Path Accumulation Extensions for the LOADng Routing Protocol in Sensor Networks

Thomas Clausen and Jiazi Yi

Laboratoire d'Informatique (LIX), Ecole Polytechnique, France

Abstract. The "Light-weight On-demand Ad-hoc Distance-vector Routing Protocol – Next Generation" (LOADng) is a simple, yet efficient and flexible routing protocol, specifically designed for use in lossy networks with constrained devices. A reactive protocol, LOADng – as a basic mode of operation – offers discovery and maintenance of hop-by-hop routes and imposes a state in intermediate routers proportional to the number of traffic paths served by that intermediate router.

This paper offers an extension to LOADng, denoted LOADng-PA (Path Accumulation). LOADng-PA is designed with the motivation of requiring even less state in each intermediate router, and with that state being independent on the number of concurrent traffic flows carried. Another motivation the design of LOADng-PA is one of monitoring and managing networks: providing more detailed topological visibility of traffic paths through the network, for either traffic or network engineering purposes.

1 Introduction

Since the late 90s, the Internet Engineering Task Force (IETF)[1] has embarked upon a path of developing routing protocols for networks with increasingly more fragile and low-capacity links, with less pre-determined connectivity properties and with increasingly constrained router resources. In '97, by chartering the MANET (Mobile Ad hoc Networks) working group, then subsequently in 2006 and 2008 by chartering the 6LoWPAN (IPv6 over Low power WPAN) and ROLL (Routing Over Low power and Lossy networks) working groups.

The MANET working group converged on the development of two protocol families: reactive protocols, including AODV (Ad hoc On-demand Distance Vector routing [1]), and proactive protocols, including Optimized Link State Routing (OLSR) [2,3]. A distance vector protocol, AODV operates in an *on-demand* fashion, acquiring and maintaining paths only while needed for carrying data, by way of a Route Request/Route Reply exchange. A link state protocol, OLSR is based on periodic control messages exchanges, and each router proactively maintaining a routing table with entries for all destinations in the network, which provides low delays but constant control overhead.

LOAD [4] is a protocol derived from AODV [1], simplified for LLNs, and standardised by the ITU-T as part of the G3-PLC standard [5] for mesh-under

[1] `http://www.ietf.org`

R.C.-H. Hsu and W. Shangguang (Eds.): IOV 2014, LNCS 8662, pp. 150–159, 2014.

routing for utility (electricity) metering networks. The emergence of LLNs thus triggered a renewed interest in AODV-derived protocols for specific scenarios, resulting in continued work within the IETF [6] and [7] for the purpose of standardisation of a successor to LOAD – denoted LOADng (the Lightweight On-demand Ad hoc Distance-vector Routing Protocol – Next Generation). LOADng incorporates the experiences from deploying LOAD – including, but not only, in LLNs – and has been accepted as part of an update to the G3-PLC ITU-T standard for communication in the "smart grid" [8].

1.1 The Lightweight On-demand Ad Hoc Distance Vector Routing Protocol - Next Generation (LOADng) Overview

A reactive protocol, the basic operations of LOADng [15,6] include generation of Route Requests (RREQs) by a LOADng Router (originator) and flooded through the network when discovering a route to a destination, and Route Replies (RREPs) generated by the sought destination ad transmitted to the originator by way of unicasts. When an intermediate router forwards a RREQ, it installs temporary routing table information towards the originator of the RREQs – the "reverse route" from the destination to the originator. When the sought destination receives a RREQ, it will respond by an unicast RREP, which is forwarded along this installed reverse route – and the forwarding of which will serve to install a "forward route" from the originator to the destination. Thus, for each bidirectional path through a LOADng router, four entries are thus maintained in the routing table: for directions, an entry is recorded for the "next hop" and for the "destination" via that "next hop".

