality depends on a number of factors including the network topology, the routing metrics in use and the prevalent conditions in the network. A network designer may take in consideration both the benefits (potentially better routes; no need to maintain routes proactively) and costs (control messages generated

during the route discovery process) when using this mechanism.

5. Functional Overview

This section contains a high level description of the route discovery mechanism proposed in this document.

The route discovery begins with the origin generating a "Discovery" message. The origin indicates in the message:

- The target;
- The relevant routing metrics;
- The constraints on how far the Discovery message may travel (henceforth called the propagation constraints);
- Additional constraints that the target must enforce (henceforth called the route constraints);
- The direction (forward: from the origin to the target; backward: from the target to the origin; or bidirectional) of the route being discovered;
- The desired number of routes (in case forward/bidirectional routes are being discovered);
- Whether the route is a source route or a hop-by-hop one.

The Discovery message propagates via IPv6 link-local multicast with a receiving router discarding the message if it does not satisfy the propagation constraints or if the hop-by-hop routes are desired and the router cannot store the state for such a route. As a copy of the Discovery message travels towards the target, it accumulates the relevant routing metric values as well as the route it takes. When the target receives a Discovery message, it applies both the propagation constraints and the route constraints on the routing metrics inside the Discovery message. Thus, the discovered routes satisfy both the propagation constraints as well as the route constraints, although the propagation of Discovery messages is guided by propagation constraints alone. Using only a subset of the constraints as propagation constraints simplifies the operation of intermediate routers, an important consideration in many LLN application domains [RFC5826][RFC5867].

The route discovery process may result in the discovery of several routes. This document does not specify how the target selects routes

among the ones discovered. Example selection methods include selecting routes as they are discovered or selecting the best routes discovered over a certain time period.

If the origin had requested the discovery of backward source-routes, the target caches one or more discovered source-routes. Additionally, the target sends one or more "Discovery Reply" messages to the origin to acknowledge the discovery of these routes.

If the origin had requested the discovery of "n" forward source-routes, the target sends "n" discovered source-routes it selects to the origin in one or more Discovery Reply messages.

If the origin had requested the discovery of "n" bidirectional source-routes, the target caches "n" discovered source-routes it selects and also sends these routes to the origin in one or more Discovery Reply messages.

If the origin had requested the discovery of "n" forward/backward/ bidirectional hop-by-hop routes, the target sends out a Discovery Reply message to the origin for each one of the "n" discovered routes it selects. The Discovery Reply message travels towards the origin along the discovered route. As this message travels towards the origin, it establishes appropriate forward/backward routing state in the routers on the path.

6. Propagation of Discovery Messages

RPL uses DIO message propagation to build a DAG. The DIO message travels via IPv6 link-local multicast. Each node joining the DAG determines a rank for itself and ignores the subsequent DIO messages received from lower (higher in numerical value) ranked neighbors. Thus, the DIO messages propagate outward from the DAG root rather than return inward towards the DAG root. The DIO message generation at a node is further controlled by a trickle timer that allows a node to avoid generating unnecessary messages [I-D.ietf-roll-trickle]. The link-local multicast based propagation, trickle-controlled generation and the rank-based poisoning of messages traveling in the wrong direction (towards the DAG root) provide powerful incentives to use the DIO message as the Discovery message and propagate the DIO/ Discovery message by creating a "temporary" DAG. Such an approach also allows reuse of the routing metrics, objective function and packet forwarding framework developed for RPL. The routing metrics used for the creation of this temporary DAG SHOULD be same as (or be a subset of) the routing metrics being used for route discovery. Similarly, the objective function, used for rank calculation in the temporary DAG, SHOULD be same as the objective function that

determines the aggregated cost of a route when limited to the routing metrics being used for temporary DAG creation.

The propagation constraints limit the spread of the temporary DAG. The temporary DAG restricts the network topology within which the route discovery takes place. The routes accumulated by the DIOs lie within this restricted topology and implicitly satisfy the propagation constraints. As the target receives a DIO, it additionally applies the route constraints on the accumulated route. Thus, for successful route discovery, the propagation constraints and the route constraints MUST be compatible. The division of the overall constraints in the two categories is an implementation specific decision. If desired, an implementation MAY consider all the constraints as propagation constraints and keep the set of route constraints empty.

