RPL: The IP routing protocol designed for low power and lossy networks

Internet Protocol for Smart Objects (IPSO) Alliance

JP Vasseur, Cisco Fellow, Cisco Systems
Navneet Agarwal, Technical Leader, Cisco Systems
Jonathan Hui, Software Engineer, Cisco Systems
Zach Shelby, Chief Nerd, Sensinode
Paul Bertrand, Founder, VP, Watteco SAS
Cedric Chauvenet, Watteco SAS

April 2011

1. Introduction: the unique routing requirements of IP smart object networks

IP smart object networks are undoubtedly one of the key components of the next wave of the Internet, with an endless number of new opportunities and applications thanks to newly designed IP based protocols (see [1], [2] and [3]).

Still such networks also present a number of technical challenges that have been explored in various IPSO white papers. The aim of this white paper is to exclusively focus on the routing aspects of IP Smart Object Networks.

Such technical challenges include issues related to power consumption, small form factors and communication challenges (low speed, high error rates, …) used to form networks. The challenges are further complicated as there is interdependence between the issues. For example, the level of communication within the smart objects networks impacts the power consumption in the smart object. Also, the network protocol design should be cognizant of the power consumption and how much data to send. Additional factors such as cost, available power and form factor of the device limits the amount of computing resources that can be put in the smart object. For example, a typical off-the-shelf available smart object would have only tens of kilobytes of RAM/flash with small micro-controllers/processors. Thus the software on the smart objects must not only be power-efficient but must be able to run in a small memory footprint device.

Smart object networks are potentially very large scale consisting of potentially (hundreds of) thousands of nodes often operating in harsh and remote environments. Individual networks built with thousands of smart objects are common. Data collection is typically sampled less frequently but the reporting, collection and analysis of this data leads to scaling issues as the network is designed to work for years. Note also that such nodes are usually unattended and must support some forms of auto-configuration and management.

Another factor related to power consumption is the communication medium of the smart object network. In order to optimize on the power consumption these devices normally use media operating on low power communication standards. These could include low-power wireless communication as well as power line communication protocols where communication happens over the same set of media that carries electricity. It is worth noting some low power link layers designed for LLNs that are working with RPL: IEEE 802.15.4, Wavenis, IEEE P1901.2, ITU G.hnem.... (see [4]). The communication over this type of media is unreliable as it is uncertain if the intended device received a message sent by another device. The message could have been disrupted partially or completely due to physical obstruction in the 'line-of-sight' or due to various sources of noise and interference. The unreliable nature of the communications in the context of smart object networks is referred to as being 'lossy' and this is one of the inherent characteristics, which should be taken into account during any software or communication/network protocol design. Such networks are also referred to as Low power and Lossy Networks (LLNs).

Routing is the process by which the network determines what path(s) the messages should take through the network. Routing in LLN has to be cognizant of the above issues and treat this as input requirements for design. The routing protocol design in this type of network should be sensitive to how much data a network can handle, the speed and the devices' capabilities. For example, in smart object networks consisting of battery-powered nodes, the act of communication consumes energy and nodes that communicate more frequently drain their energy faster.

LLNs are known to be lossy, as we have seen. This "lossyness" may be transient and unpredictable. Thus the routing protocol must be robust and be prepared to deal with these network characteristics. In traditional networks any loss of connectivity triggers a desire to quickly re-converge and find alternate routing paths. This is desirable so that data traffic is re-routed around network failures as quickly as possible and with as little loss of data. To that end, a number of techniques such as Fast Reroute for IP/MPLS have been designed for link state protocols such as OSPF or ISIS. This action is seen as overkill in LLNs due to the lossyness being transient and would unavoidably lead to lack of stability and unacceptable control plane overhead. A preferred model should be to 'under-react' to smooth over the transient loss of connectivity and have a confidence-monitoring model before triggering a full re-convergence.

Furthermore, routing in LLNs should be able to self manage to a large extent and be able to heal itself without requiring manual intervention. For example, it is not possible for a system administrator to assign an address manually or be able to enter passwords for accessing the network. In addition to large addressing spaces provided by IPv6, such auto-configuration capability (known as IPv6 stateless auto-configuration) make IPv6 an ideal candidate for LLNs.

Smart object networks operate on a variety of link types having variable quality, which is unpredictable due to various surrounding environment factors. Unlike traditional links that have a low bit-error rate (BER) the packet delivery ratios (PDR) in LLNs show wide variations. This is true for not only wireless links but PLC links as well which are impacted by impedance variations, interferences etc. Figure 1 shows a typical Packet Delivery Ratio over a low power wireless link.

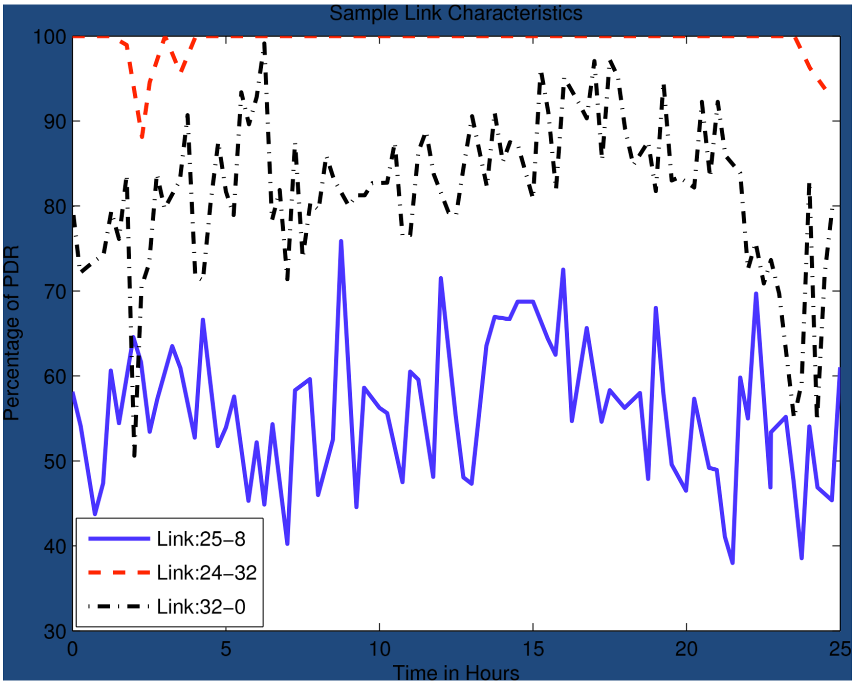


Figure 1 – PDR Variation over time on IEEE 802.15.4

The link quality has direct implication on routing protocol design with respect to convergence time. While traditional routing protocols are designed to minimize the convergence time due to the voice and video traffic requirements a similar approach for LLN would potentially lead to routing instabilities, various oscillations and routing loops. Furthermore, smart objects do not send a large amount of traffic unlike voice and video traffic on high-speed IP networks. In these environments, it is a reasonable expectation that during transient instabilities the traffic is locally redirected to an alternate next hop without triggering a global re-convergence.

