Collection Tree Extension of LOADng Protocol for Low-power and Lossy Networks

Jiazi Yi, Thomas Clausen
Laboratoire d'Informatique - LIX, Ecole Polytechnique, France
jiazi@jiaziyi.com, thomas@thomasclausen.org

***Abstract*—The Lightweight On-demand Ad hoc Distance-vector Routing Protocol - Next Generation (LOADng) is a reactive distance-vector protocol which is intended for use in Mobile Ad hoc Networks and Low-power and Lossy Networks to build paths between source–destination pairs. In 2013, ITU-T has ratified the recommendation G.9903 Amendment 1, which includes LOADng in a specific normative annex for routing protocol in smart grids. This paper proposes an extension to the LOADng, for efficient construction of a collection tree for data acquisition in sensor networks. The extension uses the mechanisms from LOADng, imposes minimal overhead and complexity, and enables a deployment to efficiently support "sensor-to-root" traffic, avoiding complications of uni-directional links in the collection tree. The protocol complexity, security and interoperability are examined in detail. The simulation results show that the extension can effectively improve the efficiency of data acquisition in the network.**

***Index Terms*—sensor network, collection tree, routing protocol, LOADng**

I. Introduction

"The Internet of Things" (IoT) assumes objects in our environment to be part of the Internet, communicating with users and with each other – and that these objects have communication as a commodity, rather than as their reason for existence. Communication in "The Internet of Things" is a challenge, subject to resource constraints, fragile and low-capacity links, dynamic and arbitrary topologies. Routing is among the challenges, which requires efficient protocols, able to converge rapidly even in very large networks, while exchanging limited control traffic and requiring limited memory and processing power.

One of the important applications of IoT is for data acquisition in sensor networks: a set of spatially distributed sensors that are used to monitor physical or environmental conditions, etc., and by their own impulse (either periodically or triggered by some event) transmit their data to a data concentrator (sink). These data are transmitted by way of a multi-hop network, and where the intermediary hops (routers) in that network are the sensor devices themselves. The collection of all paths from each sensor to the data concentrator form a *collection tree*. Traffic in such a collection tree is commonly described as being "sensor-to-root" traffic, or "multipoint-to-point" traffic, indicating that all traffic flows from the sensors to the data concentrator.

This paper describes a protocol for constructing such a collection tree in multi-hop sensor networks, where the protocol ensures that the resulting collection tree contains bi-directional paths between each sensor and the data concentrator. The protocol is defined as an extension to the LOADng routing protocol [1], which provides point-to-point routes between any two devices in a sensor network. Deploying both in unison permits efficient construction of both point-to-point routes and collection trees, by way of the same, simple, protocol mechanisms.

A. Background and History

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.

1) MANET Protocol Developments: The MANET working group has developed of two protocol families: reactive protocols, including AODV (Ad hoc On-demand Distance Vector Routing [2]), and proactive protocols, including OLSR (Optimized Link State Routing [3]). A distance vector protocol, AODV operates in an *on-demand* fashion, acquiring and maintaining routes only while needed for carrying data, by way of a *Route Request-Route Reply* exchange. A link state protocol, OLSR uses a periodic control messages exchanges, each router proactively maintaining a routing table with entries for all destinations in the network, which provides low delays but constant control overhead. A sizeable body of work exists, including studying the performance of these protocols in different scenarios, and justifying their complementarity [4] . For the purpose of this paper, it suffices to observe that OLSR provides low delays and predictable, constant control overhead – at expense of requiring memory in each router for maintaining complete network topology. AODV limits the memory required for routing state to that for actively used routes – at the expense of delays for the *Route Request-Route Reply* exchange to take place, and control overhead dependent on data flows.

After acquiring operational experiences, the MANET working group commenced developing successors to OLSR and

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

AODV, denoted OLSRv2 and DYMO (Dynamic MANET On-demand Routing). Whereas a relatively large and active community around OLSR thus standardized OLSRv2 [5] [6], the momentum behind DYMO withered in the MANET working group[2].

2) 6LowPAN, ROLL and related Protocol Developments: The 6LowPAN (IPv6 over Low power WPAN) working group was chartered for adapting IPv6 for operation over IEEE 802.15.4, accommodating characteristics of that MAC layer, and with a careful eye on resource constrained devices (memory, CPU, energy, ...). Part of the original charter for this working group was to develop protocols for routing in multi-hop topologies under such constrained conditions, and over this particular MAC. Two initial philosophies to such routing were explored: *mesh-under* and *route-over*. The former, mesh-under, would, as part of an adaptation layer between 802.15.4 and IP, provide Layer 2.5 multi-hop routing, *i.e.,* using link layer address for routing, and presenting an underlying mesh-routed multi-hop topology as a single IP link. The latter, route-over, would expose the underlying multi-hop topology to the IP layer, whereupon IP routing would build multi-hop connectivity.

Several proposals for routing were presented in 6LowPAN, for each of these philosophies, including LOAD (6LoWPAN Ad Hoc On-Demand Distance Vector Routing [7]). LOAD was a derivative of AODV, but adapted for link layer addresses and mesh-under routing, and with some simplifications over AODV (*e.g.,* removal of intermediate router replies and sequence numbers). However, 6LowPAN was addressing other issues regarding adapting IPv6 for IEEE 802.15.4, such as IP packet header compression, and solving the routing issues was suspended, delegated to a working group ROLL, created in 2008 for this purpose. ROLL produced a routing protocol denoted "Routing Protocol for Low-power lossy networks" (RPL) [8] in 2011 based on the idea of collection tree protocol [9].

3) Finally, Towards LOADng: RPL as a collection tree protocol has several well known issues with respect to supporting different kinds of traffic patterns, uni-direction link handling, as well as algorithmic and code complexity [10]. On the other hand, while LOAD [7] development was suspended by the 6LoWPAN working group, pending the results from ROLL and experiences with RPL, AODV derivatives live on: IEEE 802.11s [11] is based on AODV, and the ITU-T G3-PLC standard [12], published in 2011, specifies the use of [7] at the MAC layer, for providing mesh-under routing for utility (electricity) metering networks. Justifications for using an AODV derivative in preference to RPL include that the former better supports bi-directional data flows such as a request/reply of a meter reading, as well as algorithmic and code complexity reasons [10].