6.1. The Route Discovery Option

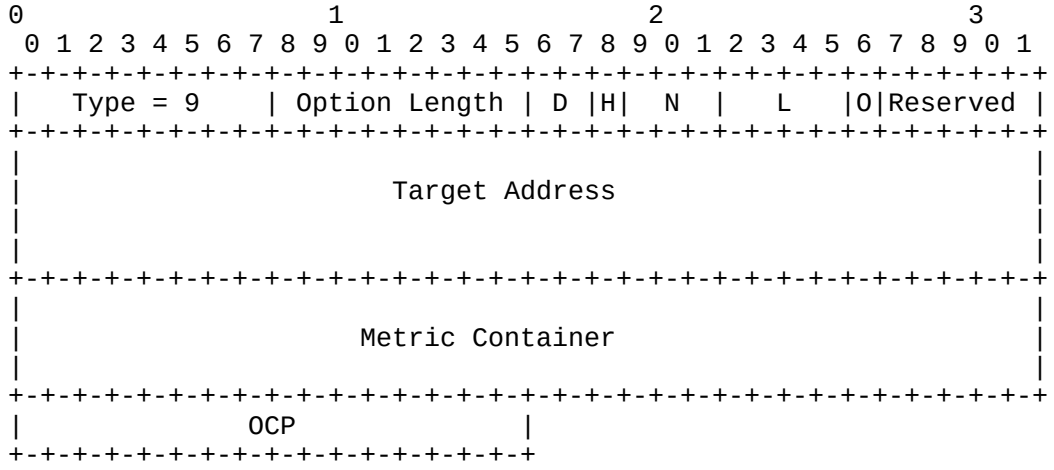


Figure 1: Format of the Route Discovery Option

In order to be used as a Discovery message, a DIO MUST carry a "Route Discovery" option illustrated in Figure 1. A DIO MUST NOT carry more than one Route Discovery options. A router MUST ignore the second and subsequent Route Discovery options carried by a DIO. A Route Discovery option consists of the following fields:

- Option Type = 0x09 (to be confirmed by IANA).
- Option Length = The length of Route Discovery option including any Metric Container and OCP fields.

- D: A 2-bit field that indicates the direction of the desired routes:

  * D = 0x00: Forward;

  * D = 0x01: Backward;

  * D = 0x02: Bidirectional.

  The D field also specifies the direction in which the link-level metrics being used for route discovery should be measured.

- H: This flag, when set, indicates if hop-by-hop routes are desired. The flag is cleared if source routes are desired.

- N: A 3-bit unsigned integer indicating the number of routes desired. Used when forward or bidirectional routes are being discovered.

- L: A 4-bit field containing an exponent of 2, such that 2 raised to the power L specifies, in units of seconds, the minimum "Life Time" of the temporary DAG, i.e., the minimum duration a router joining the temporary DAG must maintain its membership in the DAG.

- O: This flag, when set, indicates that an OCP field is present in the Route Discovery option.

- Target Address: The IPv6 address of the target.

- Metric Container: Contains the route constraints that the target MUST apply. Any metric objects contained in this metric container MUST be ignored.

- OCP: 16 bit unsigned integer. An optional field, present only if the O flag is set, This field indicates the objective function that MAY be used by the target to compare two discovered routes.

6.2. Setting a DIO Carrying a Route Discovery Option

A DIO message that carries a Route Discovery option MUST set the Base Object, described in [I-D.ietf-roll-rpl], in the following manner:

- RPLInstanceID: RPLInstanceID MUST be a local value as described in Section 5.1 of [I-D.ietf-roll-rpl]. The origin MUST ensure that different RPLInstanceID values are used in two or more concurrent route discoveries it initiates.

- Grounded (G) Flag: MUST be cleared since the objective of DAG formation is propagation of Route Discovery option. This DAG is temporary in nature and is not used for routing purpose.

- Destination Advertisement Supported (A) Flag: MUST be cleared for same reasons as described above.

- Destination Advertisement Trigger (T) Flag: MUST be cleared.

- Mode of Operation (MOP): This document suggests a new value (0x04) for this field (to be confirmed by IANA).

- DODAGPreference (Prf): TBD

- Destination Advertisement Trigger Sequence Number (DTSN): TBD

- DODAGID: IPv6 address of the origin.