Another aspect of these networks is the dynamic nature of the metrics. Traditional routing protocols have experimented and avoided the use of dynamic metrics due to risk of route oscillations and network instability. But in LLNs, node and link metrics do change over a period of time (link quality going down due to interference, CPU overloading, node switching from line power to battery power, etc) and the routing protocol should be able to adapt to it.

2. RPL: an open routing protocol for IP smart object networks standardized by the IETF

The Internet Engineering Task Force (IETF) quickly recognized the need to form a new Working Group to standardize an IPv6-based routing solution for IP smart object networks, which led to the formation of a new Working Group called ROLL (Routing Over Low power and Lossy) networks in 2008: http://www.ietf.org/html.charters/roll-charter.html.

The ROLL Working Group conducted a detailed analysis of the routing requirements focusing on several applications: urban networks including smart grid, industrial automation, home and building automation. This set of applications has been recognized to be sufficiently wide to cover most of the applications of the Internet of Things. The objective of the WG was to design a routing protocol for LLNs, supporting a variety of link layers, sharing the common characteristics of being low bandwidth, lossy and low power. Thus the routing protocol should make no specific assessment on the link layer, which could either be wireless such as IEEE 802.15.4, IEEE 802.15.4g, (low power) Wifi or Powerline Communication (PLC) using IEEE 802.15.4 such as IEEE P1901.2..

The result of this Working Group was the “Ripple” routing protocol (RPL) specification, along with supporting specifications on routing metrics, objective functions and security. The rest of this document gives an introduction to RPL and these related specifications.

Note that RPL operates at the IP layer according to the IP architecture, and thus allows for routing across multiple types of link layers, in contrast with other form of “routing” operating at lower layer (e.g. link layers).

3. An overview of RPL mode of operation

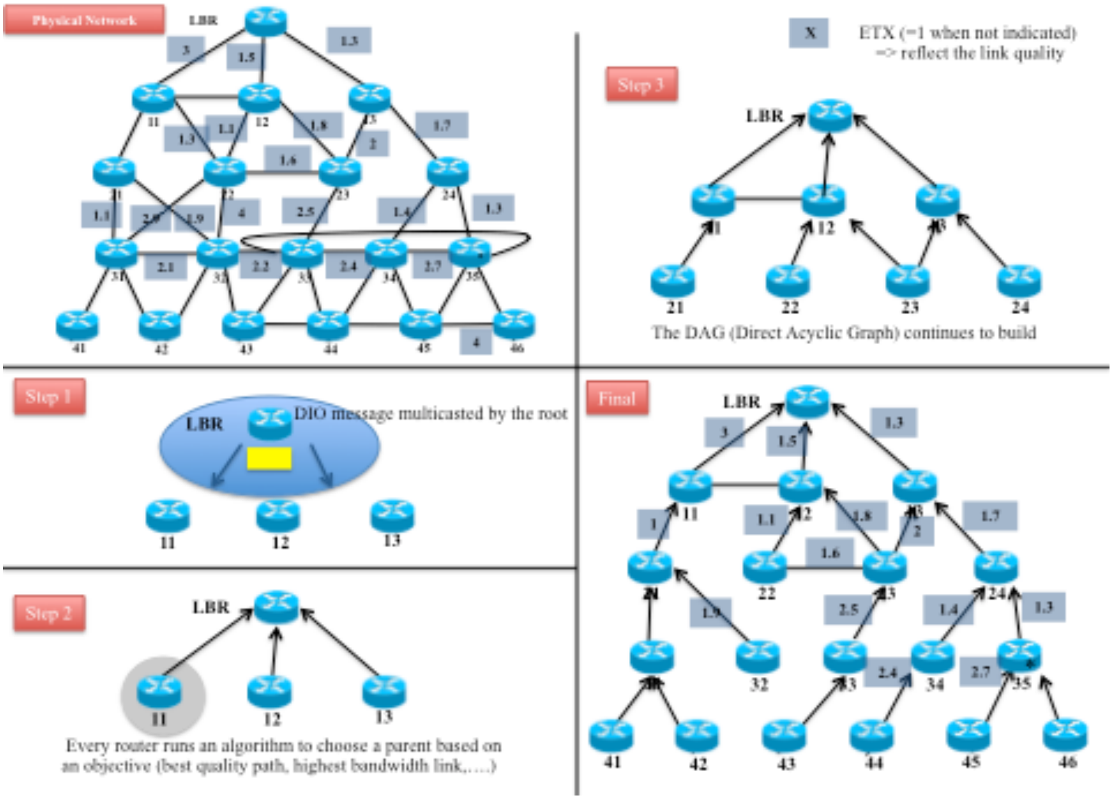
RPL is a Distance Vector IPv6 routing protocol for LLNs that specifies how to build a Destination Oriented Directed Acyclic Graph (DODAG sometimes referred to as a graph in the rest of this document) using an objective function and a set of metrics/constraints. The objective function operates on a combination of metrics and constraints to compute the ‘best’ path. There could be several objective functions in operation on the same node and mesh network because deployments vary greatly with different objectives and a single mesh network may need to carry traffic with very different requirements of path quality. For example, several DODAGs may be used with the objective to (1) ‘Find paths with best ETX [Expected Transmissions] values (metric) and avoid non-encrypted links (constraint)’ or (2) ‘Find the best path in terms of latency (metric) while avoiding battery-operated nodes (constraint)’. The objective function does not necessarily specify the metric/constraints but does dictate some rules to form the DODAG (for example, the number of parents, back-up parents, use of load-balancing, …).

The graph built by RPL is a logical routing topology built over a physical network to meet a specific criteria and the network administrator may decide to have multiple routing topologies (graphs) active at the same time used to carry traffic with different set of requirements. A node in the network can participate and join one or more graphs (in this case we call them “RPL instances”) and mark the traffic according to the graph characteristic to support QoS aware and constraint based routing. The marked traffic flows up and down along the edges of the specific graph.

DODAG Building Process

The graph building process starts at the root or LBR (LowPAN Border Router), which is configured by the system administrator. There could be multiple roots configured in the system. The RPL routing protocol specifies a set of new ICMPv6 control messages to exchange graph related information. These messages are called DIS (DODAG Information Solicitation), DIO (DODAG Information Object) and DAO (DODAG Destination Advertisement Object).

