

本文翻译者：weicq2000（2014 年 4 月 5 日译完。便于对照阅读，译文与原文页数对应。）

Internet Engineering Task Force (IETF)

Request for Comments: 6550

Category: Standards Track

ISSN: 2070-1721

T. Winter, Ed.

P. Thubert, Ed.

Cisco Systems

A. Brandt

Sigma Designs

J. Hui

Arch Rock Corporation

R. Kelsey

Ember Corporation

P. Levis

Stanford University

K. Pister

Dust Networks

R. Struik

Struik Security Consultancy

JP. Vasseur

Cisco Systems

R. Alexander

Cooper Power Systems

March 2012

RPL: 低功耗有损网络的 IPv6 路由协议

摘要

低功耗有损网络(Low-Power and Lossy Networks, LLNs)是一类网络，这类网络中路由器和它们的互连受到限制。LLN 路由器一般在处理能力、存储器和能量(电池功率)上受到限制。它们之间的互连具有高丢失率、低数据速率和不稳定的特点。LLNs 可由几十个到上千个路由器构成。支持的数据流包括点对点(LLN 内设备间)、点对多点(从中心控制点到 LLN 内的设备子集)，以及多点对点(从 LLN 内的多个设备到中心控制点)。本文档规定低功耗有损网络的 IPv6 路由协议(IPv6 Routing Protocol for Low-Power and Lossy Networks, RPL)，该协议提供的机制支持从 LLN 内多个设备到中心控制点的多点对点流量，以及从中心控制点到 LLN 内多个设备的点对多点流量。也可以支持点对点流量。

本备忘录状态

这是Internet Standards Track文档。

本文档是Internet Engineering Task Force (IETF)的成果。它代表IETF社区的共识。它已经经过公开审议，由Internet Engineering Steering Group (IESG)批准颁布。有关Internet Standards的更多信息参阅RFC 5741第2章。

关于本文档目前状态、任何勘误，以及如何对本文档提供反馈意见的有关信息可在<http://www.rfc-editor.org/info/rfc6550>上得到。

版权声明

版权(c)归 2012 IETF Trust 和本文档撰写者(们)所有。保留所有权利。

本文档遵从本文档颁布日有效的 BCP 78 和 IETF Trust 的 Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>)规定。请仔细查阅这些文档，这些文档解释了对于本文档来说，您享有的权利和受到的限制。从本文档摘录的 Code Components 必须包括如 Trust Legal Provisions 第 4.e 节所述的 Simplified BSD License 文本，如 Simplified BSD License 所述，对所提供的 Code Components 不做担保。

目录

第1章	序言	8
1-1	设计原则	8
1-2	预期的链路层类型	10
第2章	术语	10
第3章	协议综述	13
3-1	拓扑结	13
3-1-1	构造拓扑结	13
3-1-2	RRL标识符	14
3-1-3	实例、DODAGs和DODAG版本	14
3-2	上行路由和DODAG架构	16
3-2-1	目标函数(OF)	17
3-2-2	DODAG修复	17
3-2-3	安全	17
3-2-4	接地DODAG和浮动DODAG	18
3-2-5	本地DODAGs	18
3-2-6	管理偏好	18
3-2-7	数据-路径合法性和环路检测	18
3-2-8	分布式算法运行	19
3-3	下行路由和目的地通告	19
3-4	本地DODAGs路由发现	20
3-5	等级(Rank)属性	20
3-5-1	等级比较(DAGRank())	21
3-5-2	等级关系	22
3-6	RPL使用的路由度量和约束	23
3-7	环路避免	24
3-7-1	贪婪和不稳定	24
3-7-2	DODAG环路	26
3-7-3	DAO环路	27
第4章	RPL支持的流量	27
4-1	多点到点流量	27
4-2	点到多点流量	27
4-3	点到点流量	27
第5章	RPL实例	28
5-1	RPL实例ID	29
第6章	ICMPv6 RPL控制消息	30
6-1	RPL安全字段	32
6-2	DODAG信息请求(DIS)	38
6-2-1	DIS基本对象格式	38
6-2-2	安全DIS	38
6-2-3	DIS选项	38
6-3	DODAG信息对象(DIO)	38
6-3-1	DIO基本对象格式	39

6-3-2	安全DIO	41
6-3-3	DIO选项	41
6-4	目的地通告对象(DAO)	41
6-4-1	DAO 基本对象格式	42
6-4-2	安全DAO	43
6-4-3	DAO选项	43
6-5	目的地通告对象确认(DAO-ACK)	43
6-5-1	DAO-ACK 基本对象格式	44
6-5-2	安全DAO-ACK	45
6-5-3	DAO-ACK选项	45
6-6	一致性检验(CC)	45
6-6-1	CC基本对象格式	46
6-6-2	CC选项	47
6-7	RPL控制消息选项	47
6-7-1	RPL控制消息选项通用格式	47
6-7-2	Pad1	48
6-7-3	PadN	48
6-7-4	DAG度量容器	49
6-7-5	路由信息	50
6-7-6	DODAG配置	52
6-7-7	RPL目标	54
6-7-8	传送信息	55
6-7-9	请求信息	58
6-7-10	前缀信息	59
6-7-11	RPL目标描述符	63
第7章	序列计数器	63
7-1	序列计数器概述	63
7-2	序列计数器运行	64
第8章	上行路由	66
8-1	DIO基本规则	67
8-2	上行路由发现和维护	67
8-2-1	在DODAG版本中的邻居和父代	67
8-2-2	跨DODAG版本的邻居和父代	68
8-2-3	DIO消息通信	73
8-3	DIO发送	74
8-3-1	Trickle参数	75
8-4	DODAG选择	75
8-5	作为叶节点的运行	75
8-6	管理等级	76
第9章	下行路由	77
9-1	目的地通告父代	77
9-2	下行路由发现和维护	78
9-2-1	路径序列维护	79
9-2-2	DAO消息生成	79

9-3	DAO基本规则	80
9-4	DAO消息结构	80
9-5	DAO发送调度	83
9-6	触发DAO消息	83
9-7	非存储模式	84
9-8	存储模式	85
9-9	路径控制	86
9-9-1	路径控制举例	88
9-10	多播目的地通告消息	89
第10章	安全机制	90
10-1	安全综述	90
10-2	加入安全网络	91
10-3	安装密钥	92
10-4	一致性检验	93
10-5	计数器	93
10-6	离站分组发送	94
10-7	入站分组接收	95
10-7-1	时间戳密钥检验	97
10-8	完整性和保密性覆盖范围	97
10-9	密码的操作模式	98
10-9-1	CCM随机数	98
10-9-2	签名	99
第11章	分组转发和环路避免/检测	99
11-1	分组转发建议	99
11-2	环路避免和检测	101
11-2-1	源节点运行	102
11-2-2	路由器运行	102
第12章	多播运行	104
第13章	维护路由相邻性	105
第14章	目标函数指南	106
14-1	目标函数行为	106
第15章	采用邻居发现的互操作性建议	108
第16章	归纳互操作实现要求	109
16-1	通用要求	109
16-2	(仅)作为RPL叶节点运行	110
16-3	作为RPL路由器运行	110
16-3-1	(仅)支持上行路由	110
16-3-2	采用非存储模式支持上行路由和下行路由	110
16-3-3	采用存储模式支持上行路由和下行路由	111
16-4	将来规范内容	111
第17章	RPL常数和变量	112
第18章	可管理性考虑	113
18-1	引言	114
18-2	配置管理	115

18-2-1	初始化模式	115
18-2-2	DIO和DAO基本消息和选项配置	115
18-2-3	LLN中每个路由器上配置的协议参数	116
18-2-4	LLN中每个非DODAG根路由器上配置的协议参数	117
18-2-5	DODAG根上配置的参数	117
18-2-6	与基于DAO机制有关的RPL参数配置	118
18-2-7	与安全机制有关的RPL参数配置	119
18-2-8	默认值	119
18-3	监测RPL运行	120
18-3-1	监测DODAG参数	120
18-3-2	监测DODAG不一致和环路检测	121
18-4	监测RPL数据结构	121
18-4-1	候选邻居数据结构	121
18-4-2	目的地导向的有向无循环图(DODAG)表	122
18-4-3	路由表和DAO路由条目	122
18-5	故障管理	123
18-6	策略	124
18-7	故障隔离	125
18-8	对其他协议的影响	125
18-9	性能管理	126
18-10	诊断	126
第19章	安全考虑	126
19-1	综述	126
第20章	IANA考虑	128
20-1	RPL控制消息	128
20-2	RPL控制代码的新注册表	128
20-3	运行模式(MOP)的新注册表	129
20-4	RPL控制消息选项	130
20-5	目标代码点(OCP)注册表	131
20-6	安全部分算法的新注册表	131
20-7	安全部分标记的新注册表	132
20-8	Per-KIM安全等级的新注册表	132
20-9	DODAG信息请求(DIS)标记的新注册表	133
20-10	DODAG信息对象(DIO)标记的新注册表	134
20-11	目的地通告对象(DAO)标记的新注册表	134
20-12	目的地通告对象(DAO)确认标记的新注册表	135
20-13	一致性检验(CC)标记的新注册表	135
20-14	DODAG配置选项标记的新注册表	136
20-15	RPL目标选项标记的新注册表	136
20-16	传送信息选项标记的新注册表	137
20-17	请求信息选项标记的新注册表	137
20-18	ICMPv6: 源路由首部中的错误	138
20-19	本地链路范围多播地址	138
第21章	致谢	138

第22章	撰稿人	139
第23章	参考文献	139
23-1	标准类参考文献	139
23-2	信息类参考文献	140
附录A	操作举例	143
A-1	采用含有节点自有前缀的存储模式的运行举例	143
A-1-1	DIO消息和PIO	144
A-1-2	DAO消息	145
A-1-3	路由信息库	145
A-2	采用含有宽子网前缀的存储模式的运行举例	146
A-2-1	DIO消息和PIO	147
A-2-2	DAO消息	148
A-2-3	路由信息库	148
A-3	采用含有节点自有前缀的非存储模式的运行举例	149
A-3-1	DIO消息和PIO	150
A-3-2	DAO消息	150
A-3-3	路由信息库	151
A-4	采用含有宽子网前缀的非存储模式的运行举例	151
A-4-1	DIO消息和PIO	152
A-4-2	DAO消息	153
A-4-3	路由信息库	153
A-5	采用外部前缀举例	154

第 1 章 序言

低功耗有损网络(LLNs)由大量受限节点(它们在处理能力、存储能力上受到限制;以及在能量上会因用电池供电或能量净化受到限制)组成。这些路由器通过有损链路互连,一般仅支持不稳定的低速数据,分组交付率相对较低。这些网络的另一个特点是流量模式不仅有简单的点对点,而且有点对多点或多点对点多种模式。此外,这样的网络有可能由上千个节点构成。这些特点对路由解决方案提出了独特的挑战:IETF ROLL 工作组针对在[RFC5867]、[RFC5826]、[RFC5673]和[RFC5548]中规定的低功耗有损网络(LLN)路由协议,定义了特定应用路由要求。

本文档规定 LLNs 的 IPv6 路由协议(IPv6 Routing Protocol for LLNs, RPL)。注意,虽然 RPL 是按照上述多个要求文档中所列要求制定的,但是 RPL 的应用绝不限于这些应用。

1-1 设计原则

RPL 被设计满足在[RFC5867]、[RFC5826]、[RFC5673]和[RFC5548]中详细阐明的要求。网络可以同时运行多个 RPL 实例。每个这样的实例可以适应不同、以及潜在不相容的约束或性能标准。本文档定义单个实例如何运行。

为了拓宽 LLN 应用领域,RPL 把分组处理和转发从路由优化目标中分离出去。这类目标例子包括尽量减少能量、尽量缩短延时或满足约束。本文档介绍 RPL 运行模式。其他伙伴文档规定路由目标函数(Objective Functions)。RPL 实现在支持具体 LLN 应用时,将根据应用要求包括必要的 Objective Function(s)。

运行 RPL 需要双向链路。在一些 LLN 环境,这些链路可能展现不对称的特性。将路由器用作父代前需要验证路由器的可达性。RPL 期望在选择父代阶段触发外部机制,以检验链路特性和邻居可达性。邻居不可达检测(Neighbor Unreachability Detection, NUD)是这样的机制,但是也可以选用替代方案,包括双向转发检测(Bidirectional Forwarding Detection, BFD) [RFC5881]以及来自较低层的提示(通过如[RFC5184]那样的第 2 层(L2)触发器)。通常,更看重可对流量起反应的检测机制,以便尽量减小监测未使用链路的成本。

RPL 也期望在数据分组中接入和传送某些控制信息(称为“RPL Packet Information”)的外部机制。RPL Packet Information 在第 11-2 节定义, 该信息能够与带 RPL Instance 和 RPL 路由状态验证的数据分组关联。RPL 选项[RFC6553]是这类机制的例子。除了使用严格的源路由 (该路由用于在第 9 章详细介绍的 Non-Storing 模式中的下行(Downward)分组)以外, 该机制是所有分组需要的, 该机制在性质上可阻止无限循环, 减轻对 RPL Packet Information 的需要。将来的伙伴规范可能提出替代方法, 以便在 IPv6 分组中携带 RPL Packet Information, 以及可能扩展 RPL Packet Information 来支持附加功能。

RPL 提供在动态形成的网络拓扑上传播信息的机制。这种传播能够尽量减少节点中的配置, 使节点能够尽可能自主运行。这种机制使用 Trickle [RFC6206]优化传播, 如第 8-3 节所述。

在一些应用中, RPL 集合了拥有独立前缀的路由器的拓扑结构。这些前缀是否可以聚合取决于路由器的起源。路由器拥有的前缀作为 on-link 通告。

RPL 也引入一种能力, 将公共前缀与子网绑在一起, 在子网内路由。源可以注入有关由 RPL 传播的子网的信息, 并且源是那个子网的官方。因为许多 LLN 链路有非传递性, RPL 在子网上传播的公共前缀绝不能作为 on-link 通告。

尤其是, RPL 可以传播 IPv6 Neighbor Discovery (ND)信息, 诸如 Prefix Information Option (PIO) [RFC4861]和 Route Information Option (RIO) [RFC4191]。虽然 ND 信息不会与路由通告混淆并且它绝不会在另一个路由协议中被直接重新分配, 由 RPL 传播的 ND 信息保留所有它的路由器到主机(少数情况下有路由器到路由器的扩展)的原始语义。RPL 节点常常兼有主机和路由器的行为。作为主机, 它处理在[RFC4191]、[RFC4861]、[RFC4862]和[RFC6275]中规定的选项。作为路由器, RPL 节点可以通告来自特定链路所要求选项的信息, 例如, 在 ND Router Advertisement (RA)消息中, 具体操作超出本文档范围。

本规范的一组伙伴文档将对应用语句形式提供进一步指导,这些应用语句规定了适合楼宇自动化(Building Automation)、家庭自动化(Home Automation)、工业(Industrial)和城市(Urban)应用场景的一组操作要点。

1-2 预期的链路层类型

兼容 IP 分层架构, RPL 不依赖任何特定链路层技术的具体特征。RPL 被设计能运行在各种不同链路层上,包括受限、可能有损的链路层,或者典型的与极度受限的主机设备或路由器设备一起使用的链路层,这样的技术包括但不限于低功耗无线技术或电力线通信(Power Line Communication , PLC)技术。

当在 RPL 和具体链路层技术间设计链路层接口时,实施者可在[RFC3819]中获得有益帮助。

第 2 章 术语

本文档中出现的关键词“MUST”、“MUST NOT”、“REQUIRED”、“SHALL”、“SHALL NOT”、“SHOULD”、“SHOULD NOT”、“RECOMMENDED”、“NOT RECOMMENDED”、“MAY”和“OPTIONAL”的含义,遵循RFC 2119[RFC2119]描述。

此外,本文档使用来自[ROLL-TERMS]的术语,并且引入下述术语:

- DAG(Directed Acyclic Graph):

有向无循环图形,有向图有这样的性质,即,所有边缘均以没有循环存在的方式定向。所有边缘均被包含在定向到或终结在一个或多个根节点的路径中。

- DAG root:

DAG 根是没有出站边缘的 DAG 内的节点。因为图形是非循环的,根据定义,所有 DAGs 必须至少有一个 DAG 根,并且所有路径终结在 DAG 根。

- Destination-Oriented DAG (DODAG):

扎根于单个目的地的 DAG,即,扎根于没有出站边缘的单个 DAG 根(DODAG 根)的 DAG。

- **DODAG root:**

DODAG 根是 DODAG 的 DAG 根。DODAG 根可以充当 DODAG 的边缘路由器；尤其是，它可以聚合 DODAG 中的路由，可以重新分配 DODAG 路由到其他路由协议中。

- **Virtual DODAG root:**

Virtual DODAG 根是使用两个或多个 RPL 路由器的结果，例如，6LoWPAN Border Routers (6LBRs)、同步 DODAG 状态的协调，以及相对于 LLN，好像它们是单个 DODAG 根(使用多个接口)的协调一致行动。这种协调最有可能发生在可靠传输链路上加电设备间，方案实施细节超出本规范范围(在将来的伙伴规范中定义)。

- **Up:**

Up 指沿着 DODAG 边缘，从叶节点朝向 DODAG 根的方向。这与在图形和深度优先搜索(depth-first-search)中使用的通用术语意思相同，在那里，认为离根更远的端点“更深”或“在下”，认为更为贴近根的端点“更浅”或“在上”。

- **Down:**

Down 指从 DODAG 根朝向叶节点的方向，与 DODAG 边缘方向相反。这与在图形和深度优先搜索中使用的通用术语意思相同，在那里，认为离根更远的端点“更深”或“在下”，认为更为贴近根的端点“更浅”或“在上”。

- **Rank:**

节点的 Rank 定义就 DODAG 根而言，节点相对其他节点的具体位置。Rank 在向下(Down)方向严格增加，在向上(Up)方向严格减少。计算 Rank 的精确方法取决于 DAG 的 Objective Function (OF)。Rank 可以类似于跟踪简单的拓扑距离，可以作为链路度量的函数计算，可以看作是诸如约束的其他特性。

- **Objective Function (OF):**

目标函数(OF)定义路由度量(routing metrics)、优化目标和相关函数如何用于计算 Rank。此外，OF 决定如何选择 DODAG 中的父代，从而决定 DODAG 形成。

- **Objective Code Point (OCP):**

OCP 是标识符，指出 DODAG 使用哪一个 OF。

- **RPLInstanceID:**

RPLInstanceID 是网络中唯一的标识符。有相同 RPLInstanceID 的 DODAGs 共享相同 OF。

- **RPL Instance:**

RPL Instance 是一组共享 RPLInstanceID 的一个或多个 DODAG。在大多数情况下，RPL 节点归属 RPL Instance 中的一个 DODAG。每个 RPL Instance 独立于其他 RPL Instances 运行。本文档介绍在单个 RPL Instance 中的运行情况。

- **DODAGID:**

DODAGID 是 DODAG 根的标识符。在 LLN 中 RPL Instance 范围内 DODAGID 唯一。元组(RPLInstanceID、DODAGID)唯一标识 DODAG。

- **DODAG Version:**

DODAG Version 是带有给定 DODAGID 的 DODAG 的具体迭代(“Version”)。

- **DODAGVersionNumber:**

DODAGVersionNumber 是序贯计数器，它由根增加，以便形成 DODAG 的新 Version。DODAG Version 由(RPLInstanceID、DODAGID、DODAGVersionNumber)元组唯一标识。

- **Goal:**

Goal 是特定应用目标，它的定义超出 RPL 范围。任何扎根于 DODAG 的节点需要知道这个 Goal，以便决定是否能够满足该 Goal。典型的 Goal 按照特定 OF 构建 DODAG，以及保持与一组主机的连接(例如，使用这样的 OF，该 OF 尽量减少度量并连接到用于保存所收集数据的特定数据库主机)。

- **Grounded:**

如果 DODAG 根能够满足 Goal，该 DODAG 是接地的。

- **Floating:**

如果 DODAG 没有接地，它是浮动的。浮动 DODAG 预期不会有满足目标需要的性能。然而，浮动 DODAG 可以提供到 DODAG 内其他节点的连接。

- **DODAG parent:**

DODAG 内节点的父代是在朝向 DODAG 根路径上节点的直接后继者之一。DODAG 父代的 Rank 比节点的低(参阅第 3-5-1 节)。

- **Sub-DODAG:**

节点的子 DODAG 是一组其他节点，这些节点到 DODAG 根的路径经过该节点。在节点的子 DODAG 中的节点有比该节点更大的 Rank(参阅第 3-5-1 节)。

- **Local DODAG:**

Local DODAGs 包含一个且只有一个根节点，Local DODAGs 使该单个根节点能够分配和管理由本地 RPLInstanceID 标识的 RPL Instance，不需要与其他节点协调。一般，这样做是为了优化到 LLN 内目的地的路由。(参阅第 5 章)。

- Global DODAG:

Global DODAG 使用可以在几个其他节点中协调的全局 RPLInstanceID。(参阅第 5 章)。

- DIO:

DODAG Information Object(参阅第 6-3 节)

- DAO:

Destination Advertisement Object(参阅第 6-4 节)

- DIS:

DODAG Information Solicitation(参阅第 6-2 节)

- CC:

Consistency Check(参阅第 6-6 节)

当它们形成网络，对比传统 IP 网络，LLN 设备常常既是主机，又是路由器。在本文档中，“主机”指能够生成但不能转发 RPL 流量的 LLN 设备；“路由器”指能够转发也能够生成 RPL 流量的 LLN 设备；“节点”指任何 RPL 设备，或者是主机或者是路由器。

第 3 章 协议综述

本章的目标是以[RFC4101]倡导的风格介绍 RPL。协议细节在后续章节介绍。

3-1 拓扑结构

本节介绍可形成的基本 RPL 拓扑结构，介绍构建这些拓扑结构的规则，即，指导 DODAG 形成的规则。

3-1-1 构造拓扑结构

LLNs，如 Radio Networks，一般没有预先定义的拓扑结构，例如，由点对点导线强制形成的拓扑结构，所以，RPL 必须发现链路，接着有节制地选择对端。

在许多情况，由于 Layer 2 范围仅部分重叠，RPL 形成非传递/Non-Broadcast Multi-Access (NBMA)网络拓扑结构，RPL 根据这些网络拓扑计算路由。

RPL 路由优化去或来自一个或多个根的流量，这些根充当该拓扑结构的汇接点(sink)。因此，RPL 如 Directed Acyclic Graph (DAG)那样安排拓扑结构，DAG 被划分成一个或多个 Destination Oriented DAGs (DODAGs)，每个汇接点一个 DODAG。如果 DAG 有多个根，那么，可以预期这些根通过公共骨干网(如传输链路)结成联盟。

3-1-2 RPL 标识符

RPL 使用 4 个值来标识和维护拓扑结构：

- 第一个是RPLInstanceID。RPLInstanceID标识一组一个或多个Destination Oriented DAG (DODAG)。网络可以有多个RPLInstanceID，每一个定义一组独立的DODAG，可以针对不同Objective Function (OF)和/或应用优化这些DODAG。由RPLInstanceID标识的DODAGs集合被称作RPL Instance。在同一RPL Instance中的所有DODAGs使用相同的OF。
- 第 2 个是 DODAGID。DODAGID 的范围是 RPL Instance。RPLInstanceID 和 DODAGID 结合唯一标识网络中单个 DODAG。RPL Instance 可以有多个 DODAG，每一个有唯一的 DODAGID。
- 第 3 个是 DODAGVersionNumber。DODAGVersionNumber 的范围是 DODAG。有时通过递增 DODAGVersionNumber，可以根据 DODAG 根重新构建 DODAG。RPLInstanceID、DODAGID 和 DODAGVersionNumber 三者结合唯一标识 DODAG Version。
- 第 4 个是 Rank。Rank 的范围是 DODAG Version。Rank 在 DODAG Version 上建立部分顺序，定义相对 DODAG 根的单个节点位置。

3-1-3 实例、DODAGs 和 DODAG 版本

RPL Instance 包括一个或多个 DODAG 根。RPL Instance 可以提供到某些目的地前缀的路由，可以通过 DODAG 根或 DODAG 内替代路径到达。这些 DODAG 根可以独立工作，也可以在网络上协同工作，网络不局限为 LLN。

RPL Instance 可能包括：

- 有单一根的单个 DODAG
 - * 例如，为尽量减少 Home Automation 应用中单个集中照明控制器产生的延时而优化的 DODAG。

- 多个有独立根的、未经协调的 DODAG(不同 DODAGIDs)
 - * 例如，在城市数据采集应用中的多个数据采集点，这些数据采集点没有相互协调的合适连接，或者这些数据采集点使用多 DODAG 结构作为动态和自主分割网络的方法。
- 单个有虚拟根的 DODAG，该虚拟根在骨干网络上配合 LLN 汇接点(采用相同 DODAGID)。
 - * 例如，采用可靠传送链路工作的多个边界路由器，如，支持 IPv6 Low-Power Wireless Personal Area Network (6LoWPAN)应用，这些边界路由器能够作为到相同 DODAG 汇接点的逻辑等价接口。
- 适合某些应用场景的上述组合。

每个 RPL 分组与特定 RPLInstanceID(参阅第 11-2 节)关联，因此，与 RPL Instance 关联(参阅第 5 章)。提供或者自动发现 RPLInstanceID 和应用流量的类型或业务之间的映射超出本规范范围(在将来的伙伴规范中定义)。

图 1 给出包括 3 个 DODAG 的 RPL Instance 举例，其中 DODAGs 有 DODAG 根 R1、R2 和 R3。这些 DODAG 根中的每一个通告相同的 RPLInstanceID。直线表示父代和子代间是连通的。

图 2 显示 DODAGVersionNumber 增加如何导致新的 DODAG Version。此图演示导致不同 DODAG 拓扑的 DODAGVersionNumber 增加。注意，新 DODAG Version 并不总是暗示不同 DODAG 拓扑。采纳某些拓扑改变需要新 DODAG Version，本规范稍后将谈及此。

虽然当连通性支持 DODAG 结构时，该 DODAG 结构允许每个节点有多个父代，请注意，在下面例子中采取类似树形结构主要为了便于介绍。

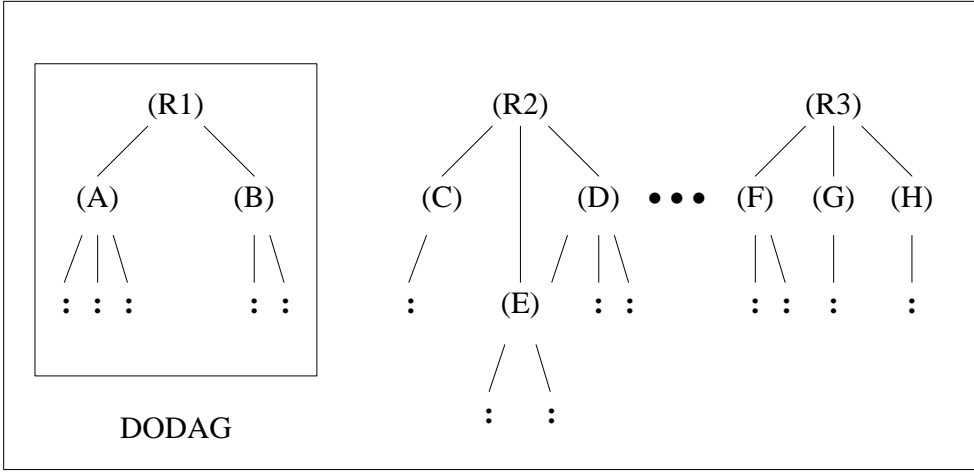


图 1 RPL Instance

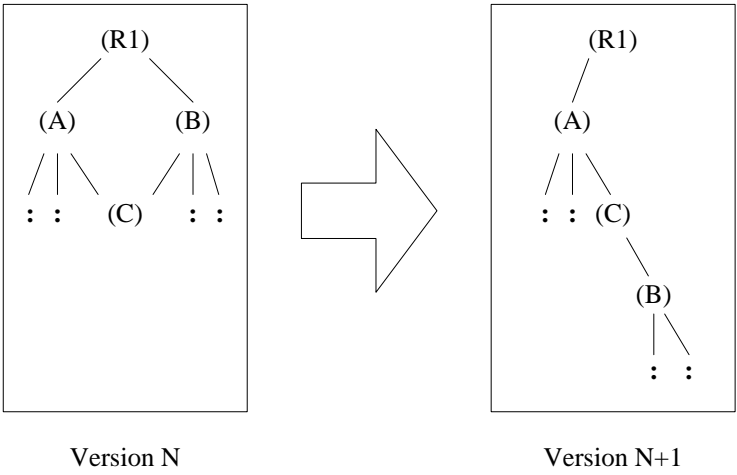


图 2 DODAG Version

3-2 上行路由和 DODAG 架构

RPL 提供向上(Up)朝向 DODAG 根的路由，形成按照 Objective Function (OF)优化的 DODAG。RPL 节点通过 DODAG Information Object (DIO)消息构建和维护这些 DODAG。

3-2-1 目标函数(OF)

目标函数(Objective Function, OF)定义 RPL 节点如何选择和优化 RPL Instance 中的路由。OF 由 DIO Configuration 选项中的 Objective Code Point (OCP)标识。OF 定义节点如何将一个或多个度量和约束(度量和约束在[RFC6551]中定义)转换成称作 Rank 的值, Rank 近似为节点离开 DODAG 根的距离。OF 也定义节点如何选择父代。更多细节参阅第 14 章、[RFC6551]、[RFC6552]和相关伙伴规范。

3-2-2 DODAG 修复

通过增加 DODAGVersionNumber, DODAG 根发起全局修复操作。这将启动新的 DODAG Version。在新 DODAG Version 中的节点可以选择新位置, 节点的 Rank 不受在旧 DODAG Version 中它们的 Rank 的限制。

RPL 也支持可以用于该 DODAG Version 内本地修复的机制。当根据 DODAG 根进行配置和由 DODAG 根中的策略进行控制时, DIO 消息规定必要的参数。

3-2-3 安全

RPL 支持消息保密性和完整性。它的设计使得当可获得且适合时, 可以使用链路层机制; 然而, 如果没有链路层机制, RPL 可以使用它自己的机制。RPL 有 3 种基本安全模式。

第一种称作“无安全的”, 发送 RPL 控制消息但不使用任何附加安全机制。无安全模式不是说 RPL 网络是不安全的: RPL 网络可以使用其他已有的安全基元(例如, 链路层安全), 满足应用安全需要。

3-2-4 接地 DODAG 和浮动 DODAG

DODAGs 可以接地或浮动：DODAG 根通告是哪一种情况。接地 DODAG 提供到主机的连接，这些主机被要求满足限定的应用目标。不预期浮动 DODAG 能够满足此目标；在大多数情况，浮动 DODAG 仅提供去 DODAG 内节点的路由。举例来讲，浮动 DODAGs 可用于修复期间保持互连。

3-2-5 本地 DODAGs

通过形成 Local DODAG (它的 DODAG 根是预期的目的地)，RPL 节点能够优化去 LLN 内目的地的路由。与全局 DAGs 不同，全局 DAGs 可由多个 DODAG 组成，本地 DAGs 有一个且仅有一个 DODAG，因此，仅有一个 DODAG 根。可根据需要构建 Local DODAGs。

3-2-6 管理偏好

实现/部署可以通过管理偏好，规定某些 DODAG 根应当先于其他 DODAG 根使用。管理偏好提供控制流量和工程 DODAG 形成的方法，以便更好地支持应用要求或需要。

3-2-7 数据-路径合法性和环路检测

LLNs 的低功耗和有损性质促使 RPL 使用按需(on demand)环路检测(使用数据分组)。因为数据流量可能是偶发的，维持始终随物理拓扑结构变动而更新的路由拓扑会浪费能源。在物理连接中 LLNs 一般呈现瞬间的、无碍流量的变化，但是，从控制平面密切跟踪这些变化代价高昂。直到有数据发送前，连接中出现的瞬态和偶发变化不需要 RPL 处理。RPL 这方面的设计汲取自现有的、高频率使用的 LLN 协议，以及关于 RPL 效用的大量实验和部署证据。

与数据分组一起传送的 RPL Packet Information 包括发射机的 Rank。分组的路由决定(上行或下行(Upward or Downward))和两个节点间 Rank 关系之间的不一致指出可能有环路。一旦收到这样的分组，节点发起本地修复操作。

例如，如果节点收到标记为上行(Upward)方向移动的分组，并且如果那个分组记载发射机有比接收节点更低的(更小的)Rank，那么，接收节点能够断定该分组不是沿上行(Upward)方向传播，以及断定 DODAG 不一致。

