Routing Protocols for 6LoWPAN

Matthias Felsche, Alexander Huhn, and Horst Schwetlick

HTW Berlin, Wilhelminenhofstr. 75A, 12459 Berlin, Germany
matthias.felsche@student.htw-berlin.de,
{Alexander.Huhn,Horst.Schwetlick}@HTW-Berlin.de

Abstract. Since the IPv6 over Low-Power Wireless Area Networks (6LoWPAN) standard does not define routing protocols, several approaches have been made to reuse and adapt existing MANET protocols. Extensive current research and development is conducted at the ROLL working group. This group defines routing requirements and provides solutions not only for 6LoWPAN but also for Low-Power and Lossy networks (LLNs) in general. The currently proposed routing protocol, RPL, is a generic distance-vector protocol which builds a distance oriented directed acyclic graph (DODAG) as topology. This paper provides a short overview over these routing protocols and algorithms, particularly for the use with IEEE 802.15.4 networks.

Keywords: 6LoWPAN, Routing, Protocol, Wireless Sensor Networks, AODV, ROLL, RPL.

1 Introduction

The IETF standard 6LoWPAN, especially together with the IEEE 802.15.4 for the physical and the MAC layer becomes increasingly important for building Wireless Sensor Networks (WSNs). These standards allow different network topologies. Multi-hop data transfer over intermediate nodes serves to increase the covered area of a network and reach remote data nodes, and simultaneously keeps the transmit power of the single node low. Routing plays a key-roll in this multi-hop data transfer. It enables vast wireless sensor networks, comprised of tiny, low-power, and cheap modules, to cover large areas densely. Since data can be delivered from every distant corner to one or more central data-sinks, the new areas of applications range from huge urban sensor networks, for example controlling traffic or street lighting, over environmental and agricultural monitoring, to house and home networks, industrial control and general sensor actor networks.

The 6LoWPAN as well as the 802.15.4 standard does not define any routing protocols. On the other hand a vast number of routing protocols have been developed. Ranging from protocols based on random-forwarding algorithms inspired by the behaviour of ants like Rumor Routing[2], over geographic or geometric approaches like GEM/VPCR [8], to hierachical clustering schemes like LEACH [7] just to name a few. Overviews, particularly for wireless sensor networks, are given in [9], [6], [10].

M.L. Reyes et al. (Eds.): IT Revolutions 2011, LNICST 82, pp. 71–83, 2012.

An overview over routing protocols applicable to 6LoW-PAN will be given in this paper. Firstly, chapter 2 will provide an introduction into routing for wireless sensor networks. The AODV and DYMO as common MANET-protocols (ad hoc networks) are treated in chapter 3 as predecessors of the proposed 6LoWPAN-protocols LOAD and DYMO-low, which are described in chapter 4.1 and 4.2. Chapter 4 as a whole will be about Routing in 6LoWPAN while the focus of this paper lies on the protocol RPL which is described in detail in chapter 4.4.

2 Routing in Wireless Sensor Networks

Enable routing and forwarding is one of the main duties of the network layer defined by the OSI-model. Others are addressing of single and multiple network-nodes and creating and maintaining a network-topology. Well-known protocols for these purposes are IP, IPv6 and ICMP (and ICMPv6) setting up the network-layer of the internet. In the internet-domain common routing-protocols are RIP, OSPF or BGP as an exterior gateway protocol.

There are two crucial points to every routing-protocol:

1. How to provide information for wise forwarding-decisions, a Routing Information Base (RIB) ?
2. To which neighbor to forward a packet not addressing a node itself? In other words: How to forward?

The RIB is mostly maintained in form of a routing table listing possible destinations and their next-hop-neighbor or the whole route. The main problem about building and maintaining a routing table are the memory-constraints of typical WSN-nodes (e.g. TI CC2430 SoC which has $2 * 4096$ Bytes of SRAM). Imagine a proactive protocol designed to always find the shortest path from sender to receiver taking the least hops and consuming the least possible energy. Such a protocol has to know all the links in the network. This is obviously no option for a WSN. Routing state should be kept as little as possible. The least possible routing information is 0 Bytes. A possible forwarding-algorithm here would be

Flooding: *Broadcast a package not addressing yourself to every neighbor you have.* As every node receiving such a package except the destination would forward the package until an eventually implemented hop-limit is reached, an unbearable amount of traffic would arise even if the sender already received the packet. Another possibility is

Gossiping: *Forward a package not addressing yourself to a random neighbor.* As is it is possible for a package to take the wrong direction it is likely to be discarded before it reaches its destination or to take a detour although a shorter path exists. At all it depends on coincidence that a package will reach its destination. We can only talk in terms of probability here.