The root starts advertising the information about the graph using the DIO message. The nodes in the listening vicinity (neighboring nodes) of the root will receive and process DIO message potentially from multiple nodes and makes a decision based on certain rules (according to the objective function, DAG characteristics, advertised path cost and potentially local policy) whether to join the graph or not. Once the node has joined a graph it has a route toward the graph (DODAG) root. The graph root is termed as the 'parent' of the node. The node computes the 'rank' of itself within the graph, which indicates the "coordinates" of the node in the graph hierarchy. If configured to act as a router, it starts advertising the graph information with the new information to its neighboring peers. If the node is a "leaf node", it simply joins the graph and does not send any DIO message. The neighboring peers will repeat this process and do parent selection, route addition and graph information advertisement using DIO messages. This rippling effect builds the graph edges out from the root to the leaf nodes where the process terminates. In this formation each node of the graph has a routing entry towards its parent (or multiple parents depending on the objective function) in a hop-by-hop fashion and the leaf nodes can send a data packet all the way to root of the graph by just forwarding the packet to its immediate parent. This model represents a MP2P (Multipoint-to-point) forwarding model where each node of the graph has reach-ability toward the graph root. This is also referred to as UPWARD routing. Each node in the graph has a 'rank' that is relative and represents an increasing coordinate of the relative position of the node with respect to the root in graph topology. The notion of "rank" is used by RPL for various purposes including loop avoidance. The MP2P flow of traffic is called the 'up' direction in the DODAG. The various steps of the graph building process are represented in Figure 2.


Physical Network
LBR
Step 3
X
ETX (=1 when not indicated)
=> reflect the link quality
The DAG (Direct Acyclic Graph) continues to build
Step 1
DIO message multicasted by the root
Step 2
Every router runs an algorithm to choose a parent based on
an objective (best quality path, highest bandwidth link,....)
Final
11
12
13
21
22
23
24
31
32
33
34
35
41
42
43
44
45
46

The DIS message is used by the nodes to proactively solicit graph information (via DIO) from the neighboring nodes should it become active in a stable graph environment using the ‘poll’ or ‘pull’ model of retrieving graph information or in other conditions.

Similar to MP2P or ‘up’ direction of traffic, which flows from the leaf towards the root there is a need for traffic to flow in the opposite or ‘down’ direction. This traffic may originate from outside the LLN network, at the root or at any intermediate nodes and destined to a (leaf) node. This requires a routing state to be built at every node and a mechanism to populate these routes. This is accomplished by the DAO (Destination Advertisement Object) message. DAO messages are used to advertise prefix reachability towards the leaf nodes in support of the ‘down’ traffic. These messages carry prefix information, valid lifetime and other information about the distance of the prefix. As each node joins the graph it will send DAO message to its parent set. Alternately, a node or root can poll the sub-dag for DAO message through an indication in the DIO message. As each node receives the DAO message, it processes the prefix information and adds a routing entry in the routing table. It optionally aggregates the prefix information received from various nodes in the sub-dag and sends a DAO message to its parent set. This process continues until the prefix information reaches the root and a complete path to the prefix is setup. Note that this mode is called the “storing” mode of operation where intermediate nodes have available memory to store routing tables. RPL also supports another mode called “non-storing” mode where intermediate node do not store any routes (discussed later in this document).

RPL also supports point-to-point (P2P) communication from any node to any other node in the graph. When a node sends a packet to another node within the LLN network, the packet travels ‘up’ to a common ancestor at which point it is forwarded in the ‘down’ direction to the destination. A technique for further optimizing (when necessary) P2P communication between nodes is being explored by the ROLL WG in [13].

RPL also provides the ability to perform multi-topology routing (MTR) thanks to the concept of a DODAG instance identified by an instance-id. The idea is to construct and identify multiple graphs (DODAGs) over the same physical topology. This provides a way to provide paths based on different optimization objectives as specified by the objective function and the routing/constraint metrics. A node can only join a single graph within an instance-id but can be associated with several instance-ids simultaneously. This is illustrated in Figure 3. This is helpful to build multiple routing topologies on a physical mesh network. For example, non-critical traffic should follow a path avoiding battery-powered nodes whereas more critical traffic should follow a path of minimum latency.

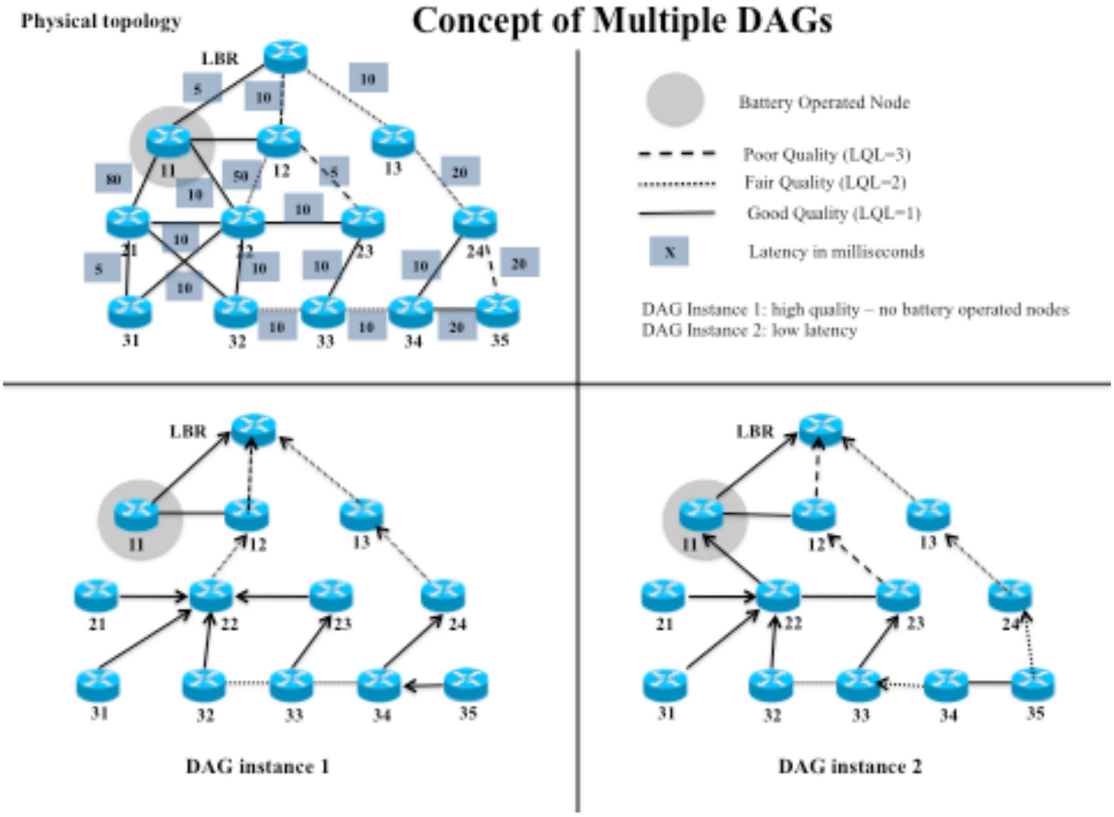


Figure 3 – Example of Multi-topology-routing using the RPL routing protocol

Another class of traffic can use paths leading to a data aggregator while other traffic should avoid nodes not supporting link encryption. So routing decisions are more complex using a variety of metrics and constraints, which could change dynamically.