3-2-8 分布式算法运行

构建 DODAG 的分布式算法的高层概述如下：

- 一些节点被配置为 DODAG 根，采用关联的 DODAG 配置。
- 节点通告它们的存在、与 DODAG 的隶属关系、路由开销，以及将本地链路多播 DIO 消息发送到 all-RPL-nodes 的相关度量。
- 节点侦听 DIOs 并使用这些 DIOs 的信息，按照规定的 Objective Function 和节点邻居的 Rank，加入新的 DODAG (从而，选择 DODAG 父代)，或维持现有的 DODAG。
- 节点通过在 DODAG Version 中它们的 DODAG 父代，为由该 DIO 消息规定的目的地提供路由表条目。决定加入 DODAG 的节点，可以提供一个或多个 DODAG 父代，作为默认路由的下一跳，以及作为大量其他关联实例的外部路由。

3-3 下行路由和目的地通告

RPL 使用 Destination Advertisement Object (DAO)消息建立下行(Downward)路由。对于需要点对多点(P2MP)或点对点(P2P)流量的应用，DAO 消息是可选功能。RPL 支持两种下行(Downward)流量模式：Storing (完全有状态)或 Non-Storing (完全源路由)；参阅第 9 章。任何给定的 RPL Instance 或者是存储模式，或者是非存储模式。在所有两种情况中，P2P 分组都要被向上(Up)传送到 DODAG 根，接着被向下(Down)传送到最终目的地(除非目的地在上行(Upward)路由上)。在 Non-Storing 情况，分组在被向下(Down)传送前将依次通过到 DODAG 根的所有路径。在 Storing 情况，分组在到达 DODAG 根之前，通过源和目的地的共同祖先，可以被向下(Down)导引向目的地。

截止撰写本规范为止,还没有在哪个实现中希望:既支持 Storing 模式又支持 Non-Storing 模式运行。大多数实现希望或者仅支持无下行(Downward)路由、Non-Storing 模式,或者仅支持 Storing 模式。其他的运行模式,诸如 Storing 和 Non-Storing 混合模式,超出本规范范围,请参阅其他伙伴规范中的介绍。

本规范介绍支持 P2P 流量的基本运行模式。注意,进一步优化的 P2P 解决方案可参阅伙伴规范介绍。

3-4 本地 DODAGs 路由发现

作为选项,RPL 网络可以支持对 LLN 内特定目的地的按需 DODAGs 发现。这类 Local DODAGs 的行为稍微不同于 Global DODAGs:它们由 DODAGID 和 RPLInstanceID 的组合唯一限定。RPLInstanceID 指出 DODAG 是否是 Local DODAG。

3-5 等级(Rank)属性

节点的 Rank 是 DODAG Version 内那个节点位置的标量表示。Rank 用于避免环路和检测环路,以及,因此,必定表现出某些属性。Rank 的精确计算留给 Objective Function。即使具体 Rank 计算由 Objective Function 完成,无论 Objective Function 怎样,Rank 必须具有通用属性。

尤其是,当朝 DODAG 目的地探索 DODAG Version 时,节点的 Rank 必须单调减小。在这一点,可将 Rank 看作是 DODAG Version 内节点位置或半径的标量表示。

虽然这样的计算可能取决于,例如,父代、链路度量、节点度量,以及节点配置和策略,Objective Function 如何计算 Rank 的细节超出本规范范围。更多信息参阅第 14 章。

虽然 Rank 的值可以从路径度量推出,并受路径度量影响,Rank 不是路径开销。Rank 有它自己的属性,这些属性不一定是所有度量的属性:

类型(Type): Rank 是抽象数字值。

功能(Function): Rank 是 DODAG Version 内与邻居间相对位置的表示, 它不一定是离开根的距离或路径开销的良好指示或适当表示。

稳定性(Stability): Rank 的稳定性决定路由拓扑结构的稳定性。**推荐**采用一些缓冲或过滤措施保持拓扑结构稳定; 因此, Rank 的改变速度不一定像某些链路度量或节点度量的改变速度一样快。新 DODAG Version 可能是协调不一致的好机会, 这些不一致或许在 DODAG Version 内的度量和 Ranks 之间随时形成。

属性(Properties): Rank 严格单调递增, 它可用于确认来自根或朝向根的进展。度量, 像带宽或抖动一样, 不一定能显示这个属性。

抽象(Abstract): Rank 没有物理单位, 而是一系列的每跳递增, 这里, 由 Objective Function 决定每次增量分配。

按照 RPL 环路避免策略将 Rank 值输入 DODAG 父代选择。一旦添加父代, 就通告节点在 DODAG 内的 Rank 值, 限制有关 DODAG 内节点的 DODAG 父代选择和移动等进一步选项, 有利于环路避免。

3-5-1 等级比较(DAGRank())

可将 Rank 看作是固定点数, 其中整数部分和小数部分间的小数点位置由 MinHopRankIncrease 决定。MinHopRankIncrease 是节点和它的任何一个 DODAG 父代间以 Rank 为单位的最小增量。DODAG 根提供 MinHopRankIncrease。MinHopRankIncrease 在跳的开销精度和网络能够支持的最大跳数间取折中。例如, 非常大的 MinHopRankIncrease 能够精确描述给定跳对 Rank 的影响, 但不能支持很多跳。

在 Objective Function 计算 Rank 时, Objective Function 在整个(即, 16 位)Rank 数值上运算。当比较 Rank 时, 例如, 确定父代关系或环路检测, 使用 Rank 的整数部分。Rank 的整数部分由 DAGRank()宏指令, 按下式计算, 式中, floor(x)是求小于或等于 x 的最大整数的函数:

$$\text{DAGRank}(\text{rank}) = \text{floor}(\text{rank}/\text{MinHopRankIncrease})$$

例如, 如果 16 位 Rank 数值是十进制数 27, MinHopRankIncrease 是十进制数 16, 那么, $\text{DAGRank}(27) = \text{floor}(1.6875) = 1$ 。Rank 整数部分是 1, 分数(小数)部分是 11/16。

遵循本文档约定, 使用的宏指令 $\text{DAGRank}(\text{node})$ 可以转换为 $\text{DAGRank}(\text{node.rank})$, 其中 node.rank 是由节点维护的 Rank 值。

如果 $\text{DAGRank}(A)$ 小于 $\text{DAGRank}(B)$, Node A 的 Rank 比 Node B 的小。

如果 $\text{DAGRank}(A)$ 等于 $\text{DAGRank}(B)$, Node A 的 Rank 等于 Node B 的。

如果 $\text{DAGRank}(A)$ 大于 $\text{DAGRank}(B)$, Node A 的 Rank 比 Node B 的大。

3-5-2 等级关系

对于 LLN 内相邻的任何节点 M 和 N, Rank 计算维持下述属性:

$\text{DAGRank}(M)$ 小于 $\text{DAGRank}(N)$:

在这种情况下, M 的位置比 N 的位置更靠近 DODAG 根。Node M 可以安全地作为 Node N 的 DODAG 父代, 没有形成环路的风险。此外, 对于 Node N, 在 DODAG 父代集中的所有父代必须具有比 $\text{DAGRank}(N)$ 小的 Rank。换句话讲, 由 Node N 呈现的 Rank **必须** 大于由它的任何父代呈现的 Rank。

$\text{DAGRank}(M)$ 等于 $\text{DAGRank}(N)$:

在这种情况下, 在 DODAG 内并且相对于 DODAG 根来说, M 的位置和 N 的位置是类似或等同的。路由通过有相等 Rank 的节点可能引起路由环路(即, 如果那个节点也选择路由通过有相等 Rank 的节点)。

DAGRank(M)大于 DAGRank(N):

在这种情况下, M 的位置比 N 的位置距离 DODAG 根更远。此外, Node M 事实上可能位于 Node N 的子 DODAG 内。如果 Node N 选择 Node M 为 DODAG 父代, 存在产生环路的风险。

例如, 如果 Objective Function 尽量减小的度量是 ETX 或是延迟时间, 采用为得是密切跟踪 ETX(预期传输计数, 即在[RFC6551]中定义并在 LLN 中使用的十分通用的路由度量)的方法计算 Rank, 或者采用适合 DODAG 内正在使用的 Objective Function 的更复杂方法计算 Rank。

3-6 RPL 使用的路由度量和约束

路由协议使用路由度量计算最短路径。内部网关协议(Interior Gateway Protocols, IGP), 如 IS-IS ([RFC5120])和 OSPF ([RFC4915])使用静态链路度量。这样的链路度量可简单反映带宽, 或者也可以按照定义不同链路特征的几种度量的多项式函数计算这些链路度量。某些路由协议支持不止一种度量: 在绝大多数情况, 每种(子)拓扑结构使用一种度量。较少情况, 在采用等价多路径(Equal Cost Multiple Paths, ECMPs)时, 第二种度量可用作平分决胜法。多度量优化称作 NP-complete 问题, 有时得到某些集中路径计算引擎支持。

比较起来, LLNs 既需要静态度量又需要动态度量支持。此外, 链路度量和节点度量都需要。对于 RPL, 甚至需要能满足所有应用情况的复合度量。

此外, RPL 支持基于约束的路由, 其中, 约束可用于链路和节点。如果链路或节点不满足要求的约束, 它将被从候选邻居集中“删除”, 从而产生受约束的最短路径。

Objective Function 规定用于计算(受约束)路径的目标。此外, 节点被配置按照在 DIO 消息中通告的度量和约束, 支持 DODAG 中的一组度量和约束, 以及选择节点的父代。向上数据流(Upstream)度量和向下数据流(Downstream)度量可以合并或分别通告, 取决于 OF 和度量。如果分别通告它们, 可能会出现 DIO 父代集不同于 DAO 父代集情况(DAO 父代是单播 DAO 消息被发送到的节点)。然而, 所有都是与 Rank 计算规则有关的 DODAG 父代。

根据 RPL 使用的路由度量和约束去耦 Objective Function。而 OF 规定如 DODAG 父代选择、负载平衡等的规则，使用的度量和/或约束集合，以及因此，确定首选路径的度量和/或约束要基于 DIO 消息内 DAG 容器选项中携带的信息。

支持的链路/节点约束和度量集合参阅[RFC6551]。

例 1：最短路径：提供最短端到端延时的路径。

例 2：最短约束路径：不穿过任何电池供电节点的路径，以及优化路径可靠性的路径。

3-7 环路避免

当发生拓扑结构变化时，RPL 努力避免形成环路，而且 RPL 包括用于检测环路(如果环路真的发生)的基于 Rank 的数据路径验证机制(更多细节参阅第 11 章)。实践中，这表示 RPL 既不保证选择的路径无环路，也不保证绷紧延迟收敛时间，但是只要使用 RPL，它能够检测和修复环路。RPL 使用这个环路检测机制确保分组在 DODAG Version 内前行，必要时触发修复。

3-7-1 贪婪和不稳定

如果节点企图在 DODAG Version 内向更深方向移动(增加 Rank)，以便增加父代集规模或改善某些其他度量，将认为节点是贪婪的。一旦节点加入 DODAG Version，RPL 不允许某些行为，包括贪婪，以防止在 DODAG Version 内产生不稳定。

假设节点愿意接收和处理来自它自己的子 DODAG 内的节点(并且，一般是比它自己更深的节点)的 DIO 消息。在这种情况下，有可能存在反馈环路，在那里两个或多个节点继续在 DODAG Version 内尝试并移动，同时企图相互对照优化。在某些情况，这将导致不稳定。正是由于此原因，RPL 将节点可以处理来自更深节点的 DIO 消息情况限制在某种本地修复形式。这种方法提出了“事件视野”概念，凭借“事件视野”，节点不会因由于节点的行动超出进入不稳定的某种限制而受到影响，这种行动可能在节点自己的子 DODAG 内发生。

3-7-1-1 举例：贪婪的父代选择和不稳定

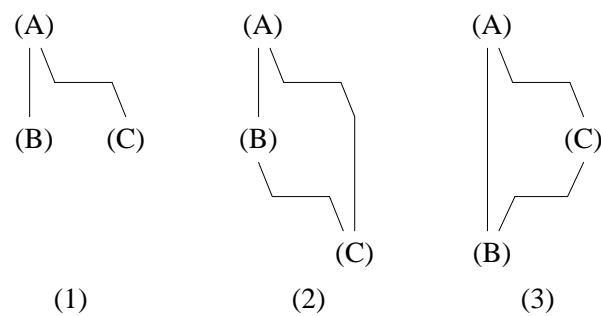


图3 贪婪的 DODAG 父代选择

图3描述3种不同配置下的DODAG。在所有3种配置中(B)和(C)间都存在可用链路。在图3-1中,Node(A)是Nodes(B)和Nodes(C)的DODAG父代。在图3-2中,Node(A)是Nodes(B)和Nodes(C)的DODAG父代,Node(B)还是Node(C)的DODAG父代。在图3-3中,Node(A)是Node(B)和Node(C)的DODAG父代,Node(C)还是Node(B)的DODAG父代。

如果RPL节点非常贪婪,除了它的那些最优先父代以外,它企图优化额外数量的父代,这会导致不稳定。考虑图3-1显示的DODAG。在这个例子中,Node(B)和Node(C)可能最优先选择Node(A)为DODAG父代,但是我们考虑它们在贪婪条件(在此条件下将尝试优化2个父代)下的运行情况。

- 设图3-1为初始条件。
- 假设Node(C)最初能够离开DODAG,并以较低的Rank重新加入,取Node(A)和Node(B)为DODAG父代,如图3-2所示。现在,Node(C)比Nodes(A)和Nodes(B)都深,并且Node(C)满足于有2个DODAG父代。
- 假设Node(B),在其贪心驱使下,愿意接收和处理来自Node(C)的DIO消息(违反RPL规则),接着,Node(B)离开DODAG,并以较低的Rank重新加入,取Node(A)和Node(C)为DODAG父代。现在,Node(B)比Nodes(A)和Node(C)都深,并且满足于有2个DAG父代。

- 接着，Node (C)，因为它也贪婪，离开并重新以更深的身份加入，再次获得 2 个父代并有比这 2 个父代更低的 Rank。
- 接下来，Node (B)再次离开并重新以更深的身份加入，再次获得 2 个父代。
- Node (C)也离开，并以更深的身份加入。
- 此过程将重复，DODAG 将在图 3-2 和图 3-3 之间振荡，直到节点计数到无穷大并再次重启循环。
- 通过 RPL 中的机制可避免这个循环：
 - * Node (B)和 Node (C)停留在足以附着到它们最优先父代(A)的 Rank 上，不再寻找任何更深(错误)的替代父代(不贪婪的节点)。
 - * Node (B)和 Node (C)不处理来自比它们自己更深的节点(因为这样的节点可能位于它们自己的子 DODAGs 内)的 DIO 消息。

在第 8-2-2-4 节对这些机制有更多介绍。

3-7-2 DODAG 环路

当节点从 DODAG 分离，又附着到该节点先前的子 DODAG 内的设备时，可能发生 DODAG 环路。实践中，这可能在丢失 DIO 消息时发生。严格使用 DODAGVersionNumber 能够消除这类环路，但是当使用某些本地修复机制时可能遇到这类环路。

例如，考虑本地修复机制(该机制使节点能够脱离 DODAG)，通告 INFINITE_RANK 的 Rank(以便抑制它的路由/通知它的子 DODAG)，接着重新附着到该 DODAG。在这些情况中的有些情况，节点可能重新附着到它自己先前的子 DODAG，引起 DODAG 环路，因为如果在 LLN 环境中 INFINITE_RANK 通告丢失，抑制可能失败。(在此情况，基于 Rank 的数据路径验证机制最终将检测和触发环路校正)。

3-7-3 DAO 环路

如果父代有根据从子代接收和处理的 DAO 消息设置的路由，但是该子代随后清除了相关的 DAO 状态，可能导致出现 DAO 环路。当 No-Path(DAO 消息，它废除了先前宣布的前缀，参阅第 6-4-3 节)被遗漏，并一直持续到所有状态被清除时，这个环路发生。RPL 包括确认 DAO 消息的可选机制，该机制可以减轻遗漏单个 DAO 消息的影响。RPL 包括环路检测机制，该机制能减轻 DAO 环路的影响并触发 DAO 环路修复。(参阅第 11-2-2-3 节)

第 4 章 RPL 支持的流量

RPL 支持 3 种基本流量：多点到点(MP2P)、点到多点(P2MP)和点到点(P2P)。

4-1 多点到点流量

在许多 LLN 应用([RFC5867]、[RFC5826]、[RFC5673]和[RFC5548])中，多点到点(MP2P)是主导流量。MP2P 流的目的地是特定的、有某种应用重要性的节点，如提供到更大的互联网或核心专有 IP 网络的连接。通过允许经 DODAG 根到达 MP2P 目的地，RPL 支持 MP2P 流量。

4-2 点到多点流量

点到多点(P2MP)是几类 LLN 应用([RFC5867]、[RFC5826]、[RFC5673]和[RFC5548])要求的流量模式。使用目的地通告机制(该机制提供朝向目的地(前缀、地址或多播组)，以及离开根的向下(Down)路由)，RPL 支持 P2MP 流量。当基础 DODAG 拓扑结构改变时，目的地通告可以更新路由表。

4-3 点到点流量

RPL DODAG 提供点到点(P2P)流量基本结构。对于支持 P2P 流量的 RPL 网络，根必须能够路由分组到目的地。网络中的节点也可以有到目的地的路由表。分组流向根，直到它到达知道去目的地的路由的祖先。正如本文档后面指出的，在最受限情况(节点不能保存路由)，那个公共祖先可以是 DODAG 根。在其他情况，公共祖先可能是既距离源，又距离目的地较近的节点。

RPL 也支持 P2P 目的地是“一跳”邻居的情况。

RPL 既没有规定也没有排除其他机制，这些机制可能用于计算和安装支持任意 P2P 流量的更佳路由。

第 5 章 RPL 实例

在给定的 LLN 内，可能有多个、逻辑上独立的 RPL Instances。RPL 节点可以属于多个 RPL Instances，RPL 节点在有些 RPL Instances 中可以充当路由器，在其他 RPL Instances 中仅作为叶节点。本文档介绍单个实例如何运行。

有两类 RPL Instances：Local 和 Global。RPL 在 Global 实例和 Local 实例间划分 RPLInstanceID 空间，以便不但可以提供协同的 RPLInstanceIDs 分配，也可以提供单向的 RPLInstanceIDs 分配。Global RPL Instances 经过协调，有一个或多个 DODAG，并且通常是长寿命的。Local RPL Instances 总是单个 DODAG，该单个 DODAG 的单一根拥有相应 DODAGID 并以单向方式分配本地 RPLInstanceID。例如，在支持进一步的按需路由解决方案中，Local RPL Instances 可用于构建 DODAGs。Local RPL Instances 的运行模式超出本规范范围，可能在其他伙伴规范中介绍。

有关 RPL Instances 的定义和提供超出本规范范围。指导原则可能涉及应用程序和具体实施，这些预计会在将来的伙伴规范中详细解释。具体操作预计是这样：即来自 RPL 网络外部的数据分组能够无歧义地分配给至少一个 RPL Instance，并在任何匹配该分组的实例上安全路由。

在 RPL 网络中的控制分组和数据分组被加上标记，以便无歧义地标识它们是哪个 RPL Instance 的一部分。

每个 RPL 控制消息有 RPLInstanceID 字段。某些 RPL 控制消息，当如下面定义的指本地 RPLInstanceID 时，也可以包括 DODAGID。

在 RPL 网络中流动的数据分组揭示 RPLInstanceID 为 RPL 要求的 RPL Packet Information 的一部分，进一步介绍参阅第 11-2 节。来自 RPL 网络外部的数据分组，入口路由器决定 RPLInstanceID 并将其放入最终分组，入口路由器将最终分组注入到 RPL 网络。

5-1 RPL 实例 ID

在整个 LLN 内，全局 RPLInstanceID 必须是唯一的。分配和提供全局 RPLInstanceID 的机制超出本规范范围。在整个网络中最多可以有 128 个 Global 实例。Local 实例总是与 DODAGID (或者采用显式，或者在某些情况下采用隐式给出 DODAGID)一起使用，可支持每个 DODAGID 最多 64 个 Local 实例。Local 实例由拥有该 DODAGID 的节点分配和管理，无需与其他节点进行任何显式协调，下面会有更多介绍。

全局 RPLInstanceID 在 RPLInstanceID 字段中编码，如图 4 所示。

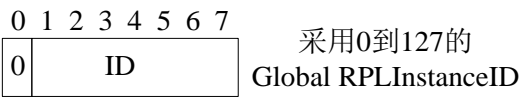


图 4 Global Instances 的 RPLInstanceID 字段格式

本地 RPLInstanceID 由拥有该 DODAGID 的节点自动配置，本地 RPLInstanceID 必须是那个 DODAGID 中唯一的。用于配置本地 RPLInstanceID 的 DODAGID 必须是该节点的可达 IPv6 地址，DODAGID 必须用作那个 Local 实例中所有通信的端点。

本地 RPLInstanceID 在 RPLInstanceID 字段中编码，如图 5 所示。

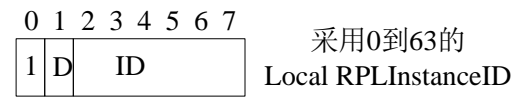


图 5 Local Instances 的 RPLInstanceID 字段格式

在本地 RPLInstanceID 中“D”标记在 RPL 控制消息中总是设置为 0。D 标记用于在数据分组中指出 DODAGID 是分组的源还是分组的目的地。如果“D”标记设置为 1，那么 IPv6 分组的目的地地址必须是 DODAGID。如果“D”标记被清 0，那么 IPv6 分组的源地址必须是该 DODAGID。

例如，考虑 Node A，它是 Local RPL Instance 的 DODAG 根，分配有本地 RPLInstanceID。根据定义，经过那个 Local RPL Instance 的所有流量或起源于或终止于 Node A。在这种情况下，该 DODAGID 是 Node A 的可达 IPv6 地址。所有流量将包括 Node A 的地址，从而包括 DODAGID，无论是在源还是在目的地地址中。因此，通过适当设置“D”标记，本地 RPLInstanceID 可以指出 DODAGID 相当于源地址，或相当于目的地地址。

第 6 章 ICMPv6 RPL 控制消息

本文档定义 RPL 控制消息,即新的 ICMPv6 [RFC4443]消息。RPL 控制消息由代码标识,由取决于该代码的基(base)(以及一系列选项)构成。

大多数 RPL 控制消息有链路范围。唯一例外是在 Non-Storing 模式中的 DAO / DAO-ACK 消息,使用多跳上的单播地址交换这些消息,因而源地址和目的地地址都使用全局或唯一本地地址。对于所有其他 RPL 控制消息,源地址是本地链路地址,目的地地址或是 all-RPL-node 多播地址,或是目的地的本地链路单播地址。all-RPL-node 多播地址是值为 ff02::1a 的新地址。

根据[RFC4443], RPL Control Message 由 ICMPv6 首部及跟在其后的消息体构成。消息体由图 6 所示的消息基和可能的一些选项构成。

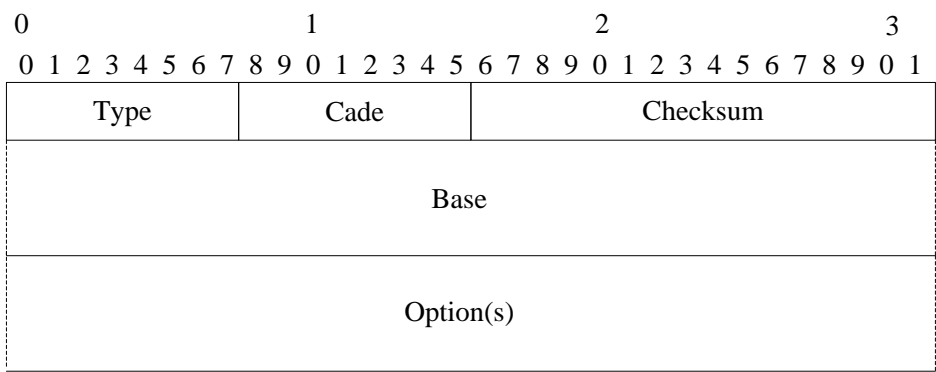


图 6 RPL 控制消息

RPL 控制消息是 Type 为 155 的 ICMPv6 信息消息。

Code 字段指出 RPL 控制消息类型。本文档定义下述 RPL 控制消息类型(参阅第 20-2 节)的代码:

- 0x00: DODAG Information Solicitation (第 6-2 节)
- 0x01: DODAG Information Object(第 6-3 节)
- 0x02: Destination Advertisement Object(第 6-4 节)
- 0x03: Destination Advertisement Object Acknowledgment(第 6-5 节)
- 0x80: Secure DODAG Information Solicitation(第 6-2-2 节)
- 0x81: Secure DODAG Information Object(第 6-3-2 节)
- 0x82: Secure Destination Advertisement Object(第 6-4-2 节)
- 0x83: Secure Destination Advertisement Object Acknowledgment(第 6-5-2 节)
- 0x8A: Consistency Check(第 6-6 节)

如果节点收到带有未知Code字段的RPL控制消息,节点**必须**不做任何进一步处理地抛弃该消息,可以提出管理告警,决不能发送任何响应消息。

按照[RFC4443]中的规定计算校验和。对于下面规定的 RPL 安全运行,校验和设置为 0,一旦包括各安全字段的 RPL 消息内容其余部分都被设置为 1, 计算校验和。

代码的高阶位(0x80)指示 RPL 消息是否启用安全性。安全 RPL 消息有如图 7 所示的支持保密性和完整性的格式。

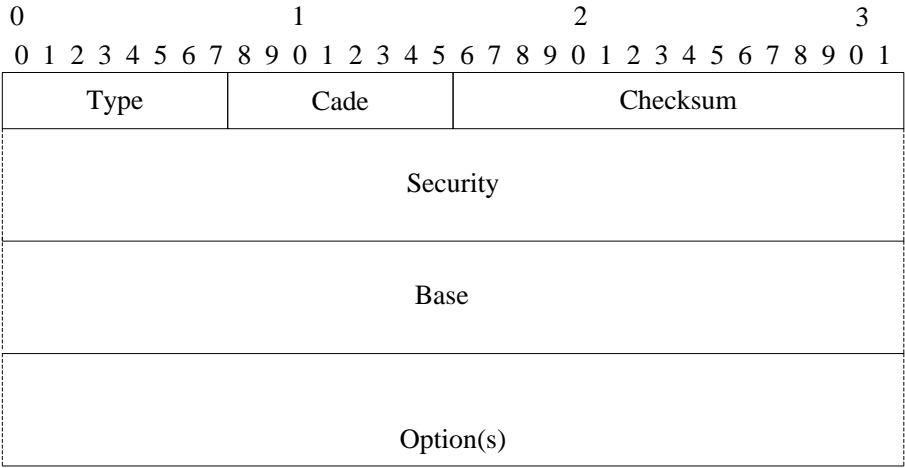


图 7 安全的 RPL 控制消息

本章其余部分介绍目前定义的 RPL 控制消息 Base(在其之后是目前定义的 RPL Control Message 选项)格式。

6-1 RPL 安全字段

每个 RPL 消息有安全变量。这些安全变量提供完整性和重放保护，以及可选的保密性和延迟保护。因为安全措施覆盖基本消息和选项，在安全消息中安全信息位于校验和及基之间，如图 7 所示。

使用的安全等级和算法在下面介绍的协议消息中指出：

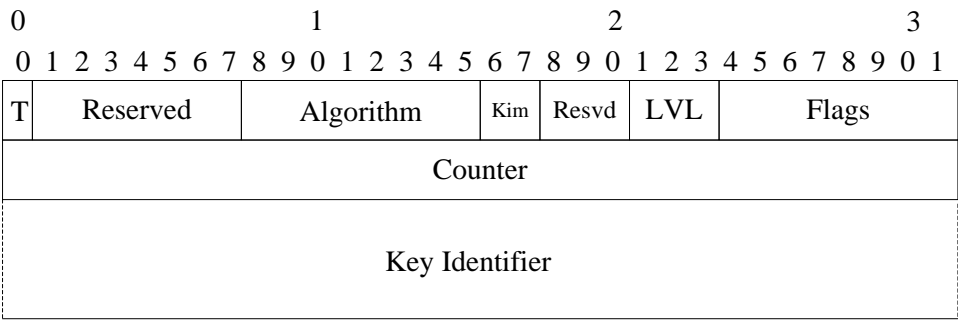


图 8 安全部分

Message Authentication Codes (MACs)和签名对整个无安全的 ICMPv6 RPL 控制消息提供认证，包括带有所有限定字段的 Security 部分，但是采用暂时设置为 0 的 ICMPv6 校验和。加密提供安全 RPL ICMPv6 消息的保密，安全 RPL ICMPv6 消息开始于 Security 部分后的第一个字节，一直持续到分组的最后一个字节。此安全性转换产生包括多个加密字段(MAC、签名，等)的安全 ICMPv6 RPL 消息。换句话讲，安全性转换自身(例如，使用的 Signature 和/或 Algorithm)详细说明如何把加密的字段整合进安全分组。Security 部分自身并没有显式携带这些加密的字段。应用 Security 部分的更详细介绍参阅第 19 章和第 10 章。

Counter 是 Time (T): 如果计数器的 Time 标记设置为 1，那么 Counter 字段是时间戳。如果该标记清零，那么计数器是递增计数器。第 10-5 节详细介绍“T”标记和 Counter 字段。

Reserved: 7 位未使用字段。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

Security Algorithm (Algorithm): Security Algorithm 字段规定网络使用的加密、MAC 和签名方案。这个字段支持的值如图 9 所示。

Algorithm	Encryption/MAC	Signature
0 1-255	带AES—128的CCM 未分配	带SHA-256的RSA 未分配

图 9 安全算法编码

第 10-9 节详细介绍该算法。

Key Identifier Mode (KIM): Key Identifier Mode 是 2 位字段，该字段指出用于分组保护的密钥是显式确定，还是隐式确定，以及指出 Key Identifier 字段的具体表示。Key Identifier Mode 的取值为图 10 给出各值之一。

Mode	KIM	含意	密钥标识符长度 (字节)
0	00	使用组密钥。 由Key Index字段确定的密钥。 Key Source不存在。 Key Index存在。	1
1	01	使用配对密钥。 由分组的源和目的地确定的密钥。 Key Source不存在。 Key Index不存在。	0
2	10	使用组密钥。 由Key Index和Key Source Identifier确定的密钥。 Key Source存在。 Key Index存在。	9
3	11	使用节点的签名密钥。 如果分组被加密, 分组使用组密钥、 Key Index和Key Source规定密钥。 Key Source可能存在。 Key Index可能存在。	0/9

图 10 Key Identifier Mode (KIM)编码

在 Mode 3 (KIM=11)中, Key Source 和 Key Identifier 的存在与否取决于下面介绍的 Security Level (LVL)。如果 Security Level 指出使用了加密, 那么这些字段存在; 如果它指出没有使用加密, 那么这些字段不存在。

Resvd: 3 位未使用字段。该字段**必须**由发送者初始化为 0, 接收者**必须**忽略该字段。

Security Level (LVL): Security Level 是 3 位字段，指出提供的分组保护。这个值可在每个分组基础上调整，可用于改变数据认证等级，以及作为选项可用于数据保密。

KIM 字段指出签名是否使用以及 Level 字段的含意。注意，分配的 Security Level 值不一定被排序---更高的 LVL 值未必等同于安全性增加。Security Level 的值为图 11 给出各值之一。

KIM = 0, 1, 2		
LVL	属性	MAC 长度
0	MAC-32	4
1	ENC-MAC-32	4
2	MAC-64	8
3	ENC-MAC-64	8
4-7	未分配	N/A

KIM = 3		
LVL	属性	Sign 长度
0	Sign-3072	384
1	ENC-Sign-3072	384
2	Sign-2048	256
3	ENC-Sign-2048	256
4-7	未分配	N/A

图 11 Security Level (LVL)编码

MAC 属性指出消息有规定长度的 MAC。ENC 属性指出消息被加密。Sign 属性指出消息有规定长度的签名。

Flags: 保留用于标记的 8 位未使用字段。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

Counter: Counter 字段指出用于构成加密机制的非重复 4 字节值，该机制实现分组保护，以及可用于提供语义安全。参阅第 10-9-1 节。

Key Identifier: Key Identifier 字段指出使用哪个密钥保护分组。这个字段提供不同粒度水平的分组保护，包括对等密钥、组密钥和签名密钥。这个字段表示的内容与 Key Identifier Mode 字段指出的内容相同，该字段格式如图 12 所示。