The emergence of LLNs thus triggered a renewed interest in AODV-derived protocols for specific scenarios, resulting in work within the IETF [1] [13] 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 (Power Line Communication) ITU-T (International Telecommunication Union – Telecommunication Standardization Sector) standard for communication in the "smart grid" [14].

[2]http://tools.ietf.org/wg/manet/minutes?item=minutes81.html

B. Statement of Purpose

There are have been a lot of protocols proposed for data acquisition in sensor networks. In [15], the authors proposed collection tree protocol that uses ETX (expected transmission count) as the routing metric to construct one-way collection tree. A CDS-Based Network Backbone for Data Collection is introduced in [16] , to balance energy consumption and prolong the router lifetime in the backbone. A Pareto based multi-optimization approach POCTP (Pareto Optimal Collection Tree Protocol) is discussed in [17] to ensure QoS such as transmission throughput, delay, loss of packets. In [18], an average transmission time (ATT) metric is applied to routing protocol, under which real-time events are transferred along the routes with the shortest transmission time expectation. Multi-channel is also used in [19] to reduce interference. Those protocols, some of them only support one-way traffic from sensor routers to one concentrator like [15], or hard to be extended for general sensor-to-sensor communications [16] [17]. Some of the protocols like [19] requires specific support from lower layers, which are hard to be applied to normal sensor equipments.

The LOADng core specification aims at finding a route between any originator-destination pairs. This kind of point-to-point traffic pattern matches the basic traffic model of the Internet. However, in the world of smart grid, another important traffic pattern, called sensor-to-root, or multipoint-to-point exists. In such kind of scenarios, there is one or more concentrators that plays as "root", and all the other routers communicate with the root. If routes from all the other routers to the root are required, it is more efficient to build a "collection tree", which is a directed graph that all edges are oriented toward and terminate at one root router.

This paper proposes an extension to LOADng, denoted LOADng Collection Tree Protocol (LOADng-CTP), for building a "collection tree" in environments, constrained in terms of computational power, memory, and in energy. An example of the design target for LOADng-CTP is the ESB (Embedded Sensor Board [20]), with a TI MSP430 low-power microcontroller, an 1MHz CPU, 2kB RAM and 60kB flash ROM. The link layers typically used in LLNs impose strict limitations on packet sizes: in IEEE 802.15.4, the maximum physical layer packet size is 127 bytes, the resulting maximum frame size at the mac-layer is 102 bytes. If link-layer security is used, this may consume up to a further 21 bytes, which leaves just 81 bytes for upper layer protocols.

The LOADng-CTP presented in this paper is thus designed to meet the following requirements:

- Effectively build a route from all sensors to the root, and the route from the root to the sensors if required.
- Uni-directional links being avoided in these routes.
- Low overhead, easy collection tree maintenance.
- Easy extension to LOADng, such that routers using only LOADng (without collection tree extension) can join the collection tree.

The remainder of this paper is organized as follows: In section II, the LOADng-CTP specification is introduced, including related message format and main operations. The protocol is further analysed in section III, from the aspect of routing complexity, security and interoperability. The simulation study is performed in section IV, in which LOADng, LOADng-CTP and RPL are compared. Section V concludes this paper .

II. LOADng-CTP Protocol Specification

LOADng Collection Tree Protocol (LOADng-CTP) is based on the operation and packet format of LOADng. Therefore, the current LOADng implementation can be easily extended to the collection tree protocol. In the following, the basic operation of LOADng is introduced briefly, followed by the single message and protocol processing required for collection tree building and maintenance.

A. LOADng Basic Operation

LOADng contains two main operations: *Route Discovery* and *Route Maintenance*.

1) Route Discovery: During *Route Discovery*, RREQ (Route Request) messages are flooded through the network. In LOADng [1], only the destination of the RREQ will reply by generating and unicasting a RREP (Route Reply) to the originator of the RREQ. All RREQ and RREP messages, generated by a LOADng router, carry a monotonically increasing sequence number, permitting both duplicate detection and detecting which of two messages contains the most "fresh" information.

2) Route Maintenance: Route Maintenance is performed when an actively used route fails. Route failure is detected by way of a data packet not being deliverable to the next hop towards the intended destination. In LOADng, the RERR is unicasted to the source of data packet. On receiving the RERR at the source of data packet, a new *Route Discovery* can be performed, in order to discover a new route to the intended destination.

Compared to AODV, LOADng has the following characteristics:

- **Modular design:** The core specification defines the simple and light-weight core functions of the protocol. LOADng is extensible, by way of a flexible packet format permitting addition of arbitrary attributes and information via new message types and/or TLV (Type-Length-Value) blocks. The LOADng protocol core is detailed in this section, with subsequent sections illustrating the use of the flexible architecture of LOADng for developing (interoperable and backwards compatible) protocol extensions.
- **Optimised Flooding:** Reducing the overhead incurred by RREQ forwarding. Jitter is employed, to reduce the probability of losses due to collisions on lower layers [21].
- **Flexible Addressing:** Address lengths from 1-16 octets are supported[3]. The only requirement is, that within a given routing domain, all addresses are of the same address length.
- **Metrics:** Different metrics are supported, to make use of link information from different layers.
- **Destination-Replies:** Intermediate LOADng Routersare explicitly prohibited from responding to RREQs, even if they may have active routes to the sought destination. All messages (RREQ or RREPs) generated by a given LOADng Router share a single unique, monotonically increasing sequence number. This also eliminates Gratuitous RREPs while ensuring loop freedom. The rationale for this simplification is reduced complexity of protocol operation and reduced message sizes – found to be without significant influence in the performance [22]. Allowing only the destination to reply to an RREQ also simplifies the task of securing the protocol, because the destination can thus sign the RREP message, and the originator could verify that it is the "real" destination that replies.
- **Reduced state:** A LOADng Router is not required to maintain a precursor list, thus when forwarding of a data packet to the recorded next hop on the path to the destination fails, an RERR is sent only to the originator of that data packet. The rationale for this simplification is an assumption that few overlapping routes are in use concurrently, and delay is not a critical issue in a given network.