Within an instance, the routing topology can be segregated among multiple graphs for various reasons such as providing greater scalability. A node can only be associated with a single graph within a specific instance but it can join multiple routing instances. However, the routing protocol allows the movement (or 'jump') of a node to a different graph within the framework of some fundamental rules. As the node moves to another graph it has to abandon its current parent set, re-compute the new rank based on its new position and do new parent selection.

Support of dynamic routing metrics and constraints

Metrics and constraints: Routing for an LLN requires a sophisticated routing metric strategy driven by type of data traffic. A metric is a scalar quantity used as input for best path selection. A constraint, on the other hand, is used as an additional criterion to prune links or nodes that do not meet the set of constraints. Metrics and constraints can be node or link based. Examples of node level metrics are node state attribute, node energy state etc., while link level metrics can be latency, reliability, link color etc. The metrics and constraints can be dynamic and the routing protocol "smoothes" and reacts to the changes in metric and constraint values (see [7]). Additionally, metrics can be recorded or accumulated. Recorded metrics carry distinct values of each path while accumulated metric is an aggregation of values along the path.

Storing and Non-storing nodes

Each node at the edge of a graph sends a DAO message to allow routing state to be built for traffic to flow in the 'down' direction known as downward routing. This means that each node in the network would have to store the prefix information from the DAO messages received from the sub-dag nodes. This has memory implications and routing table scalability at each node since each prefix entry translates to a routing entry in the routing table (or less in the presence of routing aggregation). Some nodes in the network may have significant constraints regarding memory and may be incapable of storing routing entries for downward routes. These nodes are classified as non-storing nodes while nodes capable of storing routing information are called storing nodes.

In non-storing mode, a node uses DAO messages to report its DAO to its parents all the way to the graph Root. The graph Root uses the information received to piece together a downward route to a node by using DAO parent sets from each node in the route. The nodes include the parent information in the 'transit-info' field of the DAO message. Additionally, nodes can pack DAOs by sending a single DAO message with multiple prefix information. Each prefix information can be associated with its own transit information. In this mode of operation it is expected that the root of the DODAG has the capability to store routing information while the nodes in the DODAG operate in non-storing mode. A mixed mode of operation is not allowed and all nodes in the graph have to operate in either storing or non-storing mode only.

In the non-storing mode when the root receives a packet destined to a specific destination in a non-storing graph the root adds the pieced together information in the source routing header of the packet and forwards it to the next-hop child node in the network. Each intermediate node examines the information in the source routing header and forwards the packet to the next-hop child node. This forwarding process is repeated until the packet reaches the final destination. So for example, in non-storing mode, when a node A sends a packet to a node B within the RPL domain, the packet first follows the graph up to the root where the routing information is stored. At this point, the graph root inspects the destination, consults its routing table that contains the path to the destination thanks to the DAO messages that were received, and "source-routes" the packet to its destination using a specific routing header for IPv6 (called RH4) [12].

It should be mentioned that there is a trade-off between storing and non-storing mode of operation in terms of computing resources (memory, CPU, power etc). For example, storing mode requires routing tables and uses up memory while non-storing mode, though not requiring routing tables, causes packets to increase in size which uses more power and bandwidth.

Loop Avoidance and Loop Detection

Loop detection and avoidance is one of the differentiating aspects of a routing protocol for smart object networks like RPL compared to traditional networks. In traditional networks temporary loops are formed due to topology changes and lack of synchronization between nodes. These loops need to be detected as quickly as possible to avoid packet drops (due to TTL expiry) and link congestion, therefore various optimization mechanisms have been proposed and put in place to avoid such micro-loops.

Contrasting the high data rates in traditional networks are the low data rates in LLNs. In LLNs the effect of temporary loops may have limited impact on low data rates and it is recommended to under-react, as the conditions leading to loops could be transient. Also, over-reacting to such conditions in LLNs could lead to further routing oscillations and energy consumption in nodes to process the control packets. Thus, RPL does not guarantee the absence of loops but rather tries to avoid them and specifies mechanisms to detect loops via data path validation.

RPL specifies two rules for loop avoidance. These rules rely on the 'rank' property of the nodes. Firstly, as part of the "max_depth rule", a node is not allowed to select as a parent a neighboring node that is deeper (ie whose rank is greater) such that the node will end up advertising a value node-rank+max_depth, where max_depth is a configurable value specified at the root. Secondly, a node is not allowed to be 'greedy' and attempt to move deeper in the graph to increase the number of parents.

Loops in LLNs are unavoidable hence there is a need for detecting these loops in addition to loop avoidance rules. One way to achieve this is by setting bits in the RPL routing header (RH4 [12]) and processing these bits as part of data-path validation. The idea is to set and process these bits as the packet moves up and down along the edges of the graph and check for anomalies in the values to detect loops. For example, loops in the DAO path can be detected by using a 'down' bit in the RPL routing header [12]. When a node sends a packet destined to one of its children in the 'down' direction, it sets the 'down' bit and forwards the packet to the next hop node. Upon receiving a packet with the down bit set, if the routing table lookup of the receiving node indicates that the packet has to be forwarded in the 'up' direction this indicates an inconsistency or a loop and packet needs to be discarded (a local repair needs to be triggered). Similar other optimizations are possible.

Global and Local Repair

Repair is a key feature for any routing protocol and refers to the ability to repair the routing topology when failures occur. Similarly, RPL supports graph repair mechanisms in case of link and node failures. Care must be taken to avoid triggering a re-build in transient conditions as discussed previously. RPL specifies two techniques, which are complimentary in nature and actions (known as local and global repair). When a link or neighboring node failure is detected to be unavailable and the node has no other router in the 'up' direction, a local repair is triggered to quickly find an alternate parent/path. This is a local repair with no global implication on the entire graph. As local repairs take place the graph may start to diverge from its optimum shape, at which point it might be necessary to rebuilding the graph (DODAG) thanks to a complementary mechanism called the "Global Repair".