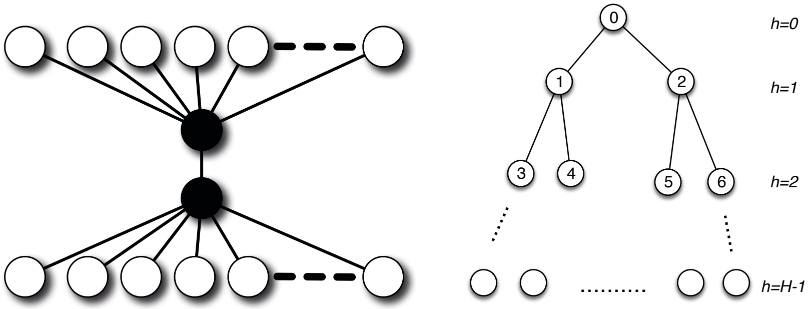
One of the key features of LOADng is, that it combines simplicity with extensibility: the core protocol provides mechanisms for discovering and maintaining bi-directional routes between pairs of routers, but offers the ability to develop functional extensions, optimising its behaviour and performance for specific deployments, topologies and traffic patterns.

LOADng provides the information required for hop-by-hop routing of data packets: a router will know the next hop towards the destination, for a data packet, but not the full path that the packet will take. To this end, the header of a data packet contains only the destination address, relying on intermediate routers to be able to make forwarding decisions based on their local knowledge of the routing topology. This provides agility for each router to make a local decision (*e.g.,* if local connectivity changes more frequently than routing updates can be propagated globally through the network) and, *e.g.,* enables "fast re-routing" mechanisms to engage and improve data packet delivery when a data packet arrives at a router without, or with outdated, topological information [12,13].

The LOADng Collection Tree Protocol (LOADng-CTP) is of the protocol extensions, developed for LOADng [9]. This extension permits efficient construction of a bi-directional collection tree, rooted in one router and covering the entire network. LOADng-CTP permits reducing the number of RREQ flooding operations from (n-1) to 2 in order to construct bi-directional paths between the root and all other routers in the network, but otherwise incurs the same state requirements as described above. This extension is applicable for deployments where a central controller, or monitoring entity is present and operating the network, *e.g.,* as is the case in a smart grid management, or for home/building/factory automation.

1.2 The Case for Source Routing and Path Accumulation

There are, however, potential downsides to hop-by-hop routing combined with a reactive protocol: by their very nature, reactive protocols acquires and maintains only topological information about actively used paths – and, at that – at their basic form only "next hops" along those paths. Thus, no router has access to a "global view" of the network topology, or even of which destinations are reachable through which paths. This may be unfortunate in some scenarios, *e.g.,* for management purposes: paths from a central unit (*e.g.,* a monitoring station) to a set of devices (*e.g.,* sensor devices) may, all pass through a faulty intermediate router rendering the network inoperative – and absence of a topological view of the paths actually used, this faulty router may be difficult to identify. Also, lack of such global information renders establishing path-diversity, *i.e.,* that data packets (as far as possible) are not concentrated across a few unfortunate routers (which may, if battery-driven, see their energy depleted prematurely, or experience congestion) or across a few unfortunate links (which may incur higher media contention and losses) hard to accomplish.



(a) Simple network topology with choke-point.

(b) Bi-directional traffic in a collection tree network.

Fig. 1. Example network topologies

A final downside is, and probably the most critical issue for memory-constrained devices, of hop-by-hop routing is, that with hop-by-hop routing and a reactive protocol, the state required in each router is proportional to the number of active paths passing through that router. Thus, any individual device in the network may, worst case, be required to maintain a routing table with entries to many (or all) destinations, as illustrated in figure 1(a): each of the white routers on top establishes a (bi-directional) path to each of the white routers in the bottom, requiring the two intermediate black routers to, each, maintain entries for each of the white routers in the network (plus, for each other, of course). In large networks with a topology in which a router finds itself an intermediary of a large number of active paths, this may put undue requirements on some routers. Figure 1(b) shows an example of such a network topology: if the root (0) is a network controller or monitoring station, and as such needs establishing bi-directional paths to all destinations in the network for command-and-control, then the two routers immediately below it (1 and 2), each, will need to maintain routing table entries for all destinations in their respective sub-trees. While, in such a case, the root (0) typically can be provisioned with sufficient resources, in an unplanned network, or a network evolving over time, it may not be possible to determine which non-root routers will thus become "choke-points" and therefore will also need to be provisioned to be able to maintain this much routing state.

