3.4 Routing

When discussing IP routing for 6LoWPAN, there are two types of routing which need to be considered: routing inside a LoWPAN, and routing between a LoWPAN and another IP network. Routing is challenging for 6LoWPAN, with low-power and lossy radio links, battery-powered nodes, multihop mesh topologies, and frequent topology change due to mobility. Successful solutions must take the specific application requirements into account, along with Internet topology and 6LoWPAN mechanisms.

As IP networks are *packet switched*, as opposed to circuit switched, forwarding decisions are made *hop-by-hop*, based on the destination address in a packet. Therefore reaching a destination node in a network from a source node requires building a path from the source to the destination node in route tables on nodes along the path. IP addresses are structured, and this structure is used to group addresses together under a single route entry. In IPv6 an address prefix is used for this purpose, which is why this is called prefix-based routing.

- LoWPAN Routers typically perform forwarding on a single wireless interface, i.e. they receive a packet on their wireless interface from one node, and then forward it to the next-hop destination using the same interface. This is an important difference to how forwarding on IP routers normally works, where packets generally are forwarded between interfaces (and thus between links). The reason for this model is that typically not all nodes in a LoWPAN are reachable in a single wireless transmission, thus IP forwarding is used to provide full connectivity over multiple hops within the same “link”.
- A LoWPAN has a *flat* address space, as all nodes in a LoWPAN share the same IPv6 prefix. This is due to the way 6LoWPAN compression is achieved, using the fact that all nodes in the network know common information to elide or compress fields. Therefore 6LoWPAN routing tables only contain entries to destination addresses in the LoWPAN, along with default routes.
- LoWPANs are stub networks, and are not meant to be transit networks below

Two kinds of routing can be performed with 6LoWPAN, *intra-LoWPAN* routing between LoWPAN Routers, and *border routing* performed at the edge of the LoWPAN by the LoWPAN Edge Router or an IPv6 router on the backbone link for Extended LoWPANs. These routing domains and the associated network-layer forwarding are illustrated in Figure 4.8 and Figure 4.9.

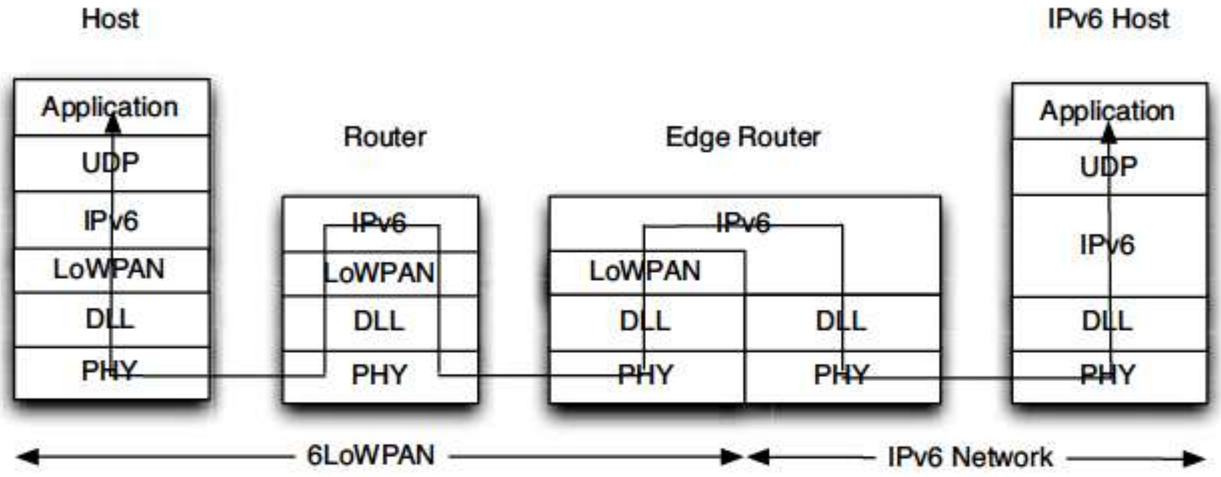


Fig: 3.4.1 Stack view of forwarding inside the LoWPAN and across the edge router.