Global repair is a repair mechanism that rebuilds the graph from scratch. It is an optimization technique but it has a cost. The global repair can be triggered only from the root and has a cost of additional control traffic in the network. Each node in the graph will rerun the objective function for preferred parent selection.

Timer Management

This is another area where RPL differs from other routing protocols that operate in less-constrained environments. In LLNs, especially when the network is made of devices that must save energy, it is imperative to limit the control plane traffic (RPL) in the network. Most routing protocols use periodic keepalives (routing protocol keepalive, protocols such as BFD) to maintain routing adjacency and to keep routing tables up to date. But this would be costly in LLNs where resources are scarce. RPL uses an adaptive timer mechanism called the "trickle timer". This mechanism controls the sending rate of DIO messages. The algorithm treats building of graphs as a consistency problem and makes use of trickle timers to decide when to multicast DIO messages. Certain events are treated as inconsistencies in the network. For example, when a node detects a loop in the network it is considered as an inconsistency in the network, or, when a node joins the network or moves within the network is considered an inconsistency in the network. Loops are detected using new bits defined in an extended IPv6 header. The interval of the trickle timer increases as the network stabilizes which results in fewer DIO messages being sent in the network. As inconsistencies are detected, the nodes reset the trickle timer and send DIOs more often. Using this mechanism the frequency of the DIO messages depends on the stability of the network and the frequency is increased in the vicinity where the inconsistency is detected. In other words, as the network becomes stable, the number of RPL messages decreases. When an inconsistency is detected (such as a loop or a change in the DODAG parameters) the timers are reset to quickly fix the issue (this can be observed in Figure 4 with the "Waves" of control traffic).

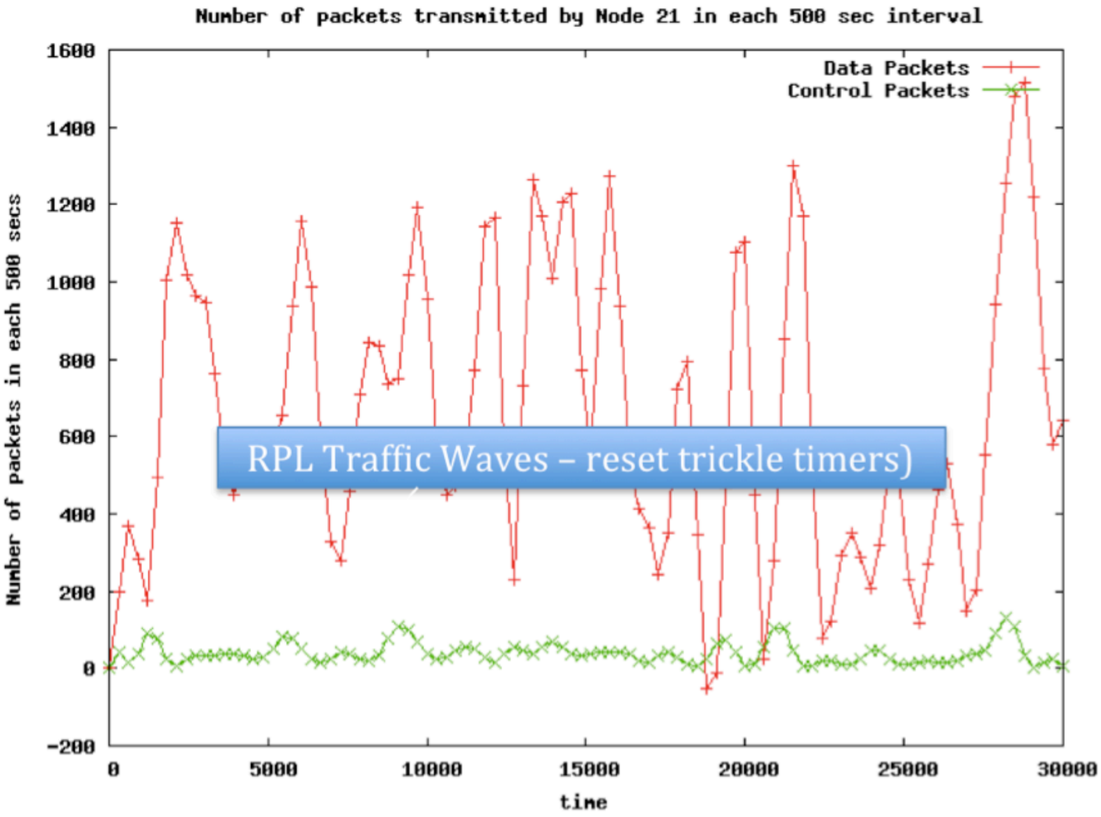


Figure 4 – RPL Control plane Traffic

One of the main advantages of the trickle timer implementation is that it does not require complex code and is fairly easy to implement. This is especially important given the constrained devices in operation.

4. RPL and 6LoWPAN

In 2005, the IETF chartered the IPv6 over Low Power, Wireless Networks (6LoWPAN) working group to standardize adaptations of IPv6 over mesh networks composed of low-power, wireless links. Link-layer datagram fragmentation and IPv6 header compression were defined to efficiently transport IPv6 datagrams within IEEE 802.15.4 frames. New mechanisms were also defined to perform IPv6 ND operations such as link-layer address resolution and duplicate address detection. While 6LoWPAN was originally chartered for IEEE 802.15.4, the working group's care to limit tight bindings to 802.15.4 allowed other link technologies (e.g. Wavenis and PLC) to utilize the same 6LoWPAN mechanisms. As such, the term "6LoWPAN networks" is often generalized to refer to mesh networks built on low-power and lossy links that utilize 6LoWPAN mechanisms.

A long-standing issue in adapting IPv6 to any link technology is whether or not to support a *single broadcast domain*, where all communication is *transitive* within the subnet (if A can send to B and B can send to C, then A can send to C) and any interface can reach any number of interfaces within the subnet by sending a single IP datagram. Emulating a single broadcast domain within a 6LoWPAN network requires link-layer routing and forwarding, often referred to as "mesh-under" since the multi-hop mesh topology is abstracted away underneath IPv6 to appear as a fully connected network. However, the IETF has not specified any mesh-under routing protocols for use in 6LoWPAN networks.

By contrast, the IETF has specified a "route-over" architecture (RPL as explained in this document) where routing and forwarding is implemented at the network layer, according to the IP architecture. Where a *mesh-under* architecture defines the extent of an IPv6 link as all nodes within the same multihop mesh, a *route-over* architecture defines the extent of an IPv6 link as immediate neighbors reachable within a single link transmission (e.g. radio range on wireless links). In other words, a route-over 6LoWPAN network would be composed of multiple overlapping link-local scopes, each node defining its own link-local scope that includes its immediate link neighbors.