The alternative to hop-by-hop routing is source-routing: the router, by which a data packet enters a routing domain, inserts the complete path information (for transversing that routing domain — which may be the complete path, up to and including the destination) into the data packet header. Consequently, intermediate routers need only inspect this information when making forwarding decisions. In order for a router to be able to insert complete path information into data packet headers, sufficient topological information must be available to construct the complete path – the task of acquiring this topological information is denoted *path accumulation*.

Literature contains a few examples of such path accumulation mechanism for reactive protocols: Dynamic Source Routing [14] and AODV-PA [16], both, use aggressive caching mechanisms to reduce the cost of route discovery. Because the routers need not only keep the routing tuples to the destinations of the route discovery, but also the routing tuples of the intermediate routers, those protocols requires maintaining more state in all routers, possibly exceeding the memory capacity of devices in a sensor network.

1.3 Paper Outline

This paper explores path accumulation extensions to LOADng, intended to – in topologies and with traffic patterns where it is appropriate – alleviate these difficulties, and (i) eliminate the requirements of routers in a strangulation point, (ii) provide visibility of the paths used and (iii) enable traffic engineering, *e.g.,* for path diversity. Consequently, the remainder of this paper is organised as follows: section 2 specifies the the path accumulation extension. The performance of LOADng with this path accumulation is, then, studied, and compared with core LOADng, by way of network simulations, and results are presented in section 3. This paper is concluded in section 4.

2 LOADng Path Accumulation

Each of the two message types used for route discovery by LOADng – RREQ and RREP – provide an option for doing path accumulation, by way of inclusion of appropriate TLVs and addresses, as the message is forwarded. While doing path accumulation in either message will result in the source learning the full path used, and while the same TLVs and general mechanism can be applied to either RREQ or RREP, the two approaches are not equivalent, and are therefore discussed independently in sections 2.1 and 2.2 below. Either way, once a path is established by way of Path Accumulation, user data is forwarded by way of source routing, carrying the complete path in each data packet: essentially, trading off router state for channel occupation.

2.1 RREQ Path Accumulation (RREQ-PA)

As indicated in section 1.1, a router seeking a path to a destination initiates Route Discovery by generating an RREQ, flooded through the network. Augmenting the RREQ so as to support Path Accumulation (RREQ-PA) requires including a flag, requesting that intermediate routers include path information when receiving the RREQ and prior to either forwarding or replying (by way of an RREP).

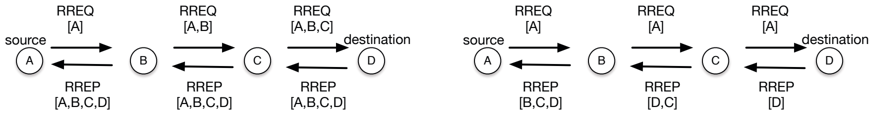
The regular LOADng processing of RREQs, specified in [6], can be modified slightly, so as to reduce the amount to state is required in each intermediate router: LOADng [6] stipulates that while forwarding RREQs, temporary "reverse path information" towards the source is installed in each intermediate router so as to facilitate forwarding of RREPs. The destination for an RREQ extend with RREQ-PA, will know the exact path that the RREQ took, and will be able to use this path for source-routing the RREP back to the initiator of the Route Discovery process. Figure 2(a) gives an example of RREQ-PA route discovery from A to D.

If the initiator of the Route Discovery process conveys additional requests – such as multiple disjoint paths, or paths avoiding/passing specific routers, the destination for a RREQ extended with RREQ Path Accumulation will be able to make intelligent decisions as to which RREQs to respond to.

The benefits of this operation are that (i) either end of the path will know the full path, all intermediaries – although only the initiator of the Route Discovery process will know if the path is actually bi-directional; (ii) traffic engineering is possible – the destination router can wait and receive multiple RREQs, and elect to send RREPs in response to multiple RREQs, for example so as to ensure that multiple non-overlapping paths are made available for carrying traffic, and (iii) no intermediate routers are required to maintain any state.

The downside to RREQ-PA is, that the over-the-channel overhead can get large: RREQs are flooded through the network, and each additional octet added is amplified by the number of routers which are participating in the network, consuming channel capacity and energy, both.