B. Message for LOADng-CTP

LOADng-CTP introduces two flags to RREQ messages, carried by a so-called RREQ flag:

- RREQ COLLECTION_TREE_TRIGGER: when set, a receiving router will be triggered to discover with which of its neighbours it has bi-directional links.
- RREQ COLLECTION_TREE_BUILD: when set, a receiving router will build the route to the root.

In addition, a HELLO message [5] is used, which includes all the 1-hop neighbours of the router generating the HELLO message. The HELLO message are broadcast, and never forwarded. It permits verification of bidirectionally of links before admitting them to the collection tree.

C. Router Parameters for LOADng-CTP

LOADng-CTP uses the following parameters for protocol functioning:

[3] *i.e.,* IPv6, IPv4, 6LowPAN short addresses, Layer-2 MAC addresses etc. are all supported by LOADng

- NET_TRAVERSAL_TIME: the maximum time that a packet is expected to take when traversing from one end of the network to the other.
- RREQ_MAX_JITTER: is the maximum jitter for RREQ message transmission. Jitter is a randomly modifying timing mechanism to control traffic transmission in wireless networks to reduce the probability of transmission collisions [21].
- HELLO_MIN_JITTER: is the minimum jitter for HELLO message transmission. HELLO_MIN_JITTER must be greater than 2× RREQ_MAX_JITTER.
- HELLO_MAX_JITTER: is the maximum jitter for HELLO message transmission.
- RREP_REQUIRED: is the flag to define if an RREP message is required on receiving RREQ_BUILD message, to build routes from the root to sensors.

D. LOADng-CTP Procedures

The collection tree is, then, build by way of the following procedure — initiated by the router wishing to be the root of the collection tree:

1) Collection tree triggering (by the root): The root generates an RREQ with *COLLECTION_TREE_TRIGGER* set (henceforth, denoted RREQ_TRIGGER). Both the originator and destination of the RREQ_TRIGGER are set to the address of the root.

When an RREQ_TRIGGER is generated, an RREQ with COLLECTION_TREE_BUILD flag set (henceforth, denoted RREQ_BUILD) is scheduled to be generated in 2×NET_TRAVERSAL_TIME.

2) Bi-directional neighbour discovery: On receiving a RREQ_TRIGGER, a router:

- Records the address of the sending router (*i.e.,*the neighbour, from which it received the RREQ_TRIGGER) in its *neighbour set*, with the status *HEARD*.
- If no earlier copy of that same RREQ_TRIGGER has been previously received:
  - The RREQ_TRIGGER is retransmitted, subject to a jitter of RREQ_MAX_JITTER, to reduce the chance of collisions (except the root router).
  - Schedules generation of a HELLO message, subject to a jitter of between HELLO_MIN_JITTER and HELLO_MAX_JITTER. When the scheduled HELLO message is generated, it lists the addresses of all the 1-hop neighbours, from which it has received a RREQ_TRIGGER.

On receiving a HELLO message, a router:

- If it finds its own address listed in the HELLO message, it records the address of the sending router (*i.e.,*the neighbour, from which it received the HELLO) in its *neighbour set*, with the status *SYM* (bi-directional).
- The HELLO message is never forwarded, but discarded silently.

Thus, each router will learn with which among its neighbour routers it has a bi-directional (SYM) or uni-directional (HEARD) link.

3) Collection tree building: 2×NET_TRAVERSAL_TIME after the RREQ_TRIGGER, the root generates a RREQ_BUILD.

On receiving a RREQ_BUILD, a router:

- Verifies if the RREQ_BUILD was received from a neighbour with which it has an bi-directional (SYM) link. If not, the RREQ_BUILD is silently discarded.
- Otherwise, if no earlier copy of that same RREQ_BUILD has been previously received, or the RREQ_BUILD indicates a short path to the root:
  - a new routing entry is inserted into the routing table, with
    - *next_hop* = previous hop of the RREQ_BUILD;
    - *destination* = root
  - The RREQ_BUILD is retransmitted, again subject to a jitter of RREQ_JITTER.

Thus, each router will record a route to the root, and this route will contain only bi-directional links. The collection tree is built, enabling upward traffic. Figure 1 illustrates the RREQ_BUILD processing.

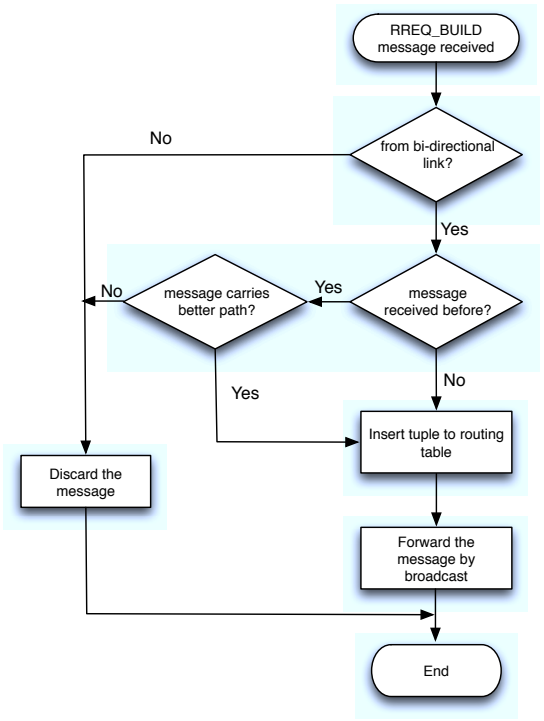


Figure 1: LOADng-CTP RREQ_BUILD message processing

4) Root-to-sensor Path Building: By exchanging of RREQ_TRIGGER and RREQ_BUILD messages, all the sensors in the network obtained a path using only bi-directional links to the root. This is sufficient for applications like environment monitoring, automatic meter reading, etc. However, in some applications, such as firmware update or remote control, the root needs to send messages to sensors in the network. The paths from root to sensors are thus desired.