The other fields in the Base Object are set as per the rules described in [I-D.ietf-roll-rpl].

The DODAG Configuration option, carried in the DIO message, specifies the parameters for the trickle timer operation that governs the generation of DIO messages by routers joining the temporary DAG. The future versions of this document will specify the default values to be used for these parameters. The other fields defined in the DODAG Configuration option are set as follows:

- The MaxRankIncrease field MUST be set to 0 to disable local repair of the temporary DAG.

- This document RECOMMENDS a value 1 for the MinHopRankInc field.

- Objective Code Point (OCP): The OCP to be used for temporary DAG formation. The objective function used for temporary DAG formation SHOULD be compatible with the objective function to determine the aggregated cost of a discovered route.

A DIO, that contains a Route Discovery option, MUST specify the propagation constraints in one or more Metric Container options placed outside the Route Discovery option. As mentioned before, the route constraints are listed in the Metric Container option placed inside the Route Discovery option. The routing metrics being used for temporary DAG formation SHOULD be same as or a subset of the routing metrics being used for route discovery. These routing metrics MUST be placed in the Metric Container options placed outside the Route Discovery option. Any link-level metrics being used for route discovery MUST be measured in the direction indicated by the D

field in Route Discovery option. Any metric object contained inside the Metric Container inside the Route Discovery option MUST be ignored.

A DIO, carrying a Route Discovery option, MUST NOT carry any Route Information or Prefix Information options described in [I-D.ietf-roll-rpl].

6.3. Joining a Temporary DAG

When a node joins a temporary DAG advertized by a DIO carrying the Route Discovery option, it MUST maintain its membership in the DAG for the Minimum Life Time duration listed in the Route Discovery option. Maintaining membership in the DAG implies remembering:

- The RPLInstanceID, the DODAGID and the DODAGVersionNumber for the temporary DAG;
- The node's rank in the temporary DAG as well as the address of at least one DAG parent;
- The propagation and the route constraints being used;
- In case of intermediate routers, the values for the routing metrics, along with the associated source route from the origin untill this node (carried in a Record Route IPv6 Extension Header proposed in [I-D.thubert-6man-reverse-routing-header]), contained in the best DIO (in terms of the routing metrics and potentially using the OCP specified in the DODAG Configuration option) received so far.

Although the main purpose of a temporary DAG's existence is to facilitate the propagation of the Route Discovery option, the temporary DAG MAY also be used for the Discovery Reply Object (defined in Section 7.1 to travel from the target to the origin. Hence, a node in a temporary DAG SHOULD also remember the address of at least one DAG parent that provides the best known path back to the origin. A node SHOULD delete information about a temporary DAG once the duration of its membership in the DAG has exceeded the DAG's minimum life time.

6.4. Processing a DIO Carrying a Route Discovery Option

The rules for DIO processing and transmission, described in Section 7 of RPL [I-D.ietf-roll-rpl], apply to DIOs carrying a Route Discovery option as well except as modified in this document.

The following rules for processing a DIO carrying a Route Discovery

Option apply to both intermediate routers and the target.

A node MUST discard a DIO with no further processing if:

- The DIO contains two or more Route Discovery options;
- The node can not evaluate one or more of the propagation constraints listed in a Metric Container inside the DIO.

A node MUST discard a DIO with no further processing if any of the following conditions are found to be true while processing a Route Discovery option contained in that DIO:

- The H field is set, i.e., hop-by-hop routes are desired, and the node chooses not to participate in a hop-by-hop route;
- The node cannot maintain its membership in the temporary DAG for the minimum life time specified in the Route Discovery option.

A node MUST update the values of link-level routing metrics included inside the DIO in accordance with the D field in the Route Discovery option. If the D field is 0x00, i.e., the forward routes are being discovered, any link-level routing metric MUST be measured in the direction towards the node receiving the DIO. If the D field is 0x01, i.e., the backward routes are being discovered, any link-level routing metric MUST be measured in the direction towards the node originating the DIO. If the D field is 0x02, i.e., the bidirectional routes are being discovered, any link-level routing metric MUST be calculated so as to take in account the metric's value in both directions. The rules for calculating bidirectional metric values will be specified in a separate document.