Either way, once a path is established by way of RREQ-PA, user data is forwarded by way of source routing, carrying the complete path in each data packet: essentially, trading off router state for channel occupation.



(a) RREQ Path Accumulation (b) RREP Path Accumuation

Fig. 2. RREQ-PA and RREP-PA examples. *A* is the source and *D* is the destination.

2.2 RREP Path Accumulation (RREP-PA)

An alternative, somewhat lighter, approach is to do path accumulation by way of RREPs: a router, seeing a path to a destination generates an RREQ, which is augmented by way of a flag requesting only the destination router to process the RREQ differently: The regular LOADng processing of RREQs, specified in [6], is maintained, reverse paths installed, etc., by all intermediate routers.

When the destination router receives an RREQ indicating "path accumulation requested", it will generate an RREP – which, also, will contain that "path accumulation requested" flag – and send it in unicast along the reverse path installed when forwarding the RREQ. Then all the intermediate routers are accumulated in RREP messages. Figure 2(b) depicts an example of RREP-PA from *A* to *D*. Processing of an RREP in intermediate routers is changed, in a fashion similar to that of RREQ processing in section 2.1: on receiving a RREP, add the address of the router, and a distance to the source of the RREP. When the router, which initiated the Route Discovery process, receives this RREP, it will have acquired the complete path traveled between itself and the destination.

The benefits results of this operation are that (i) the source – but not the destination – acquires a full path, (ii) intermediate routers need only one entry towards the source – which, in case of path accumulation being used for providing a topological view of the network to a root means, that each intermediate router will maintain exactly one entry for the "upwards route" towards that root, and (iii) the size of RREQs, which are flooded through the network, does not change as compared to LOADng – which was otherwise the case for RREQ-PA.

The potential downside to RREP-PA is, that neither traffic engineering nor path diversity is enabled: (i) the destination router, responding to an RREQ, is not able to, in any meaningful way, select to respond to one RREQ over another (except for, that is, the metrics contained in the RREQ [6], since the RREQs do not carry any detailed path information, and (ii) when generating the RREP, this is forwarded in unicast and by way of the (unique) "reverse path" installed as part of RREQ processing.

3 Performance Evaluation

In order to understand the performance impact of the path accumulation extension to LOADng, this section presents a set of ns2 simulations, comparing LOADng with and without path accumulation. Of particular interest is, of course, the size of routing table required, as well as the increase in control message size.

3.1 Simulation Settings

Simulations were made with static scenarios of 63 to 500 routers, randomly placed in a square field, but maintaining a constant density of routers. From among the routers, one is designated as “root” and the network is subject to traffic between the root and each other router in the network: sensor-to-root traffic, emulating a smart meeter sending its reading to a “controller”, and SCADA-like bidirectional data exchange every 5 seconds, lasting for 80 seconds each. The simulations were undertaken using the TwoRayGround propagation model and the IEEE 802.11 MAC[2].

As discussed in section 1.2, path accumulation is proposed, in part, to reduce memory requirements in intermediate routers, and in part to provide a topological view of the network to a central management station in the network. As a consequence, the performance LOADng-CTP alone, as described in section 1.1, is compared with LOADng-CTP combined with RREQ-PA and RREP-PA.

3.2 Simulation Results and Discussions

For packet delivery ratio and end-to-end delay, all three protocols combinations exhibit similar performance, depicted in [9].

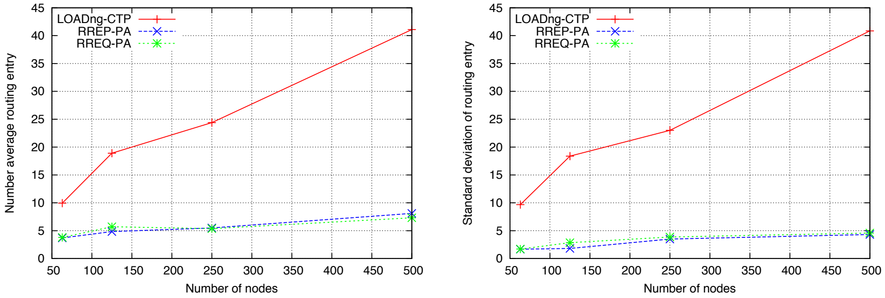
Figure 3(a) shows the average number of routing table entries required for the three protocol combinations. As expected, with path accumulation, the average number of routing table entries is (almost) constant – opposed to LOADng-CTP, where each router maintains state for its entire sub-tree. For the same reason, LOADng-CTP exhibits significant variance on the number of routing table entries, as shown in figure 3(b).