The sensors that require root-to-sensor traffic must have their RREP_REQUIRED flag set to true. On receiving

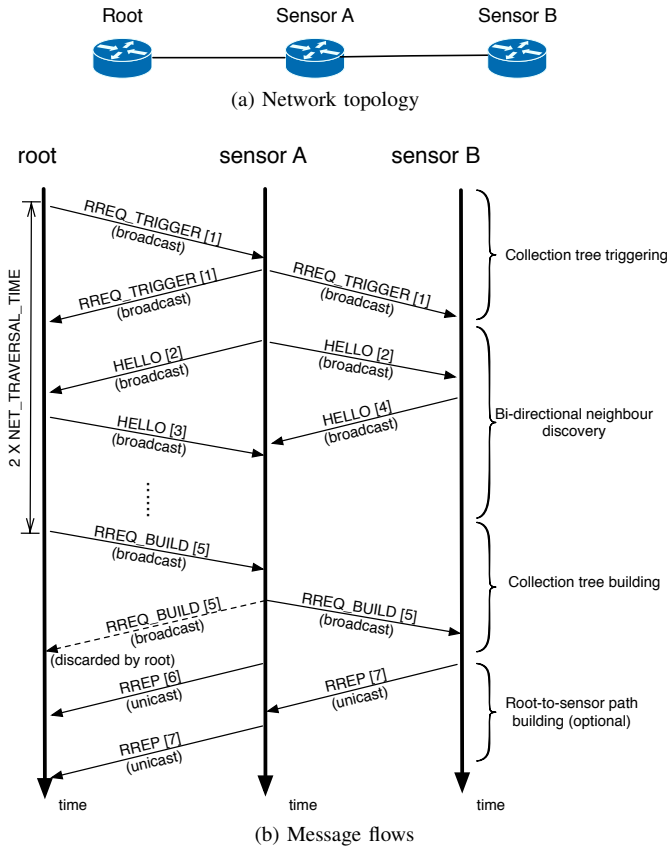
the RREQ_BUILD message, all the sensor routers with RREP_REQUIRED flag set must initiate an RREP message with content of:

- *RREP_originator* = address of the sensor router;
- *RREP_destination* = address of the root.

The RREP is thus unicast to the root, subjecting to jitter RREP_JITTER. On receiving the RREP message, a routing tuple is created in the routing table with:

- *next_hop* = previous hop of the RREP;
- *destination* = address of the RREP originator (*RREP_originator*).

Figure 2 depicts an example of root-sensor message exchange sequences by illustrating the four steps of LOADng-CTP protocol (collection tree triggering, bi-directional neighbour discovery, collection tree build and root-to-sensor path building). In the example, the *Root* router builds a collections tree connecting sensor routers *A* and *B*, with the topology shown in figure 2a. The message exchange is shown in figure 2b. The pseudo-sequence number in the brackets is used just for distinguishing different messages in this figure. In a real protocol implementation, sequence numbers are generated independently at each router.



(a) Network topology

(b) Message flows

Figure 2: Message exchange of LOADng-CTP between root and sensors

E. Collection Tree Maintenance

Based on the operation introduced in section II-D, a collection tree is built to enable data traffic transmission between the root router and all the other sensors. However, route failure could still happen, due to the "lossy" nature of sensor networks or topology changes, such as:

- Lost of control message during the collection tree building process;
- Routing entries expire because of not updated timely;
- Participation of new sensors;
- Sensors quite the network because of movement or battery drain.

LOADng-CTP supports per-path maintenance when a path failure is detected, without rebuilding the whole collection tree. A new route discovery is initiated according to usual procedures of route discovery, if:

- The data packet to be forwarded can not find a routing tuple to the desired destination in the routing table, OR
- the link to the "next hop" indicated by the routing table is detected broken.

To avoid that RREQ being broadcast through the whole network, and take benefits from that "most of other neighbour routers might have an available route to the root", a *Smart Route Request* scheme can be employed: if an intermediate router, receiving the RREQ, does not have an available route to the destination, the RREQ is forwarded as normal. If the intermediate router has a route to the root, that intermediate router will unicast the RREQ to the destination according to the routing table.

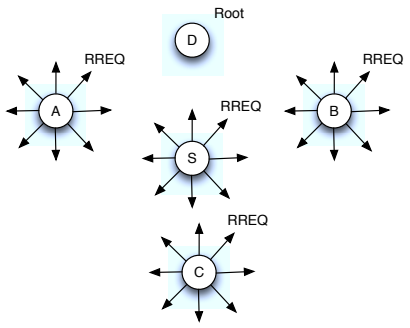
Figure 3 gives an example of path maintenance in collection tree. Router *D* is the root, the link between *S–D* is detected broken. Router *A* and *B* have already direct path to *D*, and *C* has also a routing tuple to *D* (by going through *S*). Figure 3a depicts the route discovery initiated by *S* according to LOADng basic operation. Because only the destination is allowed to reply to the RREQ message, sensor router *A, B, C* have to rebroadcast the RREQ message, even they have already routing tuples to *D*. This renders a network-wide flooding: for a network with n routers, n RREQ message retransmissions are required.

With smart route request, as shown in figure 3b, router *A, B, C* will unicast the RREQ to root *D* according to their routing tables, and *D* can choose the best path to send RREP message. By doing so, the RREQ dissemination is limited locally (4 retransmission in this example), and the routing overhead can be greatly reduced.

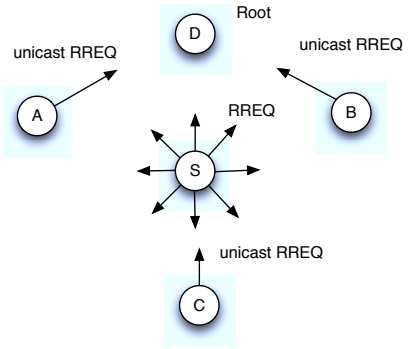
When a link on an active route to a destination is detected as broken (by way of inability to forward a data packet towards that destination), an RERR (route error) message is unicast to the source of the undeliverable data packet. Both this intermediate router and the source router need to initiate a new route discovery procedure.

III. LOADng-CTP Protocol Analyses

This section analyses the main features of the LOADng-CTP, including protocol complexity, security considerations,



(a) Route discovery without smart route request



(b) Route discovery with smart route request

Figure 3: An example of route maintenance. Router *D* is the root. The link between *S–D* is detected broken. Sensor router *A, B, C* still have routing tuple to *D*.

and its interoperability with LOADng protocol.

A. Protocol Complexity

Unlike link-state routing protocols such as OSPF [23] or OLSR[6], which require keeping a network topology locally and run the Dijkstra algorithm, LOADng and LOADng-CTP concerns only the basic additive operation when calculating link metrics. Therefore, the computational complexity is negligible. An very important concern of routing protocol for sensor networks is its routing overhead: the message required to maintain the routing table.