It is clear that an appropriate routing protocol for the wireless sensor network domain has to provide routing information in a way that the overhead of gathering and maintenance of these informations trades off against the quality of the forwarding-decisions made possible with it.

One can summarize some goals for every routing protocol applicable on wireless sensor networks:

– minimizing the amount of transmissions
– minimizing the redundancy of sent data
– minimizing additional overhead
– minimizing and balancing the overall power-consumption
– maximizing the network-lifetime

2.1 Impact on Network-Lifetime

As network-lifetime is a critical parameter to WSN-Applications it is important to know what an effect on network-lifetime an efficient routing protocol can have. For reasons of simplicity Alonso et. al. [1] assumed a multi-hop-network where nodes continuously deliver data to one or more central data sinks, each sending with the same energy. Furthermore most of the nodes are more than one hop away from any sink, no data-aggregation is done and no package shall visit a node more than once, in other words routing shall be "effective". For such networks they proved that the maximum dissipated energy is less than or equal to:

$$(2s_1 - 1) * m^T \tag{1}$$

where s_1 denotes the number of nodes in one-hop-neighborhood to a data-sink and m^T is the minimal power-consumption after time T. In other words, no "effective" routing will be worse than $2s_1 - 1$ than the possible optimum. Thus the lifetime of a network with 4 nodes close to the data-sink can be improved by factor 7 by an optimal routing-algorithm[1].

Worst routing energy-consumption will be much worse when "non-effective" algorithms like flooding are taken into account where packages are copied and do take loops and additional ways not directed at the destination. The derived optimum can be improved by applying data-aggregation and sending with adaptive energy.

3 MANET-Routing Protocols

Existing IP routing protocols were designed for connecting the internet whose topology is strictly hierarchical with a backbone and attached sub-nets. Its nodes are usually made of stationary desktop-computers, mainframes, servers, laptops etc. All devices having a static position in the network, a durable source of energy and nearly no memory- or computing-constraints. In this hierarchical network structure two kinds of routings are necessary: Interior gateway protocols for routing inside autonomous systems like OSPF or IS-IS and exterior gateway protocols like BGP for connecting the different systems. These protocols are usually located at dedicated routers which interconnect to each other on the outside

[1] Regarding the sum of necessary transmissions.

of an autonomous system and care for the hosts on the inside by forwarding their traffic to the rest of the internet. The IP-Protocol-Stack was developed for certain types of devices and a certain domain. With the development and triumph of wireless technologies and thus the upcoming of a whole new class of mobile and IP-able devices new needs arised to provide a more suitable protocol-stack to connect the new kinds of mobile networks to the stationary internet.

In 1997 the IETF formed a working group called MANET (Mobile Ad Hoc Networks) whose aim it is to develop IP-based routing protocols for wireless ad hoc networks within static and dynamic topologies supporting e.g. mobility of nodes. MANET networks are mostly mesh-networks where every node acts as host and router unlike typical IP-networks where these roles are divided between special devices.

The first proposals for 6LoWPAN routing protocols were very much inspired by the work of the MANET working group. In particular AODV and DYMO were considered to be adapted for low-power WPANs.

3.1 Ad hoc On-Demand Distance Vector (AODV) Routing

AODV was first published as an internet-draft in 1998 and became a RFC [18] in 2003. It is a reactive protocol. Routes are discovered when traffic emerges. Routing state is limited to actively used routes.

Before sending a packet Node A consults the routing table for a next hop to the destination. If there is no entry a *RREQ*-message is sent to discover a route. Every node on the way of the *RREQ*-message remembers where it came from in a reverse-routing-table. The destination B or a node having B in its routing table answers with a *RREP*-message responding successful route-discovery. This message is sent backwards to A. Every node on the way of the *RREP*-message stores Bs adress and the predecessor of the *RREP*-message in the routing-table. The route is established. Everytime a route is used the forwarding router extends its lifetime. Hence only active routes are maintained and older ones become discarded after a while to save memory.