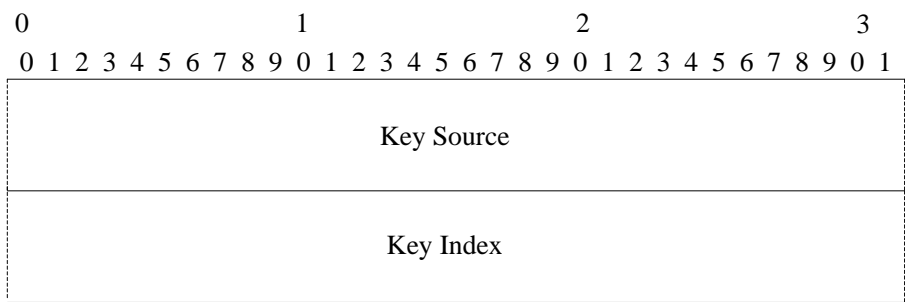


图 12 Key Identifier

Key Source: Key Source 字段，如果存在，指出组密钥发起者的逻辑标识符。如果存在，这个字段有 8 字节长。

Key Index: Key Index 字段，如果存在，允许唯一标识有相同发起者的不同密钥。每个密钥发起者有责任确保它发布的主动使用密钥有不同密钥索引，以及确保所有密钥索引有不等于 0x00 的值。0x00 值保留用于预装的、共享的密钥。如果存在，这个字段有 1 字节长。

Security 部分的未分配位保留。在发送端它们**必须**被设置为 0，在接收端**必须**忽略它们。

6-2 DODAG 信息请求(DIS)

DODAG Information Solicitation (DIS)消息可用于向 RPL 节点请求 DODAG Information Object。它的用途类似在 IPv6 Neighbor Discovery 中规定的 Router Solicitation；节点可使用 DIS 探查它附近 DODAGs 的邻居。第 8-3 节介绍节点如何响应 DIS。

6-2-1 DIS 基本对象格式

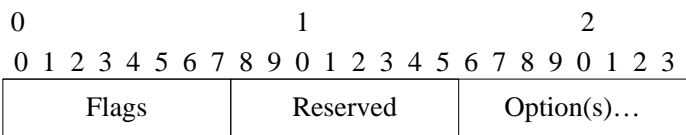


图 13 DIS Base Object

Flags: 保留用于标记的 8 位未使用字段。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

Reserved: 8 位未使用字段。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。
DIS Base 的未分配位保留。在发送端它们**必须**被设置为 0，在接收端**必须**忽略它们。

6-2-2 安全 DIS

Secure DIS 消息遵循图 7 中的格式，在图 7 中基本格式是图 13 中显示的 DIS 消息。

6-2-3 DIS 选项

DIS 消息**可以**携带合法选项。

本规范允许 DIS 消息携带下述选项：

- 0x00 Pad1
- 0x01 PadN
- 0x07 Solicited Information

6-3 DODAG 信息对象(DIO)

DODAG Information Object 携带使节点能够发现 RPL Instance、了解节点自己的配置参数、选择 DODAG 父代集，以及维护 DODAG 的信息。

6-3-1 DIO 基本对象格式

DIO Base Object 格式如图 14 所示。

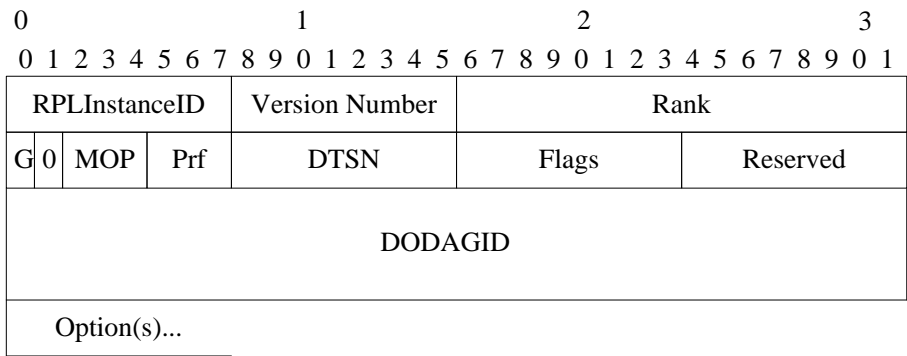


图 14 DIO Base Object

Grounded (G): Grounded“G”标记指出通告的 DODAG 是否能够满足限定应用的目标。
如果该标记置 1，DODAG 是接地的。如果该标记置 0，DODAG 是浮动的。

Mode of Operation (MOP): Mode of Operation (MOP)字段指出 RPL Instance 运行模式在管理上由 DODAG 根提供，并由 DODAG 根分发。所有加入 DODAG 的节点必须能够兑现 MOP，以便作为路由器完全参与，不然它们必须仅作为叶节点参加。MOP 编码格式如图 15 所示。

MOP	描述
0	没有由RPL维护的下行(Downward)路由
1	Non-Storing Mode运行
2	采用无多播支持的Storing Mode运行
3	采用有多播支持的Storing Mode运行
	所有其他值没有分配

值0表示目的地通告消息被禁用，DODAG仅维持上行 (Upward)路由。

图 15 Mode of Operation (MOP)编码

DODAGPreference (Prf): 3 位无符号整数，它定义这个 DODAG 根相比该实例中其他 DODAG 根有多优先。DAGPreference 的范围从 0x00(最低优先)到 0x07(最高优先)。默认值是 0(最低优先)。关于 DAGPreference 如何影响 DIP 处理，参阅第 8-2 节。

Version Number: 由DODAG根设置的相对于DODAGVersionNumber的8位无符号整数。第8-2节介绍DODAGVersionNumbers规则，以及这些规则如何影响DIO处理。

Rank: 16 位无符号整数，指出发送 DIO 消息的节点的 DODAG Rank。第 8-2 节介绍如何设置 Rank，以及 Rank 如何影响 DIO 处理。

RPLInstanceID: 由 DODAG 根设置的 8 位字段，该字段指出 DODAG 是哪个 RPL Instance 的一部分。

Destination Advertisement Trigger Sequence Number (DTSN): 由发布 DIO 消息节点设置的 8 位无符号整数。DTSN 标记用作维护下行(Downward)路由流程的一部分。这个处理在第 9 章详细介绍。

Flags: 保留用于标记的 8 位未使用字段。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

Reserved: 8 位未使用字段。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

DODAGID: 由 DODAG 根设置的 128 位 IPv6 地址, 该地址唯一标识 DODAG。DODAGID 必须是属于该 DODAG 根的、路由可达的 IPv6 地址。

DIO Base 的未分配位保留, 在发送端它们**必须**被设置为 0, 在接收端**必须**忽略它们。

6-3-2 安全 DIO

Secure DIO 消息遵循图 7 中的格式, 图 7 中的基本格式是图 14 中显示的 DIO 消息。

6-3-3 DIO 选项

DIO 消息可以携带合法选项。

本规范允许 DIO 消息携带下述选项:

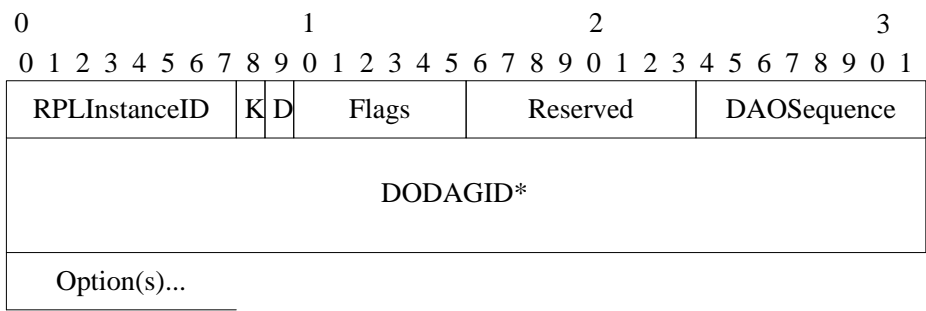
- 0x00 Pad1
- 0x01 PadN
- 0x02 DAG Metric Container
- 0x03 Routing Information
- 0x04 DODAG Configuration
- 0x08 Prefix Information

6-4 目的地通告对象(DAO)

Destination Advertisement Object (DAO)用于沿 DODAG 上行(Upward)传播目的地信息。在 Storing 模式中, DAO 消息由子代向选择的父代(们)单播。在 Non-Storing 模式中, DAO 消息向 DODAG 根单播。根据显式请求或出错, DAO 消息可选择由它的目的地, 采用 Destination Advertisement Acknowledgement (DAO-ACK)消息, 返回给该 DAO 消息的发送者一个确认。

6-4-1 DAO 基本对象格式

DAO Base Object 格式如图 16 所示。



“*” 表示DODAGID不总是存在，如下所述。

图 16 DAO Base Object

RPLInstanceID: 8 位字段，指示从 DIO 中了解到的、与 DODAG 有关的拓扑实例。

K: “K” 标记表示希望接收者返回 DAO-ACK。(参阅第 9-3 节)

D: “D” 标记表示 DODAGID 字段存在。如果使用本地 RPLInstanceID，这个标记**必须**置 1。

Flags: 在 Flags 字段中保留用于标记的 6 个未使用位。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

Reserved: 8 位未使用字段。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

DAOSequence: 随着来自节点的每个唯一 DAO 消息递增，用 DAO-ACK 消息回应。

DODAGID(可选): 由 DODAG 根设置的 128 位无符号整数，唯一标识 DODAG。仅当 “D” 标记置 1 时这个字段存在。一般当使用本地 RPLInstanceID，以便标识与该 RPLInstanceID 关联的 DODAGID 时，这个字段才存在。如果使用全局 RPLInstanceID，不需要这个字段。DAO Base 的未分配位保留。在发送端它们**必须**被设置为 0，在接收端**必须**忽略它们。

6-4-2 安全 DAO

Secure DAO 消息遵循图 7 中的格式，图 7 中的基本格式是图 16 中显示的 DAO 消息。

6-4-3 DAO 选项

DAO 消息可以携带合法选项。

本规范允许 DAO 消息携带下述选项：

- 0x00 Pad1
- 0x01 PadN
- 0x05 RPL Target
- 0x06 Transit Information
- 0x09 RPL Target Descriptor

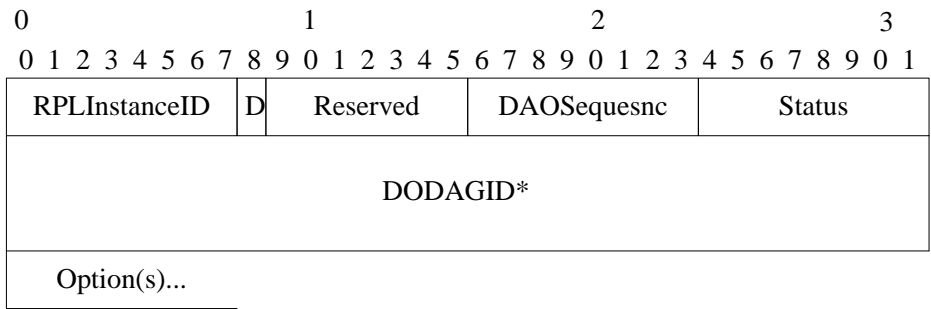
DAO 消息的特例，称作 No-Path，用于在 Storing 模式中清除下行(Downward)路由状态 (该路由状态已经通过 DAO 操作提供)。No-Path 携带 Target 选项和关联的 Transit Information 选项，其生存期的值为 0x00000000，指出不可到达那个 Target。

6-5 目的地通告对象确认(DAO-ACK)

DAO-ACK 消息作为单播分组，由 DAO 接收者(DAO 父代或 DODAG 根)发送，用以响应单播 DAO 消息。

6-5-1 DAO-ACK 基本对象格式

DAO-ACK Base Object 格式如图 17 所示。



“*” 表示DODAGID不总是存在，如下面所述。

图 17 DAO-ACK Base Object

RPLInstanceID: 8 位字段，指出从 DIO 了解到的、与 DODAG 关联的拓扑实例。

D: “D” 标记指出 DODAGID 字段存在。一般仅在使用本地 RPLInstanceID 时置 1。

Reserved: 7 位字段，保留用于标记。

DAOSequence: 随着来自节点的每个 DAO 消息递增，由接收者用 DAO-ACK 回应。
DAOSequence 用于相关 DAO 消息和 DAO ACK 消息,DAOSequence 不会与 Transit Information option Path Sequence 混淆，后者与给定的、沿 DODAG 向下(Down)的 Target 关联。

Status: 指示完成情况。状态 0 在本规范中定义为无条件接受。其余状态值保留作为拒绝代码。虽然状态代码应当按照下述指导原则在今后的规范中分配，本规范没有定义拒绝状态代码。

0: 无条件接受(即，不拒绝接收 DAO-ACK 的节点)。

1-127: 不是断然拒绝; 发送 DAO-ACK 的节点愿意充当父代, 但是建议接收节点寻找和使用替代父代。

127-255: 拒绝; 发送 DAO-ACK 的节点不愿意充当父代。

DODAGID(可选): 由 DODAG 根设置的 128 位无符号整数, 它唯一标识 DODAG。仅当 “D” 标记设置为 1 时这个字段才存在。一般当使用本地 RPLInstanceID, 以标识与该 RPLInstanceID 关联的 DODAGID 时, 这个字段才存在。如果使用全局 RPLInstanceID, 不需要这个字段。

DAO-ACK Base 的未分配位保留。在发送端它们**必须**被设置为 0, 在接收端**必须**忽略它们。

6-5-2 安全 DAO-ACK

安全 DAO-ACK 消息遵循图 7 中的格式, 在图 7 中的基本格式是在图 17 中显示的 DAO-ACK 消息。

6-5-3 DAO-ACK 选项

本规范没有定义任何由 DAO-ACK 消息携带的选项。

6-6 一致性检验(CC)

Consistency Check (CC)消息用于检验安全消息计数器和发出质询-响应。CC 消息**必须**作为安全 RPL 消息发送。

6-6-1 CC 基本对象格式

CC 基本对象格式如图 18 所示。

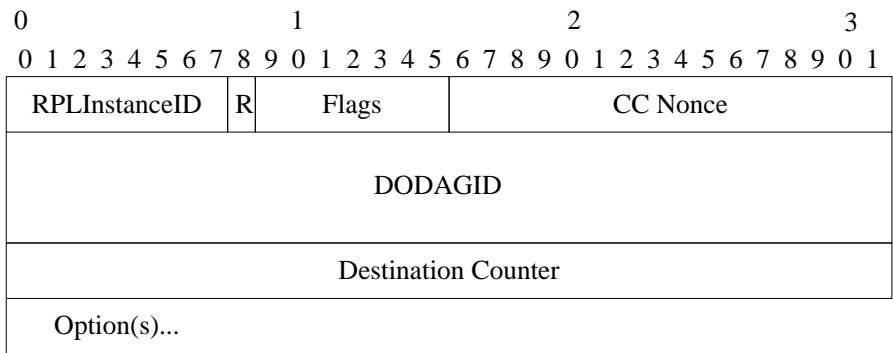


图 18 CC Base Object

RPLInstanceID: 8 位字段，指出从 DIO 了解到的、与 DODAG 关联的拓扑实例。

R: “R” 标记指出 CC 消息是否是响应。带 “R” 标记置 0 的消息是请求；带 “R” 标记置 1 的消息是响应。

Flags: 在 Flags 字段中保留用于标记的 7 个未使用位。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

CC Nonce: 由 CC 请求设置的 16 位无符号整数。相应的 CC 响应包括与请求一样的 CC 随机数值。

DODAGID: 128 位字段，包含 DODAG 根的标识符。

Destination Counter: 32 位无符号整数值，指出发送者对目前目的地安全计数器值的估计。如果发送者没有做出估计，发送者**应当**设置 Destination Counter 字段为 0。

CC Base 的未分配位保留。在发送端它们**必须**被设置为 0，在接收端**必须**忽略它们。

Destination Counter 值使新的或恢复的节点能够通过交换 CC 消息重新同步。这很重要，它能确保即使是在 Counter 状态丢失故障恢复设备事件中，给定的安全密钥的 Counter 值不重复。例如，在那里，在消息 Security 部分收到有初始化计数器的 CC 请求或其他 PRL 消息，在 CC 响应消息中提供的 Incoming Counter 使得请求节点能够重新设置它的 Outgoing Counter 到比响应节点收到的最后一个值大的值；Incoming Counter 也将根据收到的 CC 响应更新。

6-6-2 CC 选项

本规范允许 CC 消息携带下述选项：

- 0x00 Pad1
- 0x01 PadN

6-7 RPL 控制消息选项

6-7-1 RPL 控制消息选项通用格式

所有 RPL Control Message 选项都遵循图 19 所示格式。

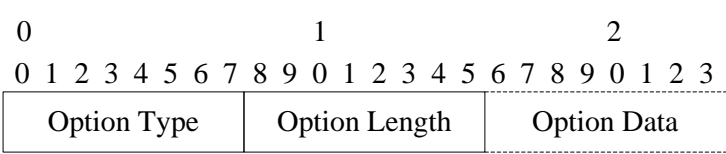


图 19 RPL Option 通用格式

Option Type: 选项类型的 8 位标识符。由 IANA 分配 Option Type 值(参阅第 20-4 节)。

Option Length: 8 位无符号整数，表示用八位位组表示的选项长度，不包括 Option Type 字段和 Length 字段。

Option Data: 可变长度字段，它包括为针对该选项的特定数据。

当处理包含选项的 RPL 消息，而接收者不能识别该选项的 Option Type 值时，接收者**必须**静默忽略不能识别的选项并继续处理接下来的选项，正确处理该消息中任何其余选项。

RPL 消息选项可能有对齐要求。遵循 IPv6 惯例，有对齐要求的选项在分组中对齐，以便每个选项的 Option Data 字段中多个八位位组值落在自然边界上(即，宽度为 n 个八位位组的字段被放置在自首部开始的 n 个八位位组的整数倍上，这里 n = 1、2、4 或 8)。

6-7-2 Pad1

Pad1 选项可以出现在 DIS、DIO、DAO、DAO-ACK 和 CC 消息中，其格式如图 20 所示。

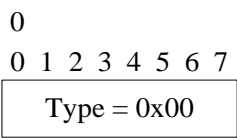


图 20 Pad1 选项格式

Pad1 选项用于将单个填充八位位组插入消息中，以便对齐选项。如需要不止一个填充八位位组，应当使用 PadN 选项，而不是使用多个 Pad1 选项。

注意！Pad1 选项格式是特殊情况---它既没有 Option Length 字段，又没有 Option Data 字段。

6-7-3 PadN

PadN 选项可以出现在 DIS、DIO、DAO、DAO-ACK 和 CC 消息中，其格式如图 21 所示。

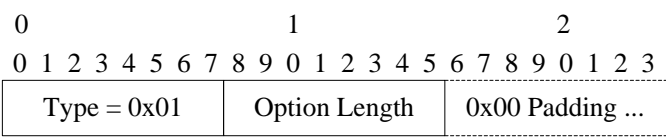


图 21 PadN 选项格式

PadN 选项用于在消息中插入两个或多个填充八位位组，以便对齐选项。接收者**必须**忽略 PadN 选项数据。

Option Type: 0x01

Option Length: 对于 N 个填充八位位组，其中 $2 \leq N \leq 7$ ，Option Length 字段包括 N-2 值。Option Length 为 0 表示 2 个八位位组的总填充。Option Length 为 5 表示 7 个八位位组的总填充，这是采用 PadN 选项允许的最大填充尺寸。

Option Data: 对于 N(N>1)个填充八位位组，Option Data 包含 N-2 个 0 值八位位组。

6-7-4 DAG 度量容器

DAG Metric Container 选项可以出现在 DIO 或 DAO 消息中,DAG Metric Container 选项的格式如图 22 所示。

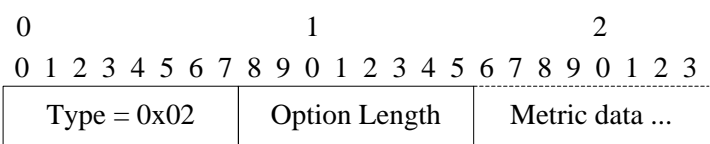


图 22 DAG Metric Container 选项格式

DAG Metric Container 用于报告沿 DODAG 的度量。根据实施者选择，DAG Metric Container 可以包括许多在[RFC6551]中规定的独立节点、链路、聚合路径度量和约束。

在同一个 RPL 控制消息中，DAG Metric Container 可以不止一次出现，例如，为了适应使用长于 256 字节 Metric Data 的情况。更多信息参阅[RFC6551]。

DAG Metric Container 的处理和传播要受实施的特定策略功能支配。

Option Type: 0x02

Option Length: Option Length 字段包含以八位位组为单位的 Metric Data 长度。

Metric Data: DAG Metric Container 数据的阶、内容和编码在[RFC6551]中规定。

6-7-5 路由信息

Route Information Option (RIO)可以出现在 DIO 消息中,它携带与在[RFC4191]中定义的 IPv6 Neighbor Discovery (ND) RIO 相同的信息。DODAG 的根是设置那条信息的官方,当沿 DODAG 向下传播时该信息不改变。RPL 路由器可以简单地将该信息转换回 ND 选项,以便在它自己的 RAs 中通告,所以附着在该 RPL 路由器的节点,将停止使用被根认为是到达分组目的地最好的那个 DODAG。除了现有的 ND 语义以外, Objective Function 有可能使用这条倾向于某 DODAG(该 DODAG 的根是特定目的地首选)的信息。为了作为 RPL 选项携带,此选项格式稍微做了修改(Type, Length, Prefix),参阅图 23。

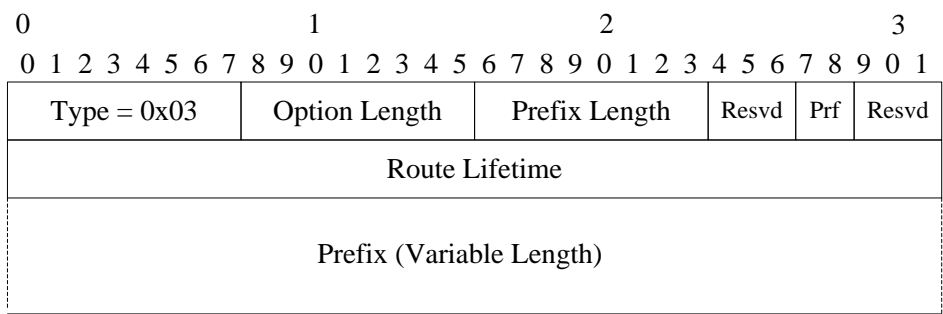


图 23 Route Information Option (RIO)格式

RIO 用于指出可从 DODAG 根连接到指定的目的地前缀。

如果 RPL 控制消息需要规定连接到一个以上的目的地,可以重复该 RIO。

就 RIO 而言,应当查询作为权威参考文献的[RFC4191]标准。这里为了方便,转录该字段说明如下:

Option Type: 0x03

Option Length: 可变, 以八位位组为单位、不包括 Type 字段和 Length 字段的选项长度。
注意, 这个长度以单个八位位组为单位表示, 与 IPv6 ND 不同。

Prefix Length: 8 位无符号整数。合法前缀中前导位数目。取值范围为 0 到 128。Prefix 字段有根据 Option Length 字段推导出的字节数, 并且必须至少是 Prefix Length。
注意, 在 RPL 中, 这意味着 Prefix 字段可以有除了 0、8 或 16 以外的长度。

Prf: 2 位带符号整数。Route Preference 指出当收到多个相同前缀(代表不同路由器)时, 是否相比其他路由器, 更倾向与这个前缀关联的路由器。如果收到 Reserved (10) 值, 必须**忽略**该 RIO。按照[RFC4191], **不允许**发送 Reserved (10)值。([RFC4191] 把 Preference 限制在仅仅 3 个值, 以便强调 Preference 不是度量。)

Resvd: 3 位未使用字段。该字段**必须**由发送者初始化为 0, 接收者**必须**忽略该字段。

Route Lifetime: 32 位无符号整数。以秒为单位的时间长度(相对于发送分组时刻), 在此时间长度内前缀适合路由确定。全 1 的位值(0xFFFFFFFF)表示无穷大。

Prefix: 可变长度字段, 包含 IP 地址或 IPv6 地址前缀。Prefix Length 字段包含前缀中有效前导位的数目。在前缀中前缀长度后的位(如果有)保留, **必须**由发送者初始化为 0, 接收者**必须**忽略。注意, 在 RPL 中, 这个字段可以有除了 0、8 或 16 以外的长度。

RIO 的未分配位保留。在发送端它们**必须**被设置为 0, 在接收端**必须**忽略它们。

6-7-6 DODAG 配置

DODAG Configuration 选项可以出现在 DIO 消息中，其格式如图 24 所示。

0																1																2																3															
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																																
Type = 0x04								Opt Length = 14								Flags				A	PCS				DIOIntDoubl.																																						
DIOIntMin.								DIORedun.								MaxRankIncrease																																															
MinHopRankIncrease																OCP																																															
Reserved								Def. Lifetime								Lifetime Unit																																															

图 24 DODAG Configuration 选项格式

DODAG Configuration 选项用于通过 DODAG 分发 DODAG Operation 配置信息。

在这个选项中传递的信息通常是静态的，并且在 DODAG 内不变，因此，在每个 DIO 中包括该信息不是必须的。这条信息在 DODAG 根配置，用 DODAG Configuration 选项分发到整个 DODAG。传播 DODAG Configuration 选项时，DODAG 根以外的节点不允许修改这条信息。偶尔，DODAG 根可以包括这个选项(如由 DODAG 根决定)，在对单播请求的响应中必须包括此选项，例如，单播 DODAG Information Solicitation (DIS) 消息。

Option Type: 0x04

Option Length: 14

Flags: Flags 字段中 4 个未使用位保留用于标记。该字段必须由发送者初始化为 0，接收者必须忽略该字段。

Authentication Enabled (A): 描述网络安全模式的 1 位标记。该位描述节点作为路由器加入网络前是否必须向密钥当局进行身份认证。如果 DIO 不是安全 DIO，“A”位必须置 0。

Path Control Size (PCS): 3 位无符号整数，用于配置可能分配给 Path Control 字段(参阅第 9-9 节)的位数。注意，当查询 PCS 以确定 Path Control 字段宽度时，值加 1，即，PCS 值为 0 导致 Path Control 字段中 1 个激活位。PCS 的默认值是 DEFAULT_PATH_CONTROL_SIZE。

DIOIntervalDoublings: 8 位无符号整数，用于配置 DIO Trickle 计时器(参阅第 8-3-1 节)的 I_{max} 。DIOIntervalDoublings 的默认值是 DEFAULT_DIO_INTERVAL_DOUBLINGS。

DIOIntervalMin: 8 位无符号整数，用于配置 DIO Trickle 计时器(参阅第 8-3-1 节)的 I_{min} 。DIOIntervalMin 的默认值是 DEFAULT_DIO_INTERVAL_MIN。

DIORedundancyConstant: 8 位无符号整数，用于配置 DIO Trickle 计时器(参阅第 8-3-1 节)的 k 。DIORedundancyConstant 的默认值是 DEFAULT_DIO_REDUNDANCY_CONSTANT。

MaxRankIncrease: 16 位无符号整数，用于配置 DAGMaxRankIncrease，即为支持本地修复而允许的 Rank 递增。如果 DAGMaxRankIncrease 为 0，禁用这个机制。

MinHopRankIncrease: 16 位无符号整数，用于配置 MinHopRankIncrease，如第 3-5-1 节所述。MinHopRankInc 的默认值是 DEFAULT_MIN_HOP_RANK_INCREASE。

Objective Code Point (OCP): 16 位无符号整数。OCP 字段标识 OF，并由 IANA 管理。

Reserved: 7 位未使用字段。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

Default Lifetime: 8 位无符号整数。这个生存期用做所有 RPL 路由的默认值。它以 Lifetime Units 为单位表示，例如，以秒为单位的默认生存期是 $(\text{Default Lifetime}) \times (\text{Lifetime Unit})$ 。

Lifetime Unit: 16 位无符号整数。提供以秒计的单位，用于表示 RPL 中的路由生存期。对于非常稳定的网络，Lifetime Unit 可以是数小时到数天。

6-7-7 RPL 目标

RPL Target 选项可以出现在 DAO 消息中，其格式如图 25 所示。

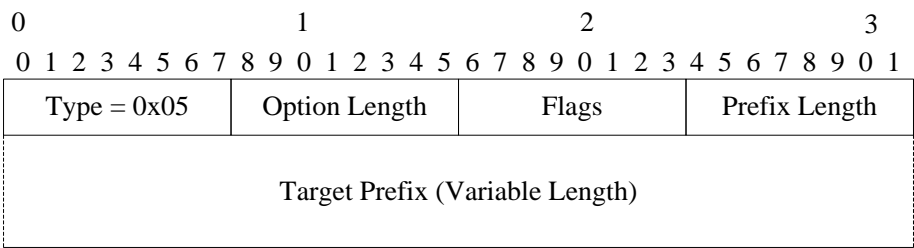


图 25 RPL Target 选项格式

RPL Target 选项用于指出 Target IPv6 address、前缀或者可到达的或沿 DODAG 查询的多播组。在 DAO 中，RPL Target 选项指出可达性。

RPL Target 选项可以选择与符合目标的 RPL Target Descriptor 选项(参见图 30)配对。

在 DAO 消息中，一组的一个或多个 Transit Information 选项(参阅第 6-7-8 节)可以直接跟在一组的一个或多个 Target 选项之后(在 DAO 消息中，每个 Target 选项可以与 RPL Target Descriptor 选项配对，如上所述)。DAO 消息结构，详述 Target 选项如何与 Transit Information 选项联合使用的更多内容参阅第 9-4 节。

RPL Target 选项如果需要可以重复，以便指示多个目标。

Option Type: 0x05

Option Length: 可变，以八位位组为单位、不包括 Type 字段和 Length 字段的选项长度。

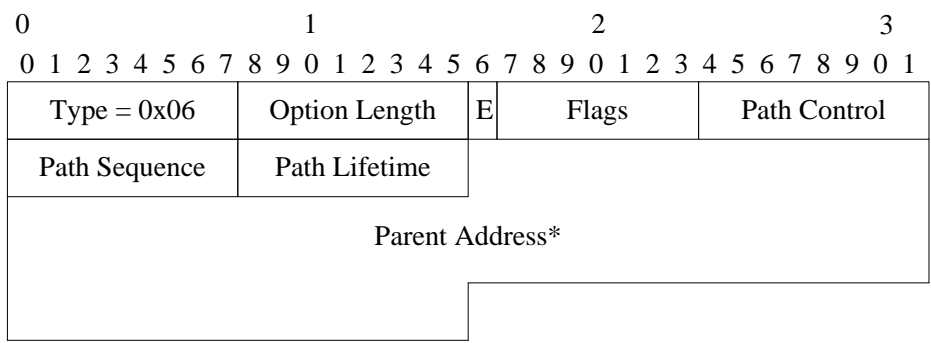
Flags: 保留用于标记的 8 位未使用字段。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

Prefix Length: 8 位无符号整数。IPv6 Prefix 中有效前导位数目。

Target Prefix: 可变长度字段，用于指出 IPv6 目的地地址、前缀或多播组。**Prefix Length** 字段包括前缀中有效前导位数目。在前缀中前缀长度后的位(如果有)保留，并且在发送端它们**必须**被设置为 0，在接收端**必须**忽略它们。

6-7-8 传送信息

Transit Information 选项可以出现在 DAO 消息中，它的格式如图 26 所示。



“*” 表示DODAG Parent Address子字段不一定总存在，如下所述。

图 26 Transit Information 选项格式

Transit Information 选项用于节点，以指出到一个或多个目的地的路径的属性。由紧靠 Transit Information option(s)前的一个或多个 Target 选项指出该目的地。

Transit Information 选项可用于节点，以指出到达祖先的它的那些 DODAG 父代，该祖先正在收集 DODAG 路由信息，一般是为了构建源路由。在 Non-Storing 运行模式中，这个祖先是 DODAG 根，并且这个选项由 DAO 消息携带。在 Storing 运行模式中，不需要 DODAG Parent Address 子字段，因为 DAO 消息被直接发送到父代。此选项长度用于确定 DODAG Parent Address 子字段是否存在。