For simplicity, a balanced tree model is considered: there is a single root in the tree, with total height of H. The height of root is 0, and the leaf nodes are with height $H-1$. Every node in the tree (except the leaf nodes) has C children ($C > 1$). Figure 4 gives an example of balanced tree with $C = 2$.

The number of nodes at height $h(0 \leq h \leq H-1)$ is $n_h = C^h$. The total number of nodes in the tree is

$$N = 1 + C + C^2 + ... + C^{(H-1)} = \frac{1 - C^H}{1 - C},\ (C > 1) \quad (1)$$

In LOADng-CTP, the message required for collection tree building is the sum of RREQ_TRIGGER, HELLO and RREQ_BUILD:

$$RREQ = 3N \quad (2)$$

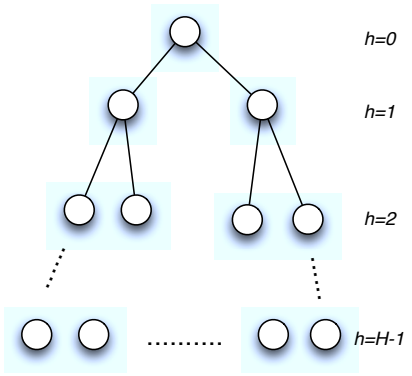


Figure 4: An example of balanced tree. Every parent has 2 children ($C = 2$)

If root-to-sensor paths are required, every sensor also has to unicast an RREP message to the root.

The number of RREP messages forwarded by all the routers at height h is

$$RREP_h = C^h \sum_{i=0}^{H-h-1} C^i = C^h \frac{1 - C^{H-h}}{1 - C} \quad (3)$$

The total number of RREP can thus be given by

$$RREP_{All} = \sum_{h=1}^{H-1} RREP_h = \sum_{h=1}^{H-1} \frac{C^h}{1 - C} - \sum_{h=1}^{H-1} \frac{C^H}{1 - C} \quad (4)$$

Considering equation 1, the total number of RREP forwarding is

$$\begin{aligned} RREP_{All} &= \frac{1}{1 - C} \frac{C - C^H}{1 - C} - \frac{(H-1)C^H}{1 - C} \\ &= NH - N + \frac{N - H}{1 - C} \end{aligned} \quad (5)$$

Considering $H = \lfloor log_C N \rfloor$, the total number of RREP messages thus scales with $O(NlogN)$.

For the basic LOADng protocol, by which only point-to-point route build is supported, the number of RREQ message forwarding required to build path from all the sensors to the root is

$$RREQ = N^2 \quad (6)$$

The RREP message is always needed in LOADng basic operation, which is the same with equation (5).

Based on equation (2), (5) and (6), it can be concluded that LOADng-CTP reduced routing overhead from $O(N^2)$ to $O(N)$ compared to basic LOADng mechanism, if only sensor-to-root paths are needed, or $O(NlogN)$, if root-to-sensor paths are also required.

B. Security Considerations

1) Protocol vulnerability: The collection tree building process relies on strictly ordered message sequences: RREQ_TRIGGER message for triggering the building process, then HELLO message for bi-direction neighbour check, and RREQ_BUILD message for collection tree build in the end. The message emission is controlled by router parameters like NET_TRAVERSAL_TIME, RREQ_JITTER, and HELLO_JITTER.

The receiving order can be expected if those parameters are set correctly – however, in real implementations, there might exist mis-configured routers, or even compromised routers that emit messages out of order. For example, if a router sends a HELLO message before it receives all the RREQ_TRIGGER messages from its neighbours, or an RREQ_BUILD message is received before the HELLO message exchange finished, the router cannot identify its bi-directional neighbours correctly – thus is not able to join the collection tree as expected.

In addition to message mis-ordering, LOADng-CTP is also prone to attacks like block-hole or spoofing attacks [24], [25]. Malicious control traffic can have severe impact on the network stability.

2) Security framework: One of the main objectives when specifying LOADng was to provide a modular architecture with a core module that is easily extensible. The rationale for this decision was that rarely "one-size-fits-all" in the area of constrained networks. This is particularly true for security extensions: some networks may not require any level of Layer 3 security, e.g., because physical access is limited or lower layer protection is sufficient. Other networks require integrity protection with a lightweight cipher suite due to limited precessing power and memory of routers. In some cases, security requirements are tighter and confidentiality as well as strong cryptographic ciphers are required.

The IETF has standardized a security framework for protocols using the message and packet format defined in [26][4], which is used by LOADng-CTP. [27] specifies a syntactical representation of security-related information in TLVs for use with [26] addresses, messages, and packets. That specification does not represent a stand-alone protocol, but is intended for use by MANET routing protocols, or security extensions thereof, such as LOADng-CTP.

Figure 5 depicts the architecture of a module for LOADng-CTP that provides integrity and non-repudiation for LOADng, using the framework specified in [27].

Incoming RFC5444 packets are first parsed by the RFC5444 parser that demultiplexes messages and sends them to the protocol "owning" the message type. As each RFC5444 packet may contain multiple messages that are used by different protocols on a router, the message type is used to demultiplex and send the message to the appropriate protocol instance. A message intended for LOADng-CTP will then be forwarded

[4]Note that this framework is currently being revised in a succeeding document that will obsolete RFC6622 once approved: http://tools.IETF.org/html/draft-IETF-manet-rfc6622-bis