To summarize: A mesh-under approach places routing functions in the link layer to emulate a single broadcast domain where all devices appear as immediate neighbors to the network layer. In contrast, a route-over approach places all routing functions at the network layer.

An expected use case for RPL is to support 6LoWPAN networks in a route-over configuration. With RPL, 6LoWPAN routers operate as IPv6 routers and form routes using RPL. Border routers that connect 6LoWPAN networks to other IP networks will typically operate as RPL DODAG roots. Nodes then utilize RPL to form one or more routing topologies so that they can forward IPv6 datagrams to their destination.

A route-over 6LoWPAN network typically does not configure any on-link prefixes due variable connectivity and neighbor relationships that is common within LLNs. As such, 6LoWPAN hosts must explicitly indicate their presence to neighboring attachment routers in one of two ways. In the first option, a host can operate a subset of the RPL protocol, by receiving DIO messages, choosing preferred parents based on advertised metrics and constraints, and communicating DAO messages to the root. The RPL-aware host does not transmit DIO messages because it is not providing any routing functionality.

Alternatively, a 6LoWPAN host may be routing-protocol agnostic by using the 6lowpan-nd protocol to discover neighboring routers, choose attachment routers, and notify one or more of those routers of their existence.

The interaction with Neighbor Discovery and RPL is important to take into account. This is especially true in a LoWPAN, where 6LoWPAN ND optimizations [11] change the interaction model and the LoWPAN network architecture demands more from ND in a route-over topology.

Hosts play a special role in LoWPANs, and the ND bootstrapping process allows them to attach to a LoWPAN without the need to participate in routing, thus reducing complexity. 6LRs (6lowpan Routers), which act either as RPL routers or leaf nodes, respond to Router Solicitation (RS) messages from 6LNs (6lowpan Nodes - other hosts or routers) with Router Advertisement (RA) messages. RAs contain the needed prefix and context information for a node to discover the LoWPAN and autoconfigure its addresses. In a LoWPAN, neighbor information is maintained by having nodes register with their default next-hop routers. This is done using a unicast Neighbor Solicitation/Neighbor Advertisement (NS/NA) exchange carrying an Address Registration Option. These exchanges are shown in Figure 5. 6LRs use ND in the same manner to bootstrap onto the network with a neighbor router, and then to register with other routers.

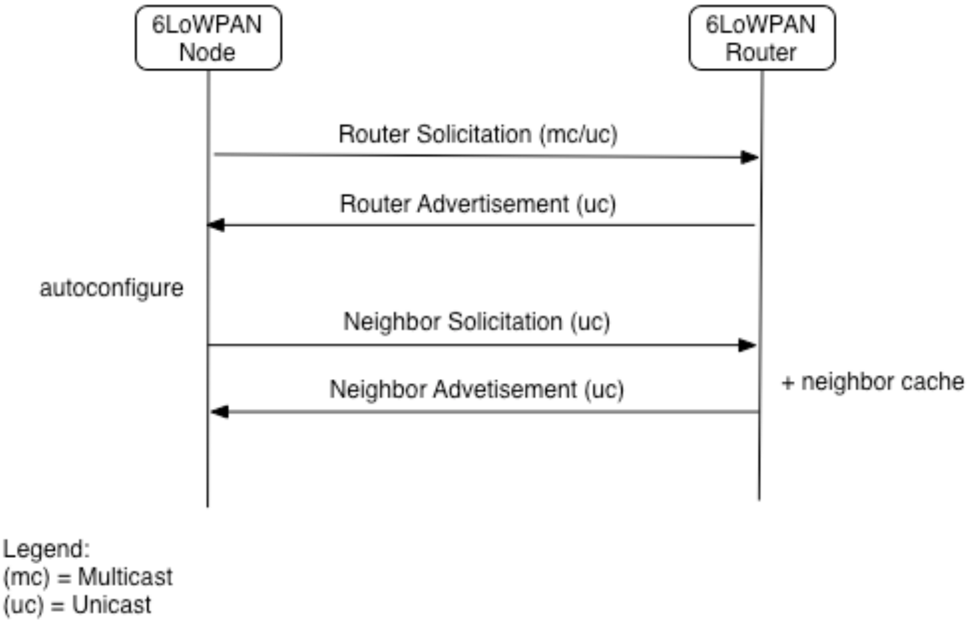

6LoWPAN Node
6LoWPAN Router
Router Solicitation (mc/uc)
Router Advertisement (uc)
autoconfigure
Neighbor Solicitation (uc)
Neighbor Advetisement (uc)
+ neighbor cache
Legend:
(mc) = Multicast
(uc) = Unicast

Figure 5 – Basic ND message exchanges in a LoWPAN.

A LoWPAN functions properly only when its prefix information and the set of compression contexts (if any), used for further compressing addresses, is in sync for all nodes in the LoWPAN. On an IPv6 link this is trivial as all nodes on the link can receive RAs from the same router. In a route-over LoWPAN the link is non-transitive, thus every 6LR in the LoWPAN needs a fresh set of prefix and context information. This information is then included in the RA sent in response to an RS from a neighboring node. This is achieved in a LoWPAN by using the multihop prefix distribution mechanism of [11]. Here the 6LBR originates the set of prefix and context information for the LoWPAN. This set of prefix and context information is provided with the 6LBRs IPv6 address and a version number. As RS/RA exchanges are made by 6LRs in the network, this information is slowly distributed throughout the LoWPAN. By following a simple set of rules, the 6LBR is able to update the set of information while keeping all nodes in the LoWPAN in sync.

In turn, RPL routers serving as attachment routers must inject host routes into the RPL domain by including information about those hosts that have registered via 6lowpan-nd in DAO messages.

As you may have noticed, applying RPL to 6LoWPAN networks does not require any considerations that are different from any other link technology. From a technical perspective, it allows the formation of a single cohesive routing graph that does not suffer from unintended cross-protocol or cross-layer interactions. From an operational perspective, running a single routing protocol across different link technologies reduces operator burden in having to understand and manage a routing protocol for each specific link technology. Within a RPL domain, one or more RPL routers are configured to serve as roots and initiate the graph building process. Other RPL routers participate in the iterative graph building process and generate DAOs toward the root to advertise reachable prefixes within their subgraphs. In storing mode, RPL routers maintain state for prefixes within their subgraph.