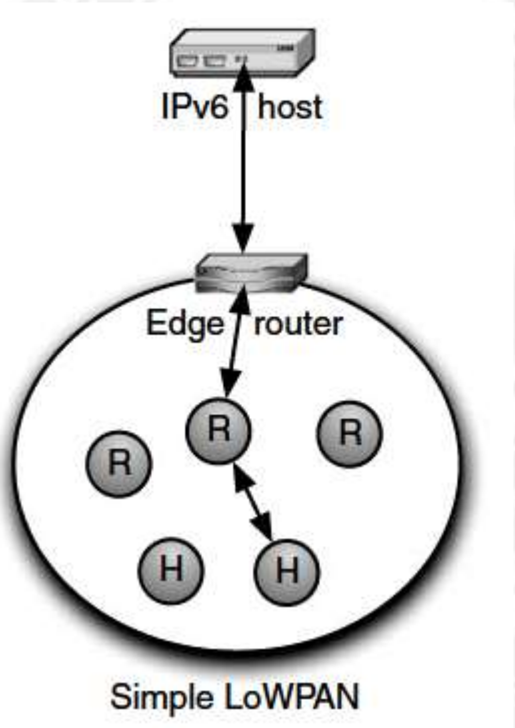


Fig 3.4.2 Topology view of forwarding inside the LoWPAN and across the edge router.

There are two main classes of routing protocols useful for 6LoWPAN: *distance-vector* routing and *link-state* routing.

Distance-vector routing:
These algorithms are based on variations of the Bellman–Ford algorithm. Using this approach, each link (and possibly node) is assigned a cost, using appropriate route metrics. When sending a packet from node A to node B, the path with the lowest cost is chosen. The routing table of each router keeps soft-state route entries for the destinations it knows about, with the associated path cost. Routing information is updated either *proactively* (a priori) or *reactively* (on-demand) depending on the routing algorithm. Owing to their simplicity, low signaling overhead and local adaptive nature, distance-vector algorithms are commonly applied to 6LoWPAN.

Link-state routing: In this approach, each node acquires complete information about the entire network, called a graph. To do this, each node floods the network with information about its link information to nearby destinations. After receiving link-state reports from sufficient nodes, each node then calculates a tree with the shortest-path (least cost) from itself to each destination using e.g. Dijkstra's algorithm. This tree is used either to maintain the routing table in each node for hop-by-hop forwarding, or to include a source-route in the header of the IP packet. Link-state algorithms incur a large amount of overhead, especially in networks with frequent topology change. They require substantial memory resources for the amount of state needed by each node. Thus they are not suitable for distributed use among LoWPAN Nodes [ID-roll-survey].

Link-state algorithms may be usefully applied off-line on LoWPAN Edge Routers which have sufficient memory capacity if the signaling overhead for collecting the link-state information is reasonable. In this way only a single tree is constructed from the ER to nodes.

In order to update routing information throughout a network or along a path, routing protocols make use of either proactive or reactive signaling techniques. These terms can be defined as:

Proactive routing: Algorithms using a proactive approach build up routing information on each node before the routes are needed. Thus they proactively prepare for the data traffic by learning routes to all possible or likely destinations. Most protocols that are used in inter-domain or intra-domain IP routing use a proactive approach as topologies are stable. Examples of proactive algorithms can also be found from MANET, for example optimized link-state routing (OLSR) and topology dissemination based on reverse-path forwarding (TBRPF). The advantage of this approach is that routes are immediately available when needed, but this comes at the cost of increased signaling overhead especially with frequent topology changes and increased state for routers.

Reactive routing: Reactive routing protocols store little or no routing information after autoconfiguration of the routing protocol. Instead, routes are discovered dynamically only at the time they are needed. Thus a process called *route discovery* is executed when a router receives a packet to an unknown destination. Examples of reactive algo-rithms include the MANET ad hoc on-demand distance vector (AODV) [RFC3561] and dynamic MANET on-demand (DYMO) [ID-manet-dymo] protocols, along with the ZigBee routing algorithm derived from AODV [ZigBee]. The advantage of this approach is that signaling and route state grows only as needed, and it is especially well suited to ad hoc networks with frequent topology change and mainly peer-to-peer communications.