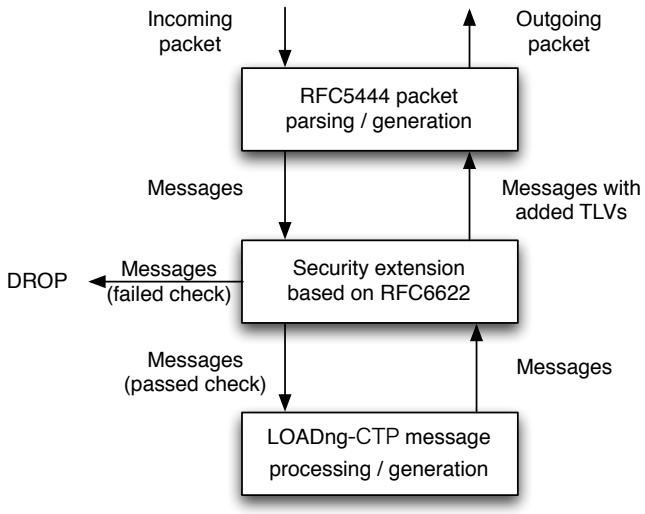


Figure 5: Relationship with RFC5444, RFC6622 and LOADng-CTP

to the security extension module that verifies the signature contained in a signature TLV inside the message. As the TLV contains additional information, such as the hash function (e.g., SHA-256, Secure Hash Algorithm) and the cryptographic function (e.g., AES, Advanced Encryption Standard), the module can choose the correct key and verify the integrity protection. If the message signature is correct, the message is handed over to the LOADng-CTP module, otherwise it is rejected. Similarly, outgoing messages from LOADng-CTP are handed over to the security module, which in turn adds the TLV containing the digital signature of the message. Then the message is handed over to the RFC5444 module that multiplexes it into a packet.

During the message signature generation as well as verification process, [27] takes special consideration for mutable fields, such as hop count and hop limit. In addition to hop count and limit, the route metric contained in a metric TLV is also updated along the path of a message, and can therefore not be protected by a digital signature. LOADng-CTP lists these mutable fields explicitly. While this is a security problem that needs to be addressed in addition to a pure message signature (and is not discussed in this paper), based on the message format of LOADng-CTP messages, at least the calculation of signature is easy. This is because the message size does not change as no field is added or removed during the forwarding process of a message through the network (and therefore no other fields, such as message size or TLV block size, need to be recalculated). The metric can simply be replaced by a sequence of zeros before calculating the signature, and is then restored afterwards.

In addition to message integrity, packets may also be digitally signed. As packets are used hop-by-hop, *i.e.,* are never forwarded, this is useful to authenticate the previous hop along the path of a message. Otherwise, a router not having

any credentials may, for example, simply forward a correctly signed RREP message from one adjacent router to another and increase the hop count. As the hop count is excluded from the signature calculation, the message integrity would still be valid. Packet signatures mitigate this problem at the expense of increased overhead on the channel. Note also that it is difficult to detect simple forwarding of a frame without modifying the content, also known as "wormhole attack".

C. Interoperability Considerations

As sensor networks and low power and lossy networks are generally decentralized system, devices would possibly work in a heterogeneous environment: there are might be old devices with basic functions, and newly jointed devices with extensions in the same routing domain. This requires interoperability between routers using LOADng-CTP and LOADng routers without collection tree extension (denoted LOADng-core router).

A LOADng-core router will forward RREQ_TRIGGER and RREQ_BUILD message as normal RREQ messages, so it will not affect the collection tree building process of other routers in the network. But because LOADng-core routers cannot generate HELLO messages themselves, and are not able to be verified as bi-directional neighbour. Therefore, LOADng routers will not join the collection tree during the collection tree building process described in this section. However, these routers can participate the collection tree by initiating a new RREQ message to the root, and thus join the collection tree as "leaf nodes" (*i.e.,* nodes without children), as shown in Figure 6.

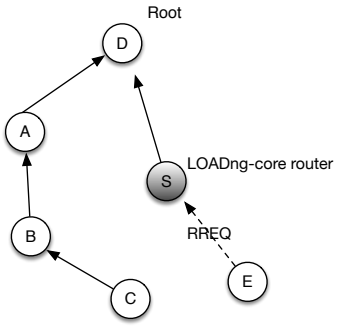


Figure 6: An example of interoperability between LOADng-CTP (white nodes) and LOADng-core routers (grey nodes).

During the collection tree building process, LOADng-core routers will not be able to function as parents of other routers. As depicted in Figure 6, router *C* will choose *B* as parent, even *S* probably provides a shorter path to the root. If the LOADng-core router is on the only path to the root – for example, router *E* has to go through *S* to reach the root, a new RREQ will be initiated to the root.

The existence of LOADng-core routers will possibly increase the routing overhead in the network by initiating more route discoveries. But with the smart RREQ introduced in section II-E, the RREQ dissemination can be kept locally, thus without introducing much influence in the networks.

IV. Simulation and Performance Analyses

A. Simulation Settings

In order to understand the performance impact of the collection tree extension to LOADng, this section presents a set of ns2 simulations, comparing LOADng, LOADng-CTP and RPL, with the parameters of the trickle timer in RPL is set according to [8]. Simulations were made with varying numbers of routers from 63 to 500 and placed statically randomly in a square field. The networks have consistent density of nodes, *i.e.,*, the simulation field grows as the number of routers increases: 1100m×1100m for 63 nodes, 1580m×1580m for 125 nodes, 2230m×2230m for 250 nodes, 3160m×3160m for 500 nodes. This simulates smart grid in suburban areas. As the size of the network grows, the scalability of the protocol can be tested.

The network is subject to sensor-to-root traffic, like periodic meter reading: all routers generate traffic, for which the destination always is a single, fixed router in the network. Each data source transmits a 512-byte data packet every 5 seconds, in bursts lasting for 80 seconds each, for a total simulation time of 100s.

For the purpose of this study, router mobility was not considered. Simulations were conducted using the TwoRayGround propagation model and the IEEE 802.11 MAC. Although there are various low-layer technologies more commonly (and, perhaps, more viably) used for LLNs (power line communication, 802.15.4, low-power wifi, bluetooth low energy, etc.), 802.11 provides basic distributed mechanisms for channel access, such as DCF (Distributed Coordination Function), CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), etc. Therefore, general behaviour of a protocol can be inferred from simulations using 802.11.

In the simulations, three types of routing protocols are compared:

- LOADng core specification [1], referred as LOADng in the following of the section. The routes are build reactively when there are data packets need to be send.
- LOADng with collection tree extension, referred as LOADng-CTP. The collection tree is triggered and build before the sending of data packets.
- RPL with trickle timer, referred as RPL. The parameters of trickle timer is set according to [8].

B. Simulation Results

Figure 7 depicts the delivery ratio of three protocols. Both LOADng-CTP and RPL obtain delivery ratios close to 100%, regardless of number of nodes. LOADng, initiating route discovery for every router (network-wide broadcast), incurs a high number of collisions on the MAC layer (shown in figure 8), and thus a lower data delivery ratio, especially in larger scenarios.