5. An example using RPL over a Low-Rate Low-Power Powerline Communication (LR-LP-PLC) for Home area Network.

Because RPL aims to offer a routing protocol for LLNs, it is by definition not restricted to any specific link layer. According to the initial requirements, RPL is a layer 3 routing protocol not tied to a specific link layer technology. As presented in the “Low Power Link Layer” white Paper [5], link layers technologies other than IEEE 802.15.4 may fit with the LLN definition, and PLC is a particularly good candidate.

The aim of this section is simply to provide an example of RPL operation using a PLC link layer for a home area network. Needless to say that LLN may also use low power wireless links or a mix of link layers too.

Similarly to low power wireless link layers, PLC links suffer from variable link quality and are not designed to be a broadcast-based technology like Ethernet.

Basically, limitations come from:

- The strong absorption of the media itself, not designed to support high frequency transmissions.
- Appliance’s power supply presenting low impedance at high frequencies thus impacting significantly signal propagation.
- EMC regulation and power consumption that limit emission levels and coverage.

Compared to regular PLC systems, LR-LP-PLC is a particular implementation (currently being standardized within ETSI), dedicated to command & control or M2M applications, designed to optimize power consumption including to the detriment of range and data rate. Some LR-LP-PLC implementations have been shown to consume similar power consumption levels as wireless radio systems like 6LowPan nodes.

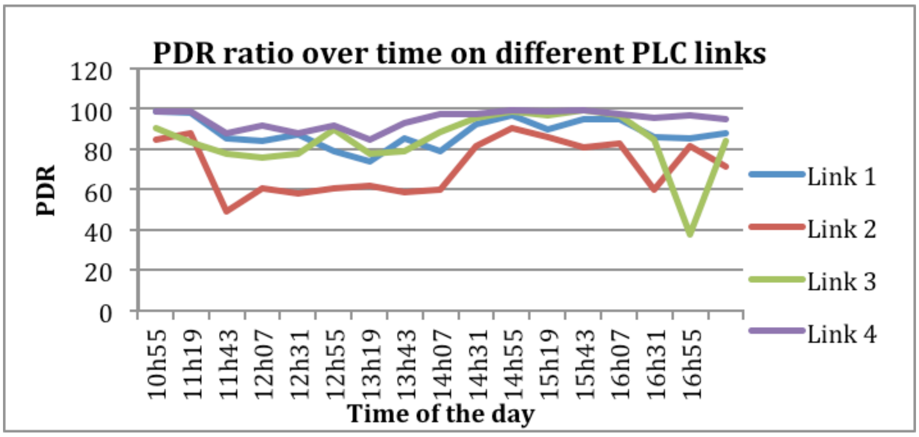


Figure 6 – Packet Delivery Ratio (PDR) variation over time on several LR-LP-PLC links

As shown in figure 6 in comparison with figure 1, the PLC link may be subject to as many disturbances as a wireless link, because every electrical device may inject noise and/or absorb the signal. Considering the number of electrical devices in an electrical network like a multi-dwelling unit and their varying electrical behaviors that disturb the communication, the routing mechanism over PLC networks has to cope with very lossy links. Furthermore, these noise/fading generators create asymmetric links that add routing complexity.

As explained in the previous sections, RPL computes multi-hop path according to a given metric. This enables the choice of the metric-optimum path to a particular node. RPL can also help in fading issues with packet forwarding, enabling the repeating of the message along the path. The use of the Expected Transmission Count (ETX) metric in RPL networks over LR-LP-PLC will help the RPL nodes to use more reliable paths to reach the root.

Note that the link quality measurement may depend on the packet size, larger size packets taking more time to transmit and therefore more exposed to disturbances.

Considering LR-LP-PLC technologies as presented in [5], experience tells us that all nodes of a single network cannot be reached with a single hop. Due to multiple and varying disturbances on the link, LR-LP-PLC networks may take advantage of mesh topologies to provide path diversity and RPL recovery mechanisms. RPL address these challenges by building a graph (DODAG), and maintaining a topology according to a set of Constraints/Metrics computed by low layers. An experiment of a RPL network over an LR-LP-PLC implementation subject to real life activity (multiphase in a multi-dwelling environment) with the ETX metric, achieved a 97% average transmission success instead of less then 50% without RPL (note that this is based on a real-life Watteco experiment and may vary with the network).

Because LR-LP-PLC links have limited throughput, the under-reactive behavior of RPL helps to maintain a reliable topology with keeping the traffic control overload very low. Being over-reactive would result in a global repair for every electrical event that may change the network topology.

Because the electrical network behavior cannot be known a priori, and because there is no strong relation between wiring and logical connections, self-configuration is the key to build a routing topology.

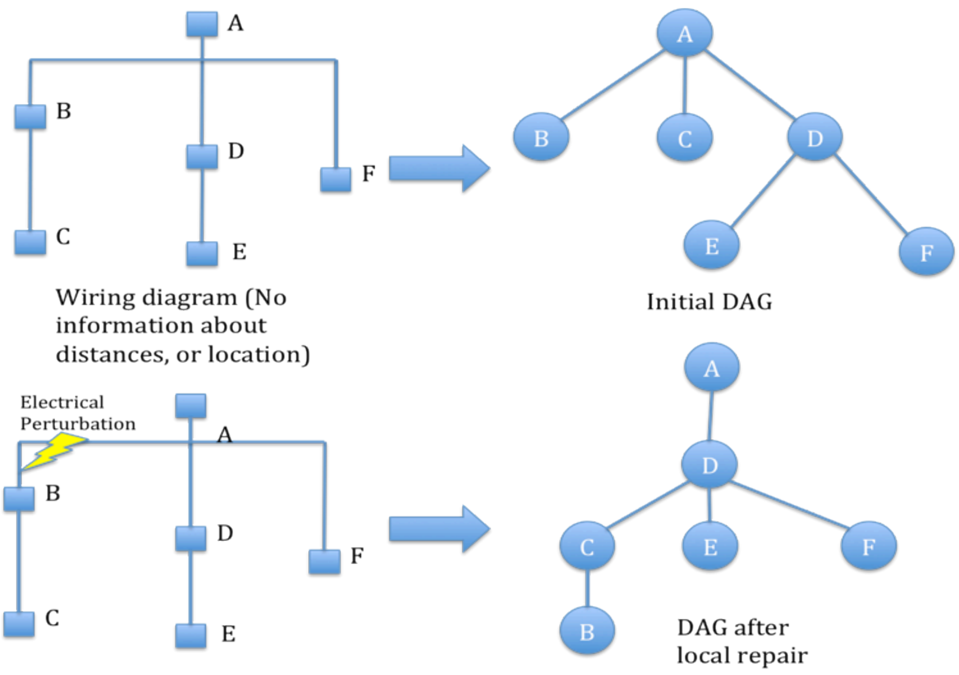


Figure 7 – Example of a local repair in the RPL routing topology after an electrical perturbation, affecting link to node B.