有不只一个 DAO 父代的非存储节点可以包括每个 DAO 父代的 Transit Information 选项，作为非存储目的地通告操作的一部分。该节点可以在不同 DAO 父代组间分发 Path Control 字段中的位，从而通报父代间的优先权。当为构建下行(Downward)路由在替代父代/路径间进行选择时，优先权可能影响 DODAG 根的决定。

一个或多个 RPL Target 选项**必须**位于一个或多个 Transit Information 选项前。采用这种方法，RPL Target 选项指出子代节点，Transit Information 选项(们)列出 DODAG 父代。此 DAO 消息结构，Target 选项如何与 Transit Information 选项联合使用的更多介绍，参阅第 9-4 节。

一般非存储节点使用多个 Transit Information 选项，并发送因此直接形成的到根的 DAO 消息。一般存储节点使用一个没有父代字段的 Transit Information 选项，并发送因此形成的、对稍后详细介绍的 Path Control 有额外调整的、到一个或多个父代的 DAO 消息。

例如，采用Non-Storing运行模式，让Tgt(T)表示Target T的Target选项。让Trnst(P)表示包含父代地址P的Transit Information选项。考虑非存储Node N，Node N通告自己拥有的目标N1和N2，它有父代P1、P2和P3。在这种情况下，预期该DAO消息包括序列((Tgt(N1)、Tgt(N2))、(Trnst(P1)、Trnst(P2)、Trnst(P3)))，因此，Target选项组{N1, N2}被Transit Information选项描述为有父代{P1, P2, P3}。接着，非存储节点将直接处理那个到DODAG根的DAO消息，并通过父代：P1、P2或P3中的一个转发那个DAO消息。

Option Type: 0x06

Option Length: 可变，取决于 DODAG Parent Address 子字段是否存在。

External (E): 1 位标记。“E” 标记置 1 指出父代路由器把外部目标重新分配到 RPL 网络。外部 Target 是通过备用协议获知的 Target。外部目标罗列在 Target 选项中，这些选项紧靠Transit Information选项前。不指望外部Target支持RPL消息和选项。

Flags: Flags 字段中保留用于标记的 7 个未使用位。该字段**必须**由发送者初始化为 0，接收者**必须**忽略该字段。

Path Control: 8 位字段。Path Control 字段限制 DAO 父代数量，这里的 DAO 父代是指通告到特定目的地的连通性的 DAO 消息可能发送到的 DAO 父代，以及提供一些相对优先级指示的 DAO 消息可能发送到的 DAO 父代。此限制对 LLN 内整个 DAO 消息扇出提出了某种界限。在 Path Control 中位的分配和排序也用于沟通优先权。按照 DODAG Configuration 中的 PCS，不一定所有这些位都启用。Path Control 字段分成 4 个子字段，每个子字段包含 2 位：PC1、PC2、PC3 和 PC4，如图 27 所示。这些子字段按优先级排序，PC1 有最高优先级，PC4 优先级最低。在子字段内，没有优先级排序。通过将父代成组(如在 ECMP 中)和排序父代，父代可与 Path Control 字段中在一定程度上优先沟通的特定位关联。

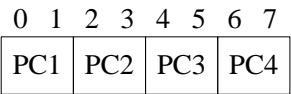


图 27 Path Control Preference 子字段编码

Path Sequence: 8 位无符号整数。当拥有 Target 前缀的节点发布 RPL Target 选项时(即，在 DAO 消息中)，那个节点建立 Path Sequence，并在每次它用被更新的信息发布 RPL Target 选项时递增 Path Sequence。

Path Lifetime: 8 位无符号整数。以 Lifetime Units 为单位的时间长度(从 Configuration 选项获得)，在此时间长度内前缀适合路由确定。当看到新的 Path Sequence 时，周期开始。全 1 位值(0xFF)代表无穷大。全 0 位值(0x00)表示不可达到。在本文档中，DAO 消息被称为 No-Path(对于那个 Target)，该 DAO 消息包含 Target 的 Path Lifetime 为 0x00 的 Transit Information 选项。

Parent Address(可选): 最初发布 Transit Information 选项的节点的 DODAG 父代 IPv6 地址。根据运行的 DODAG 模式(Storing 或 Non-Storing), 可能没有这个字段, 由 Transit Information 选项长度指出。

Transit Information 选项的未分配位保留。在发送端它们**必须**被设置为 0, 在接收端**必须**忽略它们。

6-7-9 请求信息

Solicited Information 选项可以存在于 DIS 消息中，其格式如图 28 所示。

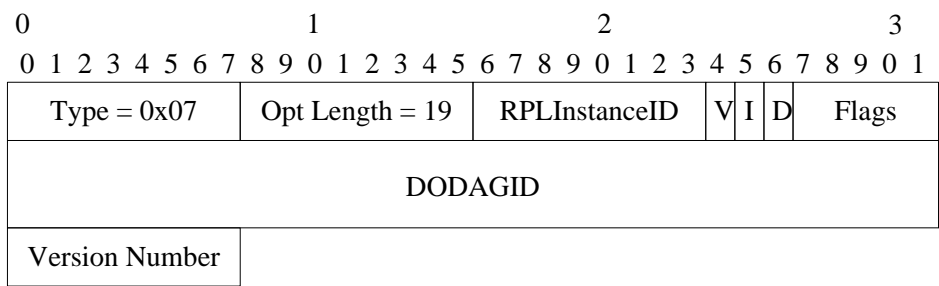


图 28 Solicited Information 选项格式

Solicited Information 选项供节点用于从相邻节点子集请求 DIO 消息。Solicited Information 选项可能规定一些匹配接收节点的属性准则。这可由请求者用于限制来自“非感兴趣”节点的响应数目。这些属性影响节点是否重新设置它的 DIO Trickle 计时器，参阅第 8-3 节。

Solicited Information 选项包括，指出节点在决定是否重新设置它的 Trickle 计时器时应当检验那些属性的标记。如果所有属性真实，节点重新设置它的 Trickle 计时器。如果标记置 1，那么 RPL 节点**必须**检验关联属性。如果标记清零，那么**不允许** RPL 节点检验关联属性(如果标记清零，RPL 节点假设关联属性真实。)

Option Type: 0x07

Option Length: 19

V: “V”标记即 Version 属性。如果接收者的 DODAGVersionNumber 匹配要求的 Version Number, Version 属性真实。如果 “V” 标记清零, 那么 Version 字段无效, 并且在发送端**必须**将 Version 字段设置为 0, 在接收端**必须**忽略它们。

I: “I” 标记即 InstanceID 属性。如果 RPL 节点的目前 RPLInstanceID 匹配要求的 RPLInstanceID, InstanceID 属性真实。如果 “I” 标记清零, 那么 RPLInstanceID 字段无效, 并且在发送端**必须**将 RPLInstanceID 字段设置为 0, 在接收端**必须**忽略它们。

D: “D” 标记即 DODAGID 属性。如果 RPL 节点的父代集有与 DODAGID 字段相同的 DODAGID, DODAGID 属性真实。如果 “D” 字段清零, 那么 DODAGID 字段无效, 并且在发送端**必须**将 DODAGID 字段设置为 0, 在接收端**必须**忽略它们。

Flags: Flags 字段中保留用于标记的 5 个未使用位。该字段**必须**由发送者初始化为 0, 接收者**必须**忽略该字段。

Version Number: 8 位无符号整数, 包含处于有效期内且正在请求的 DODAGVersionNumber 值。

RPLInstanceID: 8 位无符号整数, 包含处于有效期内且正在请求的 RPLInstanceID。

DODAGID: 128 位无符号整数, 包含处于有效期内且正在请求的 DODAGID。Solicited Information 选项的未分配位保留。在发送端它们**必须**被设置为 0, 在接收端**必须**忽略它们。

6-7-10 前缀信息

Prefix Information Option (PIO)可以出现在 DIO 消息中, 携带为[RFC4861]、[RFC4862]和[RFC6275]中 IPv6 ND Prefix Information 选项规定的、由 RPL 节点和 IPv6 主机使用的信息。尤其是, RPL 节点可以根据由[RFC4862]中规定的父代通告的前缀, 将这个选项用于 Stateless Address Autoconfiguration (SLAAC), 以及如[RFC6275]中规定的通告它自己的地址。DODAG 的根是设置这条信息的官方。这条信息沿 DODAG 向下无改变传播, 有一个例外是如果 “R” 标记置 1 以便在 PIO 中指出该 RPL 路由器的完整地址, 该路由器可以重写 Interface ID。此选项的格式做了修改(Type、Length、Prefix), 以便由和下述相同的 RPL 选项携带:

如果 DIO 中所接收 PIO 的预期作用仅仅是提供可通达接收节点的父代节点全局地址，那么发送者重新将“A”位和“L”位置 1 并将“R”位置 1。一旦收到，RPL 不会自动配置地址，或自动配置根据该前缀连接的路由[RFC4862]。因为在所有情况下，如果“L”位未置 1，RPL 节点可以在 PIOs 中包括它发送给它的子代的前缀。

0								1								2								3																																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																												
Type = 0x08								Opt Length = 30								Prefix Length								L	A	R	Reserved1																																
Valid Lifetime																																																											
Preferred Lifetime																																																											
Reserved2																																																											
Prefix																																																											

图 29 Prefix Information Option (PIO)格式

PIO 可用于分发给在 DODAG 内部使用的前缀，例如，地址自动配置。

关于 PIO，应当查询作为权威参考文献的[RFC4861]和[RFC6275]。这里，为了方便，转
录字段说明如下：

Option Type: 0x08

Option Length: 30。注意, 这个长度以单个八位位组为单位表示, 与 IPv6 ND 中的不同。

Prefix Length: 8 位无符号整数。在有效 Prefix 字段中前导位数目。此值范围从 0 到 128。

Prefix Length 字段提供用于 on-link 确定的必要信息(如果与 PIO 中的“L”标记结合)。如[RFC4862]中规定的, Prefix Length 字段还协助地址自动配置, [RFC4862]中对前缀长度可能有更多限制。

L: 1 位 on-link 标记。如果置 1, 它指出这个前缀可用于 on-link 确定。如果不置 1, 该通告没有说明前缀的 on-link 或 off-link 属性。换言之讲, 如果“L”标记不置 1, RPL 节点**决不能**断定从该前缀推导出的地址是 off-link 地址。即, 该通告**决不能**更新之前指出的该地址是 on-link 地址的结论。充当路由器的 RPL 节点**决不能**传播带有“L”标记置 1 的 PIO。充当路由器的 RPL 节点**可以**传播带有“L”标记不置 1 的 PIO。

A: 1 位自主地址配置标记。如果置 1, 它指出这个前缀可用于如[RFC4862]中规定的无状态地址配置。如果在同一链路上用两个协议(ND RAs 和 RPL DIOs)携带 PIOs, 对于无状态地址自动配置(SLAAC), RPL 节点有可能使用二者之一。也有可能通过强制在那个协议中携带的 PIOs 的“A”标记为 0, 使得对于 SLAAC 操作两个协议都不适合。

R: 1 位路由器地址标记。如果置 1, 它指出 Prefix 字段包括分配给发送路由器(该路由器可用作目标选项中的父代)的完整 IPv6 地址。指出的前缀是 Prefix 字段的开始端的前缀长度位。该路由器 IPv6 地址有与通告的前缀相同的范围和遵循与通告的前缀相同的生存期值。Prefix 字段的这个应用与在通告前缀自身中 Prefix 字段的应用一致, 因为 Prefix Advertisement 仅用于前导位。因此, 对这个标记位的解释独立于要求对 on-link (L)标记位和自主地址配置(A)标记位的处理。

Reserved1: 5 位未使用字段。该字段**必须**由发送者初始化为 0, 接收者**必须**忽略该字段。

Valid Lifetime: 32 位无符号整数。以秒为单位的时间长度(相对于分组发送时刻), 在此时间长度内对于 on-link 确定来说前缀有效。各位全为 1 的值(0xFFFFFFFF)表示无穷大。[RFC4862]也使用此 Valid Lifetime。

Preferred Lifetime: 32 位无符号整数。以秒为单位的时间长度(相对于分组发送时刻), 在此时间长度内, 经无状态地址自动配置, 根据该前缀生成的地址保持首选[RFC4862]。各位全为 1 的值(0xFFFFFFFF)表示无穷大。参阅[RFC4862]。注意, 这个字段的值**决不能超过** Valid Lifetime 字段, 以避免出现不再合法的首选地址。

Reserved2: 这个字段未使用。该字段**必须**由发送者初始化为 0, 接收者**必须**忽略该字段。

Prefix: IPv6 地址或 IPv6 地址前缀。Prefix Length 字段包括该前缀中有效前导位的数目。在前缀长度后的前缀中的位保留, **必须**由发送者初始化为 0, 接收者**必须**忽略这些位。路由器**不应当**发送本地链路前缀的前缀选项, 主机**应当**忽略这样的前缀选项。非存储节点**应当**忍住通告前缀, 直到它拥有那个前缀的地址, 然后它**应当**在这个字段中, 用“R”标记置 1, 通告它的完整地址。节点的子代, 该节点这样通告了带“R”标记置 1 的完整地址, 接着可以使用那个地址确定 Transit Information 选项的 DODAG Parent Address 子字段的内容。

PIO 的未分配位保留。在发送端它们**必须**被设置为 0, 在接收端**必须**忽略它们。

6-7-11 RPL 目标描述符

RPL Target 选项后可以紧跟一个证明那个特定目标合格的不透明描述符。

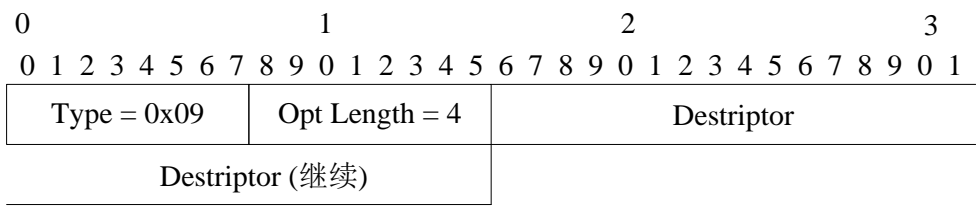


图 30 RPL Target Descriptor 选项格式

RPL Target Descriptor 选项用于证明目标，有时称为“加标记(tagging)”的事项，合格。

大多数情况，每个目标可以有一个描述符。描述符由在 RPL 网络中插入该 Target 的节点设置。在 DAO 消息中沿 DODAG 上行传播该 Target 的路由器必须只能复制，不能修改该描述符。

Option Type: 0x09

Option Length: 4

Descriptor: 32 位无符号整数。不透明。

第 7 章 序列计数器

本章介绍 RPL 中序列计数器的自举和运行总体方案，如在 DIO 消息中的 DODAGVersionNumber，在 DAO 消息中的 DAOSequence，以及在 Transit Information 选项中的 Path Sequence。

7-1 序列计数器概述

本规范使用三种不同的序列号来验证协议信息的新鲜度和同步：

DODAGVersionNumber: 这个序列计数器出现在 DIO Base 中，指示正在形成的 DODAG 的 Version。每当根决定形成新的 DODAG Version，以便重新验证完整性和允许出现全局修复时，由根单调递增 DODAGVersionNumber。当路由器加入新的 DODAG Version 时，DODAGVersionNumber 沿 DODAG 向下(Down)无改变传播。在 DODAG 中 DODAGVersionNumber 全局有效，它指出 DODAG 的 Version，路由器正在该版本中运行。一旦接收节点迁移到较新的 DODAG Version，较旧的(较低的)值表示始发路由器还没有迁移到新的 DODAG Version，并且不能用作父代。

DAOSequence: 这个序列计数器出现在 DAO Base 中,以便关联 DAO 消息和 DAO ACK 消息。对于发布供它自己使用以便检测 DAO 消息丢失和开启重新尝试的 DAO 消息的节点,DAOSequence 编号在本地有效。

Path Sequence: 这个序列计数器出现在 DAO 消息的 Transit Information 选项中。这个计数器的作用是区分这样的行动,即,根据路由冗余方案(在该方案中抵达相同目标有多条并行路由),用较新的路由代替陈旧的路由。Path Sequence 在 DODAG 中具有全局意义,它指出到所关联目标的路由的新鲜度。从始发路由器收到较旧的(较低的)值指出始发路由器持有陈旧的路由状态,始发路由器不应再被看作是到目标的潜在下一跳了。Path Sequence 由通告目标的节点计算,该节点是 Target 自身或代表主机通告 Target 的路由器。当父代路由器向根传播 DAO 内容时,Path Sequence 不改变。如果主机不向它的路由器传递计数,那么路由器负责代表该主机计算 Path Sequence,并且主机仅可以为此目的注册到一个路由器。如果考虑路由冗余,在给定时间点,发布包含相同 Target 的 DAO 消息给多个父代,那么,在那个相同目标的所有 DAO 消息中 Path Sequence 相同。

7-2 序列计数器运行

RPL 序列计数器以“棒棒糖(lollipop)”方式细分[Perlman83],其中 128 和更大的值用作线性系列,以便指出计数器的重新启动和自举,小于等于 127 的值用作大小为 128 的循环序列号码空间,如[RFC1982]所述。主要考虑从线性区转换到循环区时的运行模式。最后,在循环区运行时,如果检测到序列号相差太大,那么它们没有可比性,如下面所述。

基于 2^N 值配置比较窗口，SEQUENCE_WINDOW = 16，在本规范中 N 定义为 4。

对于给定的序列计数器：

- 1、序列计数器**应当**被初始化为具体实现定义的值，在使用前是 128 或更大。推荐值是 $240(256 - \text{SEQUENCE_WINDOW})$ 。
- 2、当序列计数器增加可能引起序列计数器超过它的最大值时，序列计数器**必须**回归到 0。当增加序列计数器到大于或等于 128 时，最大值是 255。当在小于 128 内增加序列计数器时，最大值是 127。
- 3、比较两个序列计数器时，**必须**运用下述规则：

1)如果第一个序列计数器 A 位于间隔[128..255]内，并且第二个序列计数器 B 位于 [0..127]内：

1))如果 $(256 + B - A)$ 小于或等于 SEQUENCE_WINDOW，那么 B 大于 A，A 小于 B，二者不相等。

2))如果 $(256 + B - A)$ 大于 SEQUENCE_WINDOW，那么 A 大于 B，B 小于 A，二者不相等。

例如，如果 A 是 240 并且 B 是 5，那么 $(256 + 5 - 240)$ 是 21。21 大于 SEQUENCE_WINDOW (16)；因此，240 大于 5。另一个例子，如果 A 是 250 并且 B 是 5，那么 $(256 + 5 - 250)$ 是 11。11 小于 SEQUENCE_WINDOW (16)；因此，250 小于 5。

2)两个被比较的序列计数器都小于或等于 127 情况，以及两个被比较的序列计数器都大于或等于 128 情况：

1))如果两个序列计数器之间差的绝对值小于或等于 SEQUENCE_WINDOW，那么，如[RFC1982]所述，比较用于确定大于、小于和等于关系。

2))如果两个序列计数器的差的绝对值大于 SEQUENCE_WINDOW, 那么, 不同步出现, 两个序列号不具有可比性。

4、如果确定两个序列号不具有可比性, 即, 比较的结果没有定义, 那么, 节点应当考虑比较, 就好像节点已经以这样的方式做了评估, 结果是把最近被观察到有增加的序列号放在前。如果做不到, 节点应当考虑比较, 就好像节点已经以这样的方式做了评估, 结果是尽量减小对节点自身状态的最终改变。

第 8 章 上行路由

这一章介绍 RPL 如何发现和维护上行(Upward)路由。本章介绍 DODAG Information Objects (DIOs)的使用, 该消息用于发现和维护这些路由。本章规定 RPL 如何生成和响应 DIOs。本章也介绍 DODAG Information Solicitation (DIS)消息, 它用于触发 DIO 发送。

如第 3-2-8 节所述, 决定加入 DODAG 的节点**必须**提供至少一个 DODAG 父代作为关联实例的默认路由。这个默认路由使分组能够被上行(Upward)转发, 直到它最终到达公共祖先, 从这里分组将下行(Downward)路由到目的地。如果目的地不在 DODAG 内, 那么此 DODAG 根能够借助到 DODAG 外部的连通性, 转发分组; 如果它不能向外转发分组, DODAG 根必须丢弃该分组。

DIO 消息也能够传送显式路由信息:

DODAGID: DODAGID 是根的 Global 或 Unique Local IPv6 地址。加入 DODAG 的节点**应当**提供经 DODAG 父代到达由根作为 DODAGID 使用的地址的主机路由。

RIO Prefix: 根**可以**在 DIO 消息中放置一个或多个 Route Information 选项。RIO 用于通告经过根、与优先级关联、可以到达的外部路由(如第 6-7-5 节所述), 此 RIO 包括来自[RFC4191]的 RIO。RIO Prefix 被看作是根的一种与路由通告不同的能力。尽管当分组从附着在入口 RPL 路由器的节点进入 RPL 域时, 该路由器**应当**使用 RIO Prefix 来选择 DODAG, **不允许**用另一个路由协议再次分发 RIO Prefix。Objective Function **可以**使用在 RIO 中通告的路由或这些路由的优先级, 以便偏爱相同实例的某个 DODAG 而不是其他。

8-1 DIO 基本规则

1、对于下述 DIO Base 字段，不是 DODAG 根的节点**必须**通告与它的首选 DODAG 父代(在第 8-2-1 节定义)相同的值。采用这种方法，这些值将沿 DODAG 向下(Down)传播不会被改变，并且由有到那个 DODAG 根的路由的每个节点通告。这些字段如下：

- 1)、Grounded (G)
- 2)、Mode of Operation (MOP)
- 3)、DAGPreference (Prf)
- 4)、Version
- 5)、RPLInstanceID
- 6)、DODAGID

2、在每一跳中，节点**可以**更新下述字段：

- 1)、Rank
- 2)、DTSN
- 3)、在 RPL Instance 中，每个根建立的 DODAGID 字段**必须**唯一，**必须**是属于该根的可路由 IPv6 地址。

8-2 上行路由发现和维护

上行(Upward)路由发现使节点能够通过发现属于感兴趣 DODAG 成员的邻居和认出一组父代，加入 DODAG。选择邻居和父代的精确策略取决于实现并且由目标函数(OF)驱动。这一节规定一系列规则，为了保证互操作性这些规则必须遵守。

8-2-1 在 DODAG 版本中的邻居和父代

RPL 的上行(Upward)路由发现算法和处理涉及本地链路节点的 3 个逻辑集合。首先，候选邻居集是可通过本地链路多播达到的节点的子集。这个集合的选择取决于实现和 OF。其次，父代集被限制为候选邻居集的子集。最后，首选父代是父代集的成员，该父代集是上行(Upward)路由中首选的下一跳。从概念上讲，首选父代是单个父代；不过，它可能是一组多个父代，如果这些父代优先级相等并有相同的 Rank。

更准确地讲：

- 1、DODAG 父代集**必须**是候选邻居集的子集。
- 2、DODAG 根**必须**有大小为 0 的 DODAG 父代集。
- 3、不是 DODAG 根的节点**可以**包括容量大于等于 1 的 DODAG 父代集。
- 4、节点的首选 DODAG 父代**必须**是它的 DODAG 父代集成员。
- 5、节点的 Rank **必须**大于其 DODAG 父代集的所有元素。
- 6、如果 Neighbor Unreachability Detection (NUD) [RFC4861]或等效机制，确定邻居不再可达，在 RPL 节点确认该邻居节点再次可达前，在计算和通告路由时 RPL 节点**决不能**考虑候选邻居集中的这个节点。通过不可达邻居的路由**必须**从路由表中删除。

这些规则确保对 DODAG 内的节点有一致的局部顺序。只要节点 Ranks 不变，当到根的每一跳上的 Rank 减少时，遵循上述规则可确保每个节点到 DODAG 根的路由无环路。

8-2-2 跨DODAG版本的邻居和父代

上述规则管辖单一DODAG Version。本节定义当有多个DODAG Versions时运行RPL的规则。

8-2-2-1 DODAG Version

- 1、元组(RPLInstanceID、DODAGID、DODAGVersionNumber)唯一定义DODAG Version。节点的DODAG父代集的每个元素，由从每个DODAG父代上次获知的DIO消息传送，**必须**属于相同DODAG Version。节点候选邻居集的元素**可以**属于不同DODAG Versions。

- 2、如果节点的DODAG父代集的每个元素都属于某个DODAG Version，或者如果那个节点是相应DODAG的根，节点是那个DODAG Version的成员。
- 3、节点**决不能**发送它不是其成员的DODAG Versions的DIOs。
- 4、DODAG根**可以增加**它们通告的、从而移到新的DODAG Version的DODAGVersionNumber。如果DODAG根增加它的DODAGVersionNumber，它**必须**遵循如第7章介绍的Serial Number Arithmetic惯例。触发DODAGVersionNumber增加事件在本节稍后以及在第18章介绍。
- 5、在给定的DODAG内，不是根节点**不允许**通告比它获知的最高DODAGVersionNumber 还要高的DODAGVersionNumber。更高被定义为比第7章中的运算符更高。
- 6、一旦节点通过发送DIO通告了DODAG Version，该节点**决不能**是相同DODAG (即，具有相同RPLInstanceID、相同DODAGID和更低的DODAGVersionNumber)的先前DODAG Version成员。更低被定义为比第7章中的运算符更低。

如果在非根节点上DODAG父代集变成空集，(即，最后一个父代已经移走，导致节点不再与那个DODAG关联)，那么，直到实现规定的本地计时器到期后不应当抑制DODAG信息。在抑制“旧”DODAG状态之前的间隔期间，节点能够观察到，如果DODAGVersionNumber增加应当出现新的父代。如果节点是无意中在它自己先前的子DODAG中重新加入旧的DODAG Version，这将有助于防止环路发生。

随着DODAGVersionNumber增加，新的DODAG Version从DODAG根向外扩展。通告新DODAGVersionNumber的父代可能不属于通告较旧DODAGVersionNumber的节点的子DODAG。因此，节点能够安全添加具有较新DODAGVersionNumber的任何Rank的父代，而不会形成环路。

例如，假设节点已经脱离有DODAGVersionNumber N的DODAG。假设节点有子DODAG，并打算通过通告INFINITE_RANK的Rank抑制那个子DODAG，但是这些通告可能已经在LLN内丢失。于是，如果节点观察到在那个最初的DODAG中以DODAGVersionNumber N通告位置的候选邻居，那个候选邻居可能已经在节点先前的子DODAG内，此时，添加那个候选邻居为父代有可能引起环路。在这种情况下，如果观察到那个候选邻居通告DODAGVersionNumber N+1，那么，那个候选邻居一定是安全的，因为它已经能够通过监听DODAG根来增加DODAGVersionNumber，而那个最初的节点已被删除时，它一定不在那个最初节点的子DODAG内。因此，被删除节点记住最初DODAG信息(包括DODAGVersionNumber N)是有益的。

什么时候DODAG根增加DODAGVersionNumber取决于实现，超出本规范讨论范围。举例包括定期增加DODAGVersionNumber、依靠管理介入，或在应用程序层检测连接丢失或DODAG低效。

节点转换到并通告新DODAG Version后，一旦它承担通告新DODAG Version的义务，上述规则使其不能通告先前的DODAG Version (先前的DODAGVersionNumber)。

8-2-2-2 DODAG根

- 1、不可能满足应用定义的目标的DODAG根**绝不能**将Grounded位设置为1。
- 2、DODAG根**必须**通告ROOT_RANK的Rank。
- 3、其DODAG父代集为空的节点**可以**变为浮动DODAG的DODAG根。它**也可以**设置它的DAGPreference为1，以便它有较低优先权。

在使用非LLN链路去结盟一系列LLN根的部署中，有可能在那些非RPL链路上运行RPL并使用一个路由器作为“骨干根(backbone root)”。骨干根是DODAG的虚拟根，它展示骨干网上BASE_RANK的Rank。充当那个骨干根父代的所有LLN根，包括如果自身也充当LLN根的骨干根，都展示到该LLN的ROOT_RANK的Rank。这些虚拟根是相同DODAG的一部分并且通告相同的DODAGID。它们与骨干网上的虚拟根一起协调DODAGVersionNumbers和其他DODAG参数。协调方法超出本规范讨论范围(在将来的伙伴规范中定义)。

8-2-2-3 DODAG选择

Objective Function以及通告的DAG路由度量和约束集合决定节点如何选择它的邻居集、父代集和首选父代(们)。这种选择也隐含决定DAG内的DODAG。这类选择可以包括管理偏好(preference, Prf)、度量或其他考虑。

如果节点有加入更优先DODAG的选项同时仍然满足其他优化目标，那么节点通常会寻求加入由OF确定的更优先DODAG。对于所有别的条件相同情况，留给实现决定最优先选择哪一个DODAG (提示：因为对于每个RPL Instance，节点必须仅加入一个DODAG)。

8-2-2-4 在DODAG版本中的等级和移动

- 1、节点**决不能**通告这样的Rank，该Rank低于或等于DODAG Version中节点父代集的任何成员。
- 2、节点**可以**通告这样的Rank，该Rank低于它在DODAG Version中的先前通告。
- 3、设L是给定节点已通告的DODAG Version内的最低Rank。在相同DODAG Version内，那个节点**决不能**通告高于 $L + \text{DAGMaxRankIncrease}$ 的有效Rank。INFINITE_RANK是这条规则的例外：节点**可以**在DODAG Version内不受限制地通告INFINITE_RANK。如果节点的Rank可能比 $L + \text{DAGMaxRankIncrease}$ 允许的高，当节点通告Rank时，它**必须**通告它的作为INFINITE_RANK的Rank。
- 4、节点**可以**，在任何时候，选择加入RPL Instance内的不同DODAG。这样的加入没有Rank限制，除非那个不同DODAG是这个节点先前已经是其成员的DODAG Version；在这种情况下，必须遵守前面的第3条规则。直到节点发送出指示它的新DODAG成员关系的DIO前，节点**必须**沿先前的DODAG转发分组。
- 5、获知由适当DODAG父代通告的邻接的DODAGVersionNumber后的任何时刻，节点**可以**选择迁移到DODAG内的邻接的DODAG Version。

从概念上讲，实现一直在维护DODAG Version内的DODAG父代集。移动引起DODAG父代集改变。向上(Up)移动不会引起产生环路的风险，但是向下(Down)移动有可能，所以操作要遵守额外限制。当节点迁移到邻接的DODAG Version时，需要为新的Version重新建立DODAG父代集。实现可以将迁移推迟一段合理时间，看看是否某些有潜在更好度量而Rank更高的其他邻居宣告它们自己。类似，如果节点跳进新的DODAG，节点需要为这个新DODAG构建新的DODAG父代集。

如果节点需要沿它附着的DODAG向下(Down)移动，增加它的Rank，那么节点可以限制它的路由，并且在做如第8-2-2-5节所述在移动之前延迟一段时间。

节点被允许加入任何DODAG Version，只要节点从来不是其没有任何限制的先前成员，但是如果节点一直是该DODAG Version的先前成员，那么，节点必须继续遵守在该DODAG Version生存期的任何时刻，它不可以通告比 $L + DAGMaxRankIncrease$ 高的Rank的规则。必须遵守这条规则，以便不产生漏洞，该漏洞允许节点自始至终高效将它的Rank增加到INFINITE_RANK，这可能对其他节点有影响，产生计数到无穷的资源浪费情况。

8-2-2-5 抑制

1、节点通过通告INFINITE_RANK的Rank，抑制路由。

2、在节点的父代集中，节点不允许存在有INFINITE_RANK的Rank的任何节点。

虽然出于抑制目的，实现可以通告INFINITE_RANK，这样处理与设置Rank为INFINITE_RANK不同。例如，节点可以继续发送数据分组，该分组的RPL Packet Information包括不是INFINITE_RANK的Rank，然而在该节点的DIOs中仍然通告INFINITE_RANK。

如果观察到(先前的)父代通告INFINITE_RANK的Rank，那个(先前的)父代已经与该DODAG分离，不能再充当父代，也不存在任何方法使得可以认为另一个节点有比INFINITE_RANK高的Rank。因此，那个(先前的)父代不能再充当父代，并且被从父代集中删除。

8-2-2-6 分离

- 1、在给定的DODAG Version内，节点无法保持连接到DODAG，即，不能在不违反本规范规则情况下维持非空父代集，节点**可能**会脱离这个DODAG Version。脱离的节点变成它自己的浮动DODAG的根，它**应当**立即用DIO通告这个新情况，作为抑制的替代。

8-2-2-7 跟随父代

- 1、如果节点收到来自它的DODAG父代之一的DIO，指出该父代已经离开该DODAG，如果可能，节点**应当**通过替代DODAG父代，停留在它目前的DODAG内。节点**可以**跟随该离开的父代。