6.5. Additional Processing of a DIO Carrying a Route Discovery Option At An Intermediate Router

An intermediate router MUST process a received DIO, carrying a Route Discovery option, in accordance with the following rules.

An intermediate router MUST discard the DIO with no further processing if the routing metric values do not satisfy one or more propagation constraints listed in the DIO. The router MAY check the route constraints listed inside the Route Discovery option and discard the DIO with no further processing if these constraints are not met.

An intermediate router MUST determine if this DIO is the best it has received so far for this temporary DAG in terms of the routing metrics (potentially using the OCP in the DODAG Configuration

object). If yes, the intermediate router MUST remember the routing metric values contained in this DIO along with the route travelled by the DIO so far and reset the trickle timer associated with the temporary DAG.

When the trickle timer associated with the temporary DAG fires, an intermediate router MUST generate a new DIO for this temporary DAG carrying the Route Discovery option, the best metric values it knows and the source route associated with these values (in a Record Route IPv6 extension header [I-D.thubert-6man-reverse-routing-header]).

6.6. Additional Processing of a DIO Carrying a Route Discovery Option At The Target Node

A node MUST process a received DIO, carrying a Route Discovery option that lists this node as the target, in accordance with the following rules.

A target MUST discard the DIO with no further processing if it can not evaluate the route constraints listed inside the Route Discovery option or if the routing metric values do not satisfy one or more of the propagation and route constraints.

Otherwise, the target considers the source route accumulated by the received DIO as one of the discovered routes. This document does not prescribe a particular method for selecting routes among the discovered ones. Suppose the Route Discovery option requires the discovery of "n" routes. The target may select these "n" routes in any manner it desires. Example selection methods include selecting the first "n" routes it discovers or selecting the "n" best routes discovered over a certain time period, potentially using the OCP specified in the Route Discovery option for route comparison.

After selecting one or more discovered routes, the target MUST send one or more RPL Control Messages carrying a Discovery Reply Object (defined in the next section) back to the origin (identified by the DODAGID field in the DIO Base Object) as specified in Section 7.

A target MUST NOT forward a DIO carrying a Route Discovery option any further.

7. Propagation of Discovery Reply Messages

7.1. The Discovery Reply Object (DRO)

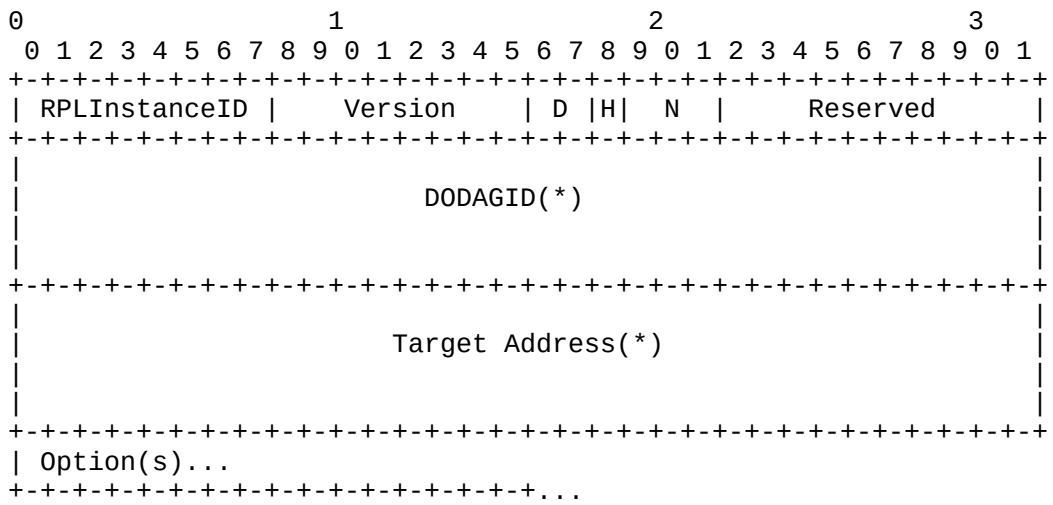


Figure 2: Format of the Discovery Reply Object (DRO)

This document defines a new RPL Control Message type, the Discovery Reply Object (DRO) with code 0x04 (to be confirmed by IANA), that serves one of the following functions:

- An acknowledgement from the target to the origin regarding the successful discovery of backward source routes;
- Carries one or more forward/bidirectional source routes from the target to the origin;
- Establishes one hop-by-hop forward/backward/bidirectional route as it travels from the target to the origin.