Advanced techniques which may be applied in 6LoWPAN routing protocols include constrained routing using compound route metrics, local route recovery, flow labeling to achieve Multi Topology Routing (MTR), forwarding on multiple paths with multipath routing, and traffic engineering. Several of these techniques are being considered for the ROLL routing algorithm.

MANET routing protocols

MANET has also produced valuable work on basic mechanisms for supporting routing in these environments. A common packet format for use by all MANET protocols. Finally, a *two-hop*neighborhood discovery protocol is currently being developed for use in collecting route information in a standard way.

AODV

The ad hoc on-demand distance vector (AODV) protocol enables mobile ad hoc multihop networks by quickly establishing and maintaining routes between nodes, even with quickly changing (dynamic) topologies. AODV creates routes to destinations when needed for data communications (reactive), and only maintains actively used routes. It includes methods for local repair, and includes a destination sequence number to ensure loop-free operation.

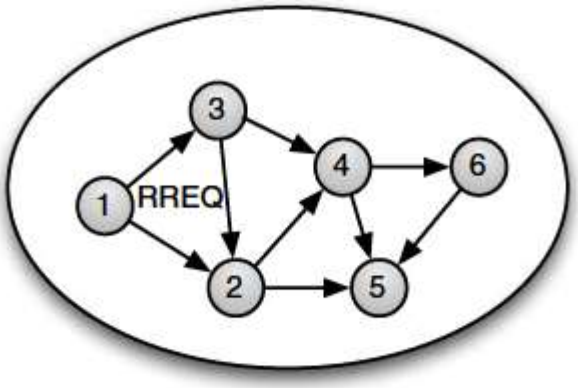
AODV is purely a route table management protocol; after routes have been established they are simply used by IP for forwarding.A small set of messages are used for discovering and maintaining routes by AODV and similar protocols. A *route request* (RREQ) is broadcast throughout the network in order to find paths to a destination. This is responded to with a *route reply* (RREP) by an intermediate router or by the destination. Figure 4.10 shows an example of reactive route discovery and forwarding in an ad hoc network. The *route error* (RERR) message is used to notify about broken links along a path. These messages are sent over UDP, one hop at a time, between the AODV processes running on ad hoc routers. AODV is specified in detail in [RFC3561].

As AODV was the first reactive distance-vector routing algorithm standardized in the IETF, it has been used as a model for many other routing algorithms. For example, the routing algorithm that ZigBee used in its network layer designs is modeled on the AODV algorithm with modifications to minimize overhead and to function on MAC addresses rather than IP addresses.

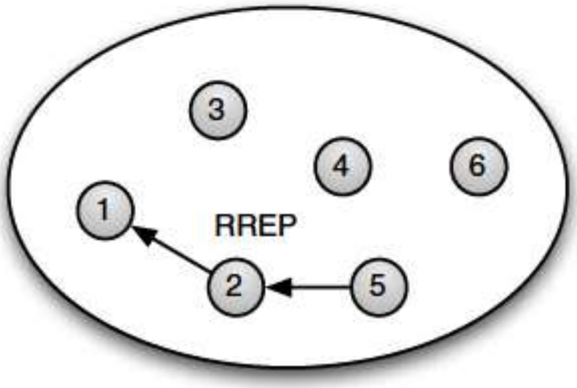
DYMO

A new reactive distance-vector routing protocol called the dynamic MANET on-demand (DYMO) protocol has been developed in the MANET WG [ID-manet-dymo], making improvements on previous protocols such as AODV and dynamic source routing (DSR). This protocol makes use of the same types of route discovery and maintenance messages as AODV. The main improvements compared to previous work include:

- improved convergence in dynamic topologies
- use of the common MANET packet format [RFC5444]
- support for a wide range of traffic flows