In case of link loss on an active route the node who recognizes it sends a *RERR*-message to all precursors using this route. These nodes delete their routing-table-entry belonging to that route and forward that message along their stored precursors of that route until it receives the original sender who has to do a Route Discovery again.

AODV uses hop-count as metrics.

3.2 DYnamic Mobile On-Demand (DYMO) Routing

The MANET working group is busy with a successor of AODV called DYMO[3]. It has been designed as an improved hybrid of AODV and DSR. It is still an internet draft.[2]

[2] All cited internet drafts are work in progress and thus if not obsoleted or expired still subject to changes.

DYMO uses the same route-discovery and maintenance mechanisms as AODV. Additionally it supports path-accumulation. *RREQ*-packages and *RREP*-packages establish routes to their originator on every node on their way, thus routing-information is gathered more quickly and with less overhead compared to AODV. But more eventually not needed routing information is gathered.

It uses the MANET packet format [4] e.g. allowing aggregation of messages contained in one package. Hop-counts are used for routing metrics by default but link costs can be added.

4 Routing in 6LoWPAN

To create inter-operable implementations of a given protocol one has to know how to encode the information stored in MAC-Frames, Network-Headers and so on. This is where standards come in handy. One of them is 6LoWPAN. 6LoWPAN stands for *IPv6 over Low power Wireless Personal Area Network*. 6LoWPAN is an IETF working group which has created several standards defining the 6LoWPAN protocol. It provides a compressed version of IPv6 for WPANs which can be connected to the internet via a border-router which acts as gateway.

As constraints are heavy on typical 6LoWPAN-devices every layer of the network stack needs to be reduced to the minimum. For example the 6LoWPAN-header has been compressed by a factor of 8 from 48 Bytes (IPv6 and UDP header) down to 6 Bytes. The 6LoWPAN Working Group at the IETF found both AODV and DYMO very interesting but not suitable as they are, so they reworked these protocols to fit the 6LoWPAN routing requirements.

The following routing protocols are no longer actively maintained. They are nonetheless presented here to provide an overview of the research done for this particular domain.

4.1 LOAD

6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD)[11] is an adaption of AODV to low-power WPANs. LOAD has been actively maintained until 2007. It is nonetheless interesting what has been done to achieve less overhead and less memory consumption.

LOAD works on top of the adaption layer using link-layer EUI-64 addresses or the 16 bit short ones. It works just like AODV despite some important abbreviations.

In route discovery mode it is only the destination who may answer a *RREQ*-message, thus loops are avoided and the package-header shrinks.

When a link is lost and a data-package cannot be forwarded, the node who cannot forward requests a new route to the destination of the data-package by sending a *RREQ*-message with the Local-Repair flag set. If the destination remains unreachable the repairing node returns a *RERR*-message to the originator of the data-package. The originator will only get a *RERR*-message, if there is no route left to the destination. Repairable link loss will be repaired without a hassle.

The *RERR*-message may only contain one unreachable destination. While AODV stores the precursors for returning RERR-messages and the routing table for usual routing, LOAD only uses the latter. *RREP*-messages are directly routed to the destination using a reverse routing table which stores currently received *RREQ*-messages. *RERR*-messages are routed directly so no precursor list is needed.

While AODV only regards hop-count as routing metrics LOAD can use route cost too. With a link-cost threshold called Link Quality Indicator(LQI) it avoids routes containing weak and unreliable links.

4.2 DYMO-Low

Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing [12] is the adaption of DYMO to the requirements and constraints exhibited by 6LoWPAN. It is located underneath the IP-Layer providing a link-layer mesh-network like Ethernet does.

Routing messages are sent via local multicast and are given a sequence number to avoid loops that could occur with this method. Every node processing routing messages (*RREQ*-messages and *RREP*-messages) creates or updates a routing table entry to its originator. Routing table entries consist of the destination, the next-hop, two timeout-timestamps explained below and a route cost to determine an optimal route. Route discovery packets may only be answered by the destination to make sure they traverse all intermediate nodes to create the full route. DYMO-low routes become invalid after a certain (configurable) time and get deleted later. These timers get updated every time a packet is successfully forwarded. Route cost and Link Quality are explicitly used as routing-metric. DYMO-low uses Link-Layer Acknowledgements to check for availability of routes. If there is no valid routing table entry for a data packet it is dropped and a *RERR*-message is sent to the data-packets originator. DYMO-low supports an energy threshold. Nodes that deceed a certain energy-level may delay processing of routing messages to wait for other ones to do that job.