The format for a Discovery Reply Object (DRO) is shown in Figure 2. A DRO consists of the following fields:

- RPLInstanceID: The RPLInstanceID of the temporary DAG used for route discovery.
- Version: The Version of the temporary DAG used for route discovery.
- D: A 2-bit field that indicates the direction of the discovered routes:

  * D = 0x00: Forward;

  * D = 0x01: Backward;

  * D = 0x02: Bidirectional.

  This field has the same value as the corresponding field in the Route Discovery option.

- o H: A flag that is set if the DRO is establishing an hop-by-hop route. If this flag is set, the DRO MUST travel from the target to the origin along the hop-by-hop route being established and MUST include one Source Route option (defined in Section 7.1.1) that contains the remaining routers on this route (as described in Section 7.4). Since the state that a node needs to maintain regarding a hop-by-hop route includes the RPLInstanceID, the DODAGID and the IPv6 address of the route's destination, a DRO with H flag set MUST also include:

  * The DODAGID of the temporary DAG used for route discovery; and

  * The Target Address if the hop-by-hop route is forward or bidirectional.

  The H flag MUST be clear if the DRO carries (or is an acknowledgement for the discovery of) one or more source routes contained in the Source Route options. The target can unicast such a DRO to the origin or send it along the temporary DAG used for route discovery. If the DRO is unicast to the origin, it MUST NOT include the DODAGID and Target Address fields. If the DRO is sent along the temporary DAG, it MUST include the DODAGID field and MUST NOT include the Target Address field.

- o N: A 3-bit field that indicates the number of source routes carried or acknowledged in the DRO. This field MUST have value 1 if the DRO is establishing a hop-by-hop route.

- o Reserved: These bits are reserved for future use. These bits MUST be set to zero on transmission and MUST be ignored on reception.

- o DODAGID: The DODAGID of the temporary DAG used for route discovery. The DODAGID also identifies the origin. This field MUST be present in the DRO if the H flag is set or if the H flag is clear but the DRO needs to travel along the temporary DAG. Otherwise, this field need not be present in the DRO. The RPLInstanceID, the Version and the DODAGID together uniquely identify the temporary DAG used for route discovery and can be copied from the Base Object of the DIO advertizing the temporary

DAG.

- Target Address: The IPv6 address of the target generating the Discovery Reply Object. This field MUST be present in the DRO if the H flag is set and the hop-by-hop route being established is forward or bidirectional.

- Options: The Discovery Reply Object MAY carry up to N Source Route options (defined in the next section) with each such option carrying a source route and optionally followed by a Metric Container option that lists the aggregated values for the routing metrics for the source route.

7.1.1. The Source Route Option

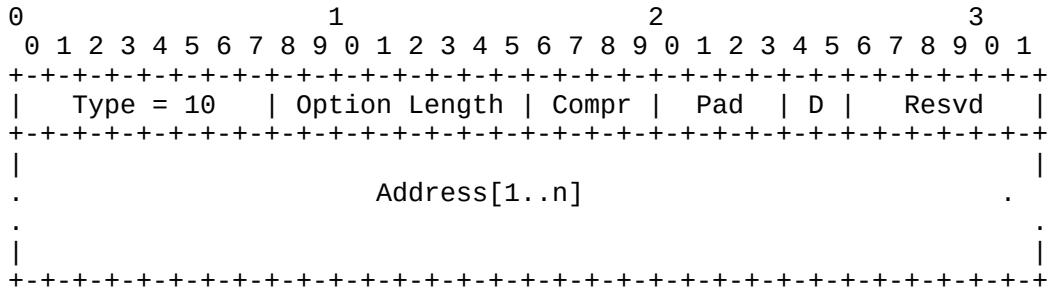


Figure 3: Format of the Source Route Option

The Source Route option, illustrated in Figure 3, carries a source route. When a Source Route option carries a complete source route between the origin and the target, it MAY be immediately followed by a Metric Container option that contains the aggregated values of the routing metrics for this source route.