DODAG父代可以移动、迁移到邻接的DODAG Version，或跳到不同的DODAG。对保留在目前DODAG内节点应当给予一些优先权，如果可能通过替代父代，但是如果没有其他选择，应当跟随父代。

8-2-3 DIO消息通信

如果收到DIO消息，接收节点必须首先确定是否该DIO消息应当接收下来做进一步处理，如果适合，随后提交DIO消息做进一步处理。

- 1、如果DIO消息异常，那么该DIO消息不适合进一步处理，节点**必须**静默抛弃它。(参阅第18章错误日志)
- 2、如果DIO消息发送者是候选邻居集成员并且该消息正常，节点**必须**处理该DIO消息。

8-2-3-1 DIO消息处理

当从候选邻居收到DIO消息时，遵循第8-2节介绍的DODAG发现规则，可以将邻居提升为DODAG父代。如果节点将邻居放入DODAG父代集，通过此新的DODAG父代节点，节点变成附着到DODAG。

最优先的父代应当用于限制那些可能变为DODAG父代的其他节点。在DODAG父代集中的一些节点可能有低于或等于最优先DODAG父代的Rank。(例如，如果能量受限设备处于较低Rank，但按照优化目标它应当避免，这种情况可能发生，导致处于较高Rank的较优先父代。)

8-3 DIO发送

RPL节点使用Trickle计时器[RFC6206]发送DIOs。来自发送者的、带有较小DAGRank的DIO (该DIO不会改变接收者的父带集、首选父代或Rank。)应当被看作与Trickle计时器一致，

以下分组和事件**必须**被看作与Trickle计时器不一致，并引起Trickle计时器复位：

- 转发分组时如果节点检测到不一致，如第11-2节所述。
- 当节点收到没有Solicited Information选项的多播DIS消息时，除非DIS标记限制这种行为。
- 当节点收到带Solicited Information选项的多播DIS消息，并且节点匹配Solicited Information选项中的所有属性时，除非DIS标记限制这种行为。
- 当节点加入新DODAG Version (例如，通过更新它的DODAGVersionNumber，加入新RPL Instance，等)时。

注意，这个列表没有穷尽所有情况，实现**可以**考虑其他不一致消息或事件。

节点**不应当**通过复位它的DIO Trickle计时器来响应单播DIS消息。如果节点收到没有Solicited Information选项的单播DIS，它**必须**单播DIO给发送者作为响应。这个DIO**必须**包括DODAG Configuration选项。如果节点收到有Solicited Information选项的单播DIS消息并且节点匹配该Solicited Information选项的属性，节点**必须**单播DIO给发送者作为响应。这个单播DIO**必须**包括DODAG Configuration选项。因此，节点**可以**发送单播DIS消息给潜在的DODAG父代，以便探求DODAG Configuration和其他参数。

8-3-1 Trickle参数

规定Trickle计时器的配置参数为：

Imin：从 DIO 消息获知，为 $(2^{\wedge}DIOIntervalMin)ms$ 。DIOIntervalMin 的默认值为 DEFAULT_DIO_INTERVAL_MIN。

Imax：从 DIO 消息获知，为 DIOIntervalDoublings。DIOIntervalDoublings 的默认值为 DEFAULT_DIO_INTERVAL_DOUBLINGS。

k：从 DIO 消息获知，为 DIORedundancyConstant。DIORedundancyConstant 的默认值为 DEFAULT_DIO_REDUNDANCY_CONSTANT。在 RPL 内，如果 k 值为 0x00，将被看作是 RPL 内的无穷大冗余常数，即，Trickle 绝不会删除这些消息。

8-4 DODAG 选择

DODAG 选择取决于实现和 OF。为了限制无规律的移动，以及所有度量相等，节点**应当**保持它们先前的选择。此外，节点**应当**提供能检测出可用性飘忽不定的父代的方法，至少届时节点可得到更稳定的选择。

如果不能连接到接地的 DODAG，或者需要优先考虑安全或其他因素，分散的 DODAGs **可以**尽可能聚合成更大的 DODAGs，以便能够在 LLN 内连通。

节点在考虑将某个候选邻居作为 DODAG 父代前，**应当**验证与该候选邻居能够双向连通和有适当的链路质量。

8-5 作为叶节点的运行

在一些情况，RPL 节点可以仅作为叶节点附着到 DODAG。这种情况的例子之一是当节点不理解或不支持 RPL Instance 的 OF 或通告的度量/约束(策略)时。按照第 18-6 节规定的相关策略，节点可以或者作为叶节点加入 DODAG，或者不加入 DODAG。如第 18-5 节提到的，此时建议记录故障。

叶节点不扩展 DODAG 的连通;然而,在某些情况,叶节点可能仍然需要偶尔发送 DIOs,尤其是当叶节点有可能不总是叶节点时,以及检测到不一致时。

作为叶节点运行时,节点必须遵循下述规则:

- 1、它**决不能**发送含 DAG Metric Container 的 DIOs。
- 2、它的 DIOs **必须**通告 INFINITE_RANK 的 DAGRank。
- 3、它可以禁止 DIO 发送,除非由于在转发分组或响应单播 DIS 消息时检测到不一致而触发发送 DIO,在这种情况下**决不能**禁止 DIO 发送。
- 4、它可以发送单播 DAOs,如第 9-2 节所述。
- 5、它可以发送多播 DAOs 到“1 跳”邻居,如第 9-10 节所述。

需要叶节点发送 DIO 的特殊情况是,如果那个叶节点是另一个 DODAG 先前的成员以及另一个节点转发假定是旧拓扑的消息,触发不一致。叶节点需要发送 DIO,以便修复不一致。注意,由于 LLNs 的有损性质,即使叶节点在变成叶节点前,通过通告旧 DODAG 中 INFINITE_RANK 的 Rank,可能已经乐观地抑制了它的路由,那个通告可能已经丢失,叶节点必须在此之后能够发送 DIO,以修复不一致。

一般讲,叶节点**决不能**作为路由器通告自己(即,发送 DIOs)。

8-6 管理等级

在有些情况,超出根据某种实现特有策略和节点属性由 OF 计算出的 Rank,调整节点通告的 Rank 或许有利。例如,除非没有其他选择,电池能量有限的节点应当是叶节点,此外该节点可以增加由 OF 规定的 Rank 计算结果,以便揭露夸大的 Rank。

第 9 章 下行路由

本章介绍 RPL 如何发现和维护下行(Downward)路由。RPL 使用 Destination Advertisement Object (DAO)消息构建和维护下行(Downward)路由。下行(Downward)路由支持 P2MP 流，从 DODAG 根朝向叶。下行(Downward)路由也支持 P2P 流：P2P 消息可以通过上行(Upward)路由，流向 DODAG 根(或公共祖先)，接着通过下行(Downward)路由，离开 DODAG 根，朝向目的地。

本规范介绍两种模式，RPL Instance 可以选择一种来维护下行(Downward)路由。第一种模式称为“存储(Storing)”模式。在存储模式中，节点为它们的子 DODAG 保存下行(Downward)路由表。在存储网络中下行(Downward)路由上的每一跳检查它的路由表，以便决定下一跳。第二种模式称为“非存储(Non-Storing)”模式。在非存储模式中，节点不保存下行(Downward)路由表。使用由 DODAG 根提供的源路由来路由下行(Downward)分组[RFC6554]。

对于存储网络和非存储网络，RPL 允许简单的 1 跳 P2P 优化。节点可以直接发送目标为 1 跳邻居的 P2P 分组到那个节点。

9-1 目的地通告父代

为了建立下行(Downward)路由，RPL 节点沿上行(Upward)发送 DAO 消息。这些 DAO 消息的下一跳目的地是所谓的“DAO 父代”。节点的 DAO 父代集合称为“DAO 父代集”。

- 1、节点**可以使用** all-RPL-nodes 多播地址发送 DAO 消息，这提供 1 跳路由优化。一旦发送多播 DAO，**必须清 0 “K” 位**。
- 2、节点的 DAO 父代集**必须是**它的 DODAG 父代集的子集。
- 3、在采用 Storing 模式时，节点**决不能**发送单播 DAO 消息给不是 DAO 父代的节点。
- 4、在采用 Storing 模式时，DAO 消息的 IPv6 源地址和目的地地址**必须是**本地链路地址。
- 5、在采用 Non-Storing 模式时，节点**决不能**发送单播 DAO 消息给不是 DODAG 根的节点。
- 6、在采用 Non-Storing 模式时，DAO 消息的 IPv6 源地址和目的地地址**必须是**本地唯一或全局地址。

DAO 父代的选择由实现和 Objective Function 规定。

9-2 下行路由发现和维护

Destination Advertisement 可配置成完全禁用,或者采用 Storing 或 Non-Storing 模式运行,如在 DIO 消息中 MOP 内报告的。

- 1、加入 DODAG 的所有节点**必须**遵守来自根的 MOP 设置。没有能力作为路由器全面参与的节点,例如,不匹配通告的 MOP 的节点,**可以**作为叶节点加入 DODAG。
- 2、如果 MOP 为 0,指出没有下行(Downward)路由,节点**决不能**发送 DAO 消息,并且**可以**忽略 DAO 消息。
- 3、在 Non-Storing 模式中,DODAG 根**应当**保存从 DAOs 获知的目的地的源路由表条目。DODAG 根**必须**能够生成从(被保存的)DAOs 获知的这些目的地的源路由。
- 4、在 Storing 模式中,所有非根节点、非叶节点**必须**保存从 DAOs 获知的目的地的路由表条目。

如 MOP 字段定义的,DODAG 可采用几种运行模式中的一种。DODAG 或者不支持下行(Downward)路由,它通过来自 DODAG 根的源路由支持下行(Downward)路由,或者它通过在网(in-network)路由表支持下行(Downward)路由。

如果是通过来自 DODAG 根的源路由支持下行(Downward)路由,通常希望 DODAG 根保存从 DAOs 获知的源路由信息,以便构建源路由。如果 DODAG 根没有保存某些信息,那么,某些目的地可能不可达。

如果通过在网路由表支持下行(Downward)路由,可能支持,可能不支持本规范中定义的多播运行,也由 MOP 字段指出。

如果通过在网路由表支持下行(Downward)路由,如本规范所述,将认为充当路由器的节点已经被充分配置,能够保持要求的路由表状态。如果充当路由器的节点不能保持完整的路由表状态,那么该路由状态不完整,结果可能导致丢弃消息,错误会被记录(参阅第 18-5 节)。将来的 RPL 扩展可能详细说明管理这种情况的优化行动/行为。

截止撰写本规范时，RPL 不支持混合模式运行，在混合模式中，某些节点采用源路由，其他节点保存路由表：将来的 RPL 扩展可能支持这种运行模式。

9-2-1 路径序列维护

对于与节点关联(节点拥有)的每个 Target，节点负责发出 DAO 消息，以便提供下行(Downward)路由。在这些 DAO 消息中包含的 Target+Transit 信息随后沿 DODAG 上行(Up)传播。在 Transit 信息选项中的 Path Sequence 计数器用作指出新鲜度，以及更新陈旧的下行(Downward)路由信息，如第 7 章所述。

当下述条件满足时，对于与节点关联(节点拥有)的 Target，节点**必须**增加 Path Sequence 计数器，并生成新的 DAO 消息：

- 1、Path Lifetime 被更新(例如，刷新或 no-Path)。

- 2、DODAG Parent Address 子字段列表被改变。

对于与节点关联(节点拥有)的 Target，必要时，节点**可以**增加 Path Sequence 计数器并生成新的 DAO 消息，以便刷新下行(Downward)路由信息。在 Storing 模式中，节点生成这样的 DAO 给它的每个 DAO 父代，以便能够多路径。在同一时刻为同一 Target 生成的所有 DAOs **必须**在 Transit Information 中用相同 Path Sequence 发送。

9-2-2 DAO 消息生成

如果节点收到 DAO 消息，由于它的 DAO 父代集的改变，或者为了响应其他事件(如相关前缀生存期到期)，节点或许发送 DAO 消息。收到 DAOs 时，很重要的是 DAO 消息是“新的”还是包含新信息。在 Non-Storing 模式中，节点收到的每个 DAO 消息是“新的”。在 Storing 模式中，如果 DAO 消息满足下述包含 Target 的准则中的任何一条，它是“新的”：

- 1、DAO 消息有较新的 Path Sequence 编号，

- 2、DAO 消息有额外的 Path Control 位，或者

3、DAO 消息是删除通往前缀的最后一条下行(Downward)路由的 No-Path DAO 消息。

如果那条 DAO 消息不是新的，收到来自它的子 DODAG 的 DAO 消息的节点可以禁止调度 DAO 消息发送。

9-3 DAO 基本规则

- 1、如果节点发送的 DAO 消息比之前发送的 DAO 消息有较新的或不同的信息，节点**必须**至少将 DAOSequence 字段加 1。发送的 DAO 消息等同于之前发送的 DAO 消息时，可以增加 DAOSequence 字段。
- 2、DAO 消息的 RPLInstanceID 字段和 DODAGID 字段**必须**有与节点的父代集成员和节点发送的 DIOs 相同的值。
- 3、为了请求在响应中发送单播 DAO-ACK，以便确认尝试，节点可以将单播 DAO 消息中的“K”标记置 1。
- 4、收到有“K”标记置 1 单播 DAO 消息的节点**应当**用 DAO-ACK 响应。收到“K”标记没有置 1 的 DAO 消息的节点**可以**用 DAO-ACK 响应，特别是用来报告出错条件。
- 5、在单播 DAO 消息中将“K”标记置 1、但是没有在响应中收到 DAO-ACK 的节点，**可以**重新安排作为另一次尝试的 DAO 消息发送，直到达到实现规定的重试次数。
- 6、节点**应当**忽略没有较新序列编号的 DAOs，并且**决不能**进一步处理它们。

与 DIO 的 Version 字段(该字段仅由 DODAG 根增加，其他节点重复但不改变。)不同，DAOSequence 值对于每个节点是唯一的。单播消息和多播 DAO 消息的序列编号空间或者相同，或者不同。**强烈推荐**使用相同的序列编号空间。

9-4 DAO 消息结构

在存储网络和非存储网络中 DAOs 结构相同。在最一般格式中，DAO 消息可以包括几组选项，其中每组由一个或多个 Target 选项构成，跟在其后的是一个或多个 Transit Information 选项。整组 Transit Information 选项适用于整组 Target 选项。后面各节介绍每种运行模式的更多细节。

- 1、RPL 节点**必须**在它们发送的每个 DAO 消息中包括一个或多个 RPL Target 选项。如果节点需要 DODAG 提供到那个节点的下行(Downward)路由, 一个 RPL Target 选项**必须**有包括该节点 IPv6 地址的前缀。紧跟在 RPL Target 选项后的**可以**是限制它的不透明 RPL Target Descriptor 选项。
- 2、如果节点针对覆盖节点地址之一的Target选项, 更新Transit Information选项中的信息, 节点**必须**增加那个Transit Information选项中的Path Sequence编号。偶尔**可以**增加Path Sequence编号, 以刷新下行(Downward)路由。
- 3、单播 DAO 消息中的一个或多个 RPL Target 选项**必须**由一个或多个 Transit Information 选项跟随。所有传送选项用于所有紧挨在它们前面的所有 Target 选项。
- 4、多播 DAOs **决不能**在 Transit Information 选项中包括 DODAG Parent Address 子字段。
- 5、接收和处理包含特定Target信息的DAO消息的节点, 以及有那个Target先前信息的节点**必须**使用与那个Target 关联的、在Transit Information选项中的Path Sequence编号, 以便确定该DAO消息是否包含按第7章所述更新过的信息。
- 6、如果节点收到不遵循上述规则的 DAO 消息, 它**必须**不做进一步处理抛弃该 DAO 消息。

在 Non-Storing 模式中, 根建立严格的原路由首部, 一跳接着一跳, 通过递归寻找将 Target(地址或前缀)和传送地址联系在一起的一跳信息。在有些情况, 如果子代地址源于由父代拥有和通告的前缀, 为了构建源路由首部, 可以由根推断那个父代-子代关系。在所有其他情况, 有必要根据可达目标通知传送 Target 关系的根, 以便以后能够递归构建路由首部。在 DAO 消息中作为 Target 通告的地址**必须**在同一路由器中收集, 或者这个地址**必须**是拥有在关联 Transit Information 中指出的地址的路由器在 on-link 上可达的。下述附加规则适用于确保端到端源路由路径的连续性:

- 1、在传送选项中使用的父代地址**必须**取自来源于那个有“R”标记置 1 的父代的 PIO。PIO 中的“R”标记指出前缀字段实际上包含完整父代地址，但是子代**不应当**假设该父代地址是 on-link。
- 2、“A”标记置 1 的 PIO 指出 RPL 子代节点可以使用通达自动配置地址的前缀。用含“A”标记置 1 的 PIO 通告前缀的父代**必须**确保，通过作为 DAO 目标通告该 PIO 中的地址或整个前缀，在该 PIO 中的地址或整个前缀可以从根到达。如果父代也将“L”标记置 1，指出该前缀是 on-link，那么该父代**必须**在 DAO 消息中作为 Target 通告整个前缀。如果“L”标记清零，“R”标记置 1，指出父代在 PIO 中提供它自己的地址，那么该父代**必须**作为 DAO 目标通告那个地址。
- 3、在 DAO 消息中作为 Target 通告的地址**必须**在同一路由器中收集，或者这个地址**必须**是拥有在关联 Transit Information 中指出的地址的路由器在 on-link 上可到达的。
- 4、为了尽量压缩路由首部，父代**应当**在含“A”标记置 1、“L”标记清零的所有 PIOs 中将“R”标记置 1，子代**应当**作为中转优先使用在该 PIO 中发现的父代地址，该 PIO 用于自动配置在 DAO 消息中作为 Target 通告的地址。
- 5、路由器或许有一些没有被认出是父代 on-link 的目标，或者因为这些目标是位于替代接口上的地址，或者因为这些目标属于到 RPL 外部的节点，例如连接主机的节点。为了在 RPL 网络中注入这样的 Target，路由器**必须**使用那个节点 DAO 父代的 on-link 地址，在那个目标的 Transit Information 选项中作为 DODAG Parent Address 子字段通告自己。如果 Target 属于外部节点，那么路由器**必须**在 Transit Information 中将 External “E” 标记置 1。

根据含“L”标记置 1 的父代 PIO，自动配置了地址的子代节点不需要作为 DAO Target 通告那个地址，因为该父代确保整个前缀已经可从根到达。然而，如果“L”标记没有置 1，那么，在 Non-Storing 模式中，子代节点必须使用可到达的父代地址通知该父代-子代关系的根，以便能够递归构建路由首部。这是将作为中转的父代的地址与 DAO 消息中作为 Target 的子代的地址关联实现的。

9-5 DAO 发送调度

因为 DAOs 沿上行(Upward)流动,接收的单播 DAO 可能触发发送单播 DAO 到 DAO 父代。

- 1、一旦收到有被更新信息的单播 DAO 消息,诸如包含有新 Path Sequence 的 Transit Information 选项,节点**应当**发送 DAO。节点**不应当**立即发送这个 DAO 消息。节点**应当**延迟发送该 DAO 消息,以便汇集来自其他节点的 DAO 信息,该节点是这些其他节点的 DAO 父代。
- 2、节点**应当**用计时器(DelayDAO)延迟发送 DAO 消息。收到 DAO 消息启动 DelayDAO 计时器。在 DelayDAO 计时器处于激活状态下收到的 DAO 消息不会复位该计时器。如果 DelayDAO 计时器到期,节点发送 DAO。
- 3、如果节点要添加其他节点到它的 DAO 父代集,节点**应当**安排发送 DAO 消息。
DelayDAO 的值和计算取决于实现。DEFAULT_DAO_DELAY 的默认值在本规范中定义。

9-6 触发 DAO 消息

节点可以触发它们的子 DODAG,以便发送 DAO 消息。每个节点维护 DAO Trigger Sequence Number (DTSN),节点通过 DIO 消息传播 DTSN。

- 1、如果节点获知它的 DAO 父代之一增加了该父代的 DTSN,节点**必须**使用第 9-3 节和第 9-5 节介绍的规则,安排发送 DAO 消息。
- 2、在 Non-Storing 模式,如果节点获知它的 DAO 父代之一增加了该父代的 DTSN,节点**必须**增加它自己的 DTSN。

在 Storing 运行模式,作为例行路由表更新和维护的一部分,存储节点**可以**增加 DTSN,以便根据它的直接子代可靠触发一组 DAO 更新。

在 Storing 运行模式，不需要根据整个子 DODAG 触发 DAO 更新，因为那个状态信息将一跳接一跳沿 DODAG 向上(Up)传播。

在 Non-Storing 运行模式，DTSN 增加也会引起节点的直接子代依次增加它们的 DTSN，根据整个子 DODAG 触发一组 DAO 更新。在 Non-Storing 运行模式，如果需要刷新 DAO 而不希望全局修复(诸如通过增加 DODAGVersionNumber)，一般仅根单独增加 DTSN。在 Non-Storing 运行模式，一般仅在观察到它们的父代增加 DTSN 时所有非根节点才会增加它们的 DTSN。

通常，节点可以按照实现规定的逻辑，如根据检测到下行(Downward)路由不一致，或偶尔根据内部计时器，触发 DAO 更新。

在存储网络中，为触发的 DAOs 选择适当 DelayDAO 可以大大减少发送 DAOs 的次数。触发沿 DODAG 向下(Down)流动；在最理想情况，由于每个节点仅发送 DAO 消息一次，DAOs 沿 DODAG 向上(Up)流动使得叶(节点)首次发送 DAO。通过设置 DelayDAO 反比于 Rank，可近似做这样的安排。注意，这个建议是想作为一种优化，提供高效聚合(它不是一般情况下正确运行所必需的)。

9-7 非存储模式

在 Non-Storing 模式中，RPL 使用 IP 源路由沿下行(Downward)路由消息。下述规则适用于采用 Non-Storing 模式的节点。Storing 模式有一套独立的规则，参阅第 9-8 节。

- 1、Transit Information 选项的 DODAG Parent Address 子字段**必须**包括一个或多个地址。所有这些地址**必须**是发送者 DAO 父代的地址。
- 2、沿着作为父代选择的一部分设置的默认路由，DAOs 被直接发送到根。
- 3、当节点从它的 DAO 父代集内删除节点时，它可以生成新 DAO 消息，该消息含经过更新的 Transit Information 选项。

在 Non-Storing 模式中，节点使用 DAOs 向 DODAG 根报告它的 DAO 父代。使用来自该路由中每个节点的 DAO 父代集，DODAG 根能够拼成到节点的下行(Downward)路由。Path Sequence 信息可用于检测陈旧的 DAO 信息。这样计算每跳路由的目的是为了尽量减少当 DAO 父代改变时的流量。如果节点报告整个源路由，那么一旦 DAO 父代改变，整个子 DODAG 必须发送新 DAOs 到 DODAG 根。因此，在 Non-Storing 模式，节点可以发送单条 DAO，尽管节点或许选择发送不止一条 DAO 消息到多个 DAO 父代中的每一个。

节点通过发送有多个 RPL Target 选项的单条 DAO 消息组合 DAOs。每个 RPL Target 选项有它自己的、紧跟其后的 Transit Information 选项。

9-8 存储模式

在 Storing 模式中，RPL 根据 IPv6 目的地地址，沿下行(Downward)路由消息。下述规则适用于 Storing 模式中的节点：

- 1、Transmit Information 选项的 DODAG Parent Address 子字段**必须**为空。
- 2、一旦收到单播 DAO，节点**必须**计算该 DAO 是否会改变节点自己通告的前缀集合。计算中**应当**考虑与该 DAO 关联的 Transit Information 选项中的 Path Sequence 信息，以便确定该 DAO 消息是否包含较新的、取代已经保存在节点的信息。如果是这样，节点**必须**遵循第 9-5 节的规定，生成并发送新 DAO 消息。这样的改变包括接收 No-Path DAO。
- 3、如果节点生成新 DAO，它**应当**单播新 DAO 到它的每个 DAO 父代。节点**决不能**单播该 DAO 消息到不是 DAO 父代的节点。
- 4、如果节点从它的 DAO 父代集删除节点，它**应当**发送 No-Path DAO 消息(参阅第 6-4-3 节)到那个删除的 DAO 父代，以便废除现有的路由。
- 5、如果到通告的下行(Downward)地址的消息遭遇转发错误、Neighbor Unreachable Detection (NUD)，或类似故障，节点**可以**将该地址标记为不可达，并生成适当的 No-Path DAO。

DAOs 通告节点已经路由到哪个目的地地址和前缀。与 Non-Storing 模式不同，这些 DAOs 不交流有关它们自己路由的信息：这个信息保存在网络中，隐含于 IPv6 源地址。当存储模式生成 DAO 时，它使用先前收到并保存的 DAOs 状态，以便产生一组 RPL Target 选项，以及该组选项所关联的 Transmit Information 选项。

因为这条信息保存在每个节点的路由表中，在 Storing 模式，DAOs 直接与 DAO 父代沟通(这些 DAO 父代保存这条信息)。

9-9 路径控制

来自节点的 DAO 消息包括一个或多个 Target 选项。每个 Target 选项或者规定节点通告的前缀、LLN 外部可达地址的前缀、在节点的子 DODAG 中的目的地地址，或者规定在子 DODAG 中的节点正在侦听的多播组。Transit Information 选项的 Path Control 字段使节点能够请求或提供多个下行(Downward)路由。节点按以下方式构建 Transit Information 选项的 Path Control 字段：

- 1、Path Control 字段的位宽**必须**等于值($PCS + 1$)，这里 PCS 在 DODAG Configuration 选项的控制字段中规定。发送端**必须**清除大于或等于值($PCS + 1$)的位，接收端**必须**将其忽略。小于那个值的位被认为是“有效”位。
- 2、节点**必须**在逻辑上构建它的 DAO 父代分组，同时填充 Path Control 字段，在该字段中每组由优先权相等的 DAO 父代构成。接着，**必须**按照优先权排序这些组，这提供 DAO 父代到 Path Control 子字段的逻辑映射(参阅图 27)。组**可以**重复，以便扩满路径控制字段的整个位宽，但是排序(包括重复的组)**必须**保留，以便优先权正确传达。
- 3、对于描述节点自己的地址或 LLN 外部前缀的 RPL Target 选项，Path Control 字段的至少一个有效位**必须**置 1。可以将 Path Control 字段的更多有效位置 1。

- 4、如果节点收到带有相同 RPL Target 选项的多个 DAOs，节点**必须**按位或(bitwise-OR)它收到的 Path Control 字段。这个聚合的按位或代表前缀要求的下行(Downward)路由数目。
- 5、如果节点发送 DAO 消息给它的父代之一，节点**必须**在子字段中选择设置为有效的一个或多个位，根据聚合的 Path Control 字段，该子字段被映射到包含那个 DAO 父代的组。给定位仅呈现为对一个父代有效。节点发送到它自己父代的 DAO 消息**必须**将这些有效位置 1，所有其他有效位清 0。
- 6、对于 RPL Target 选项和 DAOSequence 编号，节点发送到不同 DAO 父代的 DAOs **必须**有不相交的有效 Path Control 位集合。节点**决不能**在到 2 个不同 DAO 父代的 DAOs 上设置相同有效位。
- 7、Path Control 位**应当**按照映射 DAO 父代到 Path Control 子字段的优先级分配，如此，属于特定 Path Control 子字段的有效 Path Control 位，或位组，被分配给 DAO 父代，这些 DAO 父代位于被映射到那个子字段的组中。
- 8、在 Non-Storing 运行模式中，节点**可以**不对 Path Control 字段执行任何进一步处理就放过 DAOs。
- 9、节点**决不能**单播在置 1 的 Path Control 字段中没有有效位的 DAO 消息。对于给定的 Target 选项，有可能的是节点没有足够的聚合 Path Control 位，以便发送包含那个 Target 的 DAO 消息到它的每个 DAO 父代，在这种情况下，这些处于最低优先级的 DAO Parents 可能无法获得那个 Target 的 DAO 消息。

Path Control 字段使节点能够限制产生到它的下行(Downward)路由的数量。节点在 Path Control 字段设置等同于它偏爱的下行(Downward)路由最大数量的位数。大多数情况，每个位被发送到一个 DAO 父代；一串位可发送到节点的单个 DAO 父代，以便在节点自己的 DAO 父代间分配。

提供 Target(它有关联的 Path Control 字段)的 DAO 路由的节点**应当**使用那个 Path Control 字段的内容，以便在那个 Target 的多个替代 DAO 路由间确定优先级排序。Path Control 字段的分配源自(DAO 父代的)优先级，如，在下行(Downward)方向根据 Objective Function，基于这个节点的“端到端”聚合度量最佳知识确定。在 Non-Storing 模式中，通过聚合每个收到的 DAO 的信息(包括首选 DAO 父代的 Path Control 指示)，根能够确定下行(Downward)路由。

9-9-1 路径控制举例

假设有采用 Storing 模式的 LLN，它包括有 4 个父代(P1、P2、P3 和 P4)的 Node N。让 N 在它的子 DODAG 中有 3 个子代，C1、C2 和 C3。设 PCS 为 7，如此，在 Path Control 字段将有 8 个有效位：1111111b。考虑下例：

Path Control 字段分裂成 4 个子字段，PC1(11000000b)、PC2 (00110000b)、PC3 (00001100b) 和 PC4 (00000011b)，如此，这 4 个子字段代表 4 个不同优先级水平，如图 27 所示。本例，在 Node N 的实现中，组{P1, P2}彼此有相同优先级，整体上是最高优先级组。{P3} 优先级低于 {P1, P2}，然而高于 {P4}。接着，让 Node N 执行它的 Path Control 映射，结果：

{P1, P2}→ Path Control 字段中的 PC1 (11000000b)

{P3}→ Path Control 字段中的 PC2 (00110000b)

{P4}→ Path Control 字段中的 PC3 (00001100b)

{P4}→ Path Control 字段中的 PC4 (00000011b)

注意，实现重复{P4}，以便对 Path Control 字段完整覆盖。

- 1、让 C1 发送包含 Target T(Target T 带有 Path Control 10000000b)的 DAO。Node N 保存把 10000000b 与 C1 的 Path Control 字段和 Target T 关联的条目。
- 2、让 C2 发送包含 Target T(Target T 带有 Path Control 00010000b)的 DAO。Node N 保存把 00010000b 与 C1 的 Path Control 字段和 Target T 关联的条目。
- 3、让 C3 发送包含 Target T(Target T 带有 Path Control 00001100b)的 DAO。Node N 保存把 00001100b 与 C1 的 Path Control 字段和 Target T 关联的条目。
- 4、在一段时间后，Node N 生成 Target T 的 DAO。通过将它的每个子代(这些子代已经给出 Target T 的 DAO)的贡献或(ORing)在一起，Node N 构建起聚合的 Path Control 字段。因此，聚合的 Path Control 字段有如 10011100b 的有效位集合。

5、接着 Node N 在它的父代 P1、P2、P3 和 P4 中分发聚合的 Path Control 位，以便准备 DAO 消息。

6、P1 和 P2 适合从最优先的子字段(11000000b)接收有效位。这些位是聚合的 Path Control 字段中的 10000000b。Node N 必须将该位设置为仅有的 2 个父代之一。在这种情况下，给 Node P1 分配位，Node P1 获得它的 DAO 的 Path Control 字段 10000000b。没有留下位分配给 Node P2；因此，Node P2 将有 Path Control 字段 00000000b，到 Node P2 的 DAO 不能产生，因为没有有效位。

7、第 2 最优先子字段(00110000b)有有效位 00010000b。Node N 映射 P3 到这个子字段。Node N 可以分配有效位给 P3，构建 P3 的 DAO，该 DAO 包含有值为 00010000b 的 Path Control 的 Target T。

8、第 3 最优先子字段(00001100b)有有效位 00001100b。Node N 映射 P4 到这个子字段。Node N 可以分配 2 个位给 P4，构建 P4 的 DAO，该 DAO 包含有值为 00001100b 的 Path Control 的 Target T。