The overhead of RREQ messages (each RREQ retransmission is counted) is presented in figure 4. The number of RREQs retransmitted is generally the same, however as expected, RREQ-PA incurs larger RREQ messages, with as consequence up to twice as many bytes sent across the network as the LOADng-CTP and RREP-PA.

The overhead of RREP message transmission is given in figure 5, with RREQ-PA as expected exhibiting larger RREP messages and, thus, the most bytes sent across the network.

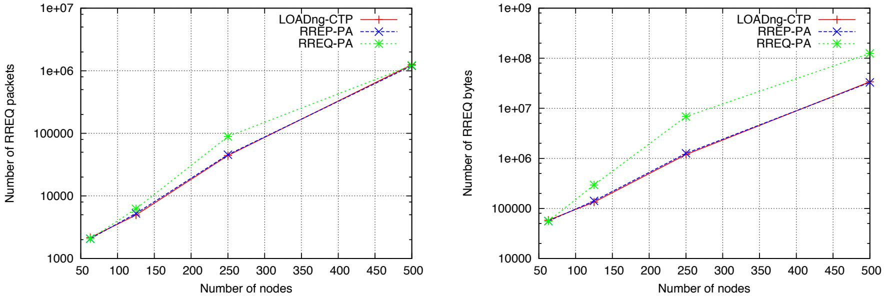
Based on the simulation results and the discussion in section 1, the characteristics of three protocol settings are summarised qualitatively in Table 1.

[2] IEEE 802.11b is, of course, not suggested as a viable interface for LLNs, but serves to illustrate general protocol behaviours.



(a) Average routing table entries (b) Standard deviation

Fig. 3. Routing table information

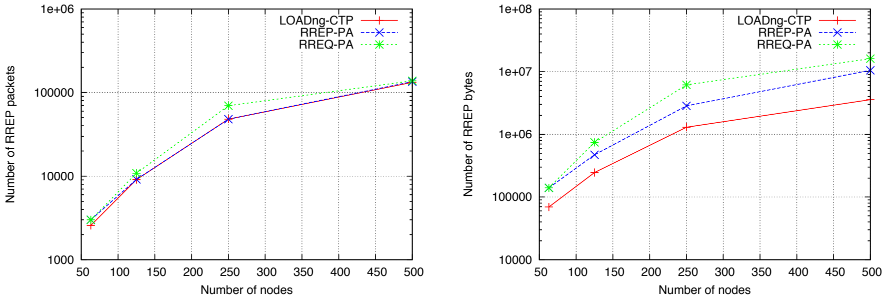


(a) Number of RREQ packets (b) Number of RREQ bytes

Fig. 4. RREQ overhead

Table 1. Characteristics of LOADng path accumulation

	LOADng-CTP	RREQ-PA	RREP-PA
Source routing	no	yes	yes
Network topology at the root	no	yes	yes
Traffic engineering	no	yes	yes
Path diversity	no	yes	no
RREQ overhead	low	high	low
RREP overhead	low	high	medium
Sensor memory requirement	high	low	low



(a) Number of RREP packets

(b) Number of RREQ bytes

Fig. 5. RREP overhead

4 Conclusion

Hop-by-hop routing requires each intermediate router to maintain routing entries – for a protocol such as LOADng, the worst case becomes four entries per active traffic flow it carries. In certain topologies, such as where there are "bottlenecks" or where all traffic originates/terminates in a single location in the network, the memory requirements that come from maintaining these entries by intermediate routers may be non-trivial to satisfy: in sensor networks, smart grid deployments, etc., devices typically measure their memory in KB (so as to keep the price, energy consumption, down), and "adding memory" to devices in such networks, once deployed, is impossible.