1. RREQ for node 5 broadcast over multiple hops.



2. RREP unicast back to node 1, creates route entries.

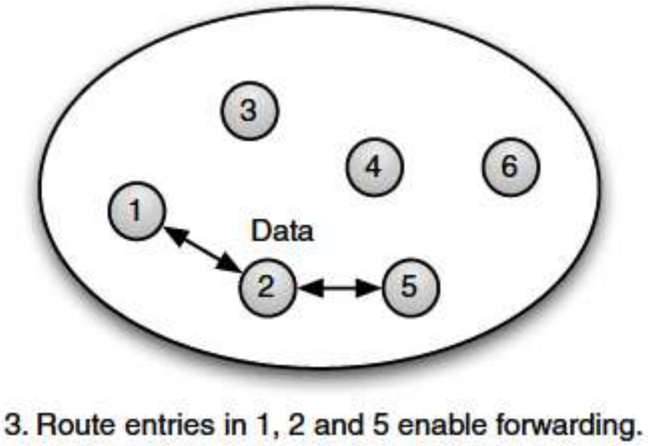


Figure 3.4 Example of reactive distance-vector routing.

OLSR

The MANET WG has also produced a proactive link-state routing protocol called the optimized link-state routing (OLSR) algorithm. Originally specified in [RFC3626], an improved OLSRv2 has been developed in the working group in [ID-manet-olsrv2]. This algorithm applies optimization to the classical link-state algorithm for use in mobile ad hoc networks. In order to build link-state tables, OLSR routers regularly exchange topology information with other routers. The flooding of this information is controlled by the use of selected multipoint relay (MPR) nodes. These MPR nodes are used as intermediate routers, and thus enable a kind of clustering technique. The OLSR algorithm makes use of the standard MANET packet format and two-hop ND techniques.

OLSR is best suited for relatively static ad hoc networks, thus minimizing the number of link-state updates throughout the network, which can cause a lot of overhead. OLSR is not very well suited to 6LoWPAN Routers because of the large amount of signaling and routing state. Link-state protocols are also not well suited to tree topologies, as often found in 6LoWPAN applications. Link-state approaches like OLSR may have possible uses for the partial optimization of larger route topologies or off-line use on border routers, for example with the ROLL routing algorithm.

The ROLL routing protocol

- Traffic patterns are not only peer-to-peer unicast flows, but more often point-to-multipoint or multipoint-to-point flows. Most applications of LLNs are Internet connected.
- Routers in LLNs have a very small, hard bound on state (limited memory).
- Most LLNs must be optimized for energy consumption.
- In most cases LLNs will be deployed over links with a limited frame size.
- Security and manageability are extremely important as LLNs are typically autonomous.
- The application spaces aimed at by ROLL are heterogeneous. Each may need a different set of features along with routing metrics to fulfill its requirements.

The following activities are going on in the ROLL working group:

- ✓ **Metrics:** The routing metrics useful for path calculation have initially been specified in [ID-roll-metrics]. In practice, work will still need to be done on algorithms for *evaluating* appropriate metrics for each specific application space.
- ✓ **Architecture:** The basic architectural requirements for ROLL have been captured in the requirement documents. Terminology for use in ROLL has been specified in [ID-roll-terminology].

- ✓ **Security:** A security framework is being developed in the working group. An overview of requirements and some techniques for ROLL security have been provided in [ID-roll-security], and some considerations for trust management are collected in [ID-roll-trust].
- ✓ **Protocol:** The goal is to design a routing protocol that can be successfully applied to fulfill the routing requirements of the four application areas identified for ROLL. Early contributions towards this goal have been made, and at the time of writing the initial ROLL protocol is being designed.