DYMO-low expired by the end of 2007.

4.3 HiLow

Hierarchical Routing over 6LoWPAN (HiLow) is a rather short and unspecific draft which exists in its second revision [13]. As LOAD and DYMO it expired by the end of 2007.

HiLow makes use of dynamic assignment of 16-bit short addresses to reduce necessary addressing-overhead. A HiLow-node keeps all its neighbors in a Neighbor Table. A relationship-flag determines a neighbors status (parent or child).

Starting from the router a tree is built where every node has got a maximum of MC children. A short-address is dynamically assigned when a node attaches itself to the existing tree according to the following function:

$$MC * A_P + 1 \tag{2}$$

where A_P is the address of the parent node.

Packets are routed up and down the tree using the 6LoWPAN Mesh-Header. Very little overhead is required because the addresses are routable with the rule above. Every node knows for any address in the network if it is an ancestor or successor of itself. If it is neither of both the packet is sent upwards until a common ancestor is found which can route it downwards. Thus every node knows for any packet where to forward it, to a parent or one of its children.

The mechanisms for route-maintenance and loss response were not worked out.

All of these drafts are no longer maintained mainly beacause the 6LoWPAN working group decided to outsource the routing-problem. Current work in routing for 6LoWPAN is done by the ROLL working group.

4.4 RPL

Because routing is a problem apart from the main tasks of 6LoWPAN the IETF decided to found a new working group to deal with that topic in particular. This new group was founded in 2008 and is called *Routing Over Low-power and Lossy networks (ROLL)*. Its aim is to prepare routing requirements and solutions for IPv6 based Low-Power and Lossy Networks(LLN) in general. In different RFCs ROLL split the wide field of LLN-domains into four subdomains for which different kinds of applications and thus different requirements are necessary:

1. *urban networks* (RFC 5548) which have to be able to deal with a number of nodes ranging from 100 to 10 mio.
2. *industrial applications* (RFC 5673) which comprise areas like process control and factory automation. Applications in this domain have to bear 10 to 200 field devices.
3. *networks in buildings* (RFC 5867) where 2000 nodes (sensors and actuators) are the minimum. This comprises applications like HVAC and general building-automation.
4. *home networks* (RFC 5826) where at least 250 nodes should be a reasonable amount.

An appropriate routing protocol for LLNs according to the ROLL working group has to fit into all of these domains which e.g. means that it has to get along with different network sizes ranging from 10 to 10 mio. nodes. So scalability is an important concern.

Further criteria have been defined in a survey which is available as (no longer active) internet draft [15] :

1. *Routing State.* Protocols which require nodes to have to store lots of routing information aren't appropriate for LLNs. Thus protocols whose effort for routing state scales with network size or density fail.

2. *Loss Response.* Link-loss should be repaired locally (at best with a single one-hop broadcast). Protocols where link-loss affects the whole network fail.
3. *Control Cost.* The transmission cost must be limited by the data-rate and a little constant (measured at one node). E.g. if a protocol uses a beacon-mode its interval has to match the data-rate.
4. *Link and Node Cost.* A LLN-protocol must consider transmission cost and take into account the state of nodes (memory, battery capacity, lifetime ...) as routing metrics.

Existing IETF-protocols were checked if they fulfil these criteria. Among them AODV and DYMO.[3] See table 1 on page 78. The results were not satisfying. So the ROLL working group decided to create a new protocol.

Table 1. Results of draft-ietf-roll-protocols-survey

Protocols	**Routing State**	**Loss Response**	**Control Cost**	**Link Cost**	**Node Cost**
OSPF/IS-IS	fail	fail	fail	pass	fail
OLSRv2	fail	?	?	pass	pass
TBRPF	fail	pass	fail	pass	?
RIP	pass	fail	pass	?	fail
AODV	pass	fail	pass	fail	fail
DYMO	pass	?	pass	?	?
DSR	fail	pass	pass	fail	fail

This new protocol is called *IPv6 Routing Protocol for Low power and Lossy Networks (RPL)* which was first published in August 2009. It is still an internet draft[21]. It currently[4] exists in its 18th revision and is evaluated by the IESG to become an internet standard (RFC).