A Source Route option consists of the following fields:

- Option Type = 0x0A (to be confirmed by IANA).

- Option Length = Variable, depending on the size of the Addresses vector.

- Compr: 4-bit unsigned integer indicating the number of prefix octets that are elided from each address. For example, Compr value will be 0 if full IPv6 addresses are carried in the Addresses vector.

- Pad: 4-bit unsigned integer. Number of octets that are used for padding between Address[n] and the end of the Source Route option.

- D: A 2-bit field that indicates the direction of the source route:

  * D = 0x00: Forward, i.e., from the origin to the target;

  * D = 0x01: Backward i. e., from the target to the origin;

  * D = 0x02: Bidirectional.

  Note that the D field in a Source Route option is independent from the D field in the DRO containing the Source Route option.

- Resvd: These bits are reserved for future use. These bits MUST be set to zero on transmission and MUST be ignored on reception.

- Address[1..n]: Vector of addresses, numbered 1 to n. Each vector element has size (16 - Compr) octets.

Note that the format of the Source Route option is very similar to that of proposed Type 4 Routing Header [I-D.ietf-6man-rpl-routing-header].

A common network configuration for an RPL domain is that all routers within an LLN share a common prefix. The Source Route option uses the Compr field to allow compaction of the Address[1..n] vector when all entries share the same prefix as the DODAGID or the Target Address of the encapsulating Discovery Reply Object. The shared prefix octets are not carried within the Source Route option and each entry in Address[1..n] has size (16 - Compr) octets. When Compr is non-zero, there may exist unused octets between the last entry, Address[n], and the end of the Source Route option. The Pad field indicates the number of unused octets that are used for padding. Note that when Compr is 0, Pad MUST be null and carry a value 0.

The Source Route option MUST NOT specify a path that visits a router more than once. When generating a Source Route option, the target may not know the mapping between IPv6 addresses and routers. Minimally, the target MUST ensure that:

- The IPv6 Addresses do not appear more than once;

- The IPv6 addresses of the origin and the target do not appear in the Address vector.

Multicast addresses MUST NOT appear in a Source Route option.

7.1.2. Processing a DRO At An Intermediate Router

When an intermediate router receives a DRO with a clear H flag, it MUST forward the DRO to a parent node in the temporary DAG.

When an intermediate router receives a DRO that has H flag set and contains multiple Source Route options, the router MUST drop the DRO with no further processing.

When an intermediate router receives a DRO that has H flag set and contains a single Source Route option, the router processes the DRO as described in Section 7.4.

7.2. DRO as Acknowledgement for Backward Source Routes

After selecting one or more backward source routes, a target MAY send a DRO message to the origin as an acknowledgement for the discovered routes. A DRO, serving as an acknowledgement for backward source route discovery, has its D field set to 0x01 (indicating backward) while the H flag is cleared (indicating source route). The N field is set to indicate the number of discovered backward source routes being acknowledged. Such a DRO message MUST NOT contain any option.

The target MAY unicast this DRO message to the origin or it MAY forward the DRO message to a parent in the temporary DAG. The target should take into consideration the minimum life time of the temporary DAG when deciding to use it to send the DRO to the origin.

7.3. DRO as Carrier of Forward/Bidirectional Source Routes

The target MUST convey the discovered forward/bidirectional source routes to the origin via the Source Route options inside one or more DRO messages. Such a DRO message MUST have its D field set to 0x00 (if it carries forward routes) or 0x02 (if its carries bidirectional routes). Also, the H flag MUST be cleared and the N field MUST indicate the number of Source Route options in the DRO. Each Source Route option inside the DRO MAY immediately be followed by a Metric Container option that carries the aggregated values of the relevant routing metrics for this source route.

The target MAY unicast this DRO message to the origin or it MAY forward the DRO message to a parent in the temporary DAG. The target should take into consideration the minimum life time of the temporary DAG when deciding to use it to send the DRO to the origin.

7.4. Establishing Hop-by-hop Routes Via DRO

In order to establish a hop-by-hop route, the target MUST send a DRO message along the discovered route, which is specified in a Source Route option. The D field in the DRO MUST reflect the direction of the discovered route. The H bit in the DRO MUST be set and the DRO MUST include the DODAGID field. If a forward or bidirectional hop-by-hop route is being established, the DRO MUST include the Target Address field as well. The N field in the DRO MUST be set to 1 and the DRO MUST include exactly one Source Route option. The target forwards the DRO to the next hop along the discovered route and includes the discovered route, excluding itself and the origin, inside the Source Route option in backward direction. Thus, the D field in the Source Route option MUST be 0x01.