9、最低优先子字段(00000011b)没有有效位。如果有有效位，这些位将添加到为 P4 构建的 DAO 的 Path Control 字段。

10、按照为这些目标收集的聚合 Path Control 字段，着手填充目的地为有其他目标(T除外)的 P1、P2、P3 和 P4 的 DAO 消息。

9-10 多播目的地通告消息

不同于单播 DAO 运行，DAO 运行的特殊情况是多播 DAO 运行，它可用于填充“1 跳 (1-hop)”路由表条目。

1、节点可以多播 DAO 消息到本地链路范围的 all-RPL-nodes 多播地址。

2、多播 DAO 消息**必须**仅用于通告有关节点自身的信息，即，直接连接到节点的前缀或节点拥有的前缀，如节点预订参加的多播组或节点拥有的全局地址。

3、多播 DAO 消息**决不能**用于从其他节点处获知的(例如，通过单播 DAO)中继连接信息。

4、节点**决不能**对收到的多播 DAO 消息执行任何其他 DAO 相关处理；尤其是，节点**决不能**根据收到多播 DAO 执行 DAO 父代行动。

- 多播 DAO 可用于开启直接 P2P 通信，不需要中继分组的 DODAG。

第 10 章 安全机制

本章介绍安全 RPL 消息的生成和处理。RPL 消息代码的高阶位标识 RPL 消息是否是安全的。除了基本控制消息(DIS、DIO、DAO、DAO-ACK)的安全版本外，RPL 有几条仅在启用了安全性的网络中才相关的消息。

实现的复杂性和规模是关注 LLNs 的核心，也就是说，在 RPL 实现中包括先进的安全措施可能是经济的，也可能是物理上难以实现的。此外，许多部署可以使用链路层或其他安全机制满足它们的安全要求，不需要在 RPL 中提供安全保证。

因此，本文档中介绍的安全措施在实施中是**可选的**。具体实现**可以**支持所介绍安全措施的子集(包括空集)，例如，实现可以支持完整性和保密，但不支持签名。实现**应当**清晰规定支持哪一种安全机制，**强烈建议**实施者仔细考虑安全要求，考虑在他们网络中安全机制的可行性。

10-1 安全综述

RPL 支持 3 种安全模式：

- 无安全。在这种安全模式中，RPL 使用基本的 DIS、DIO、DAO 和 DAO-ACK 消息，它们没有 Security 部分。由于网络能够使用其他安全机制，诸如链路层安全，无安全模式不意味着没有任何保护地发送所有消息。
- 预装。在这种安全模式中，RPL 使用安全消息。为了加入 RPL Instance，节点必须有预装的密钥。节点使用这个预装密钥提供消息保密性、完整性和真实性。使用预装密钥，节点可以作为主机或路由器加入 RPL 网络。

- 身份验证。在这种安全模式中，RPL 使用安全消息。为了加入 RPL Instance，节点必须有预装的密钥。节点使用这个密钥提供消息保密、完整性和真实性。使用这个预装密钥，节点仅可以作为主机加入网络。为了作为路由器加入网络，节点必须从密钥当局获得第 2 个密钥。这个密钥当局，在向请求者提供第 2 个密钥前，能够认证请求者被允许作为路由器。对称算法不支持身份验证模式。截止撰写本规范时，RPL 仅支持对称算法：考虑到将来潜在的加密基元，会包括身份验证模式。参阅第 10-3 节。

不管怎样，RPL Instance 使用无安全模式都由它是否使用安全 RPL 消息通告。安全网络是否使用预安装或认证模式由 DAG Configuration 选项的“A”位通告。

本规范规定 CCM，带 CBC-MAC (Cipher Block Chaining - Message Authentication Code) 的 Counter，作为 RPL 安全的密码基础[RFC3610]。在本规范中，CCM 使用 AES-128 作为它的底层密码算法。在 Security 部分中有些位留作将来规定其他算法。

所有安全 RPL 消息或者有 MAC，或者有签名。安全 RPL 消息也有针对密件的加密保护。安全 RPL 消息格式都支持完整性加密/认证方案(例如，CCM)，以及分别加密分组和认证分组的方案。

10-2 加入安全网络

RPL 安全假设希望加入安全网络的节点预先配有用于与邻居和 RPL 根沟通的共享密钥。为了加入安全 RPL 网络，节点或者侦听安全 DIOs，或者通过发送安全 DIS 触发安全 DIOs。除了第 8 章介绍的 DIO/DIS 规则以外，安全 DIO 消息和 DIS 消息要遵守以下规则：

- 1、如果发送，这个初始安全 DIS 必须将 Key Identifier Mode 字段设置为 0 (00)，必须将 Security Level 字段设置为 1 (001)。使用的密钥必须是预配置的组密钥(Key Index 0x00)。
- 2、当节点复位它的 Trickle 计时器以响应安全 DIS(参阅第 8-3 节)时，节点发送的下一个 DIO 必须是有与安全 DIS 相同的安全配置的安全 DIO。如果节点在发送 DIO 前收到多个安全 DIS 消息，该安全 DIO 必须有与节点正在响应的最后一个 DIS 相同的安全配置。

3、当节点发送 DIO 以响应单播安全 DIS(参阅第 8-3 节)时，该 DIO **必须**是安全 DIO。

上述规则使节点能够使用预配置的共享密钥加入安全 RPL Instance。一旦节点使用预配置的共享密钥加入进 DODAG，Configuration 选项的“A”位决定节点的能力。如果 Configuration 选项的“A”位被清零，那么节点可以使用这个预安装的、共享的密钥正常交换消息：它可以发布 DIOs、DAOs，等。

如果 Configuration 选项的“A”位置 1 并且 RPL Instance 采用认证模式运行：

- 1、除了用 Key Index 0x00 保证安全的安全 DIOs 中的 INFINITE_RANK 以外，节点**决不能** Rank。当处理用 Key Index 0x00 保证安全的 DIO 消息时，处理节点**必须**假定通告的 Rank 是 INFINITE_RANK。任何其他值会导致该消息被抛弃。
- 2、使用 Key Index 0x00 的安全 DAOs **决不能**有带除了节点地址以外前缀的 RPL Target 选项。如果节点收到安全 DAO 消息，该消息使用预安装的、共享的密钥，其中 RPL Target 选项不匹配 IPv6 源地址，节点**必须**抛弃该安全 DAO 消息，不做进一步处理。

上述规则意味着，在“A”位设置为 1、使用 Key Index 0x00 的 RPL Instances 中，节点可以作为主机而不是作为路由器加入 RPL Instance。节点必须与密钥当局沟通，以便获得使它能够充当路由器的密钥。

10-3 安装密钥

一旦节点已经作为主机加入网络，认证模式要求想要成为路由器的节点动态安装新密钥。已经作为主机加入的节点，使用标准 IP 通信方式与能够提供新密钥的授权服务器联系。

获得这种密钥的协议超出本规范范围，将在今后的规范中详细介绍。需要详细介绍采用认证模式的 RPL 的安全运行。

10-4 一致性检验

RPL 节点发送应对重放攻击的 Consistency Check (CC)消息并同步计数器。

- 1、如果节点收到带有“R”位清零的单播 CC 消息，并且如果节点是关联 DODAG 的成员或节点处于加入关联 DODAG 的过程中，节点**应当**用单播 CC 消息响应发送者。这个响应**必须**将“R”位置 1，并且这个响应**必须**有与它收到的消息相同的 CC 随机数字段、RPLInstanceID 字段和 DODAGID 字段。
- 2、如果节点收到多播 CC 消息，它**必须**抛弃该消息，不做进一步处理。

Consistency Check 消息使节点能够发布质询-响应，以便验证节点目前的计数器值。因为 CC 随机数由质询产生，重放消息的敌对方不太可能产生正确的响应。在 Consistency Check 响应中的计数器使质询者能够验证它获知的计数器值。

10-5 计数器

在最简单情况，计数器值是无符号整数，在每次发送安全 RPL 时节点将计数器值递增 1 或更多。计数器**可以**表示为有下述性质的时间戳：

- 1、时间戳**必须**至少有 6 个八位位组长。
- 2、时间戳**必须**是 1024Hz(二进制毫秒)粒度。
- 3、时间戳启动时间**必须**是协调世界时间(UTC)1970 年 1 月 1 日上午 12:00:00。
- 4、如果计数器表示为时间戳，计数器值**必须**是按下述方法算出的值。让 T 为时间戳，S 为所使用密钥的启动时间，E 为所使用密钥的终止时间。使用与上面描述时间戳所用的相同的 3 条规则表示 S 和 E。如果 $E > T < S$ ，那么计数器无效，节点**绝不能**生成分组。其他情况，计数器值等于 T-S。
- 5、如果计数器表示这样的时间戳，节点**可以**将安全 RPL 分组 Security 部分的“T”标记置 1。
- 6、如果 Counter 字段没有这样的时间戳，那么节点**决不能**将“T”标记置 1。

7、如果节点没有满足上述要求的本地时间戳，它**必须**忽略“T”标记。

如果节点支持这样的时间戳并且它收到有“T”标记置 1 的消息，它可以按第 10-7-1 节所述，对收到的消息进行时间检验。如果节点收到没有“T”标记置 1 的消息，它**决不能**使用这项时间检验。从应用考虑，节点的安全策略**可以**包括拒绝没有“T”标记置 1 的所有消息。

因为出于安全性、应用和其他原因，许多 LLNs 现在已经维持亚毫秒粒度上的全局时间同步，“T”标记是存在的。允许 RPL 充分利用这个已有(如果存在)的功能将极大简化某些安全问题解决方案，如延迟保护。

10-6 离站分组发送

本节介绍，在给定离站 RPL 控制分组和安全保护情况下，RPL 如何生成发送的安全分组。本节还介绍提供要求的保护的加密操作顺序。

节点的安全策略数据库决定安全保护要求，以及应用于离站 RPL 分组的安全等级。这个离站分组处理的安全策略数据库的配置取决于特定实现。

在发送安全 RPL 消息场合，RPL 节点**必须**在离站 RPL 分组中设置 Security 部分(T、Sec、KIM 和 LVL)，以便描述保护等级和使用的安全设置(参阅第 6-1 节)。RPL Message Code 字段的 Security 子字段位**必须**被设置为指示安全 RPL 消息。

构建 AES-128 CCM 随机数(参阅图 31)以保证离站分组安全而使用的计数器值，**必须**是传送到特定目的地地址的最后一个计数器的增量。

在安全策略规定延迟保护应用的场合，构建 CCM 随机数以保证离站分组安全而使用的 Timestamp 计数器**必须**按照第 10-5 节中的规则递增。使用 Timestamp 计数器的场合(用“T”标记置 1 指出)，**必须**包括作为发送安全 RPL 消息一部分的本地维护的 Timestamp 计数器。

用于保护离站分组的密码算法应由节点的安全策略数据库规定，**必须**用离站消息内 Sec 字段置 1 的值指出。

离站分组的安全策略决定规定安全密钥的可用 KIM 和 Key Identifier，这里的安全密钥用于密码分组处理，包括可选的使用签名密钥(参阅第 6-1 节)。安全策略也规定算法 (Algorithm)以及采用的认证形式或者认证和加密形式的保护等级(Level)，规定可能用于离站分组的签名。

在应用加密的场合，节点**必须**运用在分组的 Security 部分中规定的安全保护、密钥和 CCM 随机数，用那个加密的净荷替代原始分组净荷。所有安全 RPL 消息包括完整性保护。连同安全算法处理一起，节点或者得到 MAC，或者得到**必须**作为离站安全 RPL 分组一部分包括在内的签名。

10-7 入站分组接收

本节介绍安全 RPL 分组的接收和处理。给定入站安全 RPL 分组，该分组中 RPL Message Code 字段的 Security 子字段置 1，本节介绍 RPL 如何生成分组的未加密变体以及验证它的完整性。

接收者使用 RPL 安全控制字段决定必要的分组安全处理。如果不知道所描述消息类型安全等级和发起者，或不能满足本地主张的安全策略，节点**必须**抛弃该分组，不做进一步处理，节点**可以**发起管理报警，节点**决不能**发送任何响应消息。这些策略可能包括安全等级、使用的密钥、源标识符，或缺乏基于时间戳的计数器(如由“T”标记指出的)。安全策略数据库对入站分组处理的配置超出本规范范围(例如，该配置可以通过 DIO Configuration 或通过带外管理路由器配置定义)。

其中消息 Security Level (LVL)指出加密的 RPL 消息，节点使用通过 KIM 字段和 CCM 随机数识别出的密钥信息，作为对消息净荷进行去加密处理的输入。CCM 随机数从消息 Counter 字段、其他收到的和本地维持的信息中推得(参阅第 10-9-1 节)。通过调用由所接收分组 Sec 字段规定的逆加密模式操作，获得明文消息内容。

接收者使用 CCM 随机数和识别出的密钥信息检验入站分组完整性。如果相对于收到的 MAC 完整性检验失败，节点**必须**抛弃该分组。

如果收到的消息有初始化的(0 值)计数器值，并且接收者有目前为消息发起者维持的入站计数器，接收者**必须**通过发送 Consistency Check 响应消息(参阅第 6-6 节)给消息源，发起计数器再同步。应当用为特定节点地址维持的目前完整出站计数器保护 Consistency Check 响应消息。当入站计数器包括在 Consistency Check 消息净荷内时，那个出站计数器将包括在消息的安全部分内。

根据规定的安全策略，节点对收到的 RPL 消息**可以**使用重放保护。重放检验**应当**在对收到分组进行认证前进行。应当对照维持给定原始节点地址的入站计数器的水印，比较(例如从入站分组获得的)计数器。如果所接收消息计数器值为非 0，并且小于维持的入站计数器水印，这指出可能有分组重放，节点**必须**抛弃该入站分组。

如果将延迟保护被规定为入站分组安全策略检验的一部分，Timestamp 计数器用于验证所接收 RPL 分组的及时性。如果入站消息 Timestamp 计数器值指出，消息发送时间先于本地维持的发起者地址的发送时间计数器，指出有重放违规，节点**必须**抛弃该入站分组。如果收到的 Timestamp 计数器值指出消息发送时间(该时间早于 Current 时间)小于可接受的分组延迟，指出有延迟违规，节点**必须**抛弃该入站分组。

一旦消息被解密，如果适用，并且消息成功通过它的完整性校验、重放检验和可选的延迟-保护检验，节点可以更新它的本地安全信息，如针对重放比较的源预期计数器值。

当收到不能通过安全策略检验或其他应用的完整性检验、重放检验或延迟检验的消息时，节点**决不能**更新它的安全信息。

10-7-1 时间戳密钥检验

如果消息的“T”标记置 1 并且节点有符合第 10-5 节要求的本地时间戳，那么节点**可以**检验消息的时间一致性。节点通过添加计数器值到关联密钥的启动时间，计算消息发送时间。如果这个发送时间是密钥过去的结束时间，节点**可以**抛弃该信息，不做进一步处理。如果与接收机上的本地时间相比，此发送时间是太遥远的过去或未来，节点**可以**抛弃该消息，不做进一步处理。

10-8 完整性和保密性覆盖范围

对于 RPL ICMPv6 消息，整个分组处在 RPL 安全范围内。

对整个无安全 IPv6 分组计算 MACs 和签名。计算 MACs 和签名时，可考虑遵循[RFC4302] (IPsec Authenticated Header)第 3-3-3-1 节规则，将数量不定的多个 IPv6 字段用 0 填充。在较低层可能使用的任何压缩前，执行 MAC 和签名计算。

如果加密 RPL ICMPv6 消息，在 Security 部分后的第 1 个字节开始加密，一直持续到分组的最后一个字节。IPv6 首部、ICMPv6 首部和一直到 Security 部分结束的 RPL 消息不加密，因为正确解密分组需要它们。

例如，发送带有 LVL=1、KIM=0 和 Algorithm=0 消息的节点使用 CCM 算法[RFC3610]创建有 ENC-MAC-32 属性的分组：该算法加密分组并附加一个 32-bit MAC。块密码密钥由 Key Index 决定。CCM 随机数按第 10-9-1 节所述计算；认证和加密的消息是在 Security 部分后的第 1 个字节开始，在分组的最后一个字节结束的 RPL 消息。附加的认证数据以 IPv6 首部开始，以 RPL Security 部分的最后一个字节结束。

10-9 密码的操作模式

本规范介绍的密码操作模式(Algorithm = 0)基于 CCM 和块密码 AES-128 [RFC3610]。现有实现广泛支持这种操作模式。CCM 模式需要随机数(CCM 随机数)。

10-9-1 CCM 随机数

RPL 节点构建如图 31 所示的 CCM 随机数:

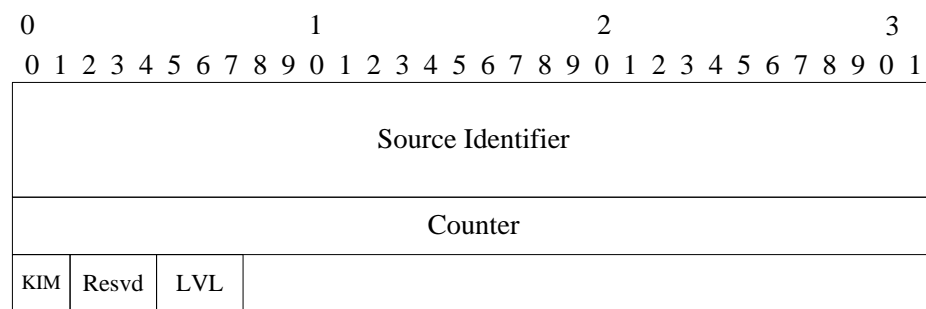


图 31 CCM 随机数

Source Identifier: 8 字节。Source Identifier 被设置为受保护分组始发者的逻辑标识符。

Counter: 4 字节。Counter 被设置为 RPL 控制消息 Security 选项中相应字段的(非压缩)值。

Key Identifier Mode (KIM): 2 位。KIM 被设置为 RPL 控制消息 Security 选项中相应字段的值。

Security Level (LVL): 3 位。Security Level 被设置为 RPL 控制消息 Security 选项中相应字段的值。

CCM 随机数的未分配位保留。构建 CCM 随机数时**必须**将它们设置为 0。

CCM 随机数的所有字段用一阶(first order)最高有效八位位组和最高有效位表示。

10-9-2 签名

如果 KIM 指出使用签名(值为 3), 那么节点添加签名到分组的数据净荷。Security Level (LVL)字段描述这个签名的长度。在 Security Mode 3 的 RPL 中, 签名方案是如[RFC3447]第 8-1 节定义的 RSA 算法实例(RSASSA-PSS)。作为公共密钥, 签名方案使用 pair (n,e) 对, 其中 n 是 2048 位或 3072 位 RSA 模, 而 $e = 2^{16} + 1$ 。签名方案使用 CCM 模式 [RFC3610]作为有 $M = 0$ (作为流密码)的加密方案。注意, 虽然[RFC3610]禁止含 $M = 0$ 的 CCM 模式, 当与签名一起使用时, RPL 明确允许含 $M = 0$ 的 CCM 模式, 因为签名提供充分的数据认证。这里, 含 $M = 0$ 的 CCM 模式按照[RFC3610]中的规定, 但是其中第 2-2 节中 M 的字段**必须**被设置为 0。签名方案使用[FIPS180]第 6-2 节中规定的 SHA-256 哈希函数。签名方案使用的消息采用[RFC3447]第 8-1 节的编码规则。

让“a”作为 6 字节的计数器表示和消息首部的串联。分组净荷是分组数据“m”和签名“s”的右侧串联。用消息部分 a 和 m 的右侧串联调用这个签名方案, 而用消息部分 a 和 m 的右侧串联以及用签名 s 调用签名验证。

这种形式的 RSA 签名对 RPL 网络提供充分的保护。如果需要, 可采用提供更简洁签名的替代签名方案, 这方面内容超出本规范范围, 可能在将来的规范中涉及。

支持采用 2048 位签名或 3072 位签名的 RSA 签署的实现**应当**支持 2048 位和 3072 位 RSA 签名的验证。这是在考虑提供 RPL 部署的升级路径。

第 11 章 分组转发和环路避免/检测

11-1 分组转发建议

本文档规定路由协议。提供这些非规范性建议是为了通过举例说明这样的实现如何与 RPL 一起工作, 为转发实现设计提供帮助。

转发分组到达目的地时, 按如下原则, 优先选择下一跳后继者:

- 1、本规范仅涉及如何根据 DODAG Version 选择后继者。该 DODAG Version 匹配正在被转发分组的 IPv6 首部中标记的 RPLInstanceID。只要附加规则落实到位，如严格的实例排序和防止产生环路的路由协议，实例外部的路由可以使用。这些规则可能在单独文档中定义。
- 2、如果本地管理偏好倾向于从不同路由协议而不是从 RPL 获知的路由，那么使用那个后继者。
- 3、如果分组首部如[RFC6554]中规定的，通过包括 RH4 首部规定源路由，那么使用那条路由。如果节点不能转发带规定的源路由的分组，那么应当抛弃那个分组。节点可以记录出错。节点可以用 Source Routing Header 消息发送 ICMPv6 出错给该分组的源(参阅第 20-18 节)。
- 4、如果在路由表中有匹配从多播目的地通告中获知的目的地的条目(例如，目的地是一跳邻居)，那么使用那个后继者。
- 5、如果在路由表中有匹配从单播目的地通告中获知的目的地的条目(例如，目的地位于向下的(Down)子 DODAG 中)，那么使用那个后继者。如果有与多个后继者关联的多个 DAO Path Control 位，那么，查询这些 Path Control 位，以便选择时按优先权排序这些后继者。如果，对于给定的 Path Control 位，据记录有多个后继者要求那个 Path Control 位，优先权应当给予最近要求该位的那个后继者。
- 6、如果有提供到匹配该目的地的前缀的路由的 DODAG Version，那么按照 OF 和路由度量选择这些 DODAG 父代之一作为后继者。
- 7、如果没有更好的匹配，任何其他至今还没有尝试过的 DODAG 父代都可以选作转发单播分组的下一个目标。
- 8、最后，抛弃分组。可以调用 ICMP Destination Unreachable(检测到不一致)。

按照[RFC2460]，转发时，Hop Limit 必须递减。

注意, 选择的后继者**决不能**是在此之前刚刚传递分组的邻居(水平分割), 除非是根据做出改变节点的路由表的决定, 打算将分组传送从上行(Upward)方向改为下行(Downward)方向, 如, 当接近目的地时, 从 DIO 路由切换到 DAO 路由以继续向目的地行进。

11-2 环路避免和检测

RPL 环路避免机制维持简单运行, 为的是尽量减小剧烈变动(churn)和状态数量。形成环路的原因很多, 例如, 控制分组丢失。RPL 包括避免崩盘和触发受损路径修复的反应式环路检测技术。

RPL 环路检测使用在数据分组内传送的 RPL Packet Information, 依靠如[RFC6553]那样的外部机制, 该外部机制放置在 IPv6 Hopby-Hop 选项首部中 RPL Packet Information 内。

RPL Packet Information 内容定义如下:

Down “O” : 1 位标记, 指出预期分组是向上(Up)传播还是向下(Down)传播。如果预期分组向下(Down)传播(使用 DAO 路由), 路由器将“O”标记置 1, 如果是向 DODAG 根转发(到有较低 Rank 的节点), 将其清零。主机或 RPL 叶节点**必须**将“O”标记设置为 0。

Rank-Error “R” : 1 位标记, 指出是否检测到 Rank 错误。如果在相对 Ranks 中有错配并且有如“O”位指出的方向, 检测到 Rank 错误。主机或 RPL 叶节点**必须**将“R”位设置为 0。

Forwarding-Error “F” : 1 位标记, 指出这个节点不能向目的地进一步转发分组。“F”位或许由于子代节点设置, 该子代节点没有到含 Down “O” 位设置为 1 的分组的目的地路由。主机或 RPL 叶节点**必须**将“F”位设置为 0。

RPLInstanceID: 8 位字段, 指出 DODAG 实例, 分组沿其发送。

SenderRank: 16 位字段, 由源设置为 0, 并由在 RPL 网络内负责转发的路由器设置为 DAGRank(rank)。

11-2-1 源节点运行

如果源意识到 RPLInstanceID 是分组的首选，那么源**必须**相应设置与该分组关联的 RPLInstanceID 字段；否则，源**必须**将其设置为 RPL_DEFAULT_INSTANCE。

11-2-2 路由器运行

11-2-2-1 实例转发

RPLInstanceID 与拥有分组的源关联。这个 RPLInstanceID **必须**匹配被任何节点将分组放置在其上的 RPL Instance，无论该节点是主机还是路由器。RPLInstanceID 是 RPL Packet Information 的一部分。

在 RPL 网络内转发分组的 RPL 路由器**必须**核查分组是否包括 RPL Packet Information。如果不包括，那么 RPL 路由器**必须**插入 RPL Packet Information。如果路由器是将分组注入 RPL 网络的入口路由器，该路由器**必须**设置 RPL Packet Information 中的 RPLInstanceID 字段。路由器如何定义到 RPLInstanceID 的映射细节超出本规范范围，留待将来的规范解决。

转发分组出 RPL 网络的路由器**必须**删除 RPL Packet Information。

如果路由器收到规定了给定 RPLInstanceID 的分组，并且该节点(即路由器)能够沿着关联到那个实例的 DODAG 转发分组，那么该路由器**必须**这样做，并且保留 RPLInstanceID 值不变。

如果任何节点不能沿着关联该 RPLInstanceID 的 DODAG 转发分组，那么该节点**应当**抛弃该分组并发送 ICMP 出错消息。

11-2-2-2 DAG 不一致环路检测

如果分组方向不匹配 Rank 关系，DODAG 出现不一致。如果接收者收到有下述特征之一的分组，接收者检测出不一致：

来自 Rank 较高节点的被置 1 (对于向下(to Down))的“O”位。

来自 Rank 较低节点的被清 0(对于向上(for Up))的“O”位。

当 DODAG 根增加 DODAGVersionNumber 时，可能在下一个 DODAG Version 和前一个 DODAG Version 间形成临时 Rank 不连续，尤其是，如果节点正在调整它们在下一个 DODAG Version 中的 Rank，以及正在推迟它们到下一个 DODAG Version 的迁移时。仍然是前一个 DODAG Version 成员的路由器可能选择转发分组到是下一个 DODAG Version 中的(将来的)父代。在某些情况，这将引起该父代检测到不一致，因为前一个 DODAG Version 中的 Rank 排序不一定与下一个 DODAG Version 中的相同，分组可能被判断为不能向前转发。如果发送路由器意识到所选择的后继者已经加入下一个 DODAG Version，那么，当发送路由器跨越此不连续转发分组到下一个 DODAG Version，以避免 Rank 不一致虚假检测时，发送路由器**必须**更新 SenderRank 到 INFINITE_RANK。

不认为沿路径出现一个不一致是重大错误，分组可以继续。然而，沿同一分组路径第二次检测到不一致不应当发生，并且**必须**抛弃该分组。

这个过程由与分组关联的 Rank-Error 位控制。当对分组的检测发现不一致时，如果 Rank-Error 位没有置 1，那么将 Rank-Error 位置 1。如果该位置 1，**必须**抛弃该分组并且**必须**将 Trickle 计时器复位。

11-2-2-3 DAO 不一致检测和恢复

DAO 不一致环路恢复是仅适用于 Storing 运行模式的机制。

在 Non-Storing 模式中，分组被源路由到目的地，DAO 不一致不在本地纠正。相反，将带有新代码“Error in Source Routing Header”的 ICMP 错误发送回根。“Error in Source Routing Header”消息的格式和“Destination Unreachable Message”的相同，如[RFC4443]所述。调用在 ICMP 消息中被发回分组的部分应当至少记录到路由首部，并且该路由首部应当被这个节点吸收，以便使 IPv6 首部中的目的地是这个节点不可能到达的下一跳。

如果路由器有下行(Downward)路由，该路由是先前经子代从 DAO 消息获知的，但是那条下行(Downward)路由在该子代中不再合法(例如因为子代中那个相关状态已经被清零)，DAO 不一致发生。使用 DAO 不一致环路恢复，分组可用于沿子 DODAG，递归挖掘和清理过时的 DAO 状态。

一般情况，下(Down)行分组绝不应该再上(Up)行。如果使用 DAO 不一致环路恢复，那么路由器**应当**发送分组回到父代，该父代用 Forwarding-Error “F” 位置 1 和 “O” 位保持不变传递过该分组。否则，路由器**必须**静默抛弃该分组。

一旦收到有 Forwarding-Error 位设置为 1 的分组，节点**必须**删除引起转发到那个邻居的路由状态，清零 Forwarding-Error 位，并尝试再次发送该分组。用户可配置的实现特定计时器到期后，可将分组发送到替代邻居。如果经过这个节点的那个替代邻居仍然出现不一致 DAO 状态，过程将重复，这个节点将把 Forwarding-Error “F” 位置 1，在替代邻居中的路由状态也将清零。

第 12 章 多播运行

本章介绍 IPv6 RPL 网络上多播路由运行，特别是单播 DAOs 怎样用于中继组注册。可用相同 DODAG 结构转发单播和多播流量。本章仅限于介绍组注册如何交换，以及转发基础设施如何运行。本章不提供 LLN 内多播的完整介绍，尤其是不介绍专门针对多播的 DODAGs 生成或运行多播 RPL 的细节---它们是将来规范的任务。

除了传送的地址类型以外，多播组注册使用与单播等同的 DAO 消息。主要不同是下行多播流量被复制到已经用多播组注册的所有子代，而单播流量仅传递给一个子代。

支持 RPL Storing 运行模式的节点也**应当**支持下面介绍的多播 DAO 操作。不认为仅支持 Non-Storing 运行模式的节点支持本章内容。

多播运行由 DIO 中的 MOP 字段控制。

- 如果 MOP 字段要求多播支持，那么作为路由器加入 RPL 网络的节点，必须按照本章介绍的 RPL 网络内多播信令和转发运行。不支持由 MOP 字段要求的多播运行的节点可以仅作为叶节点加入。

- 如果 MOP 字段不要求多播支持，那么多播由超出本规范范围的某些其他方法处理。(具体例子可能包括一系列单播复制或有限范围洪泛路由)。

路由器或许仅选择传递侦听器注册 DAO 消息到它的首选父代；在这种情况下，如果在那条链路上发送失败，对于该路由器的所有子 DODAGs，返回的多播分组或许丢失。另一方面，路由器在处理通告单播目的地的 DAO 消息时或许选择复制额外的父代；在这种情况下，或许有需要路由器删改的重复内容。

因此，多播路由状态被放置在从侦听器到 DODAG 根路径上的每个路由器中，使得根能够复制多播分组到所有它的子代路由器，这些子代路由器已经发布了包括那个多播组的 Target 选项的 DAO 消息。

对于源自 DODAG 内的多播分组，分组被传递到首选父代，如果失败，那么传递到该 DODAG 内的替代父代。除了传递该分组的子代以外，分组也被复制到所有注册的字代。最后，如果在外部基础设施中有侦听器，那么 DODAG 根必须进一步将分组传播到该外部基础设施。

因此，DODAG 根充当 RPL 网络的自动代理 Rendezvous Point，并对起始于 RPL 域内的所有多播流充当朝向非 RPL 域的源。所以，无论根是否实际附着到非 RPL 域，也无论 DODAG 是接地还是悬浮，根能够在任何时间分发内部多播流。

第 13 章 维护路由相邻性

沿着沿 DODAG 上行(Up)的默认路径，或沿着从沿 DODAG 下行(Down)的目的地通告中获知的路径的后继者选择，形成路由相邻性，路由相邻性需要维护。

在 IGP 中，诸如 OSPF [RFC4915]或 IS-IS [RFC5120]，维护路由相邻性涉及应用保持激活机制(Hellos)，或其他协议，诸如 Bidirectional Forwarding Detection (BFD) [RFC5881] 和 MANET Neighborhood Discovery Protocol (NHDP) [RFC6130]。不幸的是，这样的积极主动方法在受限环境常常不可取，因为在受限环境，相比数据流量这样的方法会导致过量控制流量，对链路负载和节点资源都形成负面影响。

与这些路由协议相比，RPL 没有定义任何检测路由相邻性失败的保持激活机制：这是因为在多数情况，这样的机制会占用过多带宽，更为重要的是能量(电池供电设备没有能力定期发送保持激活信息)。RPL 更需要能检测到邻居不再可达的外部机制。这样的机制应当最好可对流量起反应，以尽量减少维护路由相邻性的开销，专注实际使用的链路。

可用的反应机制例子包括：

Neighbor Unreachability Detection [RFC4861]机制。

基于如关联状态事件和层 2 确认事件的层 2 触发[RFC5184]。

第 14 章 目标函数指南

Objective Function (OF)，结合路由度量和约束，可用于 DODAG 加入选择，用于在那个 DODAG 中选择一些同行作为父代的。OF 用于计算父代排序列表。OF 也负责计算 DODAG Version 内设备的 Rank。