RPL is a proactive, hierarchical and generic distance vector protocol. It is mainly intended for use in wireless sensor networks. There are lots of additional drafts related to RPL describing main functionalities like a secuity framework, p2p-routes, routing metrics etc.[5] RPL is designed to fit into all of the four application-domains (s.o.). This is made possible by the definition of a generic core which can be extended and configured by an application-specific Objective Function (OF). The generic core defines the topology of the network, the means to establish and maintain it and the routing and forwarding algorithm. In all of these mechanisms the core consults the OF for specific decisions.

RPL assumes a typical 6LoWPAN-network with lots of LLN-nodes and one or more border-routers connecting them to the outer internet. The model behind the main network topology is a Distance Oriented Directed Acyclic Graph (DODAG). This is a directed graph where no route visits a node twice and the

[3] DYMO-low, LOAD and HiLow weren't considered.
[4] November 20, 2010.
[5] See the page of ROLL working group: `http://datatracker.ietf.org/wg/roll/`

destination-node is the only one without any downstream connections. There are lots of protocols based on DAGs: GEM[8], VPCR[14], the whole Link-Reversal-Routing family, among them Lightweight Mobile Routing (LMR)[5] and Temporally Ordered Routing Algorithm (TORA)[17]. This convenient hierarchical structure makes it possible to reduce control overhead because information about a nodes position is already given by the tree-like structure.

The DODAG-tree is established hop by hop starting from the border-router. The position in the tree is controlled by a node's rank which is computed by the OF and reflects the distance from the DODAG-root in terms of the OF. The rank can be a simple hop count but can also regard transmission cost and/or other parameters. This results in a tree with weighted graphs. First the border-router, then every attached node sends DIO-messages via link-local multicast. These messages include the DODAG root's identity, the used Objective Function, the routing metrics and constraints, in short the networks configuration, and the rank of the originator. Nodes in range use this information to attach themselves to the growing tree. They choose a node as parent if attaching to it results in the lowest rank compared to other candidates. If there is more than one DODAG in a network a node attaches to the one where it has most parents. Like this every node joins the tree and becomes configured according to the settings in the border-router. Upward routes are built. Traffic from the nodes to the border-router or the outer internet can flow. This tree-creation procedure is repeated according to a configurable trickle-timer[16].

Downward routes are maintained by every node sending DAO-messages upwards. These messages are generated every time a node joins a tree or when there are new informations arriving from a nodes subtree. They contain routing information for the whole subtree of the sending node. There are two operational modes for RPL: storing mode and non-storing mode. In storing mode nodes store routing-information for their subtree and packets are forwarded hop by hop. Non-storing mode is suitable if memory constraints do not allow routing tables on every node. Here DAO-messages are forwarded to the border-router who is the only one to store routing information. Downward packages are routed in a source routing manner. This enables traffic from the router to the nodes.

If the receiver of a package is in range, no routing is required and a simple data-package is sent to the link-layer address of that particular neighbor. Point-to-point (p2p) traffic has to go up the tree until it finds a common ancestor (storing mode) or up to the DODAG-root (non-storing mode) until it flows downwards to its destination.

With these routes established RPL basically allows traffic flows up and down the tree. While it is best for traffic from and to border-routers to take these ways, p2p-traffic within the tree should not go these crooked ways when better ones exist. Unnecessary traffic-congestions near the root would follow. In the worst case a package would be dropped at the root because its destination did not send a DAO-message. The advantages of the DODAG-structure orientated towards a border-router turn out to be disadvantages for efficient p2p traffic.

But nonetheless the basic algorithm can be applied here with some benefit. To establish routes complementary to the DODAG-structure a node sends a route discovery message (DIO-variant) towards his intended destination. There is a lot of flexibility here. Source routes or hop-by-hop, unidirectional or bidirectional, single or multiple routes can be established. Route and propagation constraints of the route discovery message have to be met. To fit into RPLs routing-scheme a temporary DODAG is created rooted at the originator to all its p2p-destinations. With this mechanism RPL optimizes all kinds of relevant traffic-flows: traffic from and to the router as well as p2p-traffic within the LLN.