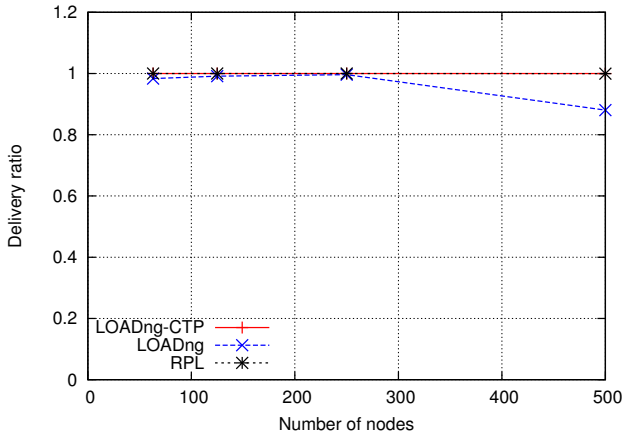


Figure 7: Packet delivery ratio

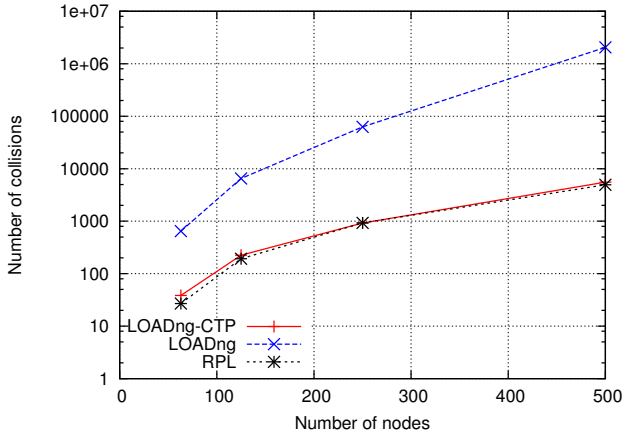


Figure 8: Number of MAC layer collisions

Figure 9 illustrates the average end-to-end delay. LOADng has longer delay mainly because the route discovery is performed reactively, *i.e.,*the data packets have to wait the finish of route discovery before being send out. LOADng-CTP and RPL have routes a-priori available, thus exhibit identical delays.

For the sensor networks, the routing overhead is also a crucial consideration. Figure 10 and Figure 11 shows the number of overhead packets per router and average overhead of network (bytes/second) respectively, which the networks are needed to converge to a stable state, *i.e.,*, every router has a route to the root.

The overhead packets of LOADng-CTP and RPL grow linearly with RPL sending twice as many packets as LOADng-CTP, and RPL sending 10 times more bytes/s as compared to LOADng-CTP, due to the RPL control packets (mainly, the DIOs) being bigger [10]: a DIO packet[5] takes up to 40 octets in these scenarios, whereas a LOADng-CTP RREQ and RREP packet typically is 10 octets. The overhead of LOADng grows exponentially as the number of nodes increases, up to

[5]Base header of 24 octets, plus other options and addresses.

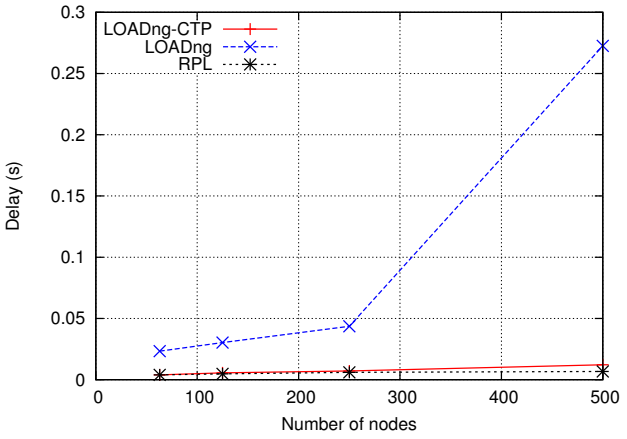


Figure 9: Average end-to-end delay

700,000 packets for scenarios of 500 nodes (not drawn in the figure). The peer-to-peer based basic LOADng mechanism is not optimized for sensor-to-root traffic.

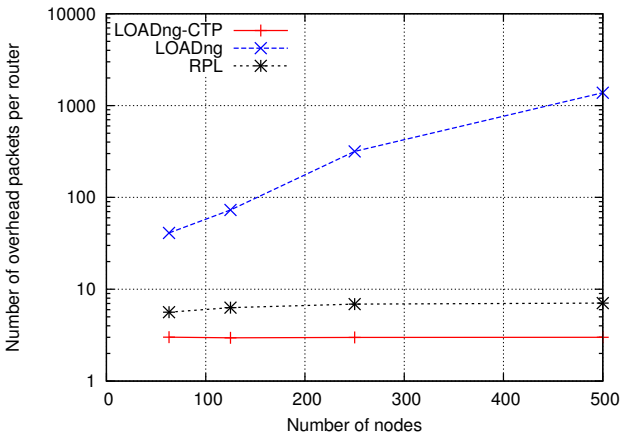


Figure 10: Number of overhead packets transmitted by each router

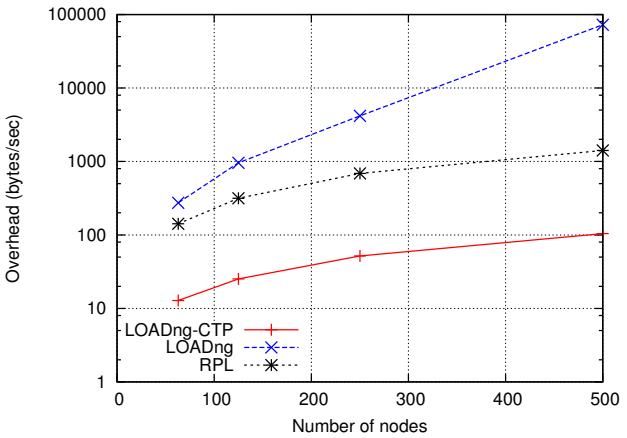


Figure 11: Overhead bytes per second in the whole network

V. Conclusion

This paper has presented a simple extension, LOADng-CTP, to the reactive LOADng routing protocol, permitting efficient and on-demand construction of collection trees for supporting sensor-to-root traffic types. LOADng-CTP permits finding paths between a root router and all the other sensor routers in the network using bi-directional links. The protocol supports per-path route maintenance without rebuilding the whole collection tree. Another key aspect of LOADng-CTP is, that any router can at any time determine that it needs to act as a root for sensor-to-root traffic, and spawn a collection tree construction; this, without requiring that said router be specifically provisioned for this purpose (no extra state, processing power, required).