Furthermore certain network deployments may require either control over which paths are used for data traffic, so as to enable traffic engineering, or information as to which paths have been used, e.g., for debugging. A reactive protocol, a core optimisation of LOADng is, that it not provides any devices with a complete topology map, but only provisions relevant "next-hop" information in intermediate routers on active paths.

This paper proposes a path accumulation extensions for LOADng (LOAng-PA), which addresses these issues. Using LOADng-PA path information is accumulated when an intermediate router forwards an RREQ or RREP message, and carried within that message towards the sought destination or to the originator of the route discovery process. Consequently, path diversity, traffic engineering, and path/topology visibility can be provided – and source routing used, eliminating the requirement of (worst case) recording four entries per active traffic flow carried across an intermediate router.

This paper has explored two different path accumulation extensions to LOADng, has enumerated their advantages and inconveniences, and has provided a performance study, by way of network simulations, quantifying their respective benefits. A key result is, that LOADng-PA effectively is able to reduce the size of routing tables in intermediate routers – with the cost of slightly increasing the routing overhead, which is also quantified in this paper.

References

1. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing, Experimental RFC 3561 (July 2003)
2. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). RFC 3626, IETF (October 2003)
3. Clausen, T., Dearlove, C., Jacquet, P.: Optimized Link State Routing Protocol (OLSR) Version 2. Standard Track RFC 7181 (April 2014)
4. Kim, K., Park, S.D., Montenegro, G., Yoo, S., Kushalnagar, N.: 6LoWPAN Ad Hoc On-Demand Distance Vector Routing, Internet Draft, work in progress, draft-daniel-6lowpan-load-adhoc-routing-03 (June 2007)
5. ITU-T G.9956: Narrow-Band OFDM power line communication transceivers - Data link layer specification (November 2011)
6. Clausen, T., de Verdiere, A.C., Yi, J., Niktash, A., Igarashi, Y., Satoh, H., Herberg, U.: The lln on-demand ad hoc distance-vector routing protocol - next generation. The Internet Engineering Task Force, internet Draft, work in progress, draft-clausen-lln-loadng (October 2011)
7. Clausen, T., Camacho, A., Yi, J., de Verdiere, A.C., Igarashi, Y., Satoh, H., Morii, Y.: Experience with the loadng routing protocol for llns. The Internet Engineering Task Force, internet Draft, work in progress, draft-lavenu-lln-loadng-interoperability-report (October 2011)
8. ITU, ITU-T G.9903: Narrow-band orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks: Amendment 1 (May 2013)
9. Yi, J., Clausen, T., de Verdiere, A.C.: Collection Tree Extension of Reactive Routing Protocol for Low-Power and Lossy Networks. In: International Journal of Distributed Sensor Networks (2014)
10. Yi, J., Clausen, T., Bas, A.: Smart Route Request for On-demand Route Discovery in Constrained Environments. In: Proceedings of the IEEE International Conference on Wireless Information Technology and Systems (September 2012)
11. Bas, A., Yi, J., Clausen, T.: Expanding Ring Search for Route Discovery in LOADng Routing Protocol. In: Proceedings of the 1st International Workshop on Smart Technologies for Energy (September 2012)
12. Clausen, T., Yi, J., Bas, A., Herberg, U.: A depth first forwarding (dff) extension for the loadng routing protocol. In: ASON 2013 Sixth International Workshop on Autonomous Self-Organizing Networks (December 2013)
13. Yi, J., Clausen, T., Herberg, U.: Depth first forwarding for low power and lossy networks: Application and extension. In: Proceedings of IEEE World Forum on Internet of Things, WF-IoT (March 2014)
14. Johnson, D., Hu, Y., Maltz, D.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks for IPv4. IETF Std. RFC 4728 (February 2007)
15. Clausen, T., Yi, J., de Verdiere, A.C.: LOADng: Towards AODV Version 2. In: VTC Fall, pp. 1–5. IEEE (2012)
16. Gwalani, S., Belding-Royer, E., Perkins, C.: Aodv-pa: Aodv with path accumulation. In: IEEE International Conference on Communications, ICC 2003, vol. 1, pp. 527–531 (May 2003)
17. Clausen, T., Dearlove, C., Dean, J., Adjih, C.: Generalized MANET Packet/Message Format, Std. Track RFC 5444 (February 2009)