RPL was designed to meet the requirements the ROLL-WG made up (see page 77). A RPL-node has to store certain routing-information for every DODAG it is part of. Beneath some global constant information about a certain DODAG, there is a set of dynamic information. The list of parent-candidate neighbors scales with network density and the routing table (in storing mode only) scales with the number of nodes in the sub-DAG, thus depends on the position relative to the DAG-root and the network size and structure. There is at least one mandatory element which scales with network density, so RPL will hardly get a *pass* for routing state.

If a packets destination does not exist in a routing table due to link loss, this package gets dropped. RPL knows two mechanisms to deal with that: *Global Repair* i.e. the DODAG-root regularly emits new DAG sequence numbers so the tree frequently rebuilds itself repairing lost links if possible. *Local Repair* is done by a node signalling that it lost its connectivity to the upper DAG-tree. Its children remove it from their parent set and wait for new DIO-messages to join a new version of the DODAG-tree. As a node in the tree is an ancestor of all the nodes in its sub-DODAG and all their upward routes traverse him, repairing a route with *Local Repair* necessarily only affects routes that use this link. So RPL gets a *pass* for Loss Response.

Due to its generic nature RPL is very flexible with its necessary amount of control messages. A minimal variant only allows traffic flows to the DAG-root. Within such a configuration the only control traffic is made of regular DIO-messages locally broadcasted by every node controlled by a trickle-timer. The message intervals are very low when there is no topology change in the network and increase when something happens. According to a RPL-simulation [19] the amount of control traffic in relation to data traffic increases the further a node is away from the DODAG-root but stays far beneath the data-rate. The worst case in the simulation is a relation of approximately 1:4 for control traffic to data traffic. So RPL gets a *pass* here.

Link and Node Cost can be configured in the OF. So RPL gets a *pass* here too.

RPL is already implemented and adopted in a number of stacks. There exists an implementation for Contiki [20], one for the Sensinode Nanostack 2.0 (called NanoMesh) and it is adapted by the ZigBee Alliance. Though RPL seems to be fully accepted by the industry it does not fully pass the roll-surveys criteria.

The Radio Engineering Group at the HTW-Berlin currently implements a wireless sensor network as a test bed for further development and practical review of protocol performance and application-design. The system consists of around 20 nodes based on the TI2430 SoC on a self-built development-board. These nodes run the 6LoWPAN-NanoStack 1.1.0 on top of freeRTOS. Since this GPL-licensed version of the NanoStack does not provide RPL-routing and version 2.0 is not available as free software the Radio Engineering Group is currently working on an RPL-implementation for version 1.1.0.

5 Summary

Different routing protocols for the 6LoWPAN standard have been reviewed and described i.e. proposals like HiLow, LOAD or DYMO-low. The latter two protocols are adapted versions of MANET protocols coming from a slightly different domain where constraints are not as strict as in low-power WPANs. A new IETF working group called Routing Over Low-power and Lossy networks (ROLL) was founded to develop routing protocols for this domain in particular. The Routing Protocol for Low-power an lossy networks (RPL) is the current proposal of the ROLL working group tailored to the constraints and requirements of low-power an lossy networks. RPL is a hierarchical and generic distance-vector protocol that can be easily configured and extended for one of the widespread application domains for Wireless Sensor Networks. It is reviewed in detail here.

RPL is an important milestone in the development of routing protocols for wireless sensor networks in general. Having RPL and 6LoWPAN there exists a complete[6] standardized, and thus well defined network-stack running on various hardware-platforms, including the IEEE 802.15.4, connecting the wireless sensor network domain to the internet.