The main features of LOADng-CTP are analysed. The routing overhead is reduced to $O(N)$ for collection tree building, compared to $O(N^2)$ of LOADng core specification (N is the number of routers in the network). An extensible security framework is proposed to protect the integrity of routing message exchange. The interoperability between collection tree extension and LOADng core specification is considered. The LOADng routers without collection tree extension can also join the collection tree by initiating a route discovery.

The performance of this extension has been studied, revealing delays and data delivery rations, comparable with RPL, are obtained while at the same time yielding considerably lower control traffic overheads. Compared to basic LOADng, the performance of the LOADng-CTP extension yields better performance: lower overhead, higher data delivery ratios, and lower delays.

Acknowledgements

The authors would like to gratefully acknowledge Ulrich Herberg, Axel Colin de Verdiere, Yuichi Igarashi, Christos Verikoukis (assigned editor) and anonymous reviewers of the manuscript for valuable comments, reviews and technical discussions.

References

[1] T. Clausen, A. C. de Verdiere, J. Yi, A. Niktash, Y. Igarashi, H. Satoh, and U. Herberg, "The lln on-demand ad hoc distance-vector routing protocol - next generation," The Internet Engineering Task Force, October 2013, internet Draft, work in progress, draft-clausen-lln-loadng.

[2] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," Experimental RFC 3561, July 2003.

[3] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," Experimental RFC 3626, October 2003.

[4] L. V. T. Clausen, P. Jacquet, "Comparative study of routing protocols for mobile ad-hoc networks." Proceedings of the IFIP MedHocNet, September, Sardinia, Italy, 2002.

[5] T. Clausen, C. Dearlove, and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)," RFC 6130, IETF, April 2010.

[6] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg, "The Optimized Link State Routing Protocol version 2," Internet Draft, draft-ietf-manet-olsrv2-19, work in progress, March 2013.

[7] K. Kim, S. D. Park, G. Montenegro, S. Yoo, and N. Kushalnagar, "6LoWPAN Ad Hoc On-Demand Distance Vector Routing," June 2007, Internet Draft, work in progress, draft-daniel-6lowpan-load-adhoc-routing-03.

[8] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks," March 2011, IETF RFC6550.

[9] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "The Collection Tree Protocol (CTP)," TinyOS, Tech. Rep., 2009. [Online]. Available: http://sing.stanford.edu/pubs/sing-09-01.pdf

[10] T. Clausen, A. C. de Verdiere, J. Yi, U. Herberg, and Y. Igarashi, "Experiences with RPL: IPv6 Routing Protocol for Low power and Lossy Networks," The Internet Engineering Task Force, February 2013, internet Draft, work in progress, draft-clausen-lln-rpl-experiences.

[11] G. Hiertz, S. Max, R. Zhao, D. Denteneer, and L. Berlemann, "Principles of ieee 802.11s," in *Proceedings of WiMAN in conjunction with the 16th ICCCN*, Honolulu, Hawaii, USA, Aug 2007, p. 6.

[12] "ITU-T G.9956: Narrow-Band OFDM power line communication transceivers - Data link layer specification," November 2011.

[13] T. Clausen, A. Camacho, J. Yi, A. C. de Verdiere, Y. Igarashi, H. Satoh, and Y. Morii, "Interoperability report for the lightweight on-demand ad hoc distance-vector routing protocol - next generation (loadng)," The Internet Engineering Task Force, December 2012, internet Draft, work in progress, draft-lavenu-lln-loadng-interoperability-report.

[14] ITU, "ITU-T G.9903: Narrow-band orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks: Amendment 1," May 2013.

[15] J.-J. Lei, T. Park, and G.-I. Kwon, "A reliable data collection protocol based on erasure-resilient code in asymmetric wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[16] X. Kui, Y. Sheng, H. Du, and J. Liang, "Constructing a cds-based network backbone for data collection in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[17] W. Yi-Zhi, Q. Dong-Ping, and H. Han-guang, "Pareto optimal collection tree protocol for industrial monitoring wsns," in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, 2011, pp. 508–512.

[18] Y. Tang, C. Bu, and A. Fan, "A transmission time based routing protocol for clustered collection tree wireless sensor networks," in *Computational and Information Sciences (ICCIS), 2010 International Conference on*, 2010, pp. 21–24.

[19] C. Buengbon, C. Tanwongvarl, and S. Chantaraskul, "Multi-channel collection tree protocol for wireless sensor networks," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, 2013, pp. 1–5.

[20] "The ESB Embedded Sensor Board," http://contiki.sourceforge.net/docs/2.6/a01781.html, 2013.

[21] T. Clausen, C. Dearlove, and B. Adamson, "Jitter Considerations in MANETs," IETF Inf. RFC 5148, February 2008.

[22] T. Clausen, J. Yi, and A. C. de Verdiere, "LOADng: Towards AODV Version 2," in *VTC Fall*. IEEE, 2012, pp. 1–5.

[23] J. Moy, "OSPF Version 2," RFC 2328, IETF, April 1998.

[24] W. Wang, Y. Lu, and B. K. Bhargava, "On vulnerability and protection of ad hoc on-demand distance vector protocol," in *Telecommunications, 2003. ICT 2003. 10th International Conference on*, vol. 1. IEEE, 2003, pp. 375–382.

[25] P. Ning and K. Sun, "How to misuse AODV: a case study of insider attacks against mobile ad-hoc routing protocols," *Ad Hoc Networks*, vol. 3, no. 6, pp. 795–819, 2005.

[26] T. Clausen, C. Dearlove, J. Dean, and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format," RFC 5444, IETF, Feburary 2009.

[27] U. Herberg and T. Clausen, "Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)," RFC 6622, IETF, May 2012.