在 DIO 消息内，使用 Objective Code Point (OCP)指出 OF，并且 OCP 指出构建 DODAG 必须使用的方法。Objective Code Point 在[RFC6552]和相关伙伴规范中规定。

14-1 目标函数行为

在节点内，预期大多数 OF 遵循相同的抽象行为：

- 每当事件指出潜在的下一跳信息被更新时触发父代选择。这可能在收到 DIO 消息、计时器到期、所有 DODAG 父代不可达，或收到指示候选邻居状态已经改变的触发时发生。
- OF 扫描节点上所有的接口。虽然大多数应用场景一般可能仅有一个接口，然而或许在一些场景中有多个接口，并且接口或许被配置为或者可用于，或者不可用于运行 RPL。接口也有可能被配置为有优先权，或者通过或许依赖链路层的某些探试方法，动态了解到此接口比别的接口更好，这些均超出本规范范围。最后，接口或许匹配或许不匹配要求的 OF 准则，例如，安全程度。因此，某些接口或许被完全排除在计算之外，例如，如果这些接口不能满足某些通告的约束，那么其他接口或许多少获得优先考虑。

- OF 扫描可能接口上的所有候选邻居，检查它们是否能够充当 DODAG 路由器。或许有许多候选邻居，候选邻居能够使用前或许需要通过某种合法性测试。尤其是，某些链路层需要与路由器一起活动的经验，以便能将该路由器作为下一跳。
- 通过在候选者的 Rank 中添加代表节点和候选者在 DODAG Version 中相对位置的值，OF 计算所比较节点的 Rank。
 - * Rank 的增加必须至少是 MinHopRankIncrease。
 - * 为在算法中保持环路避免和度量优化，Rank 的增加应当反映度量值的任何增加。例如，采用纯粹相加的度量，如，ETX，Rank 的增加可以正比于度量的增加。
 - * 不考虑将可能引起节点 Rank 增加的候选邻居选为父代。
- 忽略这样的候选邻居，它们通告的 OF 不兼容策略功能规定的 OFs 集合。
- 因为 OF 扫描所有候选邻居，OF 保持目前最好的父代，并将该父代的能力与目前的候选邻居比较。OF 定义一些对实现目标极为关键的测试。路由器间的测试确定排序关系。
 - * 如果相对于那个关系，路由器们是平等的，那么接下来的测试在路由器之间进行，
 - * 否则，两个路由器中最优的变成目前的最优父代，继续扫描下一个候选邻居。
 - * 一些 OFs 可能包括比较 Ranks 的测试，该测试会导致节点是否加入任意路由器。

- 扫描完成时，选出首选父代，计算出节点的 Rank，它是首选父代的 Rank 加上用 Rank 表示的与首选父代的步长。
- 其他轮次扫描或许有必要选出替代父代。在下一轮次中：
 - * 忽略不在相同 DODAG 中的候选邻居。
 - * 忽略 Rank 大于节点的候选邻居。
 - * 父代选择时，忽略 Rank 与节点等同的候选邻居。
 - * 首先选择 Rank 小于节点的候选邻居。

第 15 章 采用邻居发现的互操作性建议

本规范直接借用来自 IPv6 邻居发现(NeighborDiscovery, ND)的 Prefix Information Option (PIO)和 Route Information Option (RIO)。预计，由于将来的规范建立在这个基础上，利用 IPv6 ND 部分可能有额外的原因。这一章为将来的规范提供一些建议。

首先，RPL 是路由协议。当在 RPL 和 ND 间映射功能时应极为关注保持架构。RPL 仅用于路由。这就是说，有可能说服技术上的原因，允许在具体实现/部署中共享 RPL 和 IPv6 ND 间的选项。

一般讲，下述准则适用：

- 必须根据 RPL Control Message Options 注册表分配 RPL Type 代码。
- RPL Length 字段必须以单个八位位组为单位表示，这与 ND Length 字段不同。ND Length 字段以 8 个八位位组为单位表示。
- 通常不要求 RPL 选项对齐到 8 个八位位组边界。
- 当为作为 RPL 选项再分配而映射/转换 IPv6 ND 选项时，只要可能，应当删除任何填充的八位位组。例如，在 PIO 中的 Prefix Length 字段足以描述 Prefix 字段的长度。当为作为 IPv6 ND 选项再分配而映射/转换 RPL 选项时，应当恢复任何这样的填充八位位组。这个过程必须毫不含糊。

第 16 章 归纳互操作实现要求

本章归纳基本的互操作性，引用以三种主要模式之一运行的 RPL 实现规范文本。预期实现或者仅支持无下行(Downward)路由，Non-Storing 模式，或者仅支持 Storing 模式。第四种模式，作为叶节点运行，也有可能。

遵循本规范的实现可以包括不同功能子集，以适应应用场景。重要的是实施者要支持与该应用场景要求的互操作性一致的互操作性等级。针对此，可以提出超越本规范范围的进一步指导(例如，作为适用性声明)，而且可以预料，在某些情况这样的进一步指导会重复本规范的部分内容。

16-1 通用要求

一般情况，如果 RPL LLN 内的所有节点协同使用相同的 MOP、OF、度量和约束，并且因此都能充当 RPL 路由器，可以实现最高等级的互操作性。如果节点无能力成为 RPL 路由器，节点或许能够以较为受限的方式，作为 RPL 叶节点实现互操作。

所有 RPL 实现应支持使用在数据分组内传送的 RPL Packet Information(参阅第 11-2 节)。在[RFC6553]中介绍了一种这类机制。

RPL 实现应支持使用 Neighbor Unreachability Detection (NUD)或等同的机制，以便维护相邻 RPL 节点的可达性(参阅第 8-2-1 节)。可以优化替代机制，以适应实现受到约束的能力，诸如链路层提示。

本规范提供获得PIO并由此形成IPv6地址的方法。如果使用该机制，可能必须通过外部处理(如，IPv6 ND [RFC4861]或6LoWPAN ND [6LOWPAN-ND])执行地址解析和重复地址检测。

16-2 (仅)作为 RPL 叶节点运行

- (仅作为)叶节点的实现从来不作为 RPL 路由器参与。叶节点的互操作实现行为在第 8-5 节归纳。
- 不需要支持特殊 MOP 编码, 如果叶节点发送建立下行(Downward)路由的 DAO 消息, 叶节点应当采用与由 MOP 指出的运行模式一致的方法这样做。
- 不要求支持特定 OF。
- 总之, 叶节点通常不发布 DIO 消息, 它可能发布 DAO 消息和 DIS 消息。虽然叶节点一般忽略 DAO 消息和 DIS 消息, 但它接受 DIO 消息。

16-3 作为 RPL 路由器的操作

如果不能获得进一步指导, 那么 RPL 路由器实现**必须**至少支持低于 OF0 的度量 [RFC6552]。

对于一致性操作, RPL 路由器实现需要支持由 DODAG 使用的 MOP。

所有 RPL 路由器需要执行 Trickle [RFC6206]。

16-3-1 (仅)支持上行路由

仅支持上行(Upward)路由的 RPL 路由器实现支持:

- 上行(Upward)路由(参阅第 8 章)
- MOP 编码 0(参阅第 20-3 节)
- 总之, 发布 DIO 消息和 DIS 消息, 不发布 DAO 消息。接受 DIO 消息和 DIS 消息, 忽略 DAO 消息。

16-3-2 采用非存储模式支持上行路由和下行路由

采用 Non-Storing 模式支持上行(Upward)路由和下行(Downward)路由的 RPL 路由器实现支持:

- 上行(Upward)路由(参阅第 8 章)

- 下行(Downward)路由(Non-Storing) (参阅第 9 章)
- MOP 编码 1(参阅第 20-3 节)
- 源路由下行(Downward)流量([RFC6554])
- 总之，发布 DIO 消息和 DIS 消息，发布 DAO 消息到 DODAG 根。接受 DIO 消息和 DIS 消息，DAO 消息由节点而不是由 DODAG 根忽略。虽然可通过某些替代方法支持多播，采用本规范介绍的方法不能支持多播。

16-3-3 采用存储模式支持上行路由和下行路由

采用 Storing 模式支持上行(Upward)路由和下行(Downward)路由的 RPL 路由器实现支持：

- 上行(Upward)路由(参阅第 8 章)
- 下行(Downward)路由(Storing) (参阅第 9 章)
- MOP 编码 2(参阅第 20-3 节)
- 总之，发布 DIO 消息、DIS 消息和 DAO 消息。接受 DIO 消息、DIS 消息和 DAO 消息。虽然可通过某些替代方法支持多播，采用本规范介绍的方法不能支持多播。

16-3-3-1 可选的支持基本多播方案

采用下述附加方案，Storing 模式实现可因支持基本多播而得到增强：

- Basic Multicast Support(参阅第 12 章)
- MOP 编码 3(参阅第 20-3 节)

16-4 将来规范内容

许多内容留给将来的规范，包括但不限于下述内容：

- 如何匹配非 RPL 节点，如 IPv6 主机，例如，要始终如一地分发至少 PIO 材料给附着的节点。

- 如果使用认证模式，如何获得支持的认证材料 (参阅第 10-3 节)。
- 对多个并发实例的操作细节。
- 高级配置机制，如提供 `RPLInstanceIDs`、`Objective Functions` 参数化和控制安全的参数。(可以预期，这样的机制或许将 DIO 扩展为跨 DODAG 传播信息的工具)。

第 17 章 RPL 常数和变量

下面是 RPL 常数和变量归纳：

BASE_RANK：这是或许用于协调多个根的虚拟根的 Rank。BASE_RANK 值为 0。

ROOT_RANK：这是 DODAG 根的 Rank。ROOT_RANK 值为 `MinHopRankIncrease`(如由 DODAG 根通告的)，所以 `DAGRank(ROOT_RANK)` 为 1。

INFINITE_RANK：这是 Rank 恒定的最大值。INFINITE_RANK 值为 0xFFFF。

RPL_DEFAULT_INSTANCE：这是由这个协议，也是由不具有任何占优势策略的节点使用的 `RPLInstanceID`。RPL_DEFAULT_INSTANCE 值为 0。

DEFAULT_PATH_CONTROL_SIZE：这是默认值，用于在 DODAG Configuration 选项中配置 PCS，它指出在 Transit Information 选项 Path Control 字段中有效位的数目。DEFAULT_PATH_CONTROL_SIZE 值为 0。这配置了最简单情况，即限制扇出端为 1 和限制节点仅向一个父代发送 DAO 消息。

DEFAULT_DIO_INTERVAL_MIN：这是默认值，用于为 DIO Trickle 计时器配置 `Imin`。DEFAULT_DIO_INTERVAL_MIN 值为 3。这个配置导致 8 ms 的 `Imin`。

DEFAULT_DIO_INTERVAL_DOUBLINGS：这是默认值，用于为 DIO Trickle 计时器配置 `Imax`。DEFAULT_DIO_INTERVAL_DOUBLINGS 值为 20。这个配置导致 2.3 小时的最大间隔。

DEFAULT_DIO_REDUNDANCY_CONSTANT: 这是默认值，用于为 DIO Trickle 计时器配置 k 。DEFAULT_DIO_REDUNDANCY_CONSTANT 值为 10。这个配置是 Trickle 抑制机制的保守值。

DEFAULT_MIN_HOP_RANK_INCREASE: 这是 MinHopRankIncrease 的默认值。DEFAULT_MIN_HOP_RANK_INCREASE 值为 256。这个配置导致 8 位宽的 Rank 整数部分。

DEFAULT_DAO_DELAY: 这是 DelayDAO Timer 的默认值。DEFAULT_DAO_DELAY 值为 1s。参阅第 9-5 节。

DIO Timer: 每个 DODAG 一个实例，节点是该 DODAG 的成员。该计时器到期触发发送 DIO 消息。Trickle 计时器有在 $[0, \text{DIOIntervalMin}..2^{\text{DIOIntervalDoublings}}]$ 之间可变的间隔。参阅第 8-3-1 节。

DAG Version Increment Timer: 每个 DODAG 最多一个实例，节点充当该 DODAG 的 DODAG 根。可能所有实现不支持此。该计时器到期触发递增 DODAGVersionNumber，引起发送一连串新的、被更新的 DIO 消息。选择的间隔时间应适合 DODAG 的传播时间和符合应用要求(例如，与开销相比的响应时间)。

DelayDAO Timer: 每个 DODAG 的每个 DAO 父代(被选择接收目的地通告的 DODAG 父代的子集)最多一个计时器。该计时器到期触发发送 DAO 消息到该 DAO 父代。参阅第 9-5 节。

RemoveTimer: 每个邻居(即，已经发送 DAO 消息到作为 DODAG 父代的这个节点的那些邻居)的每条 DAO 条目最多一个计时器。该计时器到期可能触发 No-Path 通告，或者，如果没有 DAO 父代，立即释放该 DAO 条目。

第 18 章 可管理性考虑

本章目的是考虑 RPL 的可管理性，以及在 LLN 内如何运行 RPL。本章涉及下述可管理性方面：配置、监测、故障管理、审计，以及从[RFC5706]中陈述的建议来看的协议性能。

大多数现有的 IETF 管理标准是 MIB 模块(基于 Structure of Management Information (SMI)数据模型), 用于监测和管理网络设备。

对于许多协议, IETF 社区已经使用 IETF Standard Management Framework, 包括 Simple Network Management Protocol [RFC3410]、Structure of Management Information [RFC2578]和管理新协议的 MIB 数据模型。

如[RFC5706]中指出的, 在运行和管理方面的一般策略已经扩展到对一套工具和多个管理协议更加开放, 而不是严格依赖单一协议(如 SNMP)的策略。

2003 年, Internet Architecture Board (IAB)召开了关于 Network Management 的研讨会 [RFC3535], 会上讨论了一些 IETF 网络管理协议的优点和缺点, 将这些协议与运行需要, 尤其是配置, 相比较。

讨论的一个问题是 SNMP[RFC3410] 二进制格式用户界面不友好。对于 LLNs 情况, 应当提请注意在撰写本规范时, CoRE 工作组正在积极探索 LLNs 内的设备资源管理。尽管如此, 我们仍然认为本章应当对如何部署、运行和管理 RPL 提供重要指导。

如[RFC5706]所述:

管理信息模型应当包括如下方面的讨论, 即什么是可管理的? 协议的哪些方面需要配置? 可以允许哪些操作类型? 或许会发生什么样的特定协议事件? 哪些事件可以考虑在内? 以及哪些事件应当通知操作员?

下面各节详细讨论这些方面。

RPL 会应用于各种其资源(如存储器)变化很大的设备上, 可能从仅几千字节到几百个千字节, 甚至是兆字节。如果存储器非常受限, 不可能满足本章列出的所有要求。尽管如此, 详尽列出这些要求是有意义的, 实施者可据根据设备上的可用资源决定哪些要求能够满足。

18-2 配置管理

本节讨论配置管理，列出与配置管理有关的协议参数。

某些 RPL 参数是可选的。这些配置要求仅适用于使用的选项。

18-2-1 初始化模式

“Architectural Principles of the Internet” [RFC1958]第 3-8 节指出：“尽可能避免选项和参数。任何选项和参数应当动态配置或协商，而不是人工操作”。在 LLNs 中尤其是这样，因为 LLNs 内一般有大量设备，人工配置难以想象。这已考虑在 RPL 的设计中，DODAG 根借此向加入 DODAG 的设备提供大量参数，从而避免在路由器上的繁琐配置，避开了可能的资源错配(例如，Trickle 计时器的值，等。)。然而，有一些 RPL 实现应当允许配置的补充 RPL 参数，本节讨论它们。

18-2-1-1 启动时的 DIS 操作模式

当节点第一次加电时：

- 1、节点可以决定保持静默，等待接收来自感兴趣 DODAG 的 DIO 消息(通告支持的 OF 和度量/约束)，以及直到它加入 DODAG 不发送任何多播 DIO 消息。
- 2、节点可以决定发送一个或多个 DIS 消息(可选，特定 DODAG 需要的 DIO)作为对靠近 DODAGs 的初步试探，在某个可配置的时间段后，响应中没有 DIO 消息，节点可以决定寻找浮动 DODAG 并启动发送多播 DIO 消息。

除了要求的参数外，RPL 实现**应当**允许配置上面列出的首选运行模式(采用第二种模式：DIS 数量和相关计时器)。

18-2-2 DIO 和 DAO 基本消息和选项配置

考虑到应用 RPL 的大量场合，RPL 规定了许多协议参数。然而，已经注意到需要限制每个 RPL 路由器上必须配置的参数数量。作为替代，可以使用一些默认值，并且如果需要，这些参数可由 DODAG 根提供，从而为动态参数设置创造条件。

RPL 实现**应当**允许配置下述路由协议参数。正如上面指出的，应注意到大量参数在 DODAG 根上配置。

18-2-3 LLN 中每个路由器上配置的协议参数

RPL 实现**必须**允许配置下述 RPL 参数：

- RPLInstanceID [DIO 消息，在 DIO Base 消息内]。虽然 RPLInstanceID 必须在 DODAG 根上配置，它也必须作为对每个节点的策略进行配置，以决定节点是否应当加入具体 DODAG。注意，如果节点变成浮动 DODAG 的根，第二个 RPLInstanceID 可在该节点上配置。
- 支持的 Objective Code Points (OCPs)列表
- 支持的度量列表：[RFC6551]规定了许多用于构成 DODAG 的度量和约束。因此，RPL 实现应当允许配置节点能够接受和理解的度量列表。如果节点收到包含它不理解或不支持的度量和/或约束的 DIO，如第 8-5 节规定，节点将作为叶节点加入。
- Prefix Information，与有效的和首选的生存期及“L”标记和“A”标记一起。[DIO 消息，Prefix Information Option]。如果 Prefix Information 选项必须与 DIO 消息一起携带，以便分发自动配置的 Prefix Information，RPL 实现**应当**允许配置。在这种情况下，RPL 实现**必须**使前缀列表与相应标记一道在 PIO 中通告。
- Solicited Information [DIS 消息，在 Solicited Information 选项中]。注意，如果应当发送这样的消息，并且是在与 RPLInstance ID 的值、“V”标记/“I”标记/“D”标记一起的环境下，RPL 实现**应当**允许配置。
- “K”标记：当节点应当在 DAO 消息中将“K”标记置 1 时[DAO 消息，在 DAO Base 消息中]。
- MOP (Mode of Operation) [DIO 消息，在 DIO Base 消息中]。

- Route Information(和优先权) [DIO 消息, 在 Route Information 选项中]。

18-2-4 LLN 中每个非 DODAG 根路由器上配置的协议参数

RPL 实现**必须**允许配置 Target 前缀[DAO 消息, 在 RPL Target 选项中]。

此外, 有些情况节点可能希望指定 Target, 以方便具体 Target 处理(优先级, 等)。这类处理的规则超出本规范范围。如果使用, RPL 实现**应当**允许以每个 Target 为基础配置 Target Descriptor (例如, 使用访问列表)。

其 DODAG 父代集为空的节点可以变成浮动 DODAG 的 DODAG 根。它也可以将它的 DAGPreference 置 1, 从而降低它被首选的可能。因此, RPL 实现**必须**允许配置在这种情况下节点应当发起的各种行动:

- 启动节点自己的(浮动)DODAG: 除了它的 DAGPreference 外, 必须配置新的 DODAGID。
- 抑制破损的路径(参阅第 8-2-2-5 节中的流程)。
- 触发本地修复。

18-2-5 DODAG 根上配置的参数

此外, 其他几个参数仅在 DODAG 根上配置, 以及在 DIO 消息携带的选项中通告。如第 8-3 节规定的, RPL 实现使用多个 Trickle 计时器控制 DIO 消息发送。用一组可配置参数决定 Trickle 算法的使用, 这些参数**必须**是可配置的, 它们接着被 DODAG 根、沿 DIO 消息中的 DODAG 通告。

- DIOIntervalDoublings [DIO 消息, 在 DODAG Configuration 选项中]
- DIOIntervalMin [DIO 消息, 在 DODAG Configuration 选项中]
- DIORedundancyConstant [DIO 消息, 在 DODAG Configuration 选项中]

此外, RPL 实现**应当**为配置下述 RPL 参数集合创造条件:

- Path Control Size [DIO 消息，在 DODAG Configuration 选项中]
- MinHopRankIncrease [DIO 消息，在 DODAG Configuration 选项中]
- DODAGPreference 字段[DIO 消息，DIO Base 对象]
- DODAGID [DIO 消息，在 DIO Base 选项中]和[DAO 消息，当 DAO 消息的“D”标记置 1 时]

DAG 根行为：在有些情况，节点如果不能加入接地的 DODAG，它可能不想总是充当浮动 DODAG 根。例如，依靠电池运行的节点可能不希望长期充当浮动 DODAG 根。因此，RPL 实现可以支持这样一种能力，即对节点是否能够在配置的时间段内充当浮动 DODAG 根进行配置的能力。

DAG Version Number Increment：通过在 DODAG 根中的配置，RPL 实现可以允许通过更新 DODAGVersionNumber，刷新 DODAG 状态。RPL 实现应当允许配置由 DODAG 根使用的无论是周期触发机制还是事件触发机制，以便控制 DODAGVersionNumber 改变(该改变触发全局修复，如第 3-2-2 节所述)。

18-2-6 与基于 DAO 机制有关的 RPL 参数配置

DAO 消息是可选的，用在需要下行(Downward)路由运行的 DODAGs 中。本节介绍一组与 DAO 消息有关的参数并提供配置它们的建议。

如第 9-5 节所述，建议延迟发送到 DAO 父代的 DAO 消息，以便尽量扩大执行路由聚合的机会。因此，一旦收到 DAO 消息，节点应当启动 DelayDAO 计时器。默认值是 DEFAULT_DAO_DELAY。RPL 实现可以为配置 DelayDAO 计时器创造条件。

在 Storing 运行模式中，存储节点可以递增 DTSN，以便作为例行路由表更新和维护的一部分，根据节点的直接子代可靠触发一组 DAO 更新。RPL 实现可以为配置一组规定 DTSN 增量触发的规则(人工或基于事件)创造条件。

当 DAO 条目超时或失效时，节点应当合理地尝试向 DAO 的每个父代报告 No-Path。尝试次数可以是可配置的。

实现应当支持限制发送 DAO 消息的速率。相关参数可以是可配置的。

18-2-7 与安全机制有关的 RPL 参数配置

如第 10 章所述，本文档中介绍的安全功能可以有选择的实施，具体的实现可以支持这些安全功能的子集(包括空集)。

所以，支持介绍的安全功能的实现可以在概念上实施安全策略数据库。为了支持安全机制，RPL 实现应当为配置下述参数的子集创造条件：

- 可接受的 Security Modes [无安全模式，预装模式，认证模式]
- 可接受的 KIM 值[Secure RPL 控制消息，在 Security 部分中]
- 可接受的等级值[Secure RPL 控制消息，在 Security 部分中]
- 可接受的算法值[Secure RPL 控制消息，在 Security 部分中]
- 支持认证的或预装的密钥模式的密钥材料。

此外，RPL实现应当为配置含下述参数子集的DODAG根创造条件：

- 通告的等级值[Secure DIO 消息，在 Security 部分中]
- 通告的 KIM 值[Secure DIO 消息，在 Security 部分中]
- 通告的算法值[Secure DIO 消息，在 Security 部分中]

18-2-8 默认值

本文档为下述 RPL 变量规定了默认值：

```
DEFAULT_PATH_CONTROL_SIZE
DEFAULT_DIO_INTERVAL_MIN
DEFAULT_DIO_INTERVAL_DOUBLINGS
DEFAULT_DIO_REDUNDANCY_CONSTANT
```

DEFAULT_MIN_HOP_RANK_INCREASE
DEFAULT_DAO_DELAY

建议在协议中规定默认值；话虽如此，如[RFC5706]中讨论的，默认值可能有越来越少的感觉。RPL 是路由协议，预期使用它的网络环境具有鲜明特点，即节点数量以及链路类型和节点类型预期改变很大。因此，这些默认值极有可能随着场景和技术发展改变。事实上，LLNs 相关技术(例如，硬件，链路层)在过去几年已有长足发展，预计今后几年这些技术会有更大改变和发展。

这些建议值没有基于目前广泛采纳的最优实践，可以说偏于保守。

18-3 监测 RPL 运行

几个 RPL 参数应被监测，以便验证路由协议和网络自身运行正确。本节列出一系列有必要监测的参数。

18-3-1 监测DODAG参数

RPL实现应当提供有关下述参数的信息：

- DODAG Version编号[DIO消息，在DIO Base消息中]
- “G” 标记状态[DIO消息，在DIO Base消息中]
- MOP字段状态[DIO消息，在DIO Base消息中]
- DTSN值[DIO消息，在DIO Base消息中]
- Rank值[DIO消息，在DIO Base消息中]
- DAOSequence：按每个唯一DAO消息增加，在DAO-ACK消息中回应[DAO和DAO-ACK 消息]
- Route Information [DIO消息，Route Information Option](除了生存期和优先权外还包括每个父代的IPv6前缀列表)
- Trickle参数：

* DIOIntervalDoublings [DIO消息，在DODAG Configuration选项中]

* DIOIntervalMin [DIO消息，在DODAG Configuration选项中]

* DIORedundancyConstant[DIO消息，在DODAG Configuration选项中]

- Path Control Size [DIO消息，在DODAG Configuration 选项中]
- MinHopRankIncrease [DIO消息，在DODAG Configuration 选项中]

可以仅对DODAG根监测的值：

- Transit Information [DAO，Transit Information选项]：无论收到的Transit Information选项集合是否应当在DODAG根显示，RPL实现**应当**允许配置。在这种情况下，收到Transit Information 的RPL数据库也应当包含Path Sequence、Path Control、Path Lifetime和Parent Address。

18-3-2 监测 DODAG 不一致和环路检测

在 RPL 网络中检测 DODAG 不一致特别重要。因此，建议 RPL 实现提供适当监测工具。RPL 实现**应当**提供用于报告节点已经检测到相对于 DODAG 父代不一致的次数的计数器，例如，如果 DODAGID 已经改变。如果可能，应当提供不一致检测更细微的信息。RPL 实现**可以**提供报告下述不一致次数的计数器：

- 从具有较高 Rank 的节点收到含“O”位置 1(向下(Down))的分组
- 从具有较低 Rank 的节点收到含“O”为置 0(向上(Up))的分组
- 含“F”位置 1 的分组数量
- 含“R”位置 1 的分组数量

18-4 监测 RPL 数据结构

18-4-1 候选邻居数据结构

在候选邻居列表中的节点是用相同方法发现的节点，是潜在有条件成为父代(有足够高的本地信任)的节点。RPL 实现**应当**提供方法，为用某种反映本地信任的度量(邻居稳定程度)监测候选邻居列表创造条件，这里的本地信任是通过某种度量测定的。

如果候选邻居数量超过最大授权值，RPL 实现可以提供报告候选邻居被忽略的次数的计数器。

18-4-2 目的地导向的有向无循环图(DODAG)表

对于每个 DODAG，预期 RPL 实现保持对下述 DODAG 表值的跟踪：

- RPLInstanceID
- DODAGID
- DODAGVersionNumber
- Rank
- Objective Code Point
- 一组 DODAG 父代
- 一组沿 DODAG 上行(Upward)提供的前缀
- 用于控制 DODAG 的 DIO 消息发送的 Trickle 计时器
- DAO 父代列表
- DTSN
- 节点状态(路由器与叶节点)

RPL 实现应当为监测上面列出的一系列参数创造条件。

18-4-3 路由表和 DAO 路由条目

RPL 实现维护几个与 DODAG 有关的信息元素和 DAO 条目(对于存储节点)。在非存储节点情况，维护有限数量信息(除了如上面提到的 DODAG 特征以外，路由表基本上被缩小到一组 DODAG 父代)；而在存储节点情况，这条信息随路由条目增加。

RPL 实现**应当**为监测下述参数创造条件：

- Next Hop (DODAG 父代)
- Next Hop Interface
- 每个 DODAG 父代的路径度量值

DAO Routing Table 条目在概念上包含下述元素(仅存储节点)：

- 通告的 Neighbor Information
- IPv6 地址
- Interface ID，DAO 父代使这一条目报告给该 Interface ID。
- 重试计数器
- DAO Content 的逻辑等价物：
 - * DAO-Sequence
 - * Path Sequence
 - * DAO Lifetime
 - * DAO Path Control
- Destination Prefix (或地址或 Mcast Group)

RPL 实现**应当**提供有关每条 DAO Routing Table 条目状态的状态信息。

18-5 故障管理

故障管理是用于故障排除、验证协议运行模式正确和验证网络设计的重要部分；它也是网络性能监测的关键部分。RPL 实现**应当**允许提供下述与故障管理有关的信息：

- 存储器溢出及原因(例如，路由表溢出，等)
- 分组因标记为无效而不能发送到 DODAG 父代的次数

- 收到路由器没有相应 RPLInstanceID 的分组的次数
- 本地修复流程被触发的次数
- 由 DODAG 根触发的全局修复次数
- 收到格式不正确消息的次数
- 有转发分组同时没有下一跳(DODAG 父代)的秒数
- 没有下一跳(DODAG 父代)的秒数
- 节点因为收到含不能理解的度量/约束的 DIO 而作为叶节点加入 DODAG 并且在这种情况下节点被配置为按叶节点加入的次数(参阅第 18-6 节)

建议至少通过错误日志消息报告故障。其他协议可用于报告这种故障。

18-6 策略

RPL 实现可使用策略规则来决定是否允许节点加入由邻居通过 DIO 消息通告的特定 DODAG。

本文档规定单个 DODAG 内的运行。DODAG 由下述元组(RPLInstanceID、DODAGID)刻画。

此外，如上所述，DIO 消息用于通告其他 DODAG 特征，如用于建立到 DODAG 的路由度量和约束，以及使用的 Objective Function(由 OCP 规定)。

最重要的策略规则包括规定加入 DODAG 的 RPL 节点必须满足的下述条件：

- RPLInstanceID
- 支持的路由度量和约束列表
- Objective Function (OCP 值)

RPL 实现**必须**允许配置这些参数，**应当**规定，是否如果通告的 DODAG 与本地策略不一致，节点就必须简单忽略该 DIO，或者是否如果只要支持的路由度量和约束列表以及 OF 没有被支持，节点就应当作为叶节点加入。此外，作为策略的一部分，RPL 实现**应当**为添加 DODAGID 创造条件。

RPL 实现**应当**允许为加入 DODAG 的节点，参考 Objective Functions (OFs)的 Objective Code Points (OCPs)，配置可接受的或首选的 OFs 集合，以及如果节点的候选邻居们中没有人通告被配置的、允许的 Objective Functions 中的一个，或者如果通告的度量/约束不被理解/支持，RPL 实现**应当**允许采取哪种行动。在这种情况下可采取两种行动：

- 节点作为叶节点加入 DODAG，如第 8-5 节所述。
- 节点不加入 DODAG。

在 LLN 内的节点可以从包括 RPL 的不同路由协议获知路由信息。在这种情况下，较为理想的是通过管理优先权控制应当首先选择哪条路由。实现**应当**为详细说明路由协议(从该路由协议获知该路由)的管理优先权创造条件。

Internal Data Structures: 某些 RPL 实现可以限制候选邻居列表大小，以便限制存储器使用；在这种情况下，一些本来可用的候选邻居可能不会被考虑，并被从候选邻居列表中简单抛弃。

RPL 实现**可以**提供候选邻居列表大小指示器。

18-7 故障隔离

建议隔离这样的邻居，它们开始以不可接受的速率发送格式错误的消息。

18-8 对其他协议的影响

RPL 对其他协议的影响非常有限。在 RPL 中，路由器上需要不止一种路由协议(如 LBR)，期望设备支持路由协议间的路由重分配功能，以便为两个路由域间的可达创造条件。这样的重分配**应当**通过应用用户可配置策略来管理。

对于网络流量的影响，借助如 Trickle 计时器(参阅第 8-3 节)等机制，RPL 已经被设计成限制控制流量。因此，RPL 对其他协议的影响将非常有限。

18-9 性能管理

性能管理始终是协议的重要方面，RPL 也不例外。IP Performance Monitoring (IPPM)工作组已经规定了几个值得关注的度量：话虽如此，从设备资源和要求带宽角度考虑监测这些度量的成本，这些度量很难在 LLN 中应用。然而，RPL 实现可以支持它们中的一部分，其他值得关注的参数如下：

- 修复次数和以秒计的修复时间(平均值，方差)
- 因在设备路由表中缺乏可达邻居，使设备不能转发分组的持续时间和出现这种情况的次数
- 监测 RPL 在带宽和要求的存储器方面消耗的资源
- 发送和接收的 RPL 控制消息数量