ROLL architecture

The architecture of LLNs is very different from that considered by MANET protocols or in research work done on wireless sensor networking. In fact it has more in common with traditional intra-domain IP routing methods. The ROLL protocol can be classified as a proactive distance-vector algorithm with advanced options for constraint-based routing, multi-topology routing and traffic engineering. The key requirements or assumptions for LLNs that affect the routing architecture are:

- ✓ LLNs are Internet-connected stub networks, with support for multiple points of attachment (multiple border routers to other IP networks).
- ✓ The majority of traffic flows are going to or from border routers using unicast, point-to-multipoint or multipoint-to-point flows. Node-to-node communication is less common, but may require specific constraints.
- ✓ Support for dynamic topologies and mobility is required.
- ✓ Support is required for multipath routing, and thus multiple forwarding options.
- ✓ Support is required for multiple node and route metrics, and their application in constraint-based and multi-topology routing. The evolution of metrics and support of multiple scenarios are important.
- ✓ A course-grained depth metric is assumed for general use, which is independent of the specific scenario. It is not assumed that this metric provides absolute loop avoidance.
- ✓ Routers in LLNs have limited memory resources.
- ✓ Most applications will require enterprise-class security.

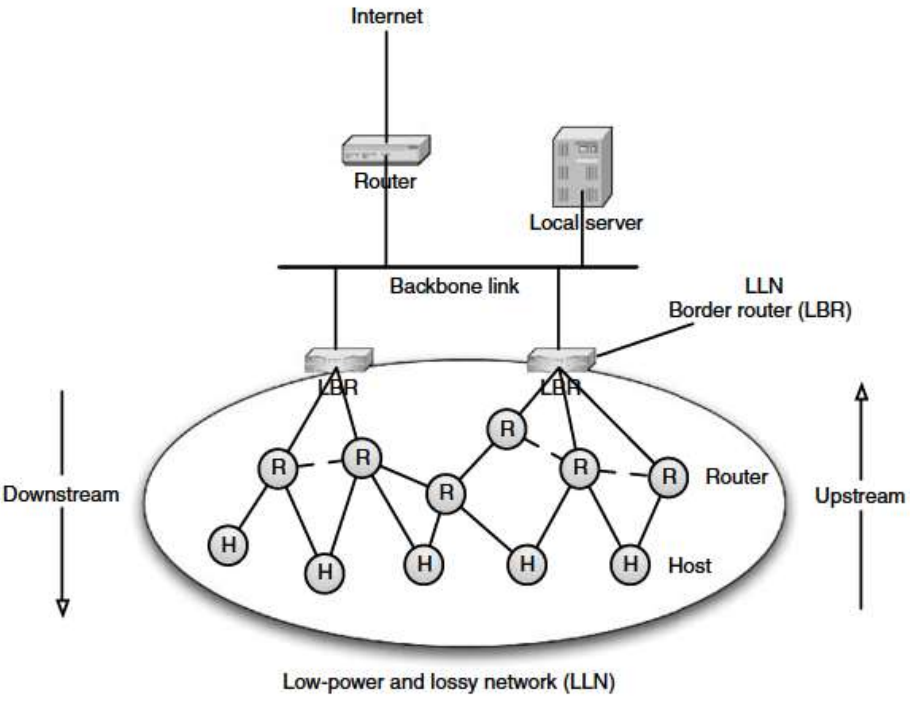


Figure 4.11 The ROLL architecture.

There are two concepts important to understanding ROLL protocol operation:

Metric granularity: ROLL uses the concept of a very granular (16–32 values) route metric called *depth*. This metric is used by the basic ROLL protocol mechanisms for building the graph, making use of siblings and for loop avoidance. The evaluation of depth is simple for all routers and nodes and is independent of the application scenario. In addition, fine-grained sets of metrics (such as those presented in Section 4.2.4) along with evaluation algorithms are applied in an application-specific manner to achieve routing on the basic graph structure.

Routing time scale: ROLL makes routing decisions on two different time scales: route-setup time and packet-forwarding time. In *route-setup time* the routing protocol maintains the basic graph topology and routing tables using static or slowly moving metrics, which is a continuous process. In addition, ROLL enables *packet-forwarding time* decisions to be made using dynamic metrics on a packet-by-packet basis, for example the immediate use of alternative next-hop routers upon failure.