Figure 7 highlights the difference between physical and logical topologies over LR-LP-PLC. This makes auto configuration a mandatory feature. The local repair mechanism in RPL provides full connectivity in the PLC network thanks to multi-hops topologies.

According to its rooted architecture, RPL enables multi physical networking. For example, the DODAG root may be used to connect the LLN to the public Internet, regardless of the media employed. For instance, a root of a LR-LP-PLC network (thus usually main-powered) running RPL may have an 802.15.4 interface and an Ethernet interface to the Internet.

6. RPL and Security

Security is critical in smart object networks but implementation complexity and size is a core concern for LLNs such that it may be economically or physically impossible to include sophisticated security provisions in a RPL implementation. Furthermore, many deployments can utilize link-layer or other security mechanisms to meet their security requirements without requiring the use of security in RPL. Therefore, the security features in RPL are available as optional extensions.

When made available, RPL nodes can operate in three security modes. In the first mode, called "unsecured," RPL control messages are sent without any additional security mechanisms. Unsecured mode implies that the RPL network could be using other security primitives (e.g. link-layer security) to meet application security requirements. In the second mode, called "pre-installed," nodes joining a RPL instance have pre-installed keys that enable them to process and generate secured

RPL messages. In the third mode, called "authenticated", nodes can join as leaf nodes using pre-installed keys as in pre-installed mode, or join as a forwarding node by obtaining a key from an authentication authority.

Each RPL message has a secure variant. The level of security (32-bit and 64-bit MAC and ENC-MAC modes are supported) and the algorithms (CCM and AES-128 are supported) in use are indicated in the protocol messages. The secure variants provide integrity and replay protection and confidentiality and delay protection as an added option.

7. Interoperability Testing

RPL has been implemented by a number of vendors during the design phase and RPL has highly benefited from return on experience as it was implemented. Furthermore both the IPSO and Zigbee/IP alliances organized several interoperability tests that were successful. Zigbee has tested the “non storing” mode of operation of RPL.

8. Simulation Results

In order to get a sense of real-life deployments several vendors are fast adopting the routing protocol specified. In addition, simulations have been done on various aspects of the algorithm to provide useful data and aid in design choices. For example, in one set of simulation results, a discrete event simulator (see [14]) has been developed based on OMNET++ and the Castalia module for wireless sensor networks within OMNET++. Hundreds of link traces were gathered to create a link failure model database of lossy links. Each link trace provided the PDR at different times. For some links received signal strength indicator (RSSI) data was available and PDR values were derived from it due to their implicit correlation.

The simulator reads the database and selects values at random thus providing fairly realistic results. When a packet is to be transmitted by a node, the PDR of the link is read from the database and the packet is dropped with a probability of 1-PDR. In this simulation the data traffic was segregated with 25% of the traffic going in the up direction to the root and 75% of the traffic going in the down direction.

Several characteristics were studied: control traffic, routing table size, path efficiency and failure handling. The following observations were made for each:

Control traffic: The control traffic is negligible compared to the data traffic and as the DODAG stabilizes the control traffic decreases significantly.
Routing Table Size: Observations were made for the number of routing entries in the absence of route aggregation. It was observed that number of routing entries increase as we get closer to the root of the DODAG.
Path Efficiency: Observation was made on the optimality of path for P2P traffic. The idea was to find out how sub-optimal the path computed by the algorithm for P2P traffic compared to an ideal routing protocol. It was observed that although the algorithm provides a fairly good quality path additional mechanism would be needed to further improve it.
Failure Handling: This observation provided critical information as it provides the protocol’s capability to compute an alternate path in the case of node or link failure.

Observations were made for local and global repair scenarios to observe the amount of time during which no path was available. It was observed that in 80% of the cases the period of time without connectivity was 20sec during local repair. Two observations were made for global repair frequency: 1hour and 1minute. These results were observed on specific networks for a given RPL parameters settings. It was observed that as the frequency interval is reduced the failure time is also reduced at the cost of increase in control traffic. It was also observed that if the global repair interval was increased to one hour and the local repair was activated the failure time was reduced significantly while the control traffic increased slightly, thus providing excellent convergence time without affecting the overall scalability (see [14] for further details).

9. References

[1] **"**Interconnecting Smart Object with IP: The next Internet" (The Morgan Kaufmann Series in Networking) by JP Vasseur and Adam Dunkels, www.thenextinternet.org, June 2010.

[2] IPSO White Paper #1 – "A survey of several low power Link layers for IP Smart Objects" by JP Vasseur, Adam Dunkels.

[3] "6LoWPAN: The Wireless Embedded Internet" (Wiley Series on Communications Networking & Distributed Systems)" by Zach Shelby and Carsten Bormann.

[4] IPSO White Paper #3 – Lightweight IPv6 Stacks for Smart Objects: the Experience of Three Independent and Interoperable Implementations by Julien Abeillé, Mathilde Durvy, Jonathan Hui, Stephen Dawson-Haggerty.

[5] IPSO White Paper #6 – "A survey of several low power Link layers for IP Smart Objects" by JP Vasseur, Paul Bertrand, Bernard Aboussouan, Eric Gnoske, Kris Pister, Roland Acra and Allen Huotori.

[6] RPL: IPv6 Routing Protocol for Low power and Lossy Networks - http://tools.ietf.org/html/draft-ietf-roll-rpl-19

[7] Routing Metrics used for Path Calculation in Low Power and Lossy Networks - http://tools.ietf.org/html/draft-ietf-roll-routing-metrics

[8] Terminology in Low power And Lossy Networks - http://tools.ietf.org/html/draft-ietf-roll-terminology

[9] "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", by G. Montenegro, N. Kushalnagar, J. Hui and D. Culler, IETF RFC4944, Sept 2007.

[10] Compression Format for IPv6 Datagrams in 6LoWPAN Networks - http://tools.ietf.org/html/draft-ietf-6lowpan-hc

[11] Neighbor Discovery Optimization for Low-power and Lossy Networks - http://tools.ietf.org/html/draft-ietf-6lowpan-nd

[12] An IPv6 Routing Header for Source Routes with RPL - http://tools.ietf.org/html/draft-ietf-6man-rpl-routing-header

[13] Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks - http://tools.ietf.org/html/draft-ietf-roll-p2p-rpl

[14] Performance Evaluation of Routing Protocol for Low Power and Lossy Networks (RPL) draft-tripathi-roll-rpl-simulation-06