References

1. Alonso, J.: Bounds on the Energy Consumption of Routings in Wireless Sensor Networks. In: Proceedings of the 2nd International Workshop on Modeling and optimization in Mobile, Ad Hoc and Wireless Networks, Cambridge, UK, März, pp. 62–70 (2004)
2. Braginsky, D., Estrin, D.: Rumor routing algorthim for sensor networks. In: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002), pp. 22–31. ACM, New York (2002)
3. Chakeres, I., Perkins, C.: Dynamic MANET On-demand (DYMO) Routing. draft-ietf-manet-dymo-21 (Internet-Draft) (Juli 2010), `http://www.ietf.org/id/draft-ietf-manet-dymo-21.txt`
4. Clausen, T., Dearlove, C., Dean, J., Adjih, C.: Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format. RFC 5444 (Proposed Standard) (Februar 2009) (Request for Comments), `http://www.ietf.org/rfc/rfc5444.txt`

[6] The transport layer is missing here but is not.

5. Corson, S., Ephremides, A.: A distributed routing algorithm for mobile radio networks. In: Conference Record. Bridging the Gap. Interoperability, Survivability, Security, Military Communications Conference, MILCOM 1989, pp. 210–213. IEEE (1989)
6. García Villalba, L.J., Sandoval Orozco, A.L., Triviño Cabrera, A., Barenco Abbas, C.J.: Routing Protocols in Wireless Sensor Networks. Sensors 9(11), 8399–8421 (2009) ISSN 1424–8220, http://www.mdpi.com/1424-8220/9/11/8399/
7. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications 1(4), 660–670 (2002)
8. James, N., Song, D.: GEM: Graph EMbedding for routing and data-centric storage in sensor networks without geographic information. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003), pp. 76–88. ACM, New York (2003)
9. Jiang, Q., Manivannan, D.: Routing protocols for sensor networks. In: First IEEE Consumer Communications and Networking Conference, CCNC 2004, pp. 93–98. IEEE (2004)
10. Karl, H., Willig, A.: Protocols and Architectures for Wireless Sensor Networks. Wiley Interscience (2007)
11. Kim, K.: 6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD). draft-daniel-6lowpan-load-adhoc-routing-03 (Internet-Draft) (Juni 2007), http://tools.ietf.org/html/draft-daniel-6lowpan-load-adhoc-routing-03
12. Kim, K.: Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing. draft-montenegro-6lowpan-dymo-low-routing-03 (Internet-Draft) (Juni 2007), http://tools.ietf.org/html/draft-montenegro-6lowpan-dymo-low-routing-03
13. Kim, K.: Hierarchical Routing over 6LoWPAN (HiLow). draft-daniel-6lowpan-hilow-hierarchical-routing-01 (Internet-Draft) (Juni 2007), http://tools.ietf.org/html/draft-daniel-6lowpan-hilow-hierarchical-routing-01
14. Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric ad-hoc routing: of theory and practice. In: Proceedings of the Twenty-Second Annual Symposium on Principles of Distributed Computing (PODC 2003), pp. 63–72. ACM, New York (2003)
15. Levis, P.: Overview of Existing Routing Protocols for Low Power and Lossy Networks. draft-ietf-roll-protocols-survey-07 (Internet-Draft) (April 2009), http://tools.ietf.org/id/draft-ietf-roll-protocols-survey-07.txt
16. Levis, P.: The Trickle Algorithm. draft-ietf-roll-trickle-08 (Internet-Draft) (August 2010), http://tools.ietf.org/id/draft-ietf-roll-trickle-04.txt
17. Park, V.D., Corson, M.S.: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In: Proceedings of the INFOCOM 1997. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution (INFOCOM 1997), p. 1405. IEEE Computer Society, Washington, DC (1997)
18. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental) (Juli 2003), http://www.ietf.org/rfc/rfc3561.txt

19. Tripathi, J.: Performance Evaluation of Routing Protocol for Low Power and Lossy Networks (RPL). draft-tripathi-roll-rpl-simulation-06 (Internet-Draft) (Januar 2011), http://tools.ietf.org/id/draft-tripathi-roll-rpl-simulation-06.txt
20. Tsiftes, N., Eriksson, J., Dunkels, A.: Poster Abstract: Low-Power Wireless IPv6 Routing with ContikiRPL. In: Proceedings of ACM/IEEE IPSN 2010 (2010)
21. Winter, T., Thubert, P.R.: IPv6 Routing Protocol for Low power and Lossy Networks. draft-ietf-roll-rpl-18 (Internet Draft) (August 2011), http://tools.ietf.org/id/draft-ietf-roll-rpl-18.txt