If the hop-by-hop route is in the backward direction, the target MUST establish the hop-by-hop state for the route before sending the DRO message. Such hop-by-hop state includes the RPLInstanceID, the DODAGID and the route's destination (in this case, the origin's address or the DODAGID).

A router receiving a DRO message MUST drop the DRO if the router cannot establish the hop-by-hop state for the route or if its own address does not appear as the first element in the Address vector in the Source Route option. Otherwise, the router MUST establish the hop-by-hop state in the direction specified in the D field in the DRO. The hop-by-hop state in the forward direction includes the RPLInstanceID, the DODAGID and the target's address. The hop-by-hop state in the backward direction includes the RPLInstanceID, the DODAGID and the origin's address. After establishing the hop-by-hop state, the router MUST remove its own address from the route contained in the Source Route option and forward the DRO to the next hop (Address[0] in the Source Route option).

8. Security Considerations

TBA

9. IANA Considerations

TBA

10. Acknowledgements

Authors gratefully acknowledge the contributions of the following

individuals (in alphabetical order) in the development of this document: Dominique Barthel, Thomas Clausen, Richard Kelsey, Zach Shelby, Pascal Thubert and JP Vasseur.

11. References

11.1. Normative References

[I-D.goyal-roll-p2p-measurement]
Goyal, M. and E. Baccelli, "A Mechanism to Measure the Quality of a Point-to-point Route in a Low Power and Lossy Network", draft-goyal-roll-p2p-measurement-01 (work in progress), October 2010.

[I-D.ietf-6man-rpl-option]
Hui, J. and J. Vasseur, "RPL Option for Carrying RPL Information in Data-Plane Datagrams", draft-ietf-6man-rpl-option-01 (work in progress), October 2010.

[I-D.ietf-6man-rpl-routing-header]
Hui, J., Vasseur, J., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with RPL", draft-ietf-6man-rpl-routing-header-01 (work in progress), October 2010.

[I-D.ietf-roll-routing-metrics]
Vasseur, J., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics used for Path Calculation in Low Power and Lossy Networks", draft-ietf-roll-routing-metrics-17 (work in progress), January 2011.

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-17 (work in progress), December 2010.

[I-D.ietf-roll-trickle]
Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", draft-ietf-roll-trickle-08 (work in progress), January 2011.

[I-D.thubert-6man-reverse-routing-header]
Thubert, P., "Reverse Routing Header",

draft-thubert-6man-reverse-routing-header-00 (work in progress), June 2010.

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

11.2. Informative References

[I-D.brandt-roll-rpl-applicability-home-building] Brandt, A., Baccelli, E., and R. Cragie, "Applicability Statement: The use of RPL in Building and Home Environments", draft-brandt-roll-rpl-applicability-home-building-01 (work in progress), November 2010.

[I-D.ietf-roll-terminology] Vasseur, J., "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-04 (work in progress), September 2010.

[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

Mukul Goyal (editor)
University of Wisconsin Milwaukee
3200 N Cramer St
Milwaukee, WI 53201
USA

Phone: +1 414 2295001
Email: mukul@uwm.edu

Emmanuel Baccelli
INRIA

Phone: +33-169-335-511
Email: Emmanuel.Baccelli@inria.fr
URI: http://www.emmanuelbaccelli.org/

Anders Brandt
Sigma Designs
Emdrupvej 26A, 1.
Copenhagen, Dk-2100
Denmark

Phone: +45-29609501
Email: abr@sdesigns.dk

Robert Cragie
Gridmerge Ltd
89 Greenfield Crescent
Wakefield WF4 4WA
UK

Phone: +44-1924910888
Email: robert.cragie@gridmerge.com

Jerald Martocci
Johnson Controls
507 E Michigan St
Milwaukee, WI 53202
USA

Phone: +1 414-524-4010
Email: jerald.p.martocci@jci.com

Charles Perkins
Tellabs Inc.

charliep@computer.org