18-10 诊断

节点有可能运行于“罗嗦”模式，以改善诊断。因此，RPL 实现应当有能力将节点置于罗嗦模式和使节点退出罗嗦模式的能力，以便获得额外诊断信息。

第 19 章 安全考虑

19-1 综述

从安全角度看，RPL 网络与任何其他网络没有什么不同。RPL 网络很容易受到被动窃取攻击，如果物理上访问到导线，主动篡改甚至不需要参与到通信中。正是自组织(ad hoc)网络的性质和它们的成本目标强加了额外的安全限制，这些限制或许使得这些网络成为最困难的安全环境。设备是低成本的，设备在计算能力、可用存储器和续航能力上有限；不能总是假设在运载工具上设备有值得信任的计算基础或高质量的随机数发生器。通信不能依靠固定基础设施的在线可用性，并且或许涉及之前从没有通信过设备间的短期联系。这些约束或许严重限制密码算法和协议的选择，影响安全架构设计，因为需要仔细处理设备间信任关系的建立和维护。此外，电池寿命和成本极大限制了这些网络能够承受的安全开销，对某些事情的关注远不如有较高带宽的网络。这些安全架构要素的大多数可以在较高层实现，因此，可以考虑排除在本规范之外。然而，需要特别关注到这些较高层接口的实现。

在本标准中的安全机制基于对称密钥和公共密钥加密，使用由较高层进程提供的密钥。这些密钥的建立和维护超出本规范范围。这些机制假设能够安全实施加密操作和密钥材料的安全可信保存。

规定的安全机制提供下述安全业务的特定组合：

数据保密：保证发送的信息只透露给信息当事人。

数据真实性：发送信息来源保证(并且，因此，信息在传送过程中没有被修改)。

重放保护：保证删除重复发送的信息。

及时性(延迟保护)：保证及时收到发送的信息。

提供的实际保护可以基于每个分组实施，并为改变数据真实性等级(尽量减小发送分组中需要的安全开销)和可选的数据保密提供方便。如果要求非一般的保护，始终提供重放保护。

通过在分组保护进程中使用非重复值(CCM 随机数)和保存某些状态信息(从那台设备最后收到的始发设备和 CCM 随机数计数器)来提供重放保护，这能够检测这个特定 CCM 随机数值是否之前由该始发设备使用过。此外，在那些机上配有并非严格的同步时钟的设备上提供所谓的延迟保护。可接受的时间延迟可以基于每个分组实施，并为改变延迟时间(便于在多跳通信路径上发送的分组中加长延迟时间)提供方便。

密码保护可在两个对等设备间使用共享密钥(链路密钥), 或者在一组设备间使用共享密钥(组密钥), 从而相对于提供的密码保护, 在密钥保存成本和密钥维护成本间提供某种灵活性和特定应用权衡。如果组密钥用于对等通信, 仅提供对外部设备的防范, 不能防范在密钥共享组内的潜在恶意设备。

可使用基于对称密钥的技术或基于公共密钥的技术提供数据真实性。采用基于公共密钥的技术(通过签名), 根据所发信息的唯一发起人, 可以证实证据, 而采用基于对称密钥的技术, 仅相对于密钥共享组中的设备提供数据真实性。因此, 在要求比仅能提供基于对称密钥认证技术更细粒度认证的场合, 如采用组通信(广播, 多播)或要求不可否认的场合, 会使用基于公共密钥的认证。

第 20 章 IANA 考虑

20-1 RPL 控制消息

RPL 控制消息是 ICMP 信息消息类型, 用于携带支持 RPL 运行的 DODAG Information Objects、DODAG Information Solicitations 和 Destination Advertisement Objects。

IANA 已经定义了 ICMPv6 Type Number Registry。RPL 控制消息的类型值是 155。

20-2 RPL 控制代码的新注册表

IANA 已经为 ICMPv6 RPL 控制消息的 Code 字段建立了注册表、RPL Control Codes。

新代码仅可以由 IETF Review 分配。每个代码用下述属性跟踪:

- Code

- 描述
- Defining RFC

目前定义的代码见附表 1。

附表 1: RPL Control Codes

代码	描述	参考
0x00	DODAG Information Solicitation	本文档
0x01	DODAG Information Object	本文档
0x02	Destination Advertisement Object	本文档
0x03	Destination Advertisement Object Acknowledgment	本文档
0x80	Secure DODAG Information Solicitation	本文档
0x81	Secure DODAG Information Object	本文档
0x82	Secure Destination Advertisement Object	本文档
0x83	Secure Destination Advertisement Object Acknowledgment	本文档
0x8A	Consistency Check	本文档

20-3 运行模式(MOP)的新注册表

IANA 已经为 3 位运行模式(Mode of Operation, MOP)建立了注册表。该注册表包含在 DIO Base 中。

新值仅可以由 IETF Review 分配。每个值用下述属性跟踪：

- Mode of Operation Value

- 能力描述
- Defining RFC

目前已定义的 4 个值参见附表 2。

附表 2 DIO Mode of Operation

MOP 值	描述	参考
0	没有由 RPL 维护的下行(Downward)路由	本文档
1	Non – Storing 运行模式	本文档
2	没有多播支持的 Storing 运行模式	本文档
3	有多播支持的 Storing 运行模式	本文档

其余部分，即十进制数 4 到 7，目前没有分配。

20-4 RPL 控制消息选项

IANA 已经为 RPL Control Message Options 建立了注册表。

新值仅可以由 IETF Review 分配。每个值用下述属性跟踪：

- Value
- Meaning
- Defining RFC

RPL Control Message Options 如附表 3 所示。

附表 3 RPL Control Message Options

值	含意	参考
0x00	Pad1	本文档
0x01	PanN	本文档
0x02	DAG Metric Container	本文档
0x03	Routing Information	本文档
0x04	DODAG Configuration	本文档
0x05	RPL Target	本文档
0x06	Transit Information	本文档
0x07	Solicited Information	本文档
0x08	Prefix Information	本文档
0x09	Target Descriptor	本文档

20-5 目标代码点(OCP)注册表

IANA 已经建立了管理 Objective Code Point (OCP)字段代码空间的注册表。

本规范没有定义 OCPs。

新代码仅可以由 IETF Review 分配。每个代码用下述属性跟踪：

- Code
- 描述
- Defining RFC

20-6 安全部分算法的新注册表

IANA 已经为 Security 部分中 8 位 Algorithm 字段的值建立了注册表。

新值仅可以由 IETF Review 分配。每个值用下述属性跟踪：

- Value
- Encryption/MAC
- Signature
- Defining RFC

目前定义的值如附表 4 所示。

附表 4 Security Section Algorithm

值	加密/MAC	签名	参考
0	采用 AES-128 的 CCM	采用 SHA-256 的 RSA	本文档

20-7 安全部分标记的新注册表

IANA 已经为 8 位 Security Section Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述
- Defining RFC

目前没有为 Security Section Flags 字段定义位。

20-8 Per-KIM 安全等级的新注册表

IANA 已经为每个已分配 KIM 值的 3 位 Security Level (LVL)字段建立了一个注册表。

对于给定的 KIM 值，新等级仅可以由 IETF Review 分配。每个等级用下述属性跟踪：

- Level
- KIM 值

- 描述
- Defining RFC

目前定义的每个 KIM 值的等级参见附表 5。

附表 5 Per-KIM Security Levels

等级	KIM 值	描述	参考
0	0	参阅图 11	本文档
1	0	参阅图 11	本文档
2	0	参阅图 11	本文档
3	0	参阅图 11	本文档
0	1	参阅图 11	本文档
1	1	参阅图 11	本文档
2	1	参阅图 11	本文档
3	1	参阅图 11	本文档
0	2	参阅图 11	本文档
1	2	参阅图 11	本文档
2	2	参阅图 11	本文档
3	2	参阅图 11	本文档
0	3	参阅图 11	本文档
1	3	参阅图 11	本文档
2	3	参阅图 11	本文档
3	3	参阅图 11	本文档

20-9 DODAG 信息请求(DIS)标记的新注册表

IANA 已经为 DIS (DODAG Informational Solicitation) Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述
- Defining RFC

目前没有为 DIS (DODAG Informational Solicitation) Flags 字段定义位。

20-10 DODAG 信息对象(DIO)标记的新注册表

IANA 已经为 8 位 DODAG Information Object (DIO) Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述
- Defining RFC

目前没有为 DIS (DODAG Informational Solicitation) Flags 字段定义位。

20-11 目的地通告对象(DAO)标记的新注册表

IANA 已经为 8 位 Destination Advertisement Object (DAO) Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述
- Defining RFC

目前定义的位参阅附表 6。

附表 6 DAO Base Flags

Bit 编号	描述	参考
0	DAO-ACK 请求(K)	本文档
1	DODAGID 字段存在(D)	本文档

20-12 目的地通告对象(DAO)确认标记的新注册表

IANA 已经为 8 位 Destination Advertisement Object (DAO) Acknowledgement Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述
- Defining RFC

目前定义的位如附表 7 所示。

附表 7 DAO-ACK Base Flags

Bit 编号	描述	参考
0	DODAGID 字段存在(D)	本文档

20-13 一致性检验(CC)标记的新注册表

IANA 已经为 8 位 Consistency Check (CC) Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述

- Defining RFC

目前定义的位如附表 8 所示。

附表 8 Consistency Check Base Flags

Bit 编号	描述	参考
0	CC Response (R)	本文档

20-14 DODAG 配置选项标记的新注册表

IANA 已经为 8 位 DODAG Configuration Option Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述
- Defining RFC

目前定义的位如附表 9 所示。

附表 9 DODAG Configuration Option Flags

Bit 编号	描述	参考
4	Authentication Enabled (A)	本文档
5-7	Path Control Size (PCS)	本文档

20-15 RPL 目标选项标记的新注册表

IANA 已经为 8 位 RPL Target Option Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述

- Defining RFC

目前没有为 RPL Target Option Flags 定义位。

20-16 传送信息选项标记的新注册表

IANA 已经为 8 位 Transit Information Option (TIO) Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述
- Defining RFC

目前定义的位如附表 10 所示。

附表 10 Transit Information Option Flags

Bit 编号	描述	参考
0	External (E)	本文档

20-17 请求信息选项标记的新注册表

IANA 已经为 8 位 Solicited Information Option (SIO) Flags 字段建立了注册表。

新的位编号仅可以由 IETF Review 分配。每个位用下述属性跟踪：

- Bit 编号(从作为最高有效位的第 0 位计数)
- 能力描述
- Defining RFC

目前定义的位如附表 11 所示。

附表 11 Solicited Information Option Flags

Bit 编号	描述	参考
0	Version Predicate 匹配(V)	本文档
1	InstanceID Predicate 匹配(I)	本文档
2	DODAGID Predicate 匹配(D)	本文档

20-18 ICMPv6：源路由首部中的错误

某些情况，如果消息不能按照它的源路由首部规定的交付，RPL 将返回 ICMPv6 出错消息。这个 ICMPv6 出错消息为 “Error in Source Routing Header” 。

IANA已经为ICMPv6 Message Types定义了ICMPv6 “代码” Fields Registry。ICMPv6 Message Type 1描述 “Destination Unreachable” 代码。根据ICMPv6 Message Type 1的 ICMPv6 Code Fields Registry，已经分配 “Error in Source Routing Header” 代码，代码值为7。

20-19 本地链路范围多播地址

分配新 IPv6 多播地址的规则在[RFC3307]中定义。本规范需要为称作全 RPL 节点 (all-RPL-nodes)、具有值为 ff02::1a 的 RPL 节点，分配具有本地链路范围的新永久多播地址。

第 21 章 致谢

本文档作者感谢来自下述诸位的审阅、反馈和意见：Emmanuel Baccelli、Dominique Barthel、Yusuf Bashir、Yoav Ben-Yehezkel、Phoebus Chen、Quynh Dang、Mischa Dohler、Mathilde Durvy、Joakim Eriksson、Omprakash Gnawali、Manhar Goindi、Mukul Goyal、Ulrich Herberg、Anders Jagd、JeongGil (John) Ko、Ajay Kumar、Quentin Lampin、Jerry Martocci、Matteo Paris、Alexandru Petrescu、Joseph Reddy、Michael Richardson、Don Sturek、Joydeep Tripathi和Nicolas Tsiftes。

本文档作者感谢 ROLL 主席、David Culler 和 JP. Vasseur，以及区域总监 Adrian Farrel 的指导和关注。

本文档作者感谢以下诸位先前做出的贡献: Robert Assimiti、Mischa Dohler、Julien Abeille、Ryuji Wakikawa、Teco Boot、Patrick Wetterwald、Bryan McLaughlin、Carlos J. Bernardos、Thomas Watteyne、Zach Shelby、Caroline Bontoux、Marco Molteni、Billy Moon、Jim Bound、Yanick Pouffary、Henning Rogge和Arsalan Tavakoli, 他们对RPL的设计提供了有益的贡献。

第 10 章、第 19 章和散布在整个文档中的 RPL Security Design, 主要由 Security Design Team 完成, 该团队成员包括 Tzeta Tsao、Roger Alexander、Dave Ward、Philip Levis、Kris Pister、Rene Struik 和 Adrian Farrel。

还要感谢 Jari Arkko 和 Ralph Droms, 感谢他们细心校阅, 尤其是关于互操作性考虑和与其他 IETF 规范的衔接。

第22章 撰稿人

Stephen Dawson-Haggerty
UC Berkeley
Soda Hall
Berkeley, CA 94720
USA
EMail: stevedh@cs.berkeley.edu

第23章 参考文献

23-1 标准类参考文献

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.
- [RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.
- [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, March 2012.
- [RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.
- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, March 2012.
- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 2012.

23-2 信息类参考文献

- [6LOWPAN-ND] Shelby, Z., Ed., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low Power and Lossy Networks (6LoWPAN)", Work in Progress, October 2011.
- [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>.

- [Perlman83] Perlman, R., "Fault-Tolerant Broadcast of Routing Information", North-Holland Computer Networks, Vol.7: p. 395-405, December 1983.
- [RFC1958] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.
- [RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, August 1996.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC3307] Haberman, B., "Allocation Guidelines for IPv6 Multicast Addresses", RFC 3307, August 2002.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [RFC3535] Schoenwaelder, J., "Overview of the 2002 IAB Network Management Workshop", RFC 3535, May 2003.
- [RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.
- [RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.
- [RFC4101] Rescorla, E. and IAB, "Writing Protocol Models", RFC 4101, June 2005.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, June 2007.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-IS)", RFC 5120, February 2008.

- [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.
- [RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, November 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. Vermeulen, "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.
- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.
- [ROLL-TERMS] Vasseur, J., "Terminology in Low power And Lossy Networks", Work in Progress, September 2011.

附录 A：操作举例

本附录提供一些例子来说明采用 RPL 传播寻址信息和前缀。这些例子描述采用 PIOs 和 RIOs 分配的信息，以及 DIO 消息和 DAO 消息的使用。注意，这个附录不是规范，RPL 寻址规划和自动配置具体细节可根据具体实现修改。RPL 仅提供分发信息的运载工具，其上的信息可以由其他机制建立和使用。

注意，这些例子演示了由 RPL 内分布的信息支持的地址自动配置方案。然而，如果实现包括另一种地址自动配置方案，RPL 节点或许被配置为在 PIO 选项中不将“A”标记置 1，尽管 PIO 仍然可用作分发前缀和寻址信息。

A-1 采用含有节点自有前缀的存储模式的运行举例

图 32 显示采用 Storing 模式运行的简单 RPL 网络逻辑寻址架构。在这个例子中，每个节点，即 Node A、B、C 和 D 都拥有自己的前缀，每个节点都使该前缀可由 on-link 设备用于地址自动配置。(这是通过在 DIO 消息的 PIO 中将“A”标记和“L”标记置 1 实现的)。Node A 拥有前缀 A::/64，Node B 拥有 B::/64，等等。Node B 相对于 Node A 自动配置 on-link 地址，A::B。类似，Nodes C 和 D 根据 Node B 的前缀，分别自动配置 on-link 地址 B::C 和 B::D。这些节点有“R”标记置 1 的选项，并在 PIO 的 Prefix 字段内发布它们的地址。

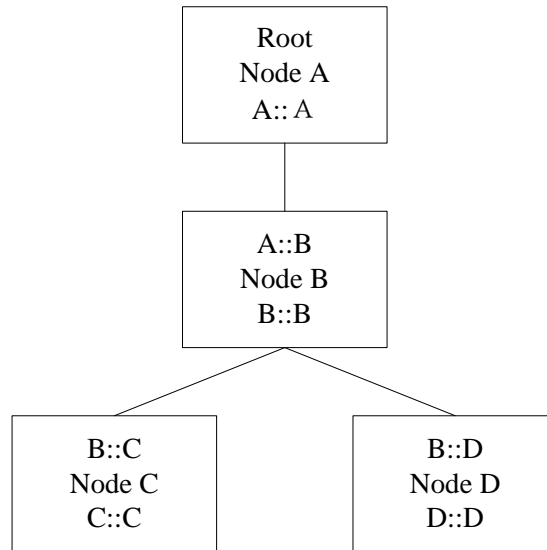


图 32 含有 Node-Owned Prefixes 的 Storing Mode

A-1-1 DIO 消息和 PIO

例如，Node A 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 1

“R” 标记：置 0

Prefix Length: 64

Prefix: A::

例如，Node B 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 1

“R” 标记：置 1

Prefix Length: 64

Prefix: B::B

例如，Node C 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 1

“R” 标记：置 0

Prefix Length: 64

Prefix: C::

例如，Node D 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 1

“R” 标记：置 1

Prefix Length: 64

Prefix: D::D

A-1-2 DAO 消息

Node B 将发送带有下列信息的 DAO 消息到 Node A：

- Target B::/64
- Target C::/64
- Target D::/64

Node C 将发送带有下列信息的 DAO 消息到 Node B：

- Target C::/64

Node D 将发送带有下列信息的 DAO 消息到 Node B：

- Target D::/64

A-1-3 路由信息库

Node A 在概念上将收集下列信息到它的 Routing Information Base (RIB)：

- 连接的 A::/64
- 通过 B 本地链路的 B::/64
- 通过 B 本地链路的 C::/64
- 通过 B 本地链路的 D::/64

Node B 在概念上将收集下列信息到它的 RIB：

- 通过 A 本地链路的 ::0
- 连接的 B::/64
- 通过 C 本地链路的 C::/64
- 通过 D 本地链路的 D::/64

Node C 在概念上将收集下述信息到它的 RIB:

- 通过 B 本地链路的 ::/0
- 连接的 C::/64

Node D 在概念上将收集下述信息到它的 RIB:

- 通过 B 本地链路的 ::/0
- 连接的 D::/64

A-2 采用含有宽子网前缀的存储模式的运行举例

图 33 显示采用 Storing 模式运行的简单 RPL 网络逻辑寻址架构。在这个例子中,根 Node A 发起前缀,该前缀用于在整个 RPL 子网上地址自动配置。(这是通过在 DIO 消息的 PIO 中将“A”标记置 1,“L”标记置 0 实现的。)Nodes A、B、C 和 D 都自动配置到前缀 A::/64。这些节点有“R”标记置 1 的选项,并在 PIO 的 Prefix 字段内发布它们的地址。

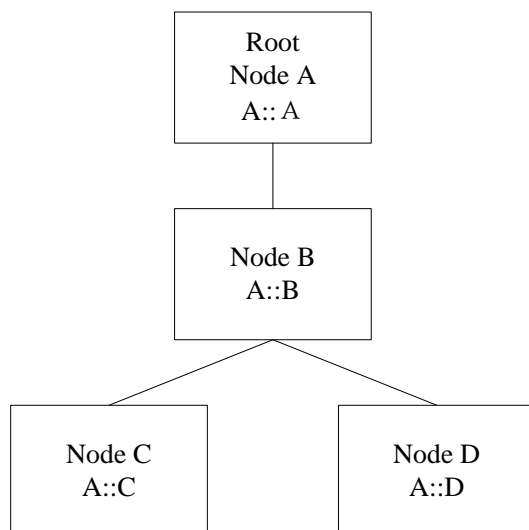


图 33 含有 Subnet-Wide Prefix 的 Storing Mode

A-2-1 DIO 消息和 PIO

例如，Node A 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置1

“L” 标记：置 0

“R” 标记：置 0

Prefix Length: 64

Prefix: A::

例如，Node B 将发 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置1

“L” 标记：置 0

“R” 标记：置 1

Prefix Length: 64

Prefix: A::B

例如，Node C 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 0

“R” 标记：置 0

Prefix Length: 64

Prefix: A::

例如，Node D 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 0

“R” 标记：置 1

Prefix Length: 64

Prefix: A::D

A-2-2 DAO 消息

Node B 将发送具有下述信息的 DAO 消息到 Node A：

- Target A::B/128
- Target A::C/128
- Target A::D/128

Node C 将发送具有下述信息的 DAO 消息到 Node B：

- Target A::C/128

Node D 将发送具有下述信息的 DAO 消息到 Node B：

- Target A::D/128

A-2-3 路由信息库

Node A 在概念上将收集下述信息到它的 RIB：

- 连接的 A::A/128
- 通过 B 本地链路的 A::B/128
- 通过 B 本地链路的 A::C/128
- 通过 B 本地链路的 A::D/128

Node B 在概念上将收集下述信息到它的 RIB：

- 通过 A 本地链路的 ::/0
- 连接的 A::B/128
- 通过 C 本地链路的 A::C/128
- 通过 D 本地链路的 A::D/128

Node C 在概念上将收集下述信息到它的 RIB:

- 通过 B 本地链路的::/0
- 连接的 A::C/128

Node D 在概念上将收集下述信息到它的 RIB:

- 通过 B 本地链路的::/0
- 连接的 A::D/128

A-3 采用含有节点自有前缀的非存储模式的运行举例

图 34 显示采用 Non-Storing 模式运行的简单 RPL 网络逻辑寻址架构。在这个例子中，每个节点，即 Node A、B、C 和 D 都拥有自己的前缀，每个节点都使该前缀可由 on-link 设备用于地址自动配置。(这是通过在 DIO 消息的 PIO 中将“A”标记和“L”标记置 1 实现的)。Node A 拥有前缀 A::/64，Node B 拥有 B::/64，等等。Node B 相对于 Node A 自动配置 on-link 地址，A::B。类似，Nodes C 和 D 根据 Node B 的前缀，分别自动配置 on-link 地址 B::C 和 B::D。这些节点有“R”标记置 1 的选项，并在 PIO 的 Prefix 字段内发布它们的地址。

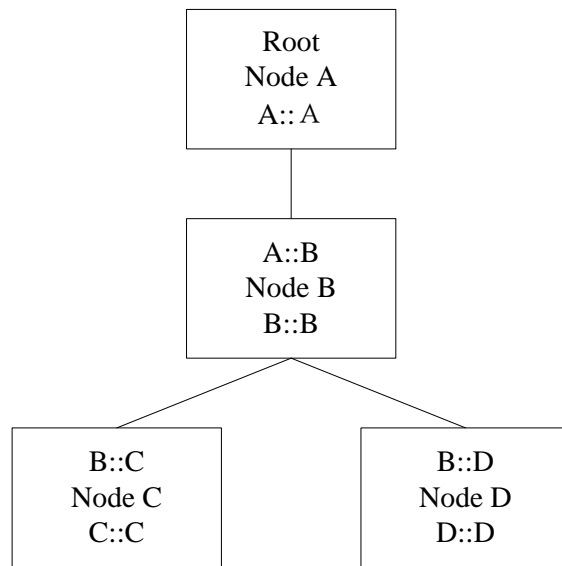


图 34 含有 Node-Owned Prefixes 的 Non-Storing Mode

A-3-1 DIO 消息和 PIO

在采用含节点自有前缀的 Non-Storing 模式的 DIO 消息中包含的 PIO，可认为等同于采用含节点自有前缀的 Storing 模式的 DIO 消息中包含的 PIO (附录 A-1-1)。

A-3-2 DAO 消息

Node B 将发送具有下述信息的 DAO 消息到 Node A:

- Target B::/64, Transit A::B

Node C 将发送具有下述信息的 DAO 消息到 Node A:

- Target C::/64, Transit B::C

Node D 将发送具有下述信息的 DAO 消息到 Node A:

- Target D::/6, Transit B::D

A-3-3 路由信息库

Node A 在概念上将收集下述信息到它的 RIB。注意，通过递归查询到 RIB，Node A 有足够信息构建源路由:

- 连接的 A::/64
- 通过 A::B 的 B::/64
- 通过 B::C 的 C::/64
- 通过 B::D 的 D::/64

Node B 在概念上将收集下述信息到它的 RIB:

- 通过 A 本地链路的 ::/0
- 连接的 B::/64

Node C 在概念上将收集下述信息到它的 RIB:

- 通过 B 本地链路的 ::/0
- 连接的 C::/64

Node D 在概念上将收集下述信息到它的 RIB:

- 通过 B 本地链路的 ::/0
- 连接的 D::/64

A-4 采用含有宽子网前缀的非存储模式的运行举例

图 35 显示采用 Non-Storing 模式运行的简单 RPL 网络逻辑寻址架构。在这个例子中，根 Node A 发起前缀，该前缀用于在整个 RPL 子网上地址自动配置。(这是通过在 DIO 消息的 PIO 中将“A”标记置 1，“L”标记置 0 实现的。) Nodes A、B、C 和 D 都自动配置到前缀 A::/64。这些节点必须将“R”标记置 1，并在 PIO 的 Prefix 字段内发布它们的地址，以便通知它们的子代，哪些地址在此传输选项中使用。

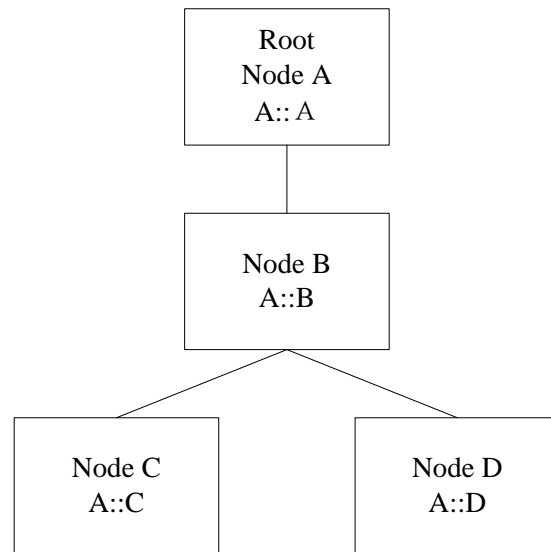


图 35 含有 Subnet-Wide Prefix 的 Non-Storing Mode

A-4-1 DIO 消息和 PIO

例如，Node A 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 0

“R” 标记：置 1

Prefix Length: 64

Prefix: A::A

例如，Node B 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 0

“R” 标记：置 1

Prefix Length: 64

Prefix: A::B

例如，Node C 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 0

“R” 标记：置 1

Prefix Length: 64

Prefix: A::C

例如，Node D 将发送 DIO 消息，该消息具有如下所述的 PIO：

“A” 标记：置 1

“L” 标记：置 0

“R” 标记：置 1

Prefix Length: 64

Prefix: A::D

A-4-2 DAO 消息

Node B 将发送具有下述信息的 DAO 消息到 Node A：

- Target A::B/128, Transit A::A

Node C 将发送具有下述信息的 DAO 消息到 Node A：

- Target A::C/128, Transit A::B

Node D 将发送具有下述信息的 DAO 消息到 Node A：

- Target A::D/128, Transit A::B

A-4-3 路由信息库

Node A 在概念上将收集下述信息到它的 RIB。注意，通过递归查询到 RIB，Node A 有足够信息构建源路由：

- 连接的 A::A/128
- 通过 A::A 的 A::B/128
- 通过 A::B 的 A::C/128
- 通过 A::B 的 A::D/128

Node B 在概念上将收集下述信息到它的 RIB：

- 通过 A 本地链路的 ::/0
- 连接的 A::B/128

Node C 在概念上将收集下述信息到它的 RIB：

- 通过 B 本地链路的 ::/0
- 连接的 A::C/128

Node D 在概念上将收集下述信息到它的 RIB:

- 通过 B 本地链路的::/0
- 连接的 A::D/128

A-5 采用外部前缀举例

考虑图 36 显示的简单网络。在这个例子中，有一组参与 RPL 网络的路由器：DODAG 根、Node A、Y 和 Z。DODAG 根和 Node Z 同时还连接到不同的外部网络域(即，到 RPL 网络外部)。注意，这些外部网络可以是 RPL 网络或者完全是另一种类型的网络。

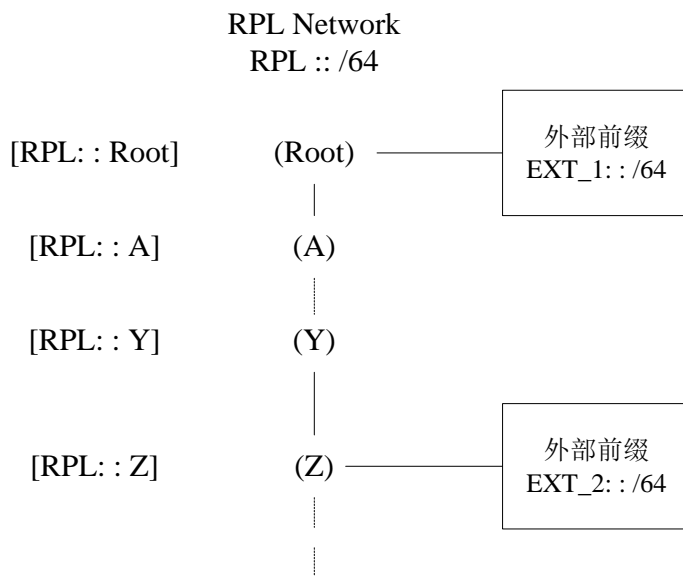


图 36 简单网络举例

在这个例子中，DODAG 根产生可用于地址自动配置的前缀。这里，整个 RPL 子网使用那个相同的前缀，RPL::/64，进行地址自动配置，然而在其他实现中可以使用更复杂/混合的方案。

DODAG 根连接到外部(相对于那个 RPL 网络)前缀 EXT_1::/64。例如,通过非 RPL 接口上显式配置或通过 IPv6 ND, DODAG 根可以获知到这个前缀的连通性。DODAG 根被配置公布到这个前缀的连接信息。

类似，Node Z 连接到外部前缀 EXT_2::/64。Node Z 也有在它下面的子 DODAG。

- 1、DODAG 根添加 RIO 到它的 DIO 消息。此 RIO 包含外部前缀 EXT_1::/64。这条信息可以在该 DODAG 内由其他节点发出的 DIO 消息中复制。因此，前缀 EXT_1::/64 的可达性沿该 DODAG 向下传播。
- 2、通过在Target选项中发送用EXT_2::/64为Target 的DAO消息，和在Transit Information 选项中发送用其自身(Node Z)为父代的DAO消息，Node Z可以通告到Target网络 EXT_2::/64的可达性。(在Storing模式，Transit Information选项不需要包括Node Z的地址)。因此，非存储根意识到构建源路由使用1跳链路(Node Z --- EXT_2::/64)。Node Z可以向在它的子DODAG中的节点补充通告它到EXT_2::/64的可达性，这是通过发送有PIO，有“A”标记置0的DIO消息实现的。

作者通讯录

Tim Winter (editor)

EMail: wintert@acm.org

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

Anders Brandt

Sigma Designs

Emdrupvej 26A, 1.

Copenhagen DK-2100

Denmark

EMail: abr@sdesigns.dk

Jonathan W. Hui

Arch Rock Corporation

501 2nd St., Suite 410

San Francisco, CA 94107

USA

EMail: jhui@archrock.com

Richard Kelsey

Ember Corporation

Boston, MA

USA

Phone: +1 617 951 1225

EMail: kelsey@ember.com

Philip Levis
Stanford University
358 Gates Hall, Stanford University
Stanford, CA 94305-9030
USA

E-Mail: pal@cs.stanford.edu

Kris Pister
Dust Networks
30695 Huntwood Ave.
Hayward, CA 94544
USA

E-Mail: kpister@dustnetworks.com

Rene Struik
Struik Security Consultancy

E-Mail: rstruik.ext@gmail.com

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

E-Mail: jpv@cisco.com

Roger K. Alexander
Cooper Power Systems
20201 Century Blvd., Suite 250
Germantown, MD 20874
USA

Phone: +1 240 454 9817

E-Mail: roger.alexander@cooperindustries.com