



UNIVERSITÀ
DEGLI STUDI
FIRENZE

FLORE

Repository istituzionale dell'Università degli Studi di Firenze

Is 6LoWPAN-ND necessary? (Spoiler alert: Yes)

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

Is 6LoWPAN-ND necessary? (Spoiler alert: Yes) / Rashid, Adnan; Pecorella, Tommaso. - In: COMPUTER NETWORKS. - ISSN 1389-1286. - ELETTRONICO. - 250:(2024), pp. 1-10. [10.1016/j.comnet.2024.110535]

Availability:

The webpage <https://hdl.handle.net/2158/1362792> of the repository was last updated on 2024-06-14T10:36:07Z

Published version:

DOI: 10.1016/j.comnet.2024.110535

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

Publisher copyright claim:

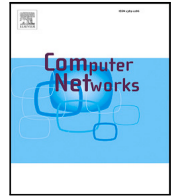
Conformità alle politiche dell'editore / Compliance to publisher's policies

Questa versione della pubblicazione è conforme a quanto richiesto dalle politiche dell'editore in materia di copyright.

This version of the publication conforms to the publisher's copyright policies.

La data sopra indicata si riferisce all'ultimo aggiornamento della scheda del Repository FloRe - The above-mentioned date refers to the last update of the record in the Institutional Repository FloRe

(Article begins on next page)



Is 6LoWPAN-ND necessary? (Spoiler alert: Yes)

Adnan Rashid^a, Tommaso Pecorella^{b,*}

^a Dpt of Electrical and Information Engineering, Politecnico di Bari, Via Edoardo Orabona 4, Bari, 70125, Italy

^b Dpt. of Information Engineering, Università di Firenze, via di S.Marta 3, Firenze, 50139, Italy

ARTICLE INFO

Keywords:

Wireless sensor networks
6LoWPAN
Neighbor discovery protocol
IPv6
Low power and lossy network
Meshunder
6LoWPAN-NDP
Internet of things
Neighbor discover optimization
Simulation

ABSTRACT

Low-Power and Lossy Networks (LLNs) are based on constrained devices. Energy conservation is one of the main constraints, and the traditional IPv6 Neighbor Discovery Protocol (IPv6-NDP) was neither designed nor suitable to cope with it. This inefficiency arises from non-transitive wireless links and heavy multicast transmission, sometimes rendering it impractical in LLNs. Substantial work has been done by the Internet Engineering Task Force (IETF) to optimize the IPv6-ND protocol, known as IPv6 over Low power Wireless Personal Area Network - Neighbor Discovery Protocol (6LoWPAN-NDP). Despite these improvements, full implementation is yet to be achieved in commercial, open-source, or proprietary sectors. In this article, we debate both Neighbor Discovery Protocols (NDPs), examining various aspects. We implemented 6LoWPAN-NDP in a well-known ns3 simulator. We discuss the complexity of 6LoWPAN-NDP and see why open-source, commercial, or proprietary sectors have not widely adopted it. We present how both protocols function optimally in meshunder and non-meshunder scenarios. We present results and analysis of both NDPs control messages' behavior. At the same time, data traffic is turned on and off, and we demonstrate the operational behavior of Link-local Unicast Address (LUA) and Global Unicast Address (GUA) in meshunder and non-meshunder scenarios. The presented implementation can be helpful in enabling large-scale simulations and evaluating scenario-specific protocol parameters, along with protocol extensions.

1. Introduction

According to market analysis forecasts, Internet of Thing (IoT) systems are poised for exponential growth, with projections reaching billions of devices in the next ten years. In contrast to computer systems or smartphones, IoT devices are usually small and possess limited resources, low cost, and prolonged operative life. However, as counter-intuitive as it might seem, the long operative life poses a problem. Wrong or sub-optimal design choices cannot simply be phased out by “natural obsolescence”, and software patches are also problematic due to the vendor's limited support and upgrade capabilities. Hence, carefully evaluating which protocols are optimal in specific scenarios is essential. Failing to conduct this type of analysis can result in systems that “work” but exhibit issues related to scalability, uneven resource consumption, etc.

IoT systems can be broadly categorized into devices equipped with an IPv6 stack and devices requiring a Gateway to connect to the Internet. In this paper, we will focus on the former. Among IoT devices using the IPv6 stack “natively”, further classification can be made based on the type of network they can utilize. Devices capable of using “IPv6-friendly” networks, such as WiFi or 5G/6G, and devices using so-called LLNs, such as Bluetooth, IEEE 802.15.4, Long Range Wide Area Network (LoRaWAN), etc.

An LLN exhibits several evident and subtle differences from a standard network. The most relevant (for the present discussion) include: (1) short frames - typically unable to efficiently carry IPv6 packets, (2) potential lack of efficient support for multicast or broadcast frames (or no support at all), (3) potential lack of uniqueness of MAC level addresses in the LLN, etc.

In order to overcome the limitations imposed by LLNs, IETF Working Groups (WGs) (in particular the IPv6 over Low power Wireless Personal Area Network (6LoWPAN) and IPv6 over Networks of Resource-constrained Nodes (6lo)) have devised several protocols. The most well-known is 6LoWPAN [1,2], a *shim* layer that hides the LLN Medium Access Control (MAC) layer from the Internet Protocol version 6 (IPv6) layer offering, for example, header and packet compression, fragmentation, and reassembly. The 6LoWPAN is a necessary choice for LLNs because the L2 protocols used by LLNs do not meet the requirements mandated by IPv6, such as minimum Maximum Transmission Unit (MTU), which for IPv6 is 1280 octets.

As shown in Fig. 1, the adaptation protocols defined by IETF are generic, meaning that they can be applied to any network protocol. Without loss of generality, in this paper we will use 802.15.4, but the analysis can be easily extended to other protocols.

* Corresponding author.

E-mail address: tommaso.pecorella@unifi.it (T. Pecorella).

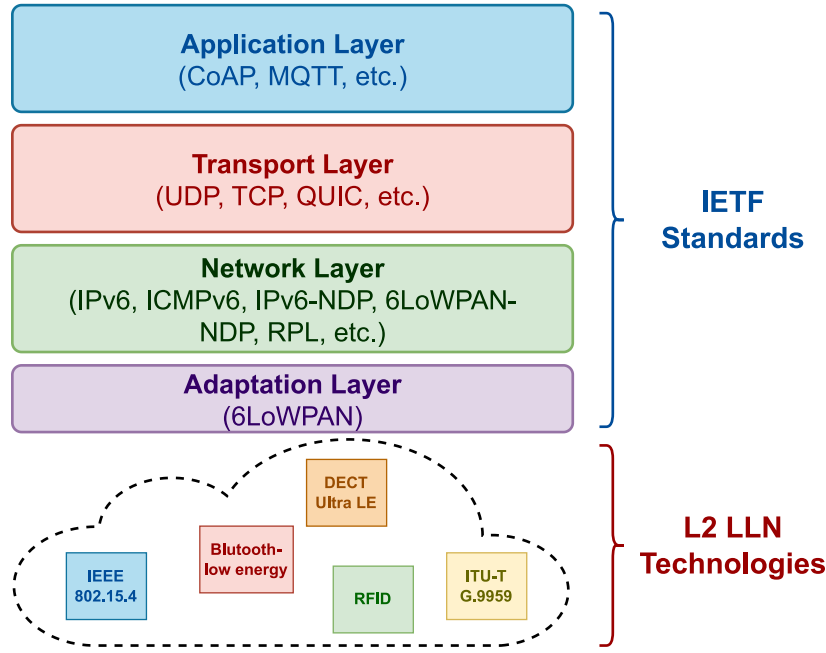


Fig. 1. IETF 6Lo stack.

Moreover, the IETF developed a substitution for IPv6-NDP [3] specifically oriented toward 6LoWPANs known as 6LoWPAN-NDP [4, 5]. The rationale is that the conventional IPv6-NDP relies heavily on multicast, making it inefficient and sometimes impractical to use multicast in LLNs. Furthermore, IPv6-NDP was not designed for non-transitive wireless links.

However, while the use of 6LoWPAN is evident, if 6LoWPAN-NDP is not utilized, the stack appears to work—there are no evident errors in the communications. This article contributes in the following ways:

- We discuss in-depth and explore various aspects of both NDPs in terms of reliability, robustness, and implicit and explicit reachability behaviors.
- We delve into the complexity of 6LoWPAN-NDP and examine why open-source, commercial, or proprietary sectors have not widely adopted it.
- We present results and analysis to assess the performance of both protocols in various configurations. The analysis is based on the behavior of NDPs control messages with data traffic enabled and disabled.
- We present the operational behavior of LUA and GUA in both meshunder and non-meshunder scenarios.
- We suggest new features to enhance the 6LoWPAN-NDP protocol, proposing optimizations to the IETF.

The rest of the paper is organized as follows: in Section 2, we discuss the 6LoWPAN-NDP standard in detail, its importance, and a survey of its implementation status. The survey of 6LoWPAN-NDP implementation status is based on open-source, proprietary, and commercial tools (e.g., simulators and operating systems). The adoption status of 6LoWPAN-NDP in the context of interoperability is discussed in Section 3. We examine the complexity of 6LoWPAN-NDP in terms of Neighbor Cache Entry (NCE) states and their optimal variations, LUA and GUA *Addresses Registration* process, and *Neighbor Unreachability Detection (NUD)* in Section 4. The performances of both NDPs are analyzed in Section 5, and Section 6 concludes this research, presenting our further optimization suggestions for the IETF.

2. 6LoWPAN-NDP standard

The communication between nodes requires that each node knows the pairing between the IP address and the link-layer address of its neighbors. The primary objective of an NDP is to build and maintain such a table, which is often dynamic because a node can change its IP or link-layer address dynamically. A secondary use of NDP is to prevent address duplication, i.e., to enable the *Duplicate Address Detection (DAD)*.

2.1. IPv6-NDP

In IPv6, NDP and DAD utilize *Internet Control Message Protocol version 6 (ICMPv6)* packets sent to multicast addresses to probe for neighbor addresses. Moreover, both protocols implicitly assume that the probability of a message not being received is negligible. This assumption, as we will see in the following, is fundamental in the context of LLNs.

In IPv6, the normal operation of a node is as follows (assuming self-assigned addresses):

- A node searches for a router through multicast Router Solicitation (RS),
- The router replies with a Router Advertisement (RA), unicast or multicast,
- The node creates its addresses, both link-local and global,
- The node performs two DADs processes, and if successful, it uses the addresses.

When a node needs to communicate with a neighbor, it utilizes the NDP to probe for the neighbor's link-local address. In an LLN, this mechanism creates several issues. Even though RS or RA messages usually cover only a single hop in LLN-specific routing protocols, the DAD operation should guarantee the address uniqueness, but in different ways for link-local and global addresses. Link-local addresses must be unique *between nodes*, meaning their uniqueness must be guaranteed only on the link (1-hop) and between effectively communicating nodes. Hence, the DAD operation for link-local addresses might be limited to

1-hop. On the contrary, global addresses must be unique in the LLN. Therefore, the DAD operation must be propagated to the entire LLN. This represents a significant issue for LLNs, as propagating a message to the whole network is energy-consuming and unreliable.

2.2. 6LoWPAN-NDP

The *Neighbor Discovery Optimization* for 6LoWPAN's, referred to as 6LoWPAN-NDP, is based on two IETF standards: RFC 6775 [4] and 8505 [5].¹ The first difference concerning the 'normal' IPv6 operations is that 6LoWPAN-NDP enforces checking if the link is bidirectional and each node has only one neighbor with a given link-layer address, preventing the case of a node having two neighbors with the same link-layer address. This case is considered unimportant for 'normal' IPv6 networks but is relevant in LLNs. In order to enforce this, the protocol mandates a registration phase, where each node registers its addresses (IP and link-local) to the neighbors.

When a node wants to join an LLN, it still performs an RS/RA, but afterwards, it selects one (or more) neighbors and performs an *Address Registration*. The Address Registration phase ensures the address uniqueness, i.e., it serves the purpose of a DAD. The Address Registration phase also serves as an NDP because it builds and keeps updated the table of the neighbors, along with their IP and link-layer addresses.

The protocol introduces three different roles for nodes participating in an LLN, namely 6LoWPAN Node (6LN), 6LoWPAN Router (6LR), and 6LoWPAN Border Router (6LBR):

1. **6LoWPAN Node (6LN):** Any host or router participating in an LLN. It is worth mentioning that most LLNs are multi-hop, so a router is any node able to forward packets.
2. **6LoWPAN Router (6LR):** An intermediate router having the ability to forward and route IPv6 packets.
3. **6LoWPAN Border Router (6LBR):** A border router located at the edge of the LLN.

Regarding the link-local address uniqueness, the only nodes involved are the 6LN and the 6LR, while the 6LBR is in charge of ensuring global address uniqueness.

2.3. 6LoWPAN-NDP new messages and options

In order to perform the Address Registration and other functionalities (as depicted in Fig. 3), the standard redefined the IPv6-NDP control options defined in RFC 4861 [3], RFC 4862 [6], RFC 7400 [7], and introduced some new ICMPv6 control messages and options [4,5], and [8]. These options and messages are briefly outlined as follows:

2.3.1. 6Lowpan-NDP new messages

- *Extended Duplicate Address Registration (EDAR)*. Used by 6LR to request a check of address uniqueness from the 6LBR on behalf of a 6LN.
- *Extended Duplicate Address Confirmation (EDAC)*. Used by the 6LBR to reply to an EDAR.

2.3.2. 6LoWPAN-NDP new options

- *Extended Address Registration Option (EARO)*. Used to request an Address Registration and to carry the registration result.
- *Authoritative Border Router Option (ABRO)*. Required when RA messages are employed to distribute prefixes and context information.
- *6LoWPAN Context Option (6CO)*. Carries prefix information utilized in 6LoWPAN header compression.
- *6LoWPAN Capability Indication Option (6CIO)*. Defines capability bits for various participating 6LN nodes in an LLN.

The above messages and options contain several fields and are quite complex in terms of implementation, pedantic network management, and processing. Interested readers can refer to RFC 6775 [4] and RFC 8505 [5] for a complete description; the only relevant point is that thanks to these options, a node can both register its address with a neighbor and check that the address is unique.

2.4. Routing-dependent 6LoWPAN-NDP operation

In a multihop LLN, two routing schemes are possible: *Meshunder* and *Route-Over*. Both routing schemes are illustrated in Fig. 2.

In the Meshunder scheme, the IP layer is unaware of the routing mechanism; it is performed by either L2 (fully transparent) or by using the 6LoWPAN MESH header (see RFC 4944 [1]). In both cases, the IP layer is unaware of the mesh topology, and all the nodes in the LLN appear to be at the IP 1-hop distance.

It is fairly evident that when the packet drop probability on a link is non-negligible, the Meshunder scheme must also implement retransmission schemes. From the IP point of view, the network will appear unbalanced, with some nodes having a small delay and high packet delivery rate while others have a high delay and low packet delivery rate. Moreover, a Meshunder scheme must implement complex mechanisms to deliver multicast packets and to efficiently route packets.

In the Meshunder scheme all nodes act as 6LN hosts and there is one 6LBR. Moreover, all the 6LNs, from the point of view of the IP layer, can communicate with the 6LBR directly (i.e., at 1-hop), because the L2 takes care of the multi-hop process.

WirelessHART and *ISA100.11a*, compared in [9,10], are two notable cases of Meshunder schemes. According to [11], in large Meshunder networks, both protocols may face scalability issues because managing and maintaining efficient communication paths can become complex and resource-intensive, leading to potential performance degradation.

In the Route-Over scheme, the routing is performed at the IP level, e.g., by *Routing Protocol for Low-Power and Lossy Networks (RPL)*. Each 6LN-only node (a.k.a leaf nodes) is guaranteed to have a 6LR at a 1-hop distance, and the 6LRs are connected through the routing scheme, to the 6LBR.

The effect of having a Meshunder or a Route-Over scheme directly impacts 6LoWPAN-NDP. As shown in Fig. 3, in a Meshunder scheme, a 6LN can register all its addresses directly with the 6LBR to ensure uniqueness. Note that if a node would want to communicate with another node in the same LLN (besides the 6LBR), it would have to register its link-local address with the destination node. In a Meshunder system, this is not performed, so direct node communication is impossible. This is the intended behavior and is the direct consequence of the assumption of non-uniqueness of the MAC address beyond the true 1-hop. The use of global addresses to communicate in the LLN is usually discouraged and will not be considered in the following.

In a Route-Over scheme, the 6LR will forward the global address registration request to the 6LBR, as shown in Fig. 3. It should be noted that the Route-Over case multicast messages are used only within 1-hop distance, and EDAR/EDAC messages are unicast.

2.5. Routing interplay

The 6LoWPAN-NDP standard should (and is) independent of the routing scheme implemented in an LLN. However, as mentioned previously, the protocol assumes that the RA messages will carry some options that are necessary for the dynamic system configuration.

In the Meshunder approach, routing in the LLN is performed below the IP layer. Hence, the nodes can use RS/RA as they would in a normal IPv6 network, and these messages can carry the necessary options. In the Route-Over approach, the specific protocol being used will have 'equivalent' messages, which can carry the options. For example, in RPL [12], the role of RS and RA is performed respectively by DODAG Information Solicitation (DIS) and DODAG Information Object (DIO) messages. However, it should be stressed that DIS or DIO are routing messages that cannot be used to build the neighbor cache tables.

¹ RFC 6775 is partially obsolete by RFC 8505.

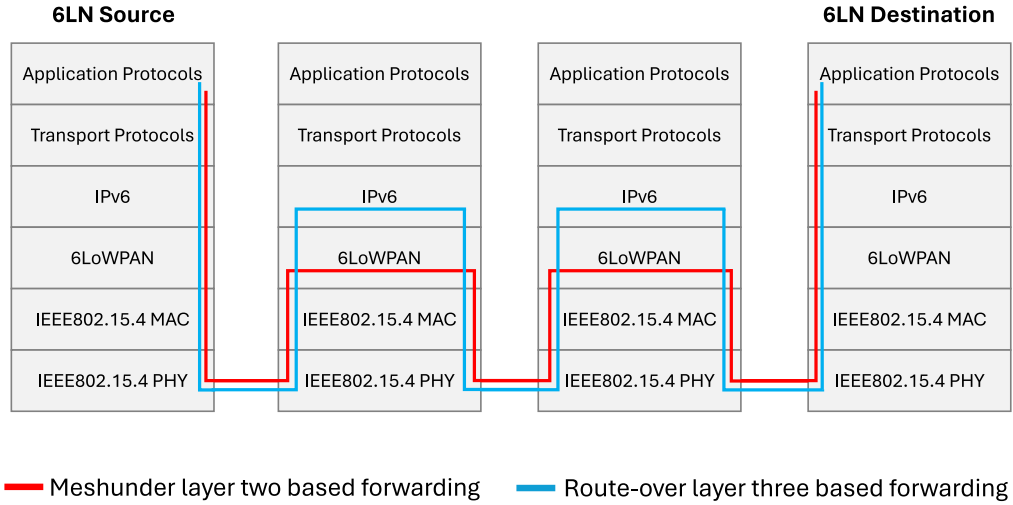


Fig. 2. Meshunder and Route-over packet flow.

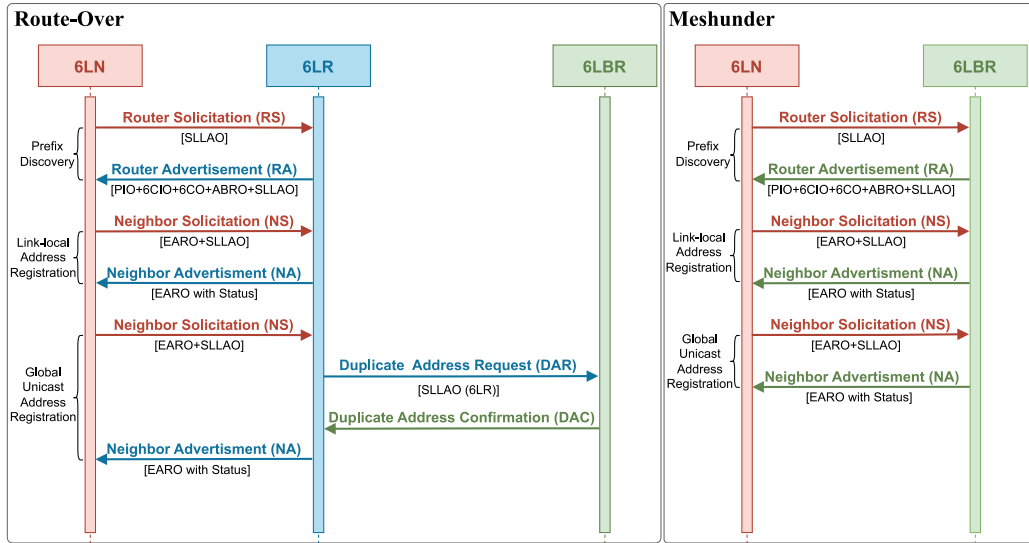


Fig. 3. Address registration message exchange.

2.6. Inter-protocol relationships

6LoWPAN-NDP is transparent to the other protocols, offering a complete replacement of IPv6-NDP. Nevertheless, it requires an active address registration of the link-local address by one of the nodes in order to allow two nodes to communicate. This procedure might fail if the neighbor node has memory limitations or even if an address collision happens.

This point might cause a slight delay in some procedures, e.g., RPL local or global repairs, or AODV route discovery, and can be mitigated by pre-registering the addresses with the neighbor nodes. Nevertheless, this point should be evaluated according to the network topology, the other protocols in the stack, and the specific scenario by performing extensive simulations in order to find the best parameter combination.

3. 6LoWPAN-NDP adoption status

The 6LoWPAN and 6LoWPAN-NDP standards can be considered mature, as RFC 4944 [1] and RFC 6775 [4] were published in 2007 and 2012, respectively. Hence, it would be logical to expect them to be implemented in LLNs, both in simulators and real protocol stacks.

However, this expectation does not match the reality: 6LoWPAN-NDP is not widely adopted.

Table 1² provides a brief overview of the implementation status of 6LoWPAN and 6LoWPAN-NDP. It is evident that the 6LoWPAN protocol is widely supported, while the opposite is true for 6LoWPAN-NDP. The difference is striking and raises the question: Is 6LoWPAN-NDP necessary? Certainly, there are several reasons for the lack of its adoption, these will be outlined in the following. Nevertheless, it is important to point out one particular element: interoperability between implementations.

Interoperability is a major concern in an LLN with devices from different vendors, possibly with different protocol stacks. Interoperability between the 6LoWPAN compression versions is possible and rather simple, as the different compression schemes use the same syntax to indicate the compression scheme being used. As a consequence, an implementation can easily support all the different schemes and be backwards compatible with nodes implementing only a subset of the compression schemes.

² Some Linux kernel versions and networking stacks may also include support for 6LoWPAN and 6LoWPAN-NDP.

Table 1

Features supported by different stacks in the market.

Stack	IPv6-NDP	6LoWPAN	6LoWPAN-NDP
[13] ns-3	✓	✓	✓✓ ^a
[14] OMNeT++	✓	✓ ^b	–
[15] NetSim	✓	✓	–
[16] Keysight	✓	✓	–
[17] Contiki	✓	✓	–
[18] Contiki-NG	✓	✓	✓ ^b
[19] OpenWSN	✓	✓	–
[20] OpenThread	✓	✓	–
[21] RIOT OS	✓	✓	✓✓ ^a
[22] Tiny OS	– ^c	✓	–
[23] Mbed-OS	✓	✓	✓✓ ^a
[24] Zephyr OS	✓	✓	–
[25] FreeRTOS	✓	–	–
[26] Netualizer	✓	✓	–
[27] Mininet	✓	✓	–

IPv6-NDP: ✓ RFC4861, RFC4862, **6LoWPAN:** ✓ RFC4944, RFC6282**6LoWPAN-NDP:** ✓ RFC6775, ✓✓ RFC8505.^a Partial support of RFC8505.^b Not part of the official release. Independent work [28,29].^c Partial support some of RFC 4861.

If the L2 protocols provide the equivalent of an EtherType, then it is possible to have coexistence between nodes using the 6LoWPAN and nodes not using it. If the L2 protocols do not have an EtherType (e.g., IEEE 802.15.4, and others do not have it), then all the nodes will have to use 6LoWPAN. However, since 6LoWPAN is an adaptation layer, and since its use is practically required in LLNs, it is widely adopted.

Regarding 6LoWPAN-NDP, the situation differs. Interoperability between nodes using it and those not using it is not guaranteed, and mixing the IPv6 NDP and 6LoWPAN-NDP is not possible. Consequently, its adoption is *limited* by the interoperability issue. From an implementation standpoint, it makes sense to avoid having a complex protocol that will most likely be disabled. There is no pressure to implement it unless it offers clear advantages over the ‘standard’ NDP. In the following sections, we will address the questions: how complex is it to implement 6LoWPAN-NDP, and does it provide clear advantages?

Both Neighbor Discovery (ND) protocols were compared and evaluated by [29,30], but only for a Route-Over topology. Work has been implemented on the Contiki OS 2.7, but only RFC 6775 [4] has been considered. [31] analyzed both NDPs procedures and their analysis was quite simple and only compared the RA and RS control messages exchanged for building topology information. Their results are very similar to previous work done in [29,30].

Although interesting, the previous works are based on outdated 6LoWPAN-NDP versions, and only consider the Route-Over scenario.

4. 6LoWPAN complexity

In this section, we will discuss 6LoWPAN-NDP in terms of the NCE states, in particular concerning the differences with the analogous states in IPv6-NDP, link-local and global Address Registration processes, and implicit and explicit NUD.

4.1. NCE states maintenance

The 6LoWPAN-NDP standard [4,5] defines three NCE states (REGISTERED, TENTATIVE, and GARBAGE-COLLECTIBLE) that are orthogonal to the states specified in the IPv6-NDP [3]. However, the roles of these new states and the transition mechanism between them are not well explained in either RFC 6775³ [4] or its updated version RFC 8505 [5]. The main problem is how these states evolve

and how these new orthogonal states influence the evolution of the IPv6-NDP NCE states. This aspect is only briefly explained in the standard, and it might create issues in an implementation.

An entry with a GARBAGE-COLLECTIBLE status strictly follows the IPv6 rules and timers. This status is used by a 6LN for the addresses of the 6LR or 6LBR nodes to which it is registered.

Conversely, a REGISTERED entry is used to indicate that a node did perform an Address Registration procedure. Consequently, it should not be refreshed by the node, as the *Registering Node* is assumed to refresh the registration periodically. A REGISTERED entry is simply removed when the registration timer expires. The TENTATIVE state is the most ambiguous one. The standard foresees its use, but there is no evident need for it. An entry with TENTATIVE status should not be used, and the only reason to include it in the standard is to use the NCE table to hold temporary data. As a matter of fact, an implementation might choose not to use it, as confirmed by the RFC Editor.

4.2. Address registration procedure

This procedure is perhaps the most challenging to handle. A node must not use an address unless it is registered *with the node it is trying to reach*. This stands in stark contrast with the ‘normal’ procedure of IPv6-NDP, where a node is allowed to discover a neighbor’s address by sending a Neighbor Solicitation (NS) and receiving a Neighbor Advertisement (NA). The data flow is shown in Fig. 3. Here we want to aim to highlight the complexities in this seemingly simple mechanism.

4.2.1. Address registration failures

The first not-so-evident point is the relationship between LUA and GUA registrations. As a matter of fact, both can fail independently. However, the registration might fail for two reasons: the address is already registered (i.e., it is a duplicate), or because the registering node cache is full. If a LUA registration fails, independently of the reason, the node should put the neighbor in a temporary blacklist. If the GUA registration fails due to duplication, then the address must be changed. In case the cache is full, the address cannot be used, nor can it be registered. The standard does not specify if the node can still use the LUA, but it appears reasonable to retain it [4,5]. Conversely, if the LUA fails, the node should also discard the global address registration result, and start over. This point is not explicitly stated in the RFCs, and it might happen when address registration is performed in parallel, i.e., the global address registration is started before the link-local address registration result, which is allowed by the standard.

4.2.2. Multiple address registrations

This situation may arise in two scenarios: when there are multiple 6LBRs in the network, and in Route-Over topologies.

If there are multiple 6LBRs, a node can register with all of them. However, the standard does not fully explain this case. It is obvious that different 6LBRs will have to use different IPv6 prefixes; However, the standard does not fully describe how to handle, for example, a link-local address registration failure with one 6LBR, i.e., whether the 6LN can (or should) use a different link-local address with different 6LBRs. This point is open to investigation.

In the Route-Over scenario, it is expected to receive multiple RAs from different 6LRs, even with one 6LBR. In this case, the 6LN can register its addresses with one or more 6LRs, without the risk of conflicts. However, this also means that the 6LBR will receive multiple EDARs, and will have to handle them accordingly, i.e., by verifying that they originate from the same 6LN in order to prevent an incorrect duplicated address reply error.

4.2.3. Address registration consequences

There are some direct consequences to the Address Registration procedure that need to be highlighted because, in our opinion, they are

³ See RFC 6775, Section 3.5.

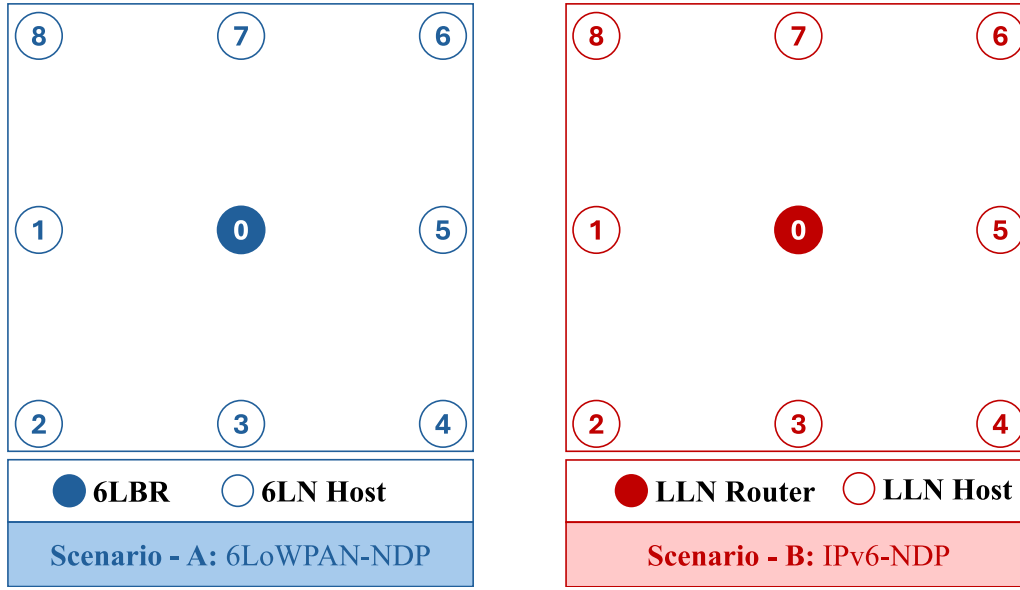


Fig. 4. Grid meshunder topology.

not immediately obvious. The first point relates to NUD. In IPv6-NDP, each node is tasked with managing its neighbor cache table and ensuring it stays up to date, taking proactive measures to prevent entries from becoming STALE. This is still true for 6LoWPAN-NDP, but only for GARBAGE-COLLECTIBLE entries. For REGISTERED entries, it is the registering node the one who is responsible for keeping the entry updated.

As an example, if a 6LN registers its addresses with a 6LR, the 6LN will have a GARBAGE-COLLECTIBLE entry for the 6LR, and the 6LR will have a REGISTERED entry for the 6LN. The 6LN must keep the 6LR entry ‘fresh’ even if it does not have anything to send to the 6LR (or any other node). Failing to do so will make the 6LN unreachable from the network. This creates a small, but continuous, overhead traffic in the network. However, it can be minimized by setting the protocol parameters (i.e., the registered entry lifetime). However, the parameter must also take into account the node mobility, so optimizing the network parameters becomes a non-trivial task.

The second point is relative to the possibility of directly communicating with neighbors. According to the protocol rules, a node cannot communicate with another node unless there is a registration. When considering two nodes, it does not matter which one registered with the other, but at least one of them must have completed the registration procedure. Mutual registration is unnecessary, but one-way registration is essential.

This situation poses a problem, especially in Meshunder topologies. If a node does not send RA messages, then the neighboring nodes will not start a registration. In a Meshunder network, only the router is responsible for sending RA messages. Hence, the IP-level topology becomes a star rather than a mesh.

However, this behavior is correct and intentional. Assuming that MAC addresses are not unique is correct and necessary in LLNs, and the registration procedure prevents this type of network error. Moreover, the registration procedure also ensures the existence of a bidirectional channel or implicit NUD, which is not guaranteed by a normal NDP.

4.3. Computational complexity and memory requirements

The requirements of 6LoWPAN-NDP in terms of memory and CPU are different, depending on the role of the nodes [32].

A node registering an address to another node (a neighbor or the 6LBR) must store the address lifetime and periodically maintain the

registration. A registrar (a neighbor or the 6LBR) needs to store the registered addresses and their states.

From a computation point of view, the protocol does not require any complex operation, except a lookup in a table, which can be optimized depending on the specific scenario. The extensions to 6LoWPAN-NDP (e.g., [8]) have different computational requirements that are not in the scope of the recent paper.

From a memory point of view, the 6LBR needs to store all the global address registrations. Hence, the memory footprint can be large in some scenarios. However, Dynamic Host Configuration Protocol version 6 (DHCPv6) [33] has a similar requirement, and the 6LBR is usually implemented in the gateway node. Hence, this point does not represent a limitation.

However, 6LoWPAN-NDP also specifies that any node must register its link-local address to its neighbors. This registration is asymmetrical, i.e., if two nodes are neighbors, only one of them needs to register with the other. Nevertheless, this point can be a limiting factor in small devices.

Note that, even though not mentioned in the standard, it is possible to implement a ‘reverse’ registration, which can be useful for nodes with a very large fan-out.

5. Protocol evaluation

In order to evaluate the performance and overhead of 6LoWPAN-NDP and IPv6-NDP, we implemented them in the ns-3 simulator [13]. The implementation, currently available publicly, is preliminary and does not yet include the EDAR/EDAC messages [34]. Nevertheless, it can successfully evaluate the protocol performances in the 1-hop scenario. Our specific objective is to analyze protocol *overhead* and *stability* in the Meshunder scenario.

While scalability is crucial for IoT, a 1-hop analysis effectively illustrates the performance of both protocols.

5.1. Simulation scenario

To analyze the performance of both NDPs, we employ a grid Meshunder topology with different node placements as shown in Fig. 4. The topology is selected to stress-test the protocols because in the ns-3 it is implemented through controlled flooding. While the method is not particularly optimized, it permits the use of multicast. Consequently,

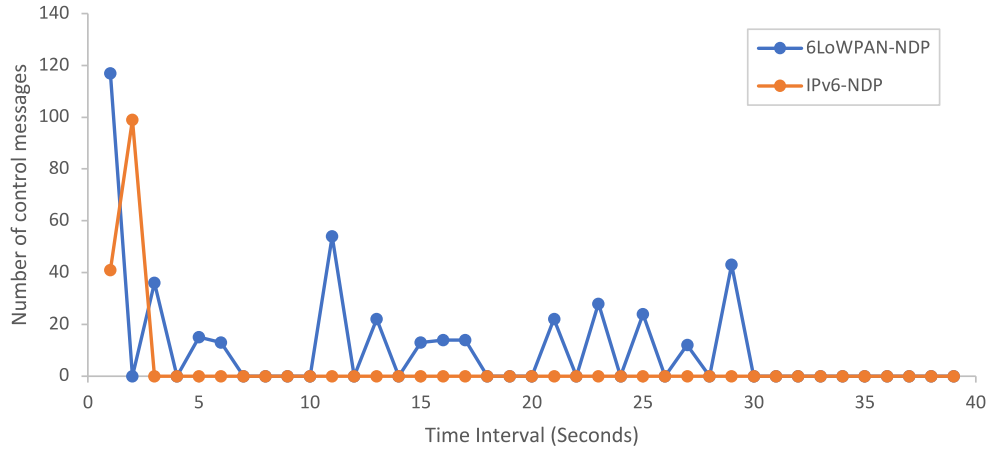


Fig. 5. Control messages.

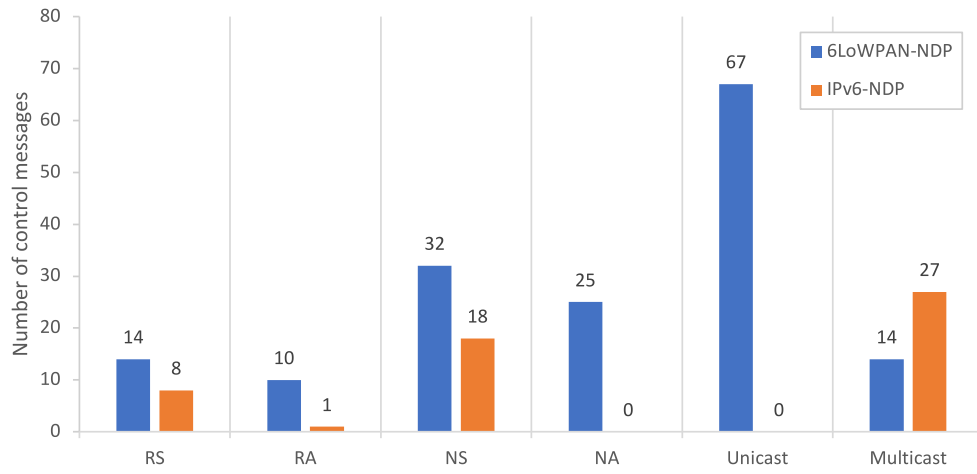


Fig. 6. Control message types.

IPv6-NDP can yield some results, indicating that the network appears to be functioning.

In our setup, the topology is based on 9 nodes. Wherein 6LoWPAN-NDP is running in Scenario-A. The central node indexed as 0 serves as a 6LBR, and all other nodes from 1–8 work as 6LN. In Scenario-B IPv6-NDP in action. The node indexed 0, functions as a LLN router, and all other nodes from 1–8 work as normal LLN nodes.

Moreover, other parameters of the simulation setup are presented in Table 2, the nodes are 30 m apart, which guarantees a 1-hop communication between the nodes. The simulations are performed using the IEEE 802.15.4, with and without active data traffic.

In order to analyze the effects of random effects in the simulations, we performed several simulations for each scenario, and we noticed that, due to the protocol behavior, the results are practically not affected by aleatory effects, with the exception of the occasional packet losses due to the channel errors. Due to the similarity between different experiments, without loss of generality, in the following we will report the result of one experiment randomly selected between the ones we did execute.

5.2. Protocol overhead analysis

The first result to highlight is shown in Figs. 5 and 6. Note that, due to the Meshunder controlled flooding, the number of messages is amplified. We ran this simulation for 40 s without active data traffic to analyze the bootstrapping behavior of both ND protocols while exchanging the control messages.

Table 2

Simulation parameters.

Parameter type	Value
Number of nodes	8 6LNs and 1 6LBR/LLN router
Radio range	About 100 m
Distance between nodes	30 m
IEEE802.15.4	Beacon-less, always on
Propagation model	Log-distance
6LoWPAN compression	RFC 6282
6LoWPAN-NDP	RFC 8505
IPv6-NDP	RFC 4861
Mobility model	Constant position mobility model
Data traffic	UDP to the central node, 12 bytes each second

It is evident from Fig. 5 that in the case of IPv6 NDP, a single RA message triggers the nodes to generate addresses and the necessary DAD procedure. This creates a temporary network spike, but afterward, the network is practically inactive. On the contrary, 6LoWPAN-NDP has more activity, primarily due to the Address Registration phase. However, Fig. 6 highlights how 6LoWPAN-NDP uses mostly unicast messages, while IPv6-NDP uses only multicast.

We should stress that the unicast messages are all 1-hop, while the multicast messages must be broadcasted to the whole network unless the Meshunder routing does perform an extremely complex routing scheme, which is usually not implemented. Hence, despite the apparent higher network load, 6LoWPAN-NDP results to be more efficient.

The Address Registration phase has another effect, shown in Fig. 7, i.e., that 6LoWPAN-NDP requires a periodic registration refresh. In the

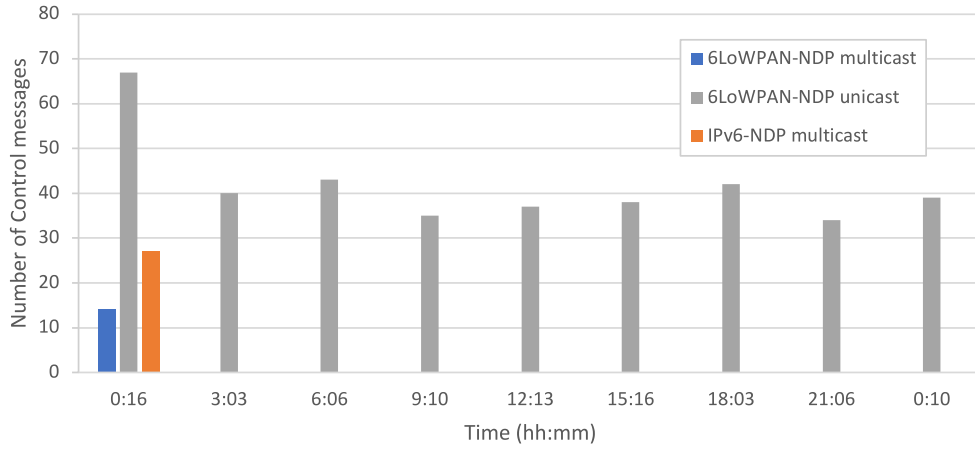


Fig. 7. Re-registration control traffic.

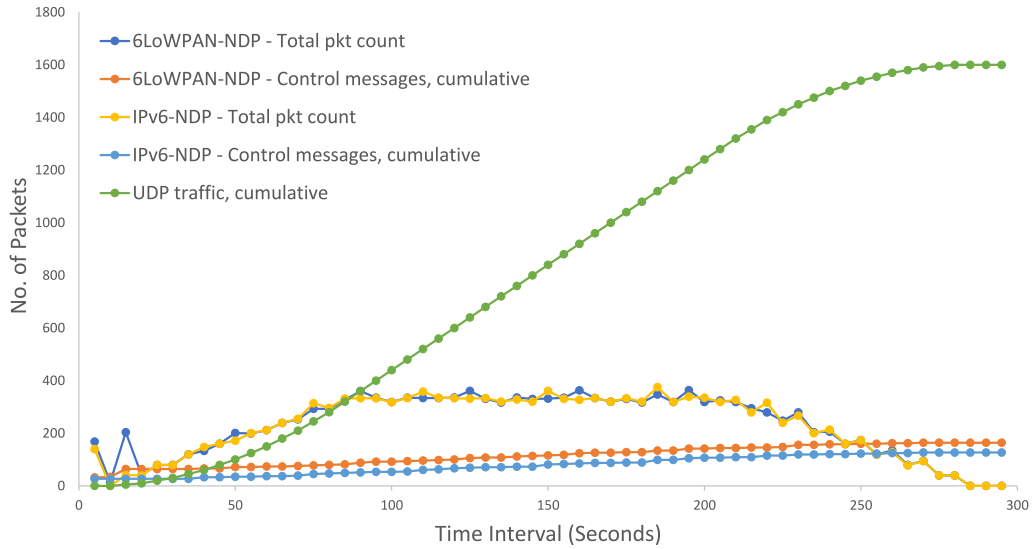


Fig. 8. Effects of data traffic on the system.

simulation we did set the Address Registration process timeout to 1 day, and the re-registration timeout to 3 h. This setup has been chosen to increase the probability of a successful re-registration, which might fail due to network errors.

The standard does not specify the policy for address re-registration (i.e., how to handle timeout errors, or how many times a node can retry), but it is evident that, in order to keep the registration active, it is mandatory to choose an adequate policy. Moreover, an efficient implementation should try to randomize the re-registration between nodes, to avoid re-registration network spikes in the network.

Toward this end, it is possible to adopt a policy similar to the one used in RPL, where Lollipop timers are used to dynamically adjust the message frequency according to the network conditions. Moreover, the address timeout and re-registration policies should take into account the node mobility and dynamic changes in the network to promptly prune non-active nodes.

Moreover, It is worth noticing that different re-registration timing policies would not lead to incompatibilities between implementations, but they would definitely affect the network efficiency. Toward this end, it is quite logical that this part is not precisely described in the standard, and it is left as implementation-dependent.

5.3. Active data traffic analysis

The simulation has been repeated with UDP active data traffic enabled, as expected in an IoT system. We outlined in Fig. 8, that the control message traffic is negligible concerning the data traffic. The figure reports both the sum of the messages sent and the total number of messages sent in a given interval. It is clear how the two NDPs mechanisms have an almost identical trend, and how the Address Registration happening at the very beginning of the simulation becomes irrelevant to the overall number of messages sent. Moreover, we can observe how the Meshunder controlled flooding policy is a sub-optimal choice, as it multiplies unnecessarily the actual number of messages in the network.

In the simulation, the data traffic triggers a refreshment of REACHABLE, GARBAGE-COLLECTIBLE entries, which are not refreshed by upper-layer traffic. The IPv6-NDP standard states that NCEs can be refreshed by upper-layer traffic, but only if there is a hint of two-way reachability. Consequently, UDP traffic cannot provide a hint, because it is inherently unidirectional, while TCP traffic can.

An implementation should consider this interaction and provide APIs to allow applications to provide the reachability hints, thus minimizing the unnecessary NDP refreshes.

5.3.1. Energy evaluation

The energy consumption of 6LoWPAN-NDP is dependent on the specific MAC and PHY protocols and is complex to generalize. Moreover, the routing protocol can also affect the results, as an efficient multicast forwarding can decrease significantly the energy consumption of IPv6-NDP.

However, it is possible to have a rough estimate of the energy consumption by measuring the number and type of packets required by the protocols. IPv6-NDP is based on heavy multicast control messages, which have to be forwarded to all the nodes. This, of course, leads to a non-negligible energy consumption. 6LoWPAN-NDP, on the contrary, only uses 1-hop multicast messages or unicast messages, which have a lower energy requirement because they do not need to be forwarded to the whole network. Moreover, the address registration lifetime can be tuned to lower energy consumption.

As a consequence, we expect 6LoWPAN-NDP to be more energy efficient than IPv6-NDP, especially in large topologies.

5.3.2. Protocol reliability

Another interesting fact found while running IPv6-NDP is NS(DAD) packet drops, highlighted in the log files:

```
d 0.021 /NodeList/4/DeviceList/0/$ns3::LrWpanNetDevice /Mac/MacTxDrop
```

Although it might be considered a minor problem, the fact that NS packets have been dropped in a scenario as simple as the one we used is deeply concerning. Losing an NS is a clear sign of protocol unreliability, and might lead to both duplicate addresses (if the DAD process fails to detect an address duplication), and to the non-reachability of an active node (if the NDP fails to return the destination address).

The second issue might lead to nodes appearing unreachable by some others, while they are reachable by others. It is obvious that detecting and fixing such erratic behavior is a network administration nightmare. Moreover, detecting such losses is non trivial, and sometimes almost impossible, in a real network.

We must stress that these two issues are completely prevented by 6LoWPAN-NDP *by design*.

6. Conclusions

The traditional classical IPv6-NDP was designed for links with a negligible loss rate, e.g., Ethernet or WiFi and is based on a reactive approach (build the neighbor cache when needed). On the contrary, 6LoWPAN-NDP uses a proactive approach and combines ND and DAD features.

Moreover, the 6LoWPAN-NDP has been designed to reduce the multicast messages, which do not guarantee a reliable reception, and are either energy inefficient in LLNs, and in some cases cannot be supported at all.

The purpose of this research was to investigate the performance and overhead of 6LoWPAN-NDP and IPv6-NDP. We also highlighted how, even in simple topologies, the IPv6-NDP can be considered unreliable, and its effectiveness in both detecting address duplication and finding neighbors is questionable, at best.

Furthermore, 6LoWPAN-NDP defines very long-lived NCEs, where the lifetime of each entry is configured by the border router. This ensures a drastic reduction in the ICMPs in the LLN.

Overall, we can state that, despite the complexity of the protocol, the 6LoWPAN-NDP usefulness has been heavily underestimated, and the lack of it in the implementations is a major issue, as the IPv6 DAD and ND are not suitable for LLNs. Not only do they use multicast, which could be unsupported at the routing level, but they also are unreliable on LLNs. Hence, fostering the 6LoWPAN-NDP adoption is mandatory. We think our work on the ns-3 simulator can be a step toward a deeper understanding of the protocol's usefulness by the IoT operating systems developers.

Finally, we want to stress that the availability of simulation tools able to perform accurate analysis on these protocols is of paramount importance. The interrelationships between the protocols at different layers are non-obvious, and the evaluation of each of them cannot be performed without taking into account the contributions of the other ones.

We should, however, note that 6LoWPAN-NDP can be improved. For example, a guideline on implementing the refresh timers would be beneficial, even if they are kept implementation-dependent.

Another potential improvement could be to have a flag in the RA options to indicate the willingness of a router to register addresses. This element could be extremely useful in constrained environments and would avoid trying an address registration on nodes that have no resources left.

CRedit authorship contribution statement

Adnan Rashid: Writing – review & editing, Writing – original draft, Validation, Software, Investigation, Formal analysis, Conceptualization.
Tommaso Pecorella: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by BRIEF—Biorobotics Research and Innovation Engineering Facilities—Missione 4, 'Istruzione e Ricerca'—Componente 2, 'Dalla ricerca all'impresa'—Linea di investimento 3.1, 'Fondo per la realizzazione di un sistema integrato di infrastrutture di ricerca e innovazione', funded by European Union—NextGenerationEU, CUP: J13C22000400007.

This work was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE0000001 - program "RESTART").

References

- [1] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC 4944 (Proposed Standard), updated by RFCs 6282, 6775, 8025, 8066, Fremont, CA, USA, Sep. 2007, <http://dx.doi.org/10.17487/RFC4944>, <https://www.rfc-editor.org/rfc/rfc4944.txt>.
- [2] J. Hui (Ed.), P. Thubert, Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, RFC 6282 (Proposed Standard), updated by RFC 8066, Sep. 2011, <http://dx.doi.org/10.17487/RFC6282>, <https://www.rfc-editor.org/rfc/rfc6282.txt>.
- [3] T. Narten, E. Nordmark, W. Simpson, H. Soliman, Neighbor Discovery for IP Version 6 (IPv6), RFC 4861 (Draft Standard), updated by RFCs 5942, 6980, 7048, 7527, 7559, 8028, 8319, 8425, Sep. 2007, <http://dx.doi.org/10.17487/RFC4861>, <https://www.rfc-editor.org/rfc/rfc4861.txt>.
- [4] Z. Shelby (Ed.), S. Chakrabarti, E. Nordmark, C. Bormann, Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 6775 (Proposed Standard), updated by RFC 8505, Nov. 2012, <http://dx.doi.org/10.17487/RFC6775>, <https://www.rfc-editor.org/rfc/rfc6775.txt>.
- [5] P. Thubert (Ed.), E. Nordmark, S. Chakrabarti, C. Perkins, Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery, Nov. 2018, <http://dx.doi.org/10.17487/RFC8505>, <https://www.rfc-editor.org/rfc/rfc8505.txt>.
- [6] S. Thomson, T. Narten, T. Jinmei, IPv6 Stateless Address Autoconfiguration, RFC 4862 (Draft Standard), updated by RFC 7527, Sep. 2007, <http://dx.doi.org/10.17487/RFC4862>, <https://www.rfc-editor.org/rfc/rfc4862.txt>.

- [7] C. Bormann, 6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 7400 (Proposed Standard), Nov. 2014, <http://dx.doi.org/10.17487/RFC7400>, <https://www.rfc-editor.org/rfc/rfc7400.txt>.
- [8] P. Thubert, B. Sarikaya, M. Sethi, R. Struik, Address-Protected Neighbor Discovery for Low-Power and Lossy Networks, RFC 8928, Nov. 2020, <http://dx.doi.org/10.17487/RFC8928>, <https://www.rfc-editor.org/info/rfc8928>.
- [9] S. Petersen, S. Carlsen, WirelessHART versus ISA100.11a: The format war hits the factory floor, IEEE Ind. Electron. Mag. 5 (4) (2011) 23–34, <http://dx.doi.org/10.1109/MIE.2011.943023>.
- [10] J.D. Adriano, E.C.d. Rosario, J.J. Rodrigues, Wireless sensor networks in industry 4.0: WirelessHART and ISA100.11a, in: 2018 13th IEEE International Conference on Industry Applications, INDUSCON, 2018, pp. 924–929, <http://dx.doi.org/10.1109/INDUSCON.2018.8627177>.
- [11] T.P. Raptis, A. Passarella, M. Conti, A survey on industrial internet with ISA100 wireless, IEEE Access 8 (2020) 157177–157196, <http://dx.doi.org/10.1109/ACCESS.2020.3019665>.
- [12] T. Winter (Ed.), P. Thubert (Ed.), A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, Mar. 2012, <http://dx.doi.org/10.17487/RFC6550>, <https://www.rfc-editor.org/rfc/rfc6550.txt>.
- [13] Ns-3: A discrete-event network simulator, 2023, URL <http://www.nsnam.org/>. (Accessed 25 December 2023).
- [14] OMNeT++: An extensible, modular, component-based C++ simulation library, 2023, URL <https://omnetpp.org/>. (Accessed 25 December 2023).
- [15] TETCOS - network simulation solutions, 2023, URL <https://www.tetcos.com/index.html>. (Accessed 25 December 2023).
- [16] Keysight technologies - network modeling, 2023, URL <https://www.keysight.com/us/en/products/network-test/network-modeling.html>. (Accessed 25 December 2023).
- [17] The contiki open source OS for the internet of things, 2024, URL <https://github.com/contiki-os/contiki/tree/master>. (Accessed 11 April 2024).
- [18] Contiki operating system - SICSLOWPAN documentation, 2024, URL <https://github.com/contiki-os/contiki/blob/master/doc/sicslowpan-doc.txt>. (Accessed 25 December 2023).
- [19] Openwsn, 2023, URL <https://openwsn.atlassian.net/wiki/spaces/OW/overview>.
- [20] OpenThread - an open-source implementation of thread, 2023, URL <https://openthread.io/>. (Accessed 25 December 2023).
- [21] RIOT operating system - 6LoWPAN neighbor discovery API documentation, 2023, URL https://api.riot-os.org/group_net_sixlowpan_nd.html. (Accessed 25 December 2023).
- [22] T. Freeforall, Blip lib6lowpan - tinyos tools, 2023, URL <https://github.com/tp-freeforall/prod/tree/tp-master/tools/tinyos/c/blip/lib6lowpan>. (Accessed 25 December 2023).
- [23] Arm mbed OS - 6LoWPAN neighbor discovery API documentation, 2023, URL <https://os.mbed.com/docs/mbed-os/v6.16/apis/6LoWPAN-ND-tech.html>. (Accessed 25 December 2023).
- [24] Zephyr project documentation - networking, 2023, URL <https://docs.zephyrproject.org/latest/connectivity/networking/index.html>. (Accessed 25 December 2023).
- [25] FreeRTOS - real time operating system for IoT, 2023, URL <https://www.freertos.org/index.html>. (Accessed 25 December 2023).
- [26] R. Herrero, Towards protocol stack virtualization in massive IoT deployments, Internet Things 14 (2021) 100396, <http://dx.doi.org/10.1016/j.iot.2021.100396>, URL <https://www.sciencedirect.com/science/article/pii/S2542660521000408>.
- [27] An instant virtual network on your laptop (or other PC), 2024, URL <https://mininet-wifi.github.io/sixlowpan/>. Accessed 11 April 2024.
- [28] M. Kirsche, J. Hartwig, A 6LoWPAN model for OMNeT++: Poster abstract, in: Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques, SimuTools '13, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, 2013, pp. 330–333.
- [29] M.A.M. Seliem, K.M.F. Elsayed, A. Khattab, Performance evaluation and optimization of neighbor discovery implementation over contiki OS, in: 2014 IEEE World Forum on Internet of Things, WF-IoT, 2014, pp. 119–123, <http://dx.doi.org/10.1109/WF-IoT.2014.6803132>.
- [30] M.A. Seliem, K.M. Elsayed, A. Khattab, Optimized neighbor discovery for 6LoWPANs: Implementation and performance evaluation, Comput. Commun. 112 (2017) 73–92, <http://dx.doi.org/10.1016/j.comcom.2017.08.013>, URL <https://www.sciencedirect.com/science/article/pii/S0140366417308988>.
- [31] N. Fathima, A. Ahammed, Rajashekarappa, R. Banu, B.D. Parameshachari, N.M. Naik, Optimized neighbor discovery in internet of things (IoT), in: 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques, ICEECCOT, 2017, pp. 1–5, <http://dx.doi.org/10.1109/ICEECCOT.2017.8284573>.
- [32] H. Ayers, P. Crews, H. Teo, C. McAvity, A. Levy, P. Levis, Design considerations for low power internet protocols, in: 2020 16th International Conference on Distributed Computing in Sensor Systems, DCOSS, 2020, pp. 103–111, <http://dx.doi.org/10.1109/DCOSS49796.2020.00027>.
- [33] T. Mrugalski, M. Siodelski, B. Volz, A. Yourtchenko, M. Richardson, S. Jiang, T. Lemon, T. Winters, Dynamic Host Configuration Protocol for IPv6 (DHCPv6), RFC 8415 (Proposed Standard), Nov. 2018, <http://dx.doi.org/10.17487/RFC8415>, <https://www.rfc-editor.org/rfc/rfc8415.txt>.
- [34] Ns-3 Development Team, Ns-3-dev repository, 2023, URL https://gitlab.com/tommypec/ns-3-dev/-/tree/6LoWPAN-ND?ref_type=heads. GitLab.



Adnan Rashid, a member of IEEE and the Internet Society, currently serves as an Assistant Professor at Politecnico di Bari, Italy, since March 2023. He earned his Ph.D. in Telecommunication and Telematics from the University of Florence in 2022 and holds an MS in Computer Engineering from CUST, Pakistan (2015), and a BS in Computer Science from FUUAST, Pakistan (2009). Adnan's teaching and research focus on IoT system security, protocols, and applications. He was a visiting faculty at PMAS-Arid Agriculture University, Pakistan, from 2017 to 2018. He also worked as a researcher at CUST on a National ICT R&D-funded project from 2013 to 2014 and as an IoT and Security Research Associate at the University of Florence from January 2019 to February 2023. He is actively engaged in the IETF working group (ippm), and he contributes to IPv6 Performance and Diagnostic Metrics Version 2. Adnan is a dedicated Technical Program Committee member for prestigious conferences and collaborates on the IETF 6LoWPAN-ND protocol's development with ns-3 simulator maintainers.



T. Pecorella received Ph.D. and M.Sc. degrees in Electronic Engineering (Telecommunications track) from the Department of Information Engineering at University of Florence (Italy) in 2000 and 1996 respectively. From 2001 to 2007 he was a researcher at Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT). Since November 2007 he is a tenure-track Assistant Professor in the Department of Information Engineering at University of Florence (Italy). In 2018 and 2019 he was also visiting professor at University of Saint Louis, Missouri (USA). He received the Best paper award at the IEEE GLOBECOM 2016, and in 2021 got the Italian Habilitation (Abilitazione Scientifica Nazionale) for Associate Professorship in Telecommunication Engineering.

He is the author of more than 90 publications between conference papers and journals. His research interests focus on IoT communication systems, network security, and application of machine learning